Extracting Depth and Matte using a Color-Filtered Aperture

Yosuke Bando*,‡
TOSHIBA Corporation
The University of Tokyo

Bing-Yu Chen†
National Taiwan University

Tomoyuki Nishita‡
The University of Tokyo

(a) (b) (c) (d) (e)

Figure 1: *(a) Top: camera lens with color filters placed in the aperture. Bottom: filter arrangement. (b) Captured image. The background color is misaligned (see Fig. 16(c) for a closeup). (c) Estimated depth (the darker, the nearer). (d) Extracted matte. (e) Composite image.*

Abstract

This paper presents a method for automatically extracting a scene depth map and the alpha matte of a foreground object by capturing a scene through RGB color filters placed in the camera lens aperture. By dividing the aperture into three regions through which only light in one of the RGB color bands can pass, we can acquire three shifted views of a scene in the RGB planes of an image in a single exposure. In other words, a captured image has depth-dependent color misalignment. We develop a color alignment measure to estimate disparities between the RGB planes for depth reconstruction. We also exploit color misalignment cues in our matting algorithm in order to disambiguate between the foreground and background regions even where their colors are similar. Based on the extracted depth and matte, the color misalignment in the captured image can be canceled, and various image editing operations can be applied to the reconstructed image, including novel view synthesis, post-exposure refocusing, and composition over different backgrounds.

Keywords: computational photography, computational camera, color filters, color correlation, depth estimation, alpha matting

1 Introduction

Rapid progress in the field of computational photography has brought forth new types of cameras and imaging systems capable of capturing additional scene properties that conventional photography misses. These properties, when combined with *computation*, extend the ability of imaging applications in many ways: increased dynamic range and resolution, depth-guided editing, post-exposure refocusing, variable lighting and reflectance, to name but a few.

*e-mail: yosuke1.bando@toshiba.co.jp
†e-mail: robin@ntu.edu.tw
‡e-mail: {ybando, nis}@is.s.u-tokyo.ac.jp

While elaborate imaging systems and optical elements continue to emerge, one of the recent trends in this field is to make a system compact, or even portable [Ng et al. 2005; Georgeiv et al. 2006], and to simplify optical elements to be attached to the conventional camera [Levin et al. 2007; Veeraraghavan et al. 2007]. The ability to easily switch from being a *computational camera* to the conventional one to capture regular photographs is also claimed as an advantage [Green et al. 2007; Liang et al. 2008]. This trend will serve as a driving force for making computational photography more commonplace and affordable for ordinary users.

To boost this trend, this paper proposes a method for automatically extracting a scene depth map and the alpha matte of a foreground object with a conventional camera body and a slightly modified camera lens with RGB color filters placed in the aperture. By dividing the aperture into three regions through which only light in one of the RGB color bands can pass, we can acquire three shifted views of a scene in the RGB planes of a captured image in a single exposure, which enables depth reconstruction. While this idea has already been proposed previously [Amari and Adelson 1992; Chang et al. 2002], we realize this idea in a hand-held camera without the need for additional equipment other than color filters. We also devise a better correspondence measure between the RGB planes which are recorded with different bands of wavelength. Moreover, we propose a method for extracting the matte of an in-focus foreground object, which is an entirely novel application of a color-filtered aperture. Color misalignment cues introduced by the filters serve to constrain the space of possible mattes that would otherwise contain erroneous mattes when foreground and background colors are similar.

The downsides of using a color-filtered aperture are that objects having only a single pure R, G, or B color cannot be handled, and that the visual quality of images is spoiled by color misalignment. We will show, however, that our method can handle many real-world objects, and we also present how to reconstruct color-aligned images using extracted depth and matte. By showing results for outdoor scenes and/or hairy foreground objects, we demonstrate the portability of our device and the effectiveness of our method, with several image editing examples such as novel view synthesis, post-exposure refocusing, and composition over different backgrounds.

2 Related Work

This section reviews several research areas that are closely related to our work. Readers can refer to [Raskar et al. 2006] for an extensive survey on computational photography.

Color-filtered aperture. The idea of using color filters in the aperture to estimate depth has been proposed previously. Amari and Adelson [1992] used a squared intensity difference measure for high-pass filtered images to estimate disparities. As they discussed in their paper, however, this measure was insufficient to compensate for intensity differences between the color planes. Their prototype was not portable, and only a single result for a textured planar surface was shown. Chang et al. [2002] normalized the intensities within a local window in each color plane before taking the sum of absolute differences between them. But as their camera was equipped with a flashbulb for projecting a speckle pattern onto the scene in order to generate strong edges in all the color planes, the performance of their correspondence measure in the absence of a flash was not shown. They also had to capture another image without flash to obtain a "normal" image. We propose a better correspondence measure between the color planes. We believe our matting method based on a color-filtered aperture is entirely new.

Coded aperture. Several researchers placed a patterned mask in the aperture to change the frequency characteristics of defocus blur to facilitate blur identification/removal and depth estimation [Levin et al. 2007; Veeraraghavan et al. 2007]. These methods offered portable imaging systems with minimal modifications to the conventional camera, which inspired us to pursue this direction. Our approach differs in that it relies on parallax cues rather than defocus cues, which introduces a view correspondence problem but escapes ambiguity between depths farther and nearer than the focused depth. We also propose a parallax-based matting method.

Single-lens multi-view image capture. Adelson and Wang [1992] showed that light rays entering a camera can be captured separately depending on their incident angle by placing a microlens array on the image sensor, and they estimated depth from multi-view images captured through a single main lens. Ng et al. [2005] realized this idea in a hand-held camera, and proposed a post-exposure refocusing method by noting that the captured multi-view images correspond to the light field inside the camera [Ng 2005]. Multi-view images can also be captured by placing an attenuation mask on the image sensor [Veeraraghavan et al. 2007], or by splitting light rays at the aperture [Green et al. 2007; Liang et al. 2008] or outside the main lens [Georgeiv et al. 2006]. Our method also splits light rays at the aperture, but requires only color filters as additional optical elements to the lens without requiring multiple exposures. Although this comes with a price of a reduced number of views (only three) each having only a single color plane, we can still obtain useful information for post-exposure manipulation of images.

Matting. In image editing, matting is an important technique for extracting foreground objects in an image so that they can be composited over other images. We only review some of the most relevant work to ours here. Interested readers can refer to Chuang's thesis [2004] for more information. The traditional approach to matting is to use a blue or green screen as a background [Vlahos 1971; Smith and Blinn 1996]. Extracting a matte from a single *natural* image (i.e., an image with general unknown background colors) requires user intervention, a typical form of which is a *trimap* that segments an image into "strictly foreground," "strictly background," and "unknown" regions. Fractional alpha values are computed in the "unknown" region based on the information from the other two regions [Chuang et al. 2001; Levin et al. 2008; Wang and Cohen 2007]. To automate matting, previous approaches used multiple images. Smith and Blinn [1996] captured images of a foreground object with two different known background colors. Alternatively, Wexler et al. [2002] used a sequence of images of a translating/rotating object. Xiong and Jia [2007] captured images from two viewpoints, and computed their stereo correspondences taking into account alpha values of a foreground object. Several methods used synchronized cameras to capture multiple images of an object [McGuire et al. 2005; McGuire et al. 2006; Joshi et al. 2006]. Our method can automatically extract alpha mattes with a single hand-held camera in a single exposure.

3 Color-Filtered Aperture

Fig. 1(a) shows our prototype camera lens with color filters in the aperture. We arranged the RGB regions so that their displacement with respect to the optical center of the lens aligns with the X and Y axes of the image sensor, as indicated by the arrows in Fig. 1(a) bottom. By this arrangement, a scene point farther than the focused depth is observed with a rightward shift in the R plane, an upward shift in the G plane, and a leftward shift in the B plane. A scene point nearer than the focused depth will be shifted in the opposite directions. Note that these color shifts come from geometric optics, not from chromatic aberration. Fig. 2 illustrates this phenomenon in 2D where the aperture is split into two (R and G) regions.

For a prototype camera lens, we cut out a disc with a triple-square-shaped hole from a piece of black cardboard, glued color filters (Fujifilter SC-58, BPB-53, and BPB-45) to it, and attached it immediately in front of the aperture diaphragm of a Canon EF 50mm f/1.8 II lens. This fabrication was done in a few hours with a box cutter and a micro screwdriver. We used an unmodified Canon EOS40D DSLR as a camera body. Fig. 3 shows the *point-spread function* (PSF) of the prototype lens, which is an image of a defocused point light source. The square shape of each filter is observed mostly as-is, with only slightly rounded corners at the horizontal extremities due to occlusion by the lens housing. Fig. 3 also shows that the three color bands are well separated. We achieved this by applying a linear transform to RGB sensor response so as to minimize crosstalk between the aperture filters and the image sensor (see Appendix A for details).

To align the RGB regions with the image sensor axes, manual adjustment was sufficient. Once this is done, pixel disparities will always align with the X and Y axes of captured images, requiring no further calibration and rectification at capture time or during post-processing. Fig. 4 shows an example photograph and its separated RGB planes. Due to the higher transmittance of the R filter, captured images shown in this paper are relatively reddish.

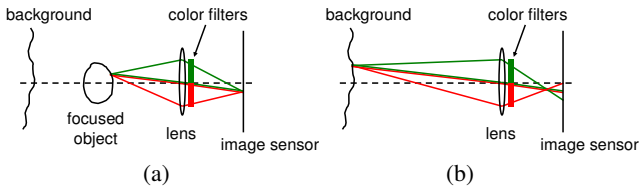


Figure 2: *2D illustration of the interactions between light rays from a scene point and a color-filtered aperture. (a) For a scene point at the focused depth, light rays in the R band and those in the G band converge to the same point on the image sensor. (b) For a scene point off the focused depth, light rays in the two bands reach different positions on the image sensor, resulting in a color shift.*

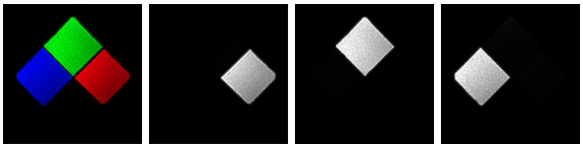


Figure 3: *Point-spread function of our lens and its RGB components. The positions of the R and B regions are opposite to those in Fig. 1(a), as the viewpoint is at the opposite side of the filters.*

Figure 4: *Example photograph taken with our lens, and its separated RGB planes. The white lines are superimposed to highlight the background color shifts. See Fig. 15(a) for a closeup view.*

4 Depth Estimation

The RGB planes I_r, I_g, and I_b of a captured image $\mathbf{I}$ correspond to three views of a scene. If we take a virtual center view (*cyclopean view*) as a reference coordinate system, the R, G, and B planes are shifted to rightward, upward, and leftward according to the arrangement of the aperture color filters. Therefore, letting d be a hypothesized disparity at (x, y), we need to measure the quality of a match between $I_r(x+d, y)$, $I_g(x, y-d)$, and $I_b(x-d, y)$.

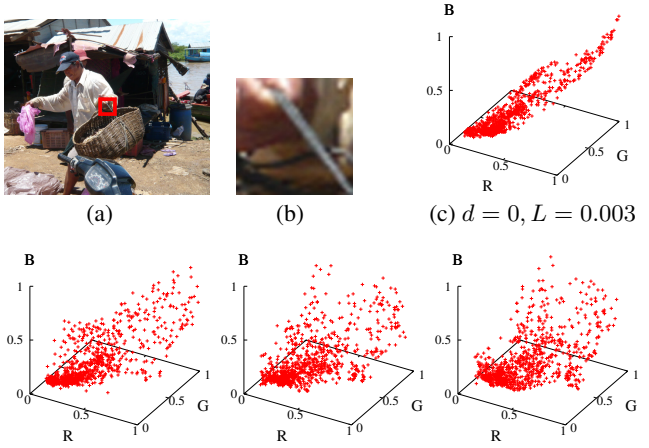
Clearly, we cannot expect these three values to have similar intensities because they are recorded with different bands of wavelength. To cope with this issue, inspired by Levin et al.'s matting approach [2008], we exploit the tendency of colors in natural images to form elongated clusters in the RGB space (*color lines model*) [Omer and Werman 2004]. We assume that pixel colors within a local window $w(x, y)$ around (x, y) belong to one cluster, and we use the magnitude of the cluster's elongation as a correspondence measure. More specifically, we consider a set $S_I(x, y; d)$ of pixel colors with hypothesized disparity d as $S_I(x, y; d) = \{(I_r(s+d, t), I_g(s, t-d), I_b(s-d, t)) \mid (s, t) \in w(x, y)\}$, and search for d that minimizes the following *color alignment measure*:

$$L(x, y; d) = \lambda_0 \lambda_1 \lambda_2 / \sigma_r^2 \sigma_g^2 \sigma_b^2, \qquad (1)$$

where λ_0, λ_1, and λ_2 ($\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq 0$) are the eigenvalues of the covariance matrix Σ of the color distribution $S_I(x, y; d)$, and σ_r^2, σ_g^2, and σ_b^2 are the diagonal elements of Σ. Note that the dependence on $(x, y; d)$ of the right-hand side of Eq. (1) is omitted for brevity. $L(x, y; d)$ is the product of the variances of the color distribution along the principal axes, normalized by the product of the variances along the RGB axes. It gets small when the cluster is elongated (i.e., $\lambda_0 \gg \lambda_1, \lambda_2$) in an oblique direction with respect to the RGB axes, meaning that the RGB components are correlated. In fact, this measure can be interpreted as an extension of *normalized cross correlation* (NCC) [Lewis 1995] so that it is applicable to three images simultaneously (see Appendix B). $L(x, y; d)$ is in the range [0, 1], with the upper bound given by Hadamard's inequality [Gradshteyn and Ryzhik 2000], since $\lambda_0 \lambda_1 \lambda_2 = \det(\Sigma)$.

To illustrate the effect of this measure, we use a sample image shown in Fig. 5(a), taken with a conventional camera lens. Since its RGB planes are aligned, the true disparity is $d = 0$ everywhere, and colors within the local window indicated by the red rectangle in Fig. 5(a) actually form an elongated cluster, as shown in Fig. 5(c). If we deliberately misalign the RGB planes by $d = 1, 3$, and 5 pixels, the distribution becomes more isotropic, and the color alignment measure becomes larger, as shown in Figs. 5(d-f).

Now that we can evaluate the quality of a match between the RGB planes, we can find the disparity d that minimizes $L(x, y; d)$ at each pixel (x, y), from a predetermined set of disparity values (-5 to 10 in our implementation). As local estimates alone are prone to error, we use the standard energy minimization framework using graph-cuts [Boykov et al. 2001] to impose spatial smoothness constraints.



(a) (b) (c) $d = 0, L = 0.003$

(d) $d = 1, L = 0.11$ (e) $d = 3, L = 0.39$ (f) $d = 5, L = 0.54$

Figure 5: *(a) Sample photograph taken with a conventional camera lens. (b) Closeup of the local window indicated by the red rectangle in (a). (c-f) Plots of the pixel colors within the local window in the RGB space. The values d and L shown below each plot are the simulated disparity and the value of Eq. (1).*

5 Matting

Matting is a problem of solving for foreground opacity $\alpha(x, y)$ at each pixel (x, y) in the following *matting equation*.

$$\mathbf{I}(x, y) = \alpha(x, y)\mathbf{F}(x, y) + (1 - \alpha(x, y))\mathbf{B}(x, y), \qquad (2)$$

which models an observed image $\mathbf{I}$ as a convex combination of a foreground color $\mathbf{F}$ and a background color $\mathbf{B}$. By capturing an image so that a foreground object is in focus, we can assume that $\alpha(x, y)$ is aligned between the RGB planes. More precisely, regions with fractional alpha values (i.e., the silhouette of a foreground object) should be within the depth-of-field of the lens. Slight violation of this assumption however does not lead to severe degradation of extracted mattes, as will be shown in Sec. 6.

Solving Eq. (2) based only on the observation $\mathbf{I}$ is an under-constrained problem, since we have only three measurements (I_r, I_g, and I_b) for seven unknowns (α, F_r, F_g, F_b, B_r, B_g, and B_b). at each pixel. Therefore, to incorporate additional constraints, we use a trimap which we automatically generate from the disparity map, and we also leverage the difference in misalignment between foreground and background colors to iteratively optimize the matte.

5.1 Matte Optimization Flow

Algorithm 1 shows our iterative matte optimization procedure. For initialization, we first roughly divide the image into foreground and background regions by thresholding the disparity map, and we dilate their border to construct a trimap having a conservatively wide "unknown" region (50-70 pixels in our implementation), as shown in Fig. 6(a). We then initialize the alpha values using a trimap-based matting method, for which we used Levin et al.'s *Closed-Form Matting* [2008]. While this often gives already good results, errors can

remain where foreground and background colors are similar (see Fig. 9(a) as an example). We detect and correct these errors in the subsequent iterative optimization using color misalignment cues. To determine how the foreground and background colors are misaligned in the "unknown" region, we make a two-layer assumption for the scene around the foreground silhouette. And we propagate the disparity values from the "strictly foreground" region to obtain foreground disparity map $d_F(x, y)$ as shown in Fig. 6(b). Similarly we obtain background disparity map $d_B(x, y)$ from the "strictly background" region (Fig. 6(c)).

In the iterative optimization, letting n denote an iteration count, we first estimate foreground and background colors $\mathbf{F}_n$ and $\mathbf{B}_n$ based on the current matte α_n, by minimizing a quadratic cost function $\sum_{(x,y)} ||\mathbf{I}(x, y) - \alpha_n(x, y)\mathbf{F}_n(x, y) - (1 - \alpha_n(x, y))\mathbf{B}_n(x, y)||^2$ derived from Eq. (2), plus smoothness constraints on foreground and background colors, similar to [Levin et al. 2008]. These estimated colors $\mathbf{F}_n$ and $\mathbf{B}_n$ have errors in the same regions as α_n has errors. We detect these erroneous regions by measuring how consistent the estimated colors are with the foreground and background disparity maps $d_F(x, y)$ and $d_B(x, y)$, as we will describe in Sec. 5.2. We then correct the alpha values around the detected regions to obtain the matte α_{n+1} for the next iteration (Sec. 5.3). We iterate this process until change in the matte is sufficiently small. Fig. 7 illustrates each step of the iterative optimization.

Algorithm 1: *Matte optimization algorithm.*

Initialization

1. Construct a trimap from the disparity map.
2. Find an initial matte α_0 based on the trimap.
3. Propagate the disparity values to obtain foreground and background disparity maps d_F and d_B.

Iterative optimization

1. Estimate foreground color $\mathbf{F}_n$ and background color $\mathbf{B}_n$ based on the current α_n.
2. Compute consistency measures C_{F_n} and C_{B_n} (Sec. 5.2).
3. Update α_{n+1} based on C_{F_n} and C_{B_n} (Sec. 5.3).
4. Repeat until convergence.

(a) (b) (c)

Figure 6: *(a) Trimap for the toy dog image in Fig. 4, constructed from the disparity map shown in Fig. 10(d) top. White: strictly foreground. Black: strictly background. Gray: unknown. (b) Propagated foreground disparity map $d_F(x, y)$. Blue indicates an undefined region. (c) Propagated background disparity map $d_B(x, y)$.*

5.2 Measuring Consistency with Disparity Maps

Similar to the color alignment measure in Eq. (1), we consider a set $S_F(x, y; d)$ of pixel colors within a local window $w(x, y)$, in this case for the foreground color $\mathbf{F}(x, y)$, not for the input image $\mathbf{I}(x, y)$, with hypothesized disparity d as $S_F(x, y; d) = \{(F_r(s+d, t), F_g(s, t-d), F_b(s-d, t)) \mid (s, t) \in w(x, y)\}$, and we define a foreground *color lines model error* as follows.

$$e_F(x, y; d) = \frac{1}{N} \sum_{i=1}^{N} l_i^2, \tag{3}$$

where $N = |S_F(x, y; d)|$, and l_i is the distance of the i-th color in $S_F(x, y; d)$ from the line fitted to the elongated color cluster (i.e., the first principal axis). Intuitively, we examine whether the colors in a local window fit the color lines model. Therefore, $e_F(x, y; d)$ becomes large when d is a wrong disparity. We define the background color lines model error $e_B(x, y; d)$ similarly. See Appendix C for more details.

As we have two possible disparities $d_F(x, y)$ and $d_B(x, y)$ at each pixel (x, y) in the "unknown" region, we define foreground and background *color consistency measures* by incorporating two values of color lines model errors at these two disparities:

$$\begin{aligned} C_F(x, y) &= \exp\left\{(e_F(x, y; d_F) - e_F(x, y; d_B))/\kappa_s\right\}, \\ C_B(x, y) &= \exp\left\{(e_B(x, y; d_B) - e_B(x, y; d_F))/\kappa_s\right\}, \end{aligned} \tag{4}$$

where κ_s is a scale parameter. If the estimated foreground color around (x, y) erroneously contains the (true) background color, $C_F(x, y)$ will be large around that region because $e_F(x, y; d_F)$ will be large and $e_F(x, y; d_B)$ will be small. The effect of the background counterpart $C_B(x, y)$ can be similarly explained.

5.3 Solving for the Matte

Following Wang and Cohen's *Robust Matting* approach [2007], we solve for $\alpha(x, y)$ as a *soft graph-labeling* problem, where each pixel (regarded as a node in a graph) has *data weights* $W_F(x, y)$ and $W_B(x, y)$, and each pair of neighboring pixels has an *edge weight* $W_e(x_0, y_0; x_1, y_1)$. The data weight $W_F(x, y)$ is responsible for pulling $\alpha(x, y)$ toward 1, whereas $W_B(x, y)$ pulls it toward 0. The edge weights impose spatial smoothness constraints on alpha values by the *Matting Laplacian* [Levin et al. 2008]. This formulation is beneficial in that it can be solved as a sparse linear system [Grady 2006], not graph-cuts, and that it guarantees $\alpha(x, y)$ to be in the range [0, 1] without additional hard constraints.

While Wang and Cohen [2007] used color samples gathered from the "strictly foreground" and "strictly background" regions to set the data weights, we instead iteratively update the data weights according to the consistency measures $C_{F_n}(x, y)$ and $C_{B_n}(x, y)$ computed for the current estimate of the foreground and background colors $\mathbf{F}_n$ and $\mathbf{B}_n$, as follows.

$$\begin{aligned} W_{F_n}(x, y) &= \kappa_\alpha \alpha_n(x, y) &&+ \kappa_c(C_{B_n}(x, y) - C_{F_n}(x, y)), \\ W_{B_n}(x, y) &= \kappa_\alpha(1 - \alpha_n(x, y)) &&+ \kappa_c(C_{F_n}(x, y) - C_{B_n}(x, y)), \end{aligned} \tag{5}$$

where κ_α and κ_c are constants. We clamp $W_{F_n}(x, y)$ and $W_{B_n}(x, y)$ at 0 to keep them non-negative. When the foreground consistency measure $C_{F_n}(x, y)$ is smaller (i.e., more consistent) than the background counterpart $C_{B_n}(x, y)$, the foreground data weight $W_{F_n}(x, y)$ is increased while the background data weight $W_{B_n}(x, y)$ is decreased, so that $\alpha(x, y)$ is pulled toward 1 from the current value $\alpha_n(x, y)$. Conversely, $\alpha(x, y)$ will be pulled toward 0 if $C_{F_n}(x, y) > C_{B_n}(x, y)$.

6 Results

For all of the results shown below, we set the local window size to 15×15 pixels, $\kappa_s = 0.1$, $\kappa_\alpha = 0.01$, and $\kappa_c = 0.02$. The matte optimization converged in about 20 iterations. The computation time for a 720×480 image was 10 sec. for depth estimation, and 2 min. for matting on an Intel Pentium 4 3.2GHz with 2GB RAM.

We first demonstrate the performance of our RGB correspondence measure. We compare our disparity estimation results with those of the previous methods [Amari and Adelson 1992; Chang et al. 2002]

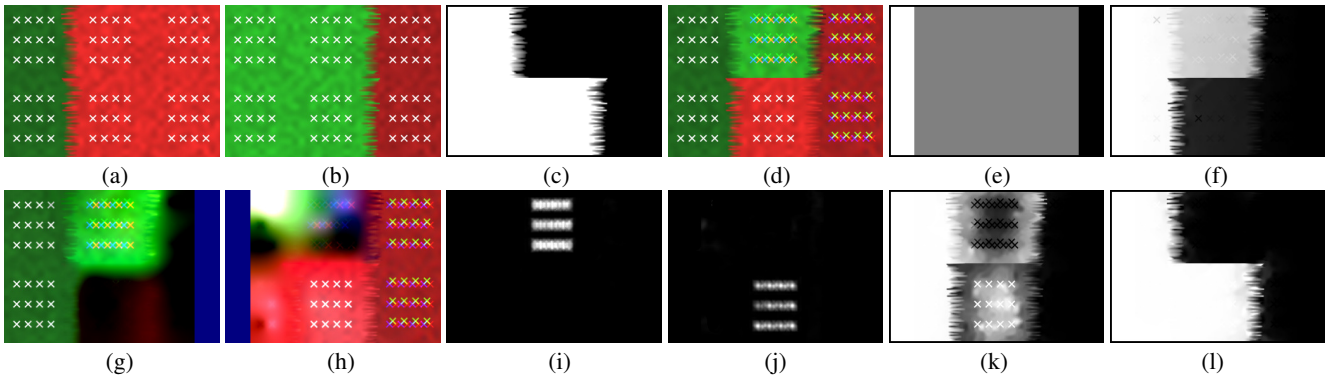


Figure 7: *Synthetic toy example demonstrating how our matte optimization works. (a) Ground truth foreground color. (b) Ground truth background color. (c) Ground truth matte. (d) Composite image from (a-c) with the background color misaligned by 5 pixels. This image is input to our matting algorithm. (e) Trimap. In this example we manually drew it in order to leave a wide "unknown" region. (f) Initialized matte α_0. The center image region has large errors because the foreground and background colors are similar. These errors will be corrected in the subsequent steps using color misalignment cues from the 'x' shaped textures. (g) Estimated foreground color $\mathbf{F}_0$ based on α_0 in (f). Blue indicates an undefined region. (h) Estimated background color $\mathbf{B}_0$ based on α_0 in (f). (i) Foreground color consistency C_{F_0} computed for $\mathbf{F}_0$ in (g). The disparity of (g) around the top center region is 5, which is inconsistent with the true foreground disparity of 0. Therefore, C_{F_0} became large around there. (j) Background color consistency C_{B_0} computed for $\mathbf{B}_0$ in (h). The disparity of (h) around the bottom center region is 0, which is inconsistent with the true background disparity of 5. Therefore, C_{B_0} became large around there. (k) Updated matte. The alpha values were pulled toward 0 where C_{F_0} in (i) is large, and toward 1 where C_{B_0} in (j) is large. (l) Final matte after convergence, which is close to the ground truth matte (c).*

in Figs. 10(a-c). In order to reveal raw performance, we show local window estimates without graph-cut optimization. As Amari and Adelson's method relies on high-pass filtering, it mostly failed to detect disparities of the defocused scene backgrounds (Fig. 10(b)). Chang et al.'s method performed better, but it handled color edges and gradations poorly, presumably because these may not be accounted for by a single intensity normalization factor within a local window (Fig. 10(c)). Our method produced better results than the previous methods (Fig. 10(a)).

We also compare our results with a mutual information-based method by Kim et al. [2003], which can handle broad types of intensity relationships between images. Since their method is coupled with iterative graph-cut optimization, our results after graph-cut optimization are also shown in Fig. 10(d). Because their correspondence measure is defined for two images, we take the average of the values for the three pairs of RGB planes (RG, GB, and BR). Their method performed well in view of the fact that it does not assume *a priori* knowledge of the intensity relationships. However, some portions of the foreground objects were not detected (Fig. 10(e)).

Next we show our matting results. Fig. 8(a) shows the extracted matte for the toy dog image in Fig. 4. The hairy silhouette was extracted successfully. We can use this matte to refine the boundary of the foreground and background regions in the depth map as shown in Fig. 8(b), by compositing the foreground and background disparity maps shown in Figs. 6(b-c). In Fig. 9, we applied existing natural image matting methods, Closed-Form Matting [Levin et al. 2008] and Robust Matting [Wang and Cohen 2007], with the trimap given by our method. These results are not for comparison because the previous methods are designed for color-aligned images, but the matte errors seen in Fig. 9 are indicative of the importance of our color consistency measure in suppressing them.

For proper comparison, we used a ground truth matte shown in Fig. 11(a) obtained by capturing an object in front of a simple background and by using *Bayesian Matting* [Chuang et al. 2001], followed by manual touch-up where needed. We created a synthetic "natural" image by compositing the object over a new background image, as shown in Fig. 11(b). We also created its color-misaligned version by shifting the background color by 3 pixels before composition. We applied the previous methods to the color-aligned synthetic image, and our method to the color-misaligned one. Though not perfect, our method produced a better matte as shown in Figs. 11(c-e). For quantitative evaluation, we conducted the same experiment for five more examples shown in Fig. 12, and we computed the mean squared errors (MSE) against the ground truth mattes, which we plotted in Fig. 13. Our method reduced MSE values by 33-86% compared to the other two methods.

As our camera is portable, and only a single exposure is required, it is easy to capture moving objects such as animals, as shown in Fig. 14. Using the camera's rapid shooting capability, we can also perform *video matting*. The supplemental video shows an extracted video matte of a walking person.

Figure 8: *(a) Extracted matte for the toy dog image in Fig. 4. (b) Refined depth map. Compare this with Fig. 10(d) top.*

Figure 9: *Results of existing natural image matting methods. (a) Closed-Form Matting. (b) Robust Matting.*

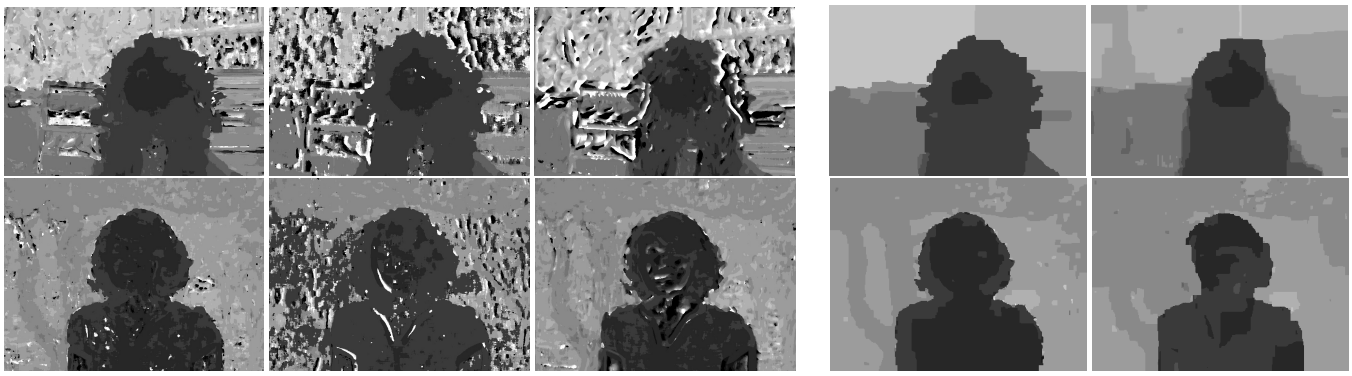


Figure 10: *Comparison of correspondence measures between the RGB planes. Larger intensities indicate larger disparities. Top row: results for the toy dog image in Fig. 4. Bottom row: results for the woman image in Fig. 1. (a) Our method (local estimate). (b) Amari and Adelson [1992] (local). (c) Chang et al. [2002] (local). (d) Our method (after graph-cut optimization). (e) Kim et al. [2003] (based on mutual information with iterative graph-cut optimization).*

Figure 11: *Comparison using a ground truth matte. (a) Ground truth matte. (b) Synthetic natural image. (c) Our method (applied to the color-misaligned version of (b)). (d) Closed-Form Matting (applied to (b)). (e) Robust Matting (applied to (b)).*

Fig. 14 also shows a portion of the foreground object (the hip of the sheep) is slightly out of the depth-of-field of the lens, violating the assumption that $\alpha(x, y)$ is aligned between the RGB planes in Eq. (2). However, degradation of the extracted matte around the region was small, as shown in Fig. 14(d).

Figure 12: *Synthetic natural images and their ground truth mattes.*

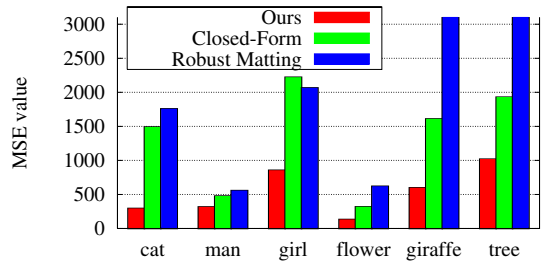
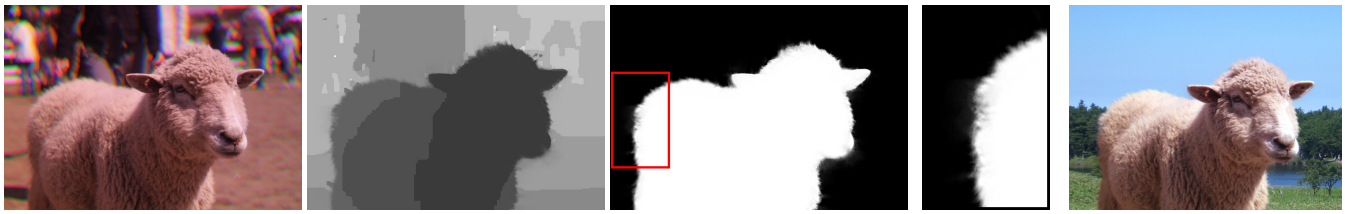


Figure 13: *MSE values of the mattes produced by our method and the previous methods for the images shown in Figs. 11 and 12.*

Finally, we show several post-exposure image editing examples based on the extracted depth and matte. First of all, we can restore a color-aligned image (Fig. 15(b)) by re-compositing the foreground and background colors after canceling their misalignment based on the foreground and background disparity maps. Specifically, if the foreground disparity at (x, y) is d, the aligned foreground color at that point is restored as: $(F_r(x+d, y), F_g(x, y-d), F_b(x-d, y))$. Moreover, as the defocus PSF is a square whose size is given by the disparity map, we can restore an all-in-focus background color by deconvolution (Fig. 15(c)). By blurring the foreground color and the all-in-focus background color differently, we can synthetically refocus the image (Fig. 15(d)) [Bando and Nishita 2007]. In the presence of hairy foreground objects, alpha mattes are indispensable for the above operations to give plausible results. The supplemental video shows interactive refocusing and view synthesis animations. We can also composite color-aligned foreground objects over other images as shown in Figs. 1(e) and 14(e), where we adjusted the objects' color to match the corresponding background. Fig. 16 shows additional color misalignment cancellation results.

7 Discussions and Conclusions

We have presented a method for automatically extracting a scene depth map and the alpha matte of an in-focus foreground object using a color-filtered aperture. Our method only modifies a camera lens with off-the-shelf color filters to capture multi-view images in a single exposure. We have proposed an effective correspondence measure between the RGB planes, and a method for employing color misalignment cues to improve the matte. We believe our concise camera design with various post-exposure image editing capabilities will make computational photography a more readily available tool for many users.



(a) (b) (c) (d) (e)

Figure 14: *Results for a sheep. (a) Captured image. (b) Depth map. (c) Matte. (d) Closeup from the red rectangle in (c). (e) Composite.*

(a) (b) (c) (d)

Figure 15: *Examples of post-exposure image editing based on the extracted depth and matte. The bottom row shows closeup views of the top row. (a) Captured image. The colors are misaligned. (b) Color misalignment canceled. (c) Defocus blur removed. (d) Refocused.*

The major limitation of our approach is that it does not work for objects having only a single pure R, G, or B color. Combining with depth-from-defocus methods may partially solve this problem. However, this does not mean that objects must have achromatic colors all over. For example, the disparity of the red box in Fig. 4 is correctly identified as shown in Fig. 10(d), thanks to the alphabets and the pictures of chocolates printed on the box. Therefore, our requirement is that objects must not be purely colored *entirely*, and we think there are many real-world objects satisfying this requirement. We would like to further investigate this limitation.

In our imaging system, the f-number is fixed to 1.8 (full aperture of our prototype lens) because a large aperture increases disparities and thus increases depth resolution. Since disparities also increase when the lens is focused near, our system typically works well for foreground objects at 0.5 to 2.5 meters away from the camera with a sufficiently distant (about twice as far away) background. For farther scenes, depth resolution will gradually decrease, and the matte quality will also deteriorate as color misalignment will be small.

By introducing color filters, the amount of incident light is decreased. Increasing the aperture filter area to compensate for this introduces more defocus. While this degrades depth estimation accuracy at defocused regions, it suppresses background clutters, which is beneficial for matting. Color filters may also affect color demosaicing of the image sensor, although we did not observe any loss of quality in our experiments, mainly because we downsampled the captured images for tractable computation time.

While our depth estimation works fairly robustly, our matting fails when the foreground and background colors are similar with little texture, as shown in Fig. 17(b), since we have few color misalignment cues. Another failure mode is that, as we use a relatively large window (15×15), we cannot recover small/thin features such as hair strands and holes in foreground objects, once they are missed in the course of optimization, as shown in Fig. 17(d). We would like to address the above issues in the future.

(a) (b) (c)

Figure 16: *More color misalignment cancellation results. (a) Restored images. (b) Closeups of (a). (c) Closeups of the original.*

(a) (b) (c) (d)

Figure 17: *Failure cases. Major errors are indicated by the arrow and circles. (a) Captured image. (b) Matte from (a). (c) Closeup of the ground truth matte for the girl image in Fig. 12. (d) Our result.*

Acknowledgments

We gratefully acknowledge helpful comments and suggestions from Takeshi Naemura, Yusuke Iguchi, Takuya Saito, and the anonymous reviewers. We would also like to thank Johanna Wolf, Zoltan Szego, Paulo Silva, and Saori Horiuchi for their help.

References

ADELSON, E. H., AND WANG, J. Y. A. 1992. Single lens stereo with a plenoptic camera. *IEEE Trans. PAMI 14*, 2, 99–106.

AMARI, Y., AND ADELSON, E. H. 1992. Single-eye range estimation by using displaced apertures with color filters. In *Proc. Int. Conf. Industrial Electronics, Control, Instrumentation, and Automation*, vol. 3, 1588–1592.

BANDO, Y., AND NISHITA, T. 2007. Towards digital refocusing from a single photograph. In *Proc. Pacific Graphics*, 363–372.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI 23*, 11, 1222–1239.

CHANG, I.-C., HUANG, C.-L., HSUEH, W.-J., LIN, H.-C., CHEN, C.-C., AND YEH, Y.-H. 2002. A novel 3-D hand-held camera based on tri-aperture lens. In *Proc. SPIE 4925*, 655–662.

CHUANG, Y.-Y., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2001. A bayesian approach to digital matting. In *Proc. CVPR*, vol. 2, 264–271.

CHUANG, Y.-Y. 2004. *New models and methods for matting and compositing*. PhD thesis, University of Washington.

GEORGEIV, T., ZHENG, K. C., CURLESS, B., SALESIN, D., NAYAR, S., AND INTWALA, C. 2006. Spatio-angular resolution tradeoff in integral photography. In *Proc. EGSR*, 263–272.

GRADSHTEYN, I. S., AND RYZHIK, I. M. 2000. *Table of integrals, series, and products (sixth edition)*. Academic Press.

GRADY, L. 2006. Random walks for image segmentation. *IEEE Trans. PAMI 28*, 11, 1768–1783.

GREEN, P., SUN, W., MATUSIK, W., AND DURAND, F. 2007. Multi-aperture photograpy. *ACM Trans. Gr. 26*, 3, 68:1–68:7.

JOSHI, N., MATUSIK, W., AND AVIDAN., S. 2006. Natural video matting using camera arrays. *ACM Trans. Gr. 25*, 3, 779–786.

KIM, J., KOLMOGOROV, V., AND ZABIH, R. 2003. Visual correspondence using energy minimization and mutual information. In *Proc. ICCV*, vol. 2, 1033–1040.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Gr. 26*, 3, 70:1–70:9.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2008. A closed-form solution to natural image matting. *IEEE Trans. PAMI 30*, 2, 228–242.

LEWIS, J. P. 1995. Fast template matching. In *Proc. Vision Interface*, 120–123.

LIANG, C.-K., LIN, T.-H., WONG, B.-Y., LIU, C., AND CHEN, H. H. 2008. Programmable aperture photography: multiplexed light field acquisition. *ACM Trans. Gr. 27*, 3, 55:1–55:10.

MCGUIRE, M., MATUSIK, M., PFISTER, H., DURAND, F., AND HUGHES, J. 2005. Defocus video matting. *ACM Trans. Gr. 24*, 3, 567–576.

MCGUIRE, M., MATUSIK, W., AND YERAZUNIS, W. 2006. Practical, real-time studio matting using dual imagers. In *Proc. EGSR*, 235–244.

NG, R., LEVOY, M., BRÉDIF, M., DUVAL, G., HOROWITZ, M., AND HANRAHAN, P., 2005. Light field photography with a hand-held plenoptic camera. Tech. Rep. CSTR 2005-02, Stanford Computer Science, Apr.

NG, R. 2005. Fourier slice photography. *ACM Trans. Gr. 24*, 3, 735–744.

OMER, I., AND WERMAN, M. 2004. Color lines: image specific color representation. In *Proc. CVPR*, vol. 2, 946–953.

RASKAR, R., TUMBLIN, J., MOHAN, A., AGRAWAL, A., AND LI, Y. 2006. Computational photography. In *Proc. Eurographics STAR*.

SMITH, A. R., AND BLINN, J. F. 1996. Blue screen matting. In *Proc. ACM SIGGRAPH 96*, 259–268.

VEERARAGHAVAN, A., RASKAR, R., AGRAWAL, A., MOHAN, A., AND TUMBLIN, J. 2007. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Gr. 26*, 3, 69:1–69:12.

VLAHOS, P., 1971. Electronic composite photography. U. S. Patent 3,595,987.

WANG, J., AND COHEN, M. F. 2007. Optimized color sampling for robust matting. In *Proc. CVPR*.

WEXLER, Y., FITZGIBBON, A., AND ZISSERMAN, A. 2002. Bayesian estimation of layers from multiple images. In *Proc. ECCV*, 487–501.

XIONG, W., AND JIA, J. 2007. Stereo matching on objects with fractional boundary. In *Proc. CVPR*.

Appendix A Color Crosstalk Suppression

Let c_r, c_g, and c_b be the mean image colors of a sheet of white paper through the R, G, and B filters, respectively. For our prototype,

$$\begin{aligned} c_r &= (1.000, 0.335, 0.025)^T, \\ c_g &= (0.153, 1.000, 0.162)^T, \\ c_b &= (0.007, 0.190, 1.000)^T, \end{aligned} \tag{A.1}$$

where the values are normalized with respect to the maximum component. Letting $M = (c_r, c_g, c_b)$, we can decompose an observed color c_o into the three aperture filters' contributions by $M^{-1}c_o$.

Appendix B Color Alignment Measure and NCC

An equivalent of Eq. (1) in 2D (e.g., in the RG space) would be:

$$L(x, y; d) = \lambda_0 \lambda_1 / \sigma_r^2 \sigma_g^2. \tag{B.1}$$

Let σ_{rg} be the covariance between the R and G components, then by $\lambda_0 \lambda_1 = \det(\Sigma) = \sigma_r^2 \sigma_g^2 - \sigma_{rg}^2$, we obtain:

$$L(x, y; d) = 1 - \sigma_{rg}^2 / \sigma_r^2 \sigma_g^2 \tag{B.2}$$

Since $\mathrm{NCC} = \sigma_{rg}/\sigma_r\sigma_g \in [0, 1]$, the 2D version of the color alignment measure L has a one-to-one correspondence to NCC.

Appendix C Computing the Color Lines Model Error

Letting c_i be the i-th color in $S_F(x, y; d)$, μ be the mean color, and v_0 be a unit vector of the fitted line (the first principal eigenvector), trigonometry gives the distance l_i of the point c_i from the line as:

$$l_i^2 = |c_i - \mu|^2 - ((c_i - \mu)^T v_0)^2. \tag{C.1}$$

The average of the first term is, by definition, the variance:

$$\frac{1}{N}\sum_{i=1}^{N} |c_i - \mu|^2 = \sigma_r^2 + \sigma_g^2 + \sigma_b^2. \tag{C.2}$$

For the second term, we have:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}((c_i - \mu)^T v_0)^2 &= v_0^T \left(\frac{1}{N}\sum_{i=1}^{N}(c_i - \mu)(c_i - \mu)^T\right) v_0 \\ &= v_0^T \Sigma v_0 = v_0^T(\lambda_0 v_0) = \lambda_0,\end{aligned} \tag{C.3}$$

by the definitions of the covariance matrix Σ and the eigenvector v_0. Therefore, the color lines model error can be computed as follows.

$$e_F(x, y; d) = \sigma_r^2 + \sigma_g^2 + \sigma_b^2 - \lambda_0. \tag{C.4}$$

This turns out to be similar to the color alignment measure of Eq. (1), but we found it more effective for matting to use this unnormalized, direct error measure. Since estimation errors of background disparities are typically larger than those of foreground disparities, we discount $e_B(x, y; d)$ by scaling it by around 0.7-0.9.