



# Matting and Compositing for Fresnel Reflection on Wavy Surfaces

Yuki Endo<sup>1</sup> Yoshihiro Kanamori<sup>1</sup> Yukio Fukui<sup>1</sup> Jun Mitani<sup>1,2</sup>

<sup>1</sup> University of Tsukuba    <sup>2</sup> JST ERATO

---

## Abstract

*This paper introduces a framework that can extract an alpha matte from a single image with Fresnel reflection, and that can composite other objects with the image such that plausible reflections are included. Our method handles reflections in a plane with small undulations, for example, a water surface with waves or a glossy tabletop. During the matting stage, our method first estimates the transmission color, which is assumed to be uniform, and then calculates a reflection image and alpha matte based on user markups. However, accurate extraction of the matte becomes challenging when a plane has small undulations because these create perturbations in the matte. We therefore propose a filter that can refine the matte effectively. In the compositing stage, the reflection of a composited object is synthesized by ray tracing in real time. We demonstrate the effectiveness of our method through comparisons with ground-truth data and results using natural images as inputs.*

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

---

## 1. Introduction

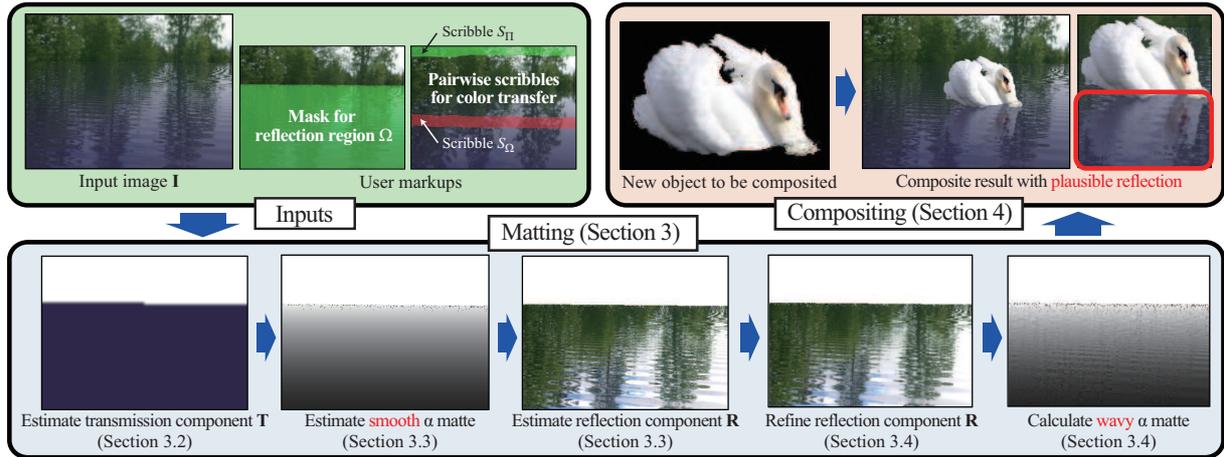
Scenes reflected on the surface of the sea or a lake are popular photographic subjects, as are seen by a quick image search on the Internet. Water surfaces can behave like a flat mirror, or can yield interesting warping in reflection images if the surface is wavy. Such effects are caused by *Fresnel reflection*, which consists of a linear combination of reflected and transmitted light, and can be observed in our life, for example, on a tabletop or glossy floor.

Editing photographs with Fresnel reflection is quite interesting, especially when an image includes reflections on wavy surfaces. Indeed, pleasing images and animations with reflections on wavy surfaces can be created from a single photograph, although this involves considerable manual labor [CGZ\*05, PHCS06]. Matting plays the key role in semi-automating this creation process; the reflected image is first separated into a reflection component, transmission component, and alpha matte, and then new objects can be composited by using this information. Matting is ill-posed, however, and is therefore a challenging problem. Applying previous matting methods for optical phenomena (e.g., shadow [CGC\*03, WTBS07] and haze [Tan08, Fat08, HST09, KN09]) to images with Fresnel reflection is difficult due to the differences between the targets and the incompatibility of the formulations. Additionally, the matting

method for *refraction* in glass [YTBK11] is inappropriate for the complicated *reflections* observed on water surfaces.

In this paper, we propose a matting method that deals with Fresnel reflection in a single image based on user markups. Targeting reflections on surfaces of deep water, glossy tabletops, floors, and the like, we introduce the following three assumptions: (1) the transmission color is approximately uniform, (2) a pair composed of a source object and its reflection can be found in the input image, and (3) the reflection surface is predominantly planar but possibly includes waves. We first estimate the transmission color based on the first and second assumptions by using *color transfer* [RAGS01]. We then roughly estimate the reflection component and alpha matte, as well as the camera parameters, by assuming that the alpha matte is smooth. However, the alpha matte will contain high-frequency regions in the case of a wavy reflection surface. We thus propose a filter that refines the alpha matte, and use ground-truth data to validate the filter. We also provide a compositing system that uses the calculated information to composite new objects, including generated plausible reflections, with the input image in real time.

Figure 1 presents an overview of our system. Our matting algorithm takes a single image with a reflection surface as an input, as well as a user-specified region of the reflection surface and a pair of scribbles for color transfer. The al-



**Figure 1:** Overview of our system. To solve matting problems, a user specifies the reflection surface and pairwise scribbles. After several matting steps, a composite result with plausible reflection can be obtained.

gorithm then computes the reflection component, transmission component, and alpha matte, and modifies the reflection component and alpha matte such that wave effects can be incorporated (Section 3). From this matting information, we subsequently render reflection images of newly composited objects by using ray tracing (Section 4).

Note that, to the authors’ knowledge, this is the first attempt to extract a complicated reflection and alpha matte for Fresnel reflection from a single image. We demonstrate the effectiveness of our method through various examples, and discuss its limitations in terms of the assumptions used (Sections 5 and 6).

## 2. Related Work

Current methods for *natural image matting* are outlined in a survey [WC08] and an evaluation website [RRW\*09]. These methods extract foreground objects and an alpha matte from a single image by the user specifying definitely foreground and background regions using a *trimap* or scribbles. Unfortunately, such definite separation is not available for reflection images, and thus we require an alternative method.

Several matting methods for optical phenomena have recently been proposed. Wu et al. [WTBS07] extracted shadows from natural images by assuming that colors in a shadowed region are products of the shadow color and ground colors based on retinex theory. Similarly, haze removal techniques suppose that an input image is a linear blending of the haze color and background colors. Tan’s method [Tan08] enhances image contrast, and does not decompose the image. Fattal’s [Fat08] dehazing method estimates a uniform haze color by assuming that the transmission and surface shading are uncorrelated locally. In our case, this statistical approach will fail in reflection regions where the transmission color is

dominant or when sky without shading is primarily reflected. Methods that use the dark channel prior [HST09, KN09] are based on the hypothesis that at least one color channel tends to be almost zero in natural images. This hypothesis is invalid in regions where a bright sky is reflected as in our case. Most importantly, applying these methods to the situation examined here is nontrivial due to the differences between the targets.

A basic technique for reproducing reflection/refraction is to use ray tracing with three-dimensional geometries. However, several methods do not use geometric information. Environment matting [ZWC99, CZH\*00, WFZ02, PD03] measures the light transport of a reflective or refractive object in the real world by means of specialized devices or multiple photographs. Khan et al. [KRFB06] proposed a method that can convert an object’s material in a photograph such that the object appears to be composed of metal or glass. The input object is assumed to be opaque, and depths in the image are estimated from luminance intensities. In contrast, the target object here is not opaque and its luminance intensity does not provide depth information.

Yeung et al. proposed a matting method for separating glossy reflections and refractions on a glass surface in an image [YTBK11]. In their method, the reflection component is limited to white highlights that are extracted as definitely foreground in a *trimap*, and an alpha matte and warping function of refraction are calculated based on user-specified strokes. We consider their method to be the most relevant to our matting method. However, the reflection component in our case consists of a various colors and definitely foreground regions are not available. Additionally, manual specification of small-scale warping caused by a wavy reflection surface is impractical.

Separating reflection and refraction on a glass surface has been studied in computer vision. While some of these methods take a single image as an input [LZW04, LW07] and others multiple images [BBZZ05, GSZ08], here we give an overview of single image methods. The key concept in these methods is that luminance edges belong to either reflection or refraction components. Nevertheless, an automated approach [LZW04] is unable to achieve adequate results in many cases, and manually specifying edges [LW07] is labor intensive. We do not deal with complicated refractions in our matting method, but target a uniform transmission color as observed in deep sea water or a nontextured tabletop.

### 3. Reflection Matting

We start by defining our reflection matting problem and our assumptions for solving the problem. We then describe the proposed matting algorithm for separating each component of an input image.

#### 3.1. Reflection Model and Assumptions

We target reflections caused by the Fresnel effect, which occurs at an interface between objects with different refractive indices. The light intensity observed at the interface between the objects is a linear blending of the reflected and transmitted light. For an image, we have

$$\mathbf{I}(\mathbf{x}) = \alpha(\mathbf{x})\mathbf{R}(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\mathbf{T}(\mathbf{x}), \quad (1)$$

where  $\mathbf{I}$ ,  $\mathbf{R}$ , and  $\mathbf{T}$  are the three-channel (RGB) input image, reflection component, and transmission component, respectively.  $\alpha \in [0, 1]$  is the Fresnel coefficient and  $\mathbf{x} = (x, y)^T$  is a pixel coordinate.

The Fresnel coefficient is dependent on the incident angle of the viewing ray. At an air/water interface, for example,  $\alpha$  monotonically increases when the incident angle is sufficiently large. Thus, the Fresnel coefficient is small at the near side of the camera and large at the far side, as shown in the alpha mattes in Fig. 1 (each pixel intensity encodes the Fresnel coefficient). Correspondingly, the transmitted color is dominant at the near side and the reflected color is dominant at the far side.

Solving Eq. (1) for a single input image  $\mathbf{I}$  is an ill-posed problem because each pixel has only three known values but seven unknowns. Although the current situation may seem similar to natural image matting [WC08], which decomposes an image into foreground and background components and a matte, or to recent refraction matting techniques [YTBK11], the targets and formulations are different, as described in Section 2. We instead focus on typical cases observed on the surfaces of deep waterbodies, glossy tabletops, or floors, which can be seen in many photographs on the Internet. Specifically, we make the following three assumptions.

1. The transmission component  $\mathbf{T}$  can be approximated as a uniform color.
2. At least one real object and its corresponding reflection can be found in an input image.
3. The reflection surface is predominantly planar but may include waves.

Throughout this paper, we use the term *real object* to describe an object that is the reflection source above the surface, in contrast to its corresponding reflection. Our matting algorithm takes a single photograph with a reflection surface as an input, as well as a user-specified mask for the reflection region and a pair of scribbles for color transfer. We denote the real-object and reflection regions in an input image by  $\Pi$  and  $\Omega$ , respectively. Our algorithm then calculates  $\mathbf{T}$ ,  $\mathbf{R}$ , and  $\alpha$  for  $\Omega$ . Moreover,  $\Pi$  is located above the upper edge of  $\Omega$ 's boundary  $\partial\Omega$ .

On the basis of these assumptions, the matting algorithm proceeds in several steps (see Fig. 1). We estimate  $\mathbf{T}$  first (Section 3.2) because this is assumed to be uniform (i.e., constant) over the entire region and is thus easier to determine than the other components. We next calculate an alpha matte according to the Fresnel equation by using incident angles of the viewing rays based on estimated camera parameters (Section 3.3). We then compute  $\mathbf{R}$  using the data generated in the previous steps (Section 3.3). Note that the reflection surface can be wavy, however, and so we refine  $\mathbf{R}$  and the alpha matte using a proposed filter (Section 3.4). In the following subsections, we describe the details of these steps.

#### 3.2. Estimating Transmission Component $\mathbf{T}$

If the transmission component is uniform, then  $\mathbf{T}$  in Eq. (1) is a constant vector:

$$\mathbf{I}(\mathbf{x}) = \alpha(\mathbf{x})\mathbf{R}(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\mathbf{T}. \quad (2)$$

We thus consider the luminance gradients of Eq. (2):

$$\nabla I(\mathbf{x}) = (R - T)\nabla\alpha(\mathbf{x}) + \alpha(\mathbf{x})\nabla R(\mathbf{x}), \quad (3)$$

where  $I$ ,  $R$ , and  $T$  are the luminance of  $\mathbf{I}$ ,  $\mathbf{R}$ , and  $\mathbf{T}$ , respectively, and  $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^T$  is the gradient operator. If the reflection surface is sufficiently flat, then the alpha matte is smooth because the differences among the incident angles are small. Thus,  $\nabla\alpha$  is negligible. We therefore approximate Eq. (3) as

$$\nabla I(\mathbf{x}) \approx \alpha(\mathbf{x})\nabla R(\mathbf{x}). \quad (4)$$

However, this approximation might be invalid for wavy surfaces since incident angles can vary greatly in this case. To avoid wavy surface effects, we downsample the input image using a Gaussian filter. For example, we scale an input image of  $600 \times 400$  pixels down to 1/4 of its original size.

If we know  $\alpha$  and  $\mathbf{R}$ , we can calculate  $\mathbf{T}$  from Eq. (2) by using only those pixels in a partial region of  $\Omega$  because  $\mathbf{T}$  is

constant. Conversely,  $\alpha$  can be computed by using  $\mathbf{I}$  and  $\mathbf{R}$  through Eq. (4). We therefore seek  $\mathbf{R}$  in  $\Omega$  locally. On the basis of our second assumption,  $\mathbf{R}$  in  $\Omega$  can ideally be obtained by determining the corresponding region in  $\Pi$  and using the real-object colors. However, accurate alignment of the pair is difficult because the colors and shapes in  $\Omega$  are changed compared with those in  $\Pi$  as a result of the Fresnel effect and waves, respectively. An exact match of the pair thus cannot be found, but we can convert the color distribution in  $\Omega$  to be similar to that in  $\Pi$  by using *color transfer* [RAGS01]. We then use the color-transferred  $\mathbf{I} \in \Omega$  as  $\mathbf{R}$ .

Pairwise regions for color transfer are specified through user-provided scribbles  $S_\Pi$  and  $S_\Omega$ . As shown in the ‘‘User markups’’ procedure of the ‘‘Inputs’’ step in Fig. 1, the user specifies a region in  $\Pi$  with a single green scribble  $S_\Pi$ , and the corresponding region in  $\Omega$  with a single red scribble  $S_\Omega$ . The colors in  $S_\Pi$  are then transferred to those in  $S_\Omega$  in order to estimate  $\mathbf{R}$  in  $\Omega$ . Note that we do not require  $\mathbf{R}$  for every pixel in  $\Omega$  during this step, we simply collect samples to estimate  $\mathbf{T}$ .

This color transfer works well.  $\alpha$  can be considered constant in  $S_\Omega$  because the region is vertically short, and so  $\mathbf{I}$  is equal to scaled  $\mathbf{R}$  with a constant offset (Eq. (2)). Explicitly,  $\mathbf{R}$  in  $S_\Omega$  can be obtained by scaling the variance of  $\mathbf{I}$  in  $S_\Pi$  and offsetting the mean. The mean and variance of  $\mathbf{I}$  in  $S_\Omega$  thus conform to those in  $S_\Pi$  through color transfer.

Finally, we calculate  $\mathbf{T}$  using the information obtained above. The  $\mathbf{R}$  pixel samples in  $S_\Omega$  are used to determine  $\alpha$  from Eq. (4) by projecting  $\nabla I(\mathbf{x})$  onto  $\nabla R(\mathbf{x})$ :

$$\tilde{\alpha}(\mathbf{x}) = \frac{\nabla I(\mathbf{x}) \cdot \nabla R(\mathbf{x})}{\|\nabla R(\mathbf{x})\|^2}. \quad (5)$$

Here,  $\tilde{\alpha}$  is an intermediate value to estimate  $\mathbf{T}$ . We omit samples if  $\|\nabla R(\mathbf{x})\|^2 < 0.01$  during this process to reduce errors caused by division. We therefore calculate  $\mathbf{T}$  for each sample through Eq. (2) using  $\tilde{\alpha}$ , and average the values to reduce errors.

### 3.3. Estimating $\alpha$ Matte and Reflection Component $\mathbf{R}$

We obtained  $\alpha$  values in only a partial region  $S_\Omega$  in the previous subsection. Next, we estimate the entire  $\alpha$  matte. The goal in this step is to generate a smooth alpha matte that captures low-frequency variations, and then calculate the corresponding  $\mathbf{R}$ . We again use the downsampled  $\mathbf{I}$  as input here to avoid wave effects. To reliably estimate a smooth alpha matte, we exploit the analytic formulation of the Fresnel coefficient. Specifically, we use the Schlick’s approximation [Sch94] for the Fresnel coefficient:

$$\alpha(\theta) = \alpha_0 + (1 - \alpha_0)(1 - \cos(\theta))^5, \quad (6)$$

and for a planar reflection surface, the incident angle  $\theta$  of a viewing ray is given by

$$\theta(y) = \gamma(y) + \theta_o, \quad \gamma(y) = \arctan\left(\frac{2y}{h} \tan\left(\frac{\phi}{2}\right)\right), \quad (7)$$

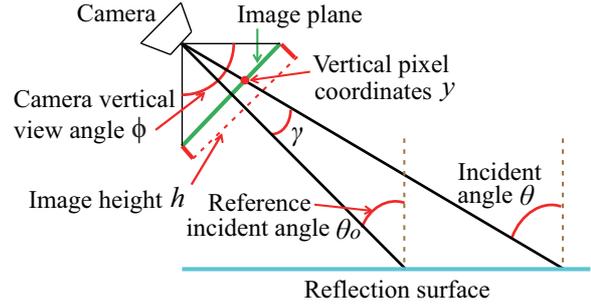


Figure 2: Relation between reflection surface and camera parameters.

where  $h$  is the image height,  $y$  is the vertical pixel coordinate,  $\phi$  is the vertical view angle of the camera, and  $\theta_o$  is the reference incident angle (Fig. 2) for the camera’s gazing direction.  $\alpha_0$  is a constant determined by the refraction index  $n$  of the material:  $\alpha_0 = \frac{n-1}{n+1}$ . In our experiments, we simply use a fixed value for  $n$  according to the material, for instance,  $n = 1.33$  for water. Approximate specification of  $n$  is applicable here because a small variation of  $n$  has little effect on the value output by Eq. (6).

To determine  $\theta$ , we must estimate camera parameters  $\theta_o$  and  $\phi$ . We therefore solve the following minimization problem by using the Levenberg-Marquardt method:

$$\operatorname{argmin}_{\theta_o, \phi} \sum_{\mathbf{p} \in P} \left| \alpha(\mathbf{p}) - \left( \alpha_0 + (1 - \alpha_0)(1 - \cos(\theta(y_p))) \right)^5 \right|^2 \quad (8)$$

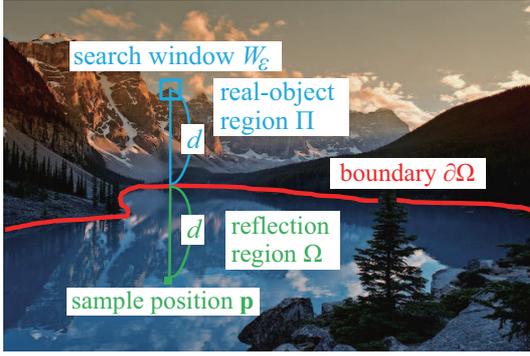
where  $P = \{\mathbf{p} = (x_p, y_p) \mid \mathbf{p} \in \Omega\}$  is a set of training sample positions. Although Eq. (8) is nonlinear, it converges to an appropriate minimum when the initial values are  $\theta_o = 70^\circ$  and  $\phi = 30^\circ$ . Note that  $\phi$  can be also specified from the exchangeable image file format (EXIF) data of the input photograph, and  $n$  can be estimated during optimization of Eq. (8).

We collect samples  $\{\alpha(\mathbf{p}), y_p\}$  throughout  $\Omega$ , and not in a partial region as in the previous subsection, to avoid bias. To compute  $\alpha(\mathbf{p})$  in each training sample, we search for an appropriate reflection-color sample in  $\Pi$ . Let  $\mathbf{R}(\mathbf{q})$  ( $\mathbf{q} \in \Pi$ ) be the color of a reflection-color sample. We seek the optimal  $\mathbf{R}(\mathbf{q})$  based on the following metric  $\varepsilon$ :

$$\varepsilon(\mathbf{R}(\mathbf{q}), \mathbf{T}) = \|\mathbf{I}(\mathbf{p}) - (\hat{\alpha}(\mathbf{p})\mathbf{R}(\mathbf{q}) + (1 - \hat{\alpha}(\mathbf{p}))\mathbf{T})\|, \quad (9)$$

$$\hat{\alpha}(\mathbf{p}) = \frac{(\mathbf{I}(\mathbf{p}) - \mathbf{T}) \cdot (\mathbf{R}(\mathbf{q}) - \mathbf{T})}{\|\mathbf{R}(\mathbf{q}) - \mathbf{T}\|^2}, \quad (10)$$

where the value of  $\hat{\alpha}(\mathbf{p})$  that minimizes  $\varepsilon$  is set as  $\alpha(\mathbf{p})$ . Eqs. (9) and (10) are derived from Eq. (2).  $\varepsilon$  measures the difference between  $\mathbf{I}(\mathbf{p})$  and the linear blending of  $\mathbf{R}(\mathbf{q})$  and  $\mathbf{T}$ . Intuitively,  $\varepsilon$  represents the distance between  $\mathbf{I}(\mathbf{p})$  and a line segment connecting  $\mathbf{R}(\mathbf{q})$  and  $\mathbf{T}$  in RGB color space. A smaller value of  $\varepsilon$  hence signifies that the color of the sample is better matched. This metric is motivated by *robust matting* [WC07], where background and foreground samples are



**Figure 3:** To calculate  $\alpha(\mathbf{p})$  in each training sample, we search for the pixel with minimal  $\varepsilon$  within  $W_\varepsilon$  (blue square; enlarged for illustrative purposes), and use it as sample  $\mathbf{R}(\mathbf{q})$ . The center of  $W_\varepsilon$  is at a vertical  $d$  from  $\partial\Omega$ .

collected based on a similar metric of natural image matting. Note that we do not use the map  $\hat{\alpha}$  as the alpha matte because it may contain substantial noise and errors.

To efficiently search for appropriate samples  $\mathbf{R}(\mathbf{q})$  in  $\Pi$ , we limit the search space such that the corresponding real object is likely to be found. Let  $d$  be the vertical distance between the pixel under examination and the boundary  $\partial\Omega$ , as shown in Fig. 3. We choose a sample with minimal  $\varepsilon$  in a search window  $W_\varepsilon$  (we use a  $7 \times 7$  window in our experiments) centered at the position vertically symmetric to  $\mathbf{p}$ :

$$\mathbf{q} = \underset{\tilde{\mathbf{q}} \in W_\varepsilon}{\operatorname{argmin}} \varepsilon(\mathbf{R}(\tilde{\mathbf{q}}), \mathbf{T}). \quad (11)$$

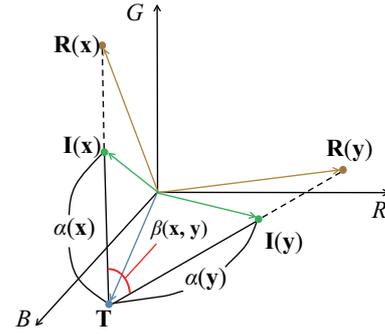
If  $d$  is sufficiently large that the search window protrudes outside of the image, we simply ignore the sample at  $\mathbf{p}$ . To precisely estimate the camera parameters, we collect 20% of the pixels in  $\Omega$  that minimize  $\varepsilon$ .

In summary, we calculate  $\alpha$  and  $\mathbf{R}$  as follows. We first estimate camera parameters through Eq. (8) and training samples based on Eqs. (9) and (10). We then obtain a smooth alpha matte for  $\Omega$  by using the estimated camera parameters and Eq. (6), and calculate  $\mathbf{R}$  by Eq. (2). We modify  $\alpha$  and  $\mathbf{R}$  in the next subsection to account for wavy reflection surfaces.

### 3.4. Modified $\mathbf{R}$ and $\alpha$

Here, we modify  $\mathbf{R}$  and  $\alpha$  to account for wavy surfaces, since the alpha matte must contain high-frequency regions because of the varying incident angles of the viewing rays. However, the alpha matte obtained in the previous subsection is smooth, and the current  $\mathbf{R}$  contains high-frequency residuals.

To solve these issues, we smooth the residuals in  $\mathbf{R}$ , and then compute  $\alpha$  by Eq. (2) using the smoothed  $\mathbf{R}$ . Note



**Figure 4:** Geometric interpretation of the proposed filter. The degree of smoothing by the filter is dependent on  $\beta$ , which indicates the angle between  $\mathbf{I}(\mathbf{x}) - \mathbf{T}$  and  $\mathbf{I}(\mathbf{y}) - \mathbf{T}$  in RGB color space, and can detect color variation of the true  $\mathbf{R}$ . Our filter with  $\beta$  can smooth waves caused only by variation of  $\alpha$ .

that we must preserve the inherent edges of the reflection and smooth only the residuals during this process. However, typical edge-preserving filters, including the bilateral filter [TM98], are not optimal because they preserve edges of not only the true  $\mathbf{R}$  (i.e., its exact form) but also high-frequency residuals, which should be smoothed. Therefore, we design a new filter as a variant of the bilateral filter, and apply it to  $\mathbf{R}$  as follows:

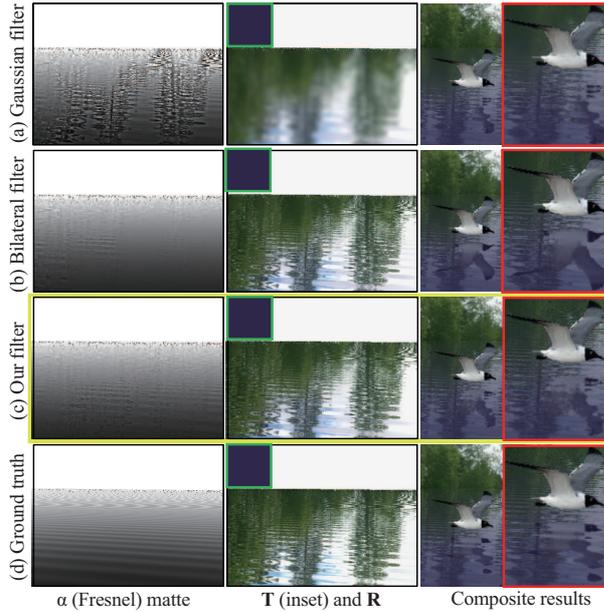
$$\mathbf{R}^*(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in W_x} G(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y}) \mathbf{R}(\mathbf{y})}{\sum_{\mathbf{y} \in W_x} G(\mathbf{x}, \mathbf{y}) w(\mathbf{x}, \mathbf{y})}, \quad (12)$$

$$w(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{(\beta(\mathbf{x}, \mathbf{y}))^2}{\sigma}\right), \quad (13)$$

$$\beta(\mathbf{x}, \mathbf{y}) = \arccos\left(\frac{(\mathbf{I}(\mathbf{x}) - \mathbf{T}) \cdot (\mathbf{I}(\mathbf{y}) - \mathbf{T})}{\|\mathbf{I}(\mathbf{x}) - \mathbf{T}\| \|\mathbf{I}(\mathbf{y}) - \mathbf{T}\|}\right). \quad (14)$$

Here,  $\mathbf{R}^*$  is the filtered  $\mathbf{R}$ ,  $W_x$  is a set of pixels in the  $K \times K$  filter kernel (with kernel size  $K$ ) centered at  $\mathbf{x}$ ,  $G$  is the spatial Gaussian weight,  $w$  is a proposed weight function, and  $\sigma$  is the variance of  $w$ . Our weight function  $w$  can detect the variability between the inherent colors of neighboring pixels in the reflection. If the difference between these colors is small, then our filter increases the smoothing of  $\mathbf{R}$ . We set  $\sigma = 0.01$  for all experiments, whereas the kernel size is adjusted for each input image according to the image-space wavelength of the wavy surface in  $\Omega$ .

We explain  $w(\mathbf{x}, \mathbf{y})$  of our filter geometrically using Fig. 4. Intuitively,  $\beta(\mathbf{x}, \mathbf{y})$  in  $w(\mathbf{x}, \mathbf{y})$  is the angle between vectors  $\mathbf{I}(\mathbf{x}) - \mathbf{T}$  and  $\mathbf{I}(\mathbf{y}) - \mathbf{T}$ . Note that we should smooth  $\mathbf{R}$  according to its true form, but we do not know the exact values of its elements. Instead, we know  $\mathbf{I}$  and  $\mathbf{T}$ , and  $\mathbf{I} - \mathbf{T}$  has the same orientation as  $\mathbf{R} - \mathbf{T}$  because  $\mathbf{I}(\mathbf{x}) - \mathbf{T} = \alpha(\mathbf{x})(\mathbf{R}(\mathbf{x}) - \mathbf{T})$  by Eq. (2). Therefore,  $\beta(\mathbf{x}, \mathbf{y})$  detects the difference between the true  $\mathbf{R}$  of  $\mathbf{x}$  and  $\mathbf{y}$  through  $\mathbf{I} - \mathbf{T}$ , and  $\beta(\mathbf{x}, \mathbf{y})$  is independent



**Figure 5:** Comparison among the results of our filter, the result of conventional filters, and ground-truth data. Our filter yields the most accurate alpha matte and most plausible reflection image in the composite result.

of  $\alpha$ . Consequently, our filter can preserve the edges of the true  $\mathbf{R}$  and smooth the residuals in the current  $\mathbf{R}$  caused by changes in  $\alpha$ .

Figure 5 compares the results of our filter with those of conventional Gaussian and bilateral filters. We use the input image in Fig. 1, which is synthesized by ray tracing with an environment map and Stokes waves. We also compare our results with ground-truth data of this synthesized input image. We set the kernel sizes of all the filters to  $27 \times 27$  because of the relatively wide waves in the input.

In the Gaussian filter results (Fig. 5(a)), the filtered  $\mathbf{R}$  is too smooth overall and  $\alpha$  includes undesired edges of the reflected objects. These errors cause an inappropriate composite result where the reflections of the trees are blurred. The bilateral filter successfully eliminates the edges of reflected objects in  $\alpha$  (Fig. 5(b)). However, it fails to smooth waves in  $\mathbf{R}$ , and thus fails to capture the waves in  $\alpha$ . As a result, the reflected image of the gull in the composite result is unnaturally flat. Our filter effectively improves the above-mentioned errors (Fig. 5(c)). Specifically, waves in the true  $\alpha$  are extracted in our results, while the reflection edges are preserved similarly to the bilateral filter. Compared with the ground-truth data (Fig. 5(d)), our results extract each component more accurately in all cases.

Finally, Algorithm 1 summarizes our matting algorithm.

---

#### Algorithm 1 Reflection matting.

---

**Input:**  $\mathbf{I}$ , reflection region  $\Omega$  and pairwise scribbles  $\{S_{\Pi}, S_{\Omega}\}$

**Output:**  $\mathbf{T}$ ,  $\mathbf{R}$  and  $\alpha$

$\mathbf{I}_{ds} \leftarrow \text{DOWNSAMPLING}(\mathbf{I});$

$\mathbf{R}_{\Omega} \leftarrow \text{COLORTRANSFER}(\mathbf{I}_{ds}, S_{\Pi}, S_{\Omega}); // \text{Sec. 3.2}$

$\mathbf{T} \leftarrow \text{CALCTRANSMISSIONCOLOR}(\mathbf{I}_{ds}, \mathbf{R}_{\Omega}); // \text{Sec. 3.2}$

$\{\alpha(\mathbf{p}), y_p\} \leftarrow \text{COLLECTSAMPLES}(\mathbf{I}_{ds}, \mathbf{T}, \Omega); // \text{Sec. 3.3}$

$(\theta_o, \phi) \leftarrow \text{CALCCAMERAPARAM}(\{\alpha(\mathbf{p}), y_p\}); // \text{Sec. 3.3}$

$\alpha \leftarrow \text{CALCALPHAMATTE}(\theta_o, \phi, \Omega); // \text{Sec. 3.3}$

**for each**  $\mathbf{x} \in \Omega$  **do**

$\mathbf{R}(\mathbf{x}) \leftarrow \frac{\mathbf{I}(\mathbf{x}) - (1 - \alpha(\mathbf{x}))\mathbf{T}}{\alpha(\mathbf{x})};$

**end for**

**if** reflection region  $\Omega$  is wavy **then**

$\mathbf{R} \leftarrow \text{APPLYSMOOTHINGFILTER}(\mathbf{R}, \Omega); // \text{Sec. 3.4}$

**for each**  $\mathbf{x} \in \Omega$  **do**

$\alpha(\mathbf{x}) \leftarrow \frac{(\mathbf{I} - \mathbf{T}) \cdot (\mathbf{R}(\mathbf{x}) - \mathbf{T})}{\|\mathbf{R}(\mathbf{x}) - \mathbf{T}\|^2};$

**end for**

**end if**

---

#### 4. Reflection Composition

After decomposing an input image, we can generate the reflections of newly composited objects by using the calculated components and camera parameters. In the compositing step, a new reflection component  $\mathbf{R}_{new}$  for a composited object is synthesized to create the final composite result  $\mathbf{I}_{new}$ :

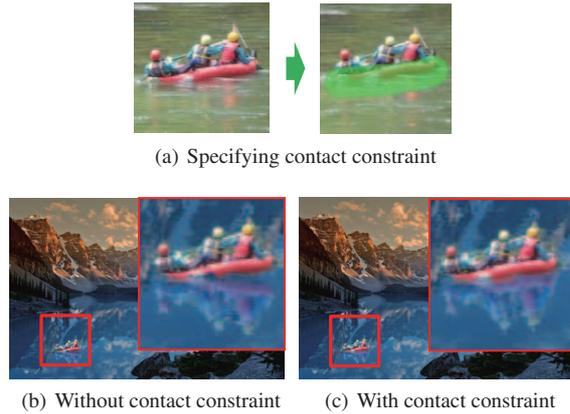
$$\mathbf{I}_{new}(\mathbf{x}) = \alpha(\mathbf{x})\mathbf{R}_{new}(\mathbf{x}) + (1 - \alpha(\mathbf{x}))\mathbf{T}. \quad (15)$$

We employ ray tracing to compute  $\mathbf{R}_{new}$ . Composited objects are represented as billboards that are facing the camera and in contact with the reflection surface. The incident angles  $\theta$  of the viewing rays can be estimated according to Eq. (6) as follows:

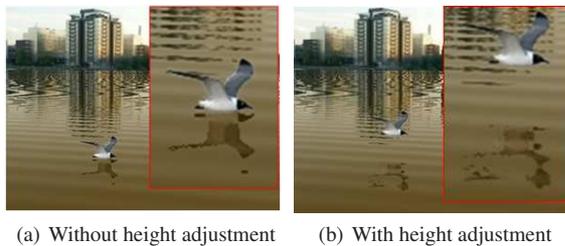
$$\theta(\mathbf{x}) = \arccos\left(1 - \left(\frac{\alpha(\mathbf{x}) - \alpha_0}{1 - \alpha_0}\right)^{\frac{1}{5}}\right), \quad (16)$$

where  $\mathbf{x}$  is the pixel position passed through by a viewing ray. Note that naively applying ray tracing causes aliasing in reflection images, especially around object silhouettes; a reflected image in a photograph must be slightly blurred because of subpixel-scale reflection due to small waves on a surface. Therefore, we apply prefiltering using a Gaussian filter when retrieving the colors of real objects.

In addition, we implemented two operations to increase the flexibility of compositing. The first operation is a contact constraint. If an object faces an oblique direction, a gap can be created between the object's bottom and the reflection surface (Fig. 6(b)). To correct this, the user can provide a scribble to roughly specify where the contact should be (Fig. 6(c)), and our system then extracts the object's bottom from within the scribble. The second operation is a height adjustment. The user can adjust the height of objects from the reflection surface using the mouse wheel. This operation



**Figure 6: Contact constraint.** (a) A user can specify an object's bottom using a green scribble, without this specification (b) the object does not make contact with the reflection surface appropriately, whereas with this specification (c) the contact is appropriate.



**Figure 7: Height adjustment.** A user can adjust the height of flying objects (for example, from (a) to (b)) using the mouse wheel.

enables us to calculate the reflection of flying objects, such as birds in the air (Fig. 7(b)).

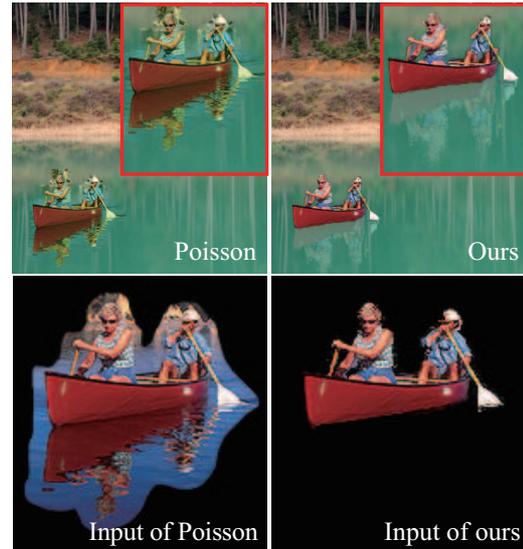
## 5. Experimental Results and Discussion

We implemented our prototype system as a single-threaded C++ program, and conducted experiments on a personal computer with an Intel i7 processor running at 2.8 GHz and with 8 GB RAM. The statistics of our matting results are summarized in Table 1. Compositing is performed in real time, as demonstrated in the video accompanying the online version of this article.

Figure 8 shows several results of our method when natural images with water surfaces are used as inputs. Reflections are appropriately generated according to the Fresnel effect. That is, reflection colors on the surface are dominant at the far side of the viewing point, whereas dark water colors are dominant at the near side. Warping by waves is also successfully handled, especially in Figs. 8(d) and (f).

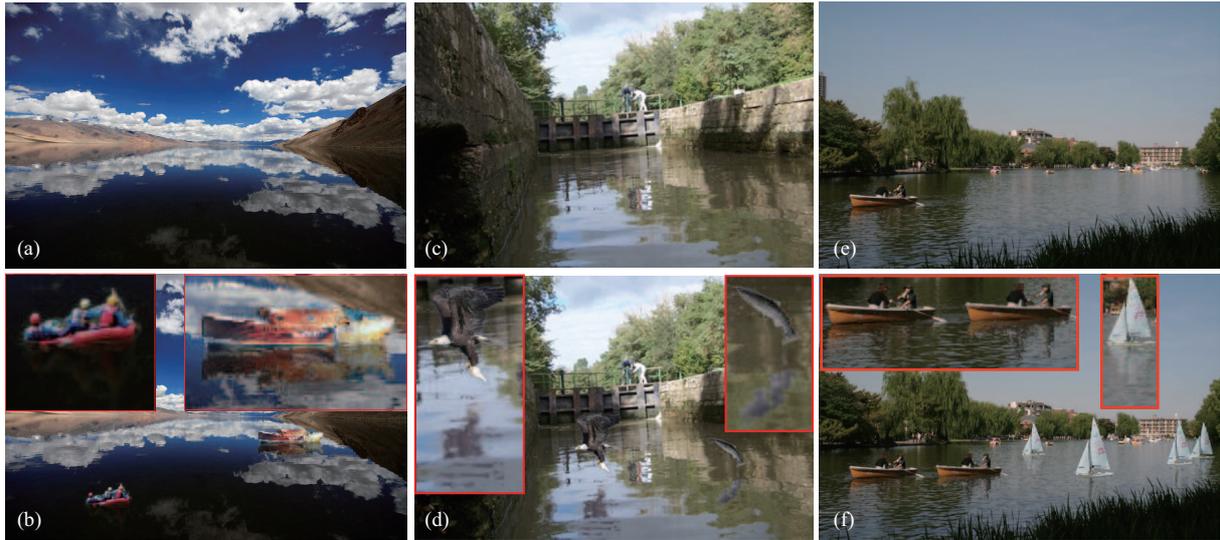
**Table 1: Computational times (s) of matting results for different image sizes (pixel) and filter kernel sizes (pixel).**

Image	Image size	Kernel size	Time
Fig.1	400 × 400	27 × 27	7.92
Fig.8(b)	640 × 428	5 × 5	0.896
Fig.8(d)	640 × 450	27 × 27	7.68
Fig.8(f)	600 × 400	13 × 13	1.40
Fig.10(b)	620 × 403	5 × 5	0.801
Fig.10(d)	615 × 413	(w/o filter)	0.532
Fig.10(f)	420 × 634	13 × 13	1.37

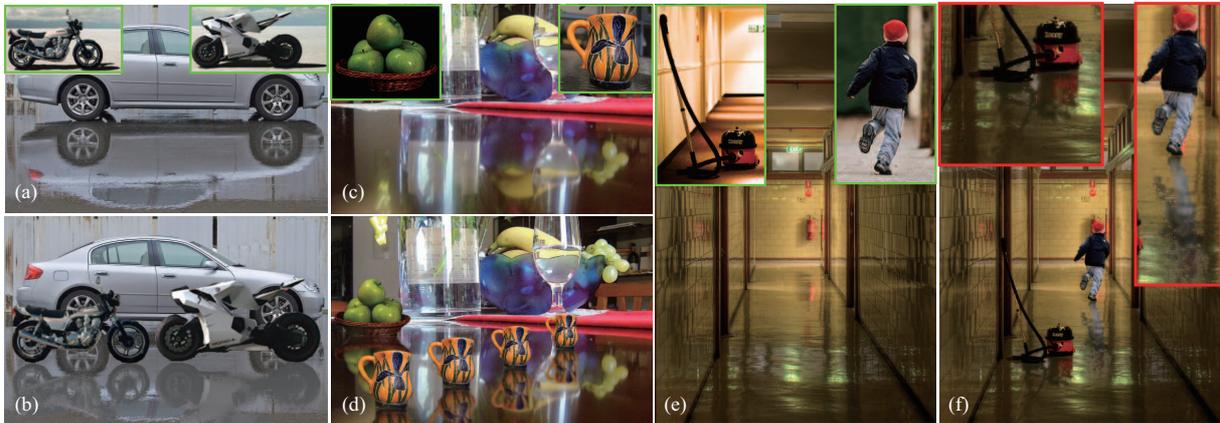


**Figure 9: Comparison between the results obtained by Poisson Image Editing [PGB03] and by our method.** While Poisson Image Editing requires a real reflection image and can produce unnatural results due to waves of different wavelength or different camera angles, our method can synthesize plausible reflections even without a real reflection image.

The left inset in Fig. 8(f) shows a comparison between our results and a real reflection in the photograph. We extracted the object producing the real reflection in the input image and synthesized a copy next to the original for a side-by-side comparison. Our result is visually plausible, but the shape of the reflection is subtly different from real one because of our billboard approximation. Such a composite might also be possible by using *Poisson Image Editing* [PGB03] if a real reflection image is available. However, that method can produce unnatural reflections for images with waves of different wavelengths or different camera angles (Fig. 9). We emphasize here that Poisson Image Editing cannot generate a reflection without a real reflection image, whereas our method can synthesize a plausible reflection even if a real reflection image does not exist.



**Figure 8:** Our synthetic results for photographs with water surfaces. Each inset shows a magnified section of the image.



**Figure 10:** Our method can handle reflections on (a) and (b) wet ground and off-specular reflections on glossy surfaces including (c) and (d) tabletops and (e) and (f) floors. Note that original objects (insets in (a), (c), and (e)) do not have reflection images.

Our method is also applicable for reflections on other surfaces, such as wet ground (Figs. 10(a) and (b)), tabletops (Figs. 10(c) and (d)), and glossy floors (Figs. 10(e) and (f)). Although the cup has a fixed size in Fig. 10(c), the relative image sizes of the cups in Fig. 10(d) are automatically adjusted according to the scene depths determined by the estimated camera parameters (Section 3.3). In the other two examples, we also consider warping by wavy reflection surfaces. Note that our method can produce plausible reflection images even though the original objects (insets in Figs. 10(a), (c), and (e)) do not have reflection images.

**Limitation of our method.** The billboard approximation used in ray tracing can result in reflections with an unnat-

ural appearance. The tops of the tea cups are rendered as reflection images in Fig. 11; but, in reality, these would not be seen. This artifact should be alleviated by incorporating the geometries of the real objects.

## 6. Conclusion and Future Work

We have presented a method of matting and compositing for reflections in images. Our system enables a user to easily edit a reflection through only a few simple interactions. Under several assumptions, our matting algorithm solves the reflection matting problem that has been shown to be intractable with existing image matting techniques. Despite the simpli-



**Figure 11:** Limitation of our method. Tops of the tea cups are rendered in reflections (e.g., as marked by the red oval) because of the billboard approximation used in ray tracing. This is physically incorrect.

fications resulting from our assumptions, we have demonstrated that the proposed method generates visually plausible composited reflections.

Our work is the first to produce mattes for complicated reflections, and we expect it to open new avenues of research, including the handling of highly complex (e.g., curved) reflection surfaces and real-object geometries, and the separation of transmission components for images with shallow water surfaces (i.e., a pond, river, or pool) where intricate bed patterns will be visible.

### Acknowledgements

We would like to thank Daisuke Miyazaki and Yosuke Bando for their valuable comments. We also thank anonymous reviewers for their insightful suggestions. This work was supported by Grant-in-Aid for Young Scientists (B) 23700101.

### References

- [BBZZ05] BRONSTEIN A. M., BRONSTEIN M. M., ZIBULEVSKY M., ZEEVI Y. Y.: Sparse ICA for blind separation of transmitted and reflected images. *International Journal of Imaging Systems and Technology* 15, 1 (2005), 84–91. 3
- [CGC\*03] CHUANG Y.-Y., GOLDMAN D. B., CURLESS B., SALESIN D. H., SZELISKI R.: Shadow matting and compositing. *ACM Trans. Graph.* 22, 3 (2003), 494–500. 1
- [CGZ\*05] CHUANG Y.-Y., GOLDMAN D. B., ZHENG K. C., CURLESS B., SALESIN D. H., SZELISKI R.: Animating pictures with stochastic motion textures. In *ACM SIGGRAPH 2005 Papers* (2005), SIGGRAPH '05, pp. 853–860. 1
- [CZH\*00] CHUANG Y.-Y., ZONGKER D. E., HINDORFF J., CURLESS B., SALESIN D. H., SZELISKI R.: Environment matting extensions: Towards higher accuracy and real-time capture. In *Siggraph 2000, Computer Graphics Proceedings* (2000). 2
- [Fat08] FATTAL R.: Single image dehazing. In *ACM SIGGRAPH 2008 papers* (2008), SIGGRAPH '08, pp. 72:1–72:9. 1, 2
- [GSZ08] GAI K., SHI Z. W., ZHANG C. S.: Blindly separating mixtures of multiple layers with spatial shifts. In *CVPR* (2008), pp. 1–8. 3

- [HST09] HE K. M., SUN J., TANG X.: Single image haze removal using dark channel prior. In *CVPR* (2009), pp. 1956–1963. 1, 2
- [KN09] KRATZ L., NISHINO K.: Factorizing scene albedo and depth from a single foggy image. In *ICCV* (2009), IEEE, pp. 1701–1708. 1, 2
- [KRFB06] KHAN E. A., REINHARD E., FLEMING R., BUELTHOFF H.: Image-based material editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 25, 3 (2006). 2
- [LW07] LEVIN A., WEISS Y.: User assisted separation of reflections from a single image using a sparsity prior. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29, 9 (2007), 1647–1654. 3
- [LZW04] LEVIN A., ZOMET A., WEISS Y.: Separating reflections from a single image using local features. In *CVPR (1)* (2004), pp. 306–313. 3
- [PD03] PEERS P., DUTRE P.: Wavelet environment matting. In *Proceedings of the 14th Eurographics Workshop on Rendering* (25–27 2003), pp. 157–166. 2
- [PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics* 22, 3 (2003), 313–318. 7
- [PHCS06] PARK J., HEO N., CHOI S., SHIN S. Y.: Tour into the picture with water surface reflection and object movements: Research articles. *Comput. Animat. Virtual Worlds* 17, 3-4 (2006), 315–324. 1
- [RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications* 21, 5 (2001), 34–41. 1, 4
- [RRW\*09] RHEMANN C., ROTHER C., WANG J., GELAUTZ M., KOHLI P., ROTT P.: A perceptually motivated online benchmark for image matting. In *In Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)* (2009). 2
- [Sch94] SCHLICK C.: An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum* 13, 3 (1994), 233–246. 4
- [Tan08] TAN R. T.: Visibility in bad weather from a single image. In *CVPR* (2008), pp. 1–8. 1, 2
- [TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *ICCV* (1998), pp. 839–846. 5
- [WC07] WANG J., COHEN M. F.: Optimized color sampling for robust matting. In *CVPR* (2007), pp. 1–8. 4
- [WC08] WANG J., COHEN M. F.: Image and video matting: a survey. In *Foundations and Trends in Computer Graphics and Vision* (2008), vol. 3. 2, 3
- [WFZ02] WEXLER Y., FITZGIBBON A., ZISSERMAN A.: Image-based environment matting. In *Proceedings of the 13th Eurographics Workshop on Rendering (RENDERING TECHNIQUES-02)* (26–28 2002), pp. 279–290. 2
- [WTBS07] WU T.-P., TANG C.-K., BROWN M. S., SHUM H.-Y.: Natural shadow matting. *ACM Trans. Graph* 26, 2 (2007). 1, 2
- [YTBK11] YEUNG S.-K., TANG C.-K., BROWN M. S., KANG S. B.: Matting and compositing of transparent and refractive objects. *ACM Transactions on Graphics* 30, 1 (2011), 2:1–2:13. 1, 2, 3
- [ZWCS99] ZONGKER D. E., WERNER D. M., CURLESS B., SALESIN D.: Environment matting and compositing. In *SIGGRAPH* (1999), pp. 205–214. 2