Brigham Young University
BYU ScholarsArchive

Faculty Publications

2010-06-01

Simultaneous Foreground, Background, and Alpha Estimation for Image Matting

Bryan S. Morse
morse@byu.edu

Brian L. Price

Scott Cohen

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Computer Sciences Commons

Original Publication Citation

B. Price, B. Morse, and S. Cohen, "Simultaneous foreground, background, and alpha estimation for image matting," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 21.

BYU ScholarsArchive Citation

Morse, Bryan S.; Price, Brian L.; and Cohen, Scott, "Simultaneous Foreground, Background, and Alpha Estimation for Image Matting" (2010). *Faculty Publications*. 99.
https://scholarsarchive.byu.edu/facpub/99

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

Simultaneous Foreground, Background, and Alpha Estimation for Image Matting

Brian L. Price
Brigham Young University
bprice@cs.byu.edu

Bryan S. Morse
Brigham Young University
morse@cs.byu.edu

Scott Cohen
Adobe Systems
scohen@adobe.com

Abstract

Image matting is the process of extracting a soft segmentation of an object in an image as defined by the matting equation. Most current techniques focus largely on computing the alpha values of unknown pixels and treat computation of the foreground and background colors as an afterthought, if at all. However, for many applications, such as compositing an object into a new scene or deleting an object from the scene, the foreground and background colors are vital for an acceptable answer. We propose a method of solving for the foreground, background, and alpha of an unknown region in an image simultaneously. This allows for novel constraints to be placed directly on the foreground and background as well as on alpha. We show through both visual results and quantitative measurements on standard datasets that this approach produces more accurate foreground and background values at each pixel while maintaining competitive results on the alpha matte.

1. Introduction

The goal of image matting is to produce a soft segmentation of an image I by computing the relative contribution of foreground F and background B at each pixel according to the equation

$$I = \alpha F + (1 - \alpha)B \tag{1}$$

where α is the opacity of the foreground. If I is a three-channel color image, the matting formula yields seven unknowns with only three equations, making the solution greatly underconstrained. Most techniques constrain the system with a user-defined trimap (which identifies known foreground and background regions and unknown regions) as well as additional terms to produce an energy function to optimize or a system of equations to solve.

Matting is important for image and video editing when one wishes to select an object exactly for editing. While some applications may only require an alpha value for the selected object, such as perhaps applying a filter that tails off as alpha drops to zero, many others require both the alpha value and the extracted foreground and/or background color. A prime example of this is composition, where a selected object may have a noticeable "halo" around the object when placed on a different-colored background if the background colors are still maintained in the foreground. Another such task is object deletion, where the background colors are needed to restore the image.

Interestingly, most existing matting algorithms do not attempt to solve for the foreground and background colors, focusing on solving for alpha exclusively [4, 7, 9, 10, 14, 15, 17, 19]. Such solutions still result in an underconstrained problem, with three equations and six unknowns, which some methods attempt to solve as a postprocess [1, 6, 12, 16]. Of course, given a trimap, no information is initially known about alpha, but only about foreground and background. Some methods will constrain the foreground and background colors [6, 7, 9, 12, 14, 17] or begin with an initial estimate of them [9, 10, 15, 17], but few attempt to solve for them alongside alpha.

In this paper, we introduce a method for solving the foreground, background, and alpha simultaneously and make the following three contributions. First, by optimizing over the foreground and background as opposed to only addressing them as a postprocess or not at all, we produce more accurate foreground and background values. Second, we produce alpha matte results that are competitive with the state-of-the-art. Third, we contribute several new ideas regarding the additional terms used to solve the matting equation. These include a novel color term that affects the foreground and background directly, as well as an important observation and new terms involving the gradient of the matting equation and smoothness assumptions.

2. Related Work

Despite the matting problem consisting of seven unknowns per pixel, the majority of matting algorithms focus only on computing one unknown, the alpha channel.

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

Bayesian Matting [3], an exception to this, computes foreground, background, and alpha to provide a complete solution to the matting equation. The colors of the known foreground and background are represented using Gaussians, and the unknown values are computed by an optimization over this information and the matting equation using *maximum a posteriori* estimation. While this approach computes the foreground and background, the imprecise representation of the known color values does not lead to robust results in the general case.

Easy Matting [5] also computes the foreground and background at each unknown pixel but limits the possible color to one of twenty samples, severely limiting the range of possible values. The alpha value is then computed using the matting equation and a smoothness prior on alpha. Similarly, Wang and Cohen [18] presuppose that the foreground and background belong to a small, unspecified sample set, and then compute an alpha using the matting equation and an alpha smoothness prior. Unfortunately, these methods often oversmooth the matte due to the alpha smoothness prior, and the limited freedom in foreground and background values limits their results.

While few algorithms explicitly optimize over the foreground and background, many do constrain those values to some extent. An early example of this is seen in the work of Ruzon and Tomasi [12]. The foreground and background are modeled as Gaussians, and the assumption is made that the color at the unknown pixels comes from a distribution between these Gaussians. An alpha matte is then computed by maximizing the probability of the image color on possible intermediary distributions.

The closed-form solution of Levin et al. [6] constrains the foreground and background colors in a local neighborhood to each be a linear combination of two colors respectively using the matting Laplacian. This allows them to eliminate the foreground and background colors from the matting equation and solve directly for alpha. While this works well in many cases, it fails when the color model may not be well represented by their linear model.

The elegant solution in [6] has inspired several improvements. Spectral matting [7] extends the framework to segment multiple layers without the use of a trimap or user interaction. The variant in [14] also extends this framework to computing mattes of multiple layers. Singaraju et al. [13] introduced an improvement to the matting Laplacian that handles cases where the colors in a neighborhood may be represented by a point rather than a line.

Another example of a paper that applies constraints to the foreground and background without solving for them is the work of Zheng and Kambhamettu [19]. They approach the matting problem in a semi-supervised machine learning paradigm. This allows them to model the matting equation using both linear and non-linear equations, and is used in conjunction with a smoothness prior on alpha to compute an alpha matte.

Several other approaches include an initial estimate of the foreground and background in their computation but do not seek to improve those values. In Robust Matting [17] an estimate of the foreground and background is computed by sampling along the nearest edge in the known foreground or background regions. This information is used to form a sparsity bias that encourages most of the unknown pixels to take the value of 0 or 1. The sparsity bias is used in conjunction with the foreground and background smoothness constraint of [6] to produce an alpha matte. While this method works well in many cases, it can fail when its initial estimation of foreground and background is not accurate.

Rhemann et al. [9] formulate matte computation similar to [17] but with several improvements. The initial foreground and background estimation is improved by taking samples from known regions that are close to the pixel in question using a geodesic distance measure. A hard segmentation is also used to help formulate the sparsity prior. Robust Matting is also improved to handle high resolution images through interactive trimap generation in [10].

Poisson Matting [15] computes an estimate of the difference of the foreground and background colors in grayscale. This estimate is updated iteratively following a computation of the matting equation using Poisson equations under the assumption that the foreground and background are smooth. The simplified method of updating the foreground-background difference in grayscale and the smoothness assumption on the foreground colors prohibit good results without excessive user interaction.

Another approach to creating an alpha matte is to forgo the matting equation and compute alpha using other means. Grady et al. [4] equates alpha to the probability that a random walker will reach a given pixel from the known foreground and background regions. Bai and Sapiro [1] compute alpha as a ratio of geodesic distances from a pixel to the known foreground and background regions. While these approaches do well in some cases, it is difficult to achieve good results over a wide range of problems without a more explicit representation of the matting equation.

Several methods compute the foreground and background as a postprocess after computing alpha. Ruzon and Tomasi [12] solve for the foreground and background by interpolating the means of their respective Gaussians. Levin et al. [6] minimizes an equation combining the matting equation and a smoothness prior on the foreground and background. Soft Scissors [16], an interactive extension of [17], computes the foreground using a random walk. Bai and Sapiro [1] solve for the foreground and background by randomly sampling the known areas looking for values that minimize the matting equation.

3. Methods

We approach the matting problem as an energy minimization problem over the seven values for each pixel in the unknown region of the trimap. By optimizing not just over alpha, but also over the foreground and background, we may introduce additional regularizations to the problem that are otherwise difficult or impossible. This leads to improved estimates of foreground and background colors compared to existing algorithms.

3.1. Problem formulation

We formulate our solution to the matting equation as an energy minimization problem of the form

$$E(X) = \sum_{p\in\Omega} \sum_{i=0}^{|U|-1} \lambda_i U_i(p) + \sum_{(p,q)\in N} \sum_{i=0}^{|V|-1} \gamma_i V_i(p,q) \quad (2)$$

where X is the set of 7-tuples $(F_r, F_g, F_b, B_r, B_g, B_b, \alpha)$ corresponding to the foreground, background, and alpha values at each pixel p in the unknown region Ω, U is the set of all unary terms being minimized, V is the set of all pairwise terms being minimized, N is the set of all pairs of tuples whose corresponding pixels are adjacent (4-connected), and λ_i and γ_i are weighting factors.

Since our formulation minimizes over foreground, background, and alpha, we may include terms that affect any of these values. The remainder of this section details the terms we use in Equation 2 and the resulting optimization.

3.2. Matting Equation

Adherence to the matting equation (Equation 1) is maintained by

$$U_0(p) = ||I(p) - [\alpha(p)F(p) + (1-\alpha(p))B(p)]||_2^2 \quad (3)$$

where $I(p)$ is the color of image I at pixel p (and similarly for F, B, and α).

3.3. Matting Derivatives

We can constrain the solution not only with respect to the matting equation, but also with respect to the derivatives of the matting equation:

$$\nabla I = \alpha \nabla F + (1-\alpha)\nabla B + (F-B)\nabla\alpha \quad (4)$$

We define the derivative between adjacent pixels p and q (in the x and y direction) as $\Delta_{pq} I_j = I_j(p) - I_j(q)$, where I_j is an image consisting of only the j^{th} channel of I (and similarly for F and B). We then derive the following term:

$$V_0(p,q) = \sum_{j\in(r,g,b)} [\Delta_{pq} I_j - \alpha(p)\Delta_{pq} F_j - (1-\alpha(p))\Delta_{pq} B_j - (F_j(p) - B_j(p))\Delta_{pq}\alpha]^2 \quad (5)$$

3.4. Smooth Matting Gradients

While Equation 1 is known to be true, it has an infinite number of solutions, corresponding to any line segment in the color cube that passes through the color at the pixel. The matting derivative of Equation 4 is derived from Equation 1 and is insufficient to completely constrain the problem. Because of this, additional terms are needed. One common practice in current methods is to add a smoothness assumption on the foreground, background, or alpha. In other words, one or more of the terms ∇F, ∇B, and $\nabla\alpha$ are assumed to be zero. The effect of this is to assume that the gradient of the image is not affected by gradients in the term that was set to zero.

Differing algorithms make different decisions about which gradient to set to zero. For example, several algorithms make the assumption that the foreground and background values are locally smooth. Poisson Matting [15] does so by directly modifying Equation 4 by setting ∇F and ∇B to zero, leaving

$$\nabla I = (F-B)\nabla\alpha \quad (6)$$

Smoothness assumptions on foreground and background are also made in Levin et al. [6] and other works incorporating its matting Laplacian [7, 9, 10, 13, 14, 17], although the smoothness assumption in [6] is admittedly more complex and elegant than that of Equation 6.

On the other hand, other papers, such as Easy Matting [5], Wang and Cohen [18], and Digital Matting [19] make the opposite assumption, that the alpha is smooth (setting $\nabla\alpha$ to 0), thereby minimizing

$$\nabla I = \alpha\nabla F + (1-\alpha)\nabla B \quad (7)$$

It is interesting that different methods simplify Equation 4 to make opposite assumptions. Of course, in places where the change in alpha values is primarily driving the image gradient, $\nabla F = \nabla B = 0$ is a better assumption to make. In areas where α is 0 or 1, or where there is smooth transparency, $\nabla\alpha = 0$ is the better assumption. In order to apply each of these assumptions where they are effective, we could instead make the smoothness assumption

$$\nabla I = \begin{cases} \alpha\nabla F & \text{if } \nabla F \mapsto \nabla I \\ (1-\alpha)\nabla B & \text{if } \nabla B \mapsto \nabla I \\ (F-B)\nabla\alpha & \text{if } \nabla\alpha \mapsto \nabla I \end{cases} \quad (8)$$

where $\nabla F \mapsto \nabla I$ indicates that the gradient at I is due primarily to the gradient in F. This leads to a term that encourages smoothness in foreground, background, and alpha where there should be smoothness, and encourages the gradient to be proportional to the image where it should be proportional:

$$V_1(p,q) = \begin{cases} \sum_{j\in(r,g,b)} g_j^F(p,q) & \text{if } \nabla F \mapsto \nabla I \\ \sum_{j\in(r,g,b)} g_j^B(p,q) & \text{if } \nabla B \mapsto \nabla I \\ \sum_{j\in(r,g,b)} g_j^\alpha(p,q) & \text{if } \nabla\alpha \mapsto \nabla I \end{cases} \quad (9)$$

where (using the notation from Equation 5)

$$g_j^F(p,q) = (\Delta_{pq} I_j - \alpha(p)\Delta_{pq} F_j)^2 + (\Delta_{pq} B_j)^2 + (\Delta_{pq}\alpha)^2 \quad (10)$$

$$g_j^B(p,q) = (\Delta_{pq} I_j - (1-\alpha(p))\Delta_{pq} B_j)^2 + (\Delta_{pq} F_j)^2 + (\Delta_{pq}\alpha)^2 \quad (11)$$

$$g_j^\alpha(p,q) = (\Delta_{pq} I_j - (F_j(p) - B_j(p))\Delta_{pq}\alpha)^2 + (\Delta_{pq} F_j)^2 + (\Delta_{pq} B_j)^2 \quad (12)$$

Equations 10, 11, and 12 are just discretizations of Equation 4 encouraging $\nabla B = \nabla\alpha = 0$, $\nabla F = \nabla\alpha = 0$, and $\nabla F = \nabla B = 0$, respectively.

To decide which of the gradients is causing the image gradient, we use the alpha values at pixels p and q. For some threshold T, if $\alpha > T$ for both p and q, then we assume $\nabla F \mapsto \nabla I$. If $\alpha < (1 - T)$ for both p and q, then we assume $\nabla B \mapsto \nabla I$. Otherwise, we assume $\nabla\alpha \mapsto \nabla I$.

3.5. Color

The color term encourages the foreground and background colors to be similar to the colors in the known foreground and background areas of the image respectively. This is achieved by imposing a cost on the foreground (background) color based on the squared distance in color space to the nearest pixel in the foreground (background) in a combined color and position space,

$$U_1(p) = ||F(p) - I^{\mathcal{F}}_{near}(p)||^2 + ||B(p) - I^{\mathcal{B}}_{near}(p)||^2 \quad (13)$$

where $I^{\mathcal{F}}_{near}(p)$ is the color of the nearest pixel to p in the known foreground region of I and $I^{\mathcal{B}}_{near}(p)$ the nearest pixel in the known background. To compute the nearest pixel, we calculate a distance for the foreground and background pixels respectively to all other possible x,y,r,g,b locations using the Maurer distance transform [8].

3.6. Sparsity in α

In most images, the number of mixed pixels is far less than the number of pixels that belong completely to foreground or background (pixels whose $\alpha = 0$ or 1). Because of this, it is reasonable to bias pixels toward having alpha values of 0 or 1. Our term to bias alpha toward 0 or 1 is given by

$$U_2(p) = \begin{cases} (1-\alpha)h(I,p,B,F) & \text{if } h(I,p,F,B) \leq T_h \\ \alpha h(I,p,F,B) & \text{if } h(I,p,B,F) \leq T_h \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where

$$h(I,p,F,B) = \frac{dist(I,p,F)}{dist(I,p,F) + dist(I,p,B)} \quad (15)$$

$dist(I,p,F) = ||I(p) - I^{\mathcal{F}}_{near}(p)||^2$ and T_h is a threshold. Here we use a value of $T_h = 0.01$. This term introduces the assumption that if the color models indicate that the color at a pixel is distinctly foreground or background only (if it has a low squared distance to foreground and high to background, or vice versa), then that pixel likely has an alpha value near 1 or 0 respectively.

3.7. Optimization Method

Many previous matting algorithms have cost functions that are specifically designed to be easy to minimize. In our formulation, we are more concerned with including the best terms possible to allow for the simultaneous computation of the foreground, background, and alpha. Unfortunately, this approach results in a difficult function to optimize.

We use gradient descent in order to minimize Equation 2. Because our objective function is not convex, we can easily fall into local minima while optimizing the energy function. To address this, we add a momentum term to help overcome shallow local minima. We also borrow an idea from graduated non-convexity [2]. These methods attempt to minimize over a non-convex objective function by smoothing the function such that it is (more) convex, finding a minimum, and then iteratively using that minimum to initialize a less smooth form of the function. We achieve a similar end by changing the matting equation weight λ_0 in order to allow for a smoother function initially before converging on the weighted terms we desire. Although the matting equation term (Equation 3) is extremely important to finding the correct solution, it may be minimized to 0 at infinitely many values of F, B, and α, and so produces many local minima. By reducing λ_0 near the beginning of the gradient descent, we can more easily avoid local minima created by this term. Specifically, we use $\lambda_0 = 0.25$ initially and increment it in steps until $\lambda_0 = 4$. We fix the other parameters at $\lambda_1 = 1$, $\lambda_2 = 0.4$, $\gamma_0 = 0.2$, $\gamma_1 = 0.8$. We also increase the threshold T used for computing the primary gradient in Equation 9. We use a $T = 0.6$ initially, and increment it at until $T = 0.99$. The lower initial value helps better compute F and B, and the later higher value enforces more smoothness in α once the foreground and background are better estimated. Note that we constrain F, B, and α to the range $[0, 1]$ at each iteration.

To initialize the gradient descent, we compute an initial estimate of the foreground, background, and alpha values. We initialize F and B by sampling the known regions similar to [17]. We initialize α by computing the ratio of the distance to the nearest foreground and background color for a given pixel as computed by Equation 15 ($\alpha = 1 - h(I,p,F,B)$). This distance metric is similar in spirit to those in [1, 4].

4. Results

We evaluate our matting technique using the two datasets introduced in [11]. The first dataset, which we will refer to as the public dataset, consists of 27 input images and

Algorithm	MAD	MSE	Gradient
Our Method	0.017	0.0066	0.016
Closed-Form Matting [6]	0.019	0.0081	0.035
Robust Matting [17]	0.037	0.0177	0.032

Table 1. Average error in α on public set from [11] for our method, Closed-Form Matting [6], and Robust Matting[17].

trimaps, and has the ground truth alpha mattes and ground truth foreground images publicly available. The second dataset, the private dataset, consist of 8 images which are overall more difficult than the public dataset, and their accompanying trimaps only. The foreground, background, and alpha ground truth images of the private dataset are not available, although one may submit their alpha mattes to www.alphamatting.com to obtain an error score over four different metrics.

4.1. Alpha Accuracy

Table 1 shows the error in α on the public dataset from [11] for our algorithm, Closed-Form Matting [6] using their publicly-available code, and Robust Matting [17] using our own implementation. The mean absolute difference (MAD), mean squared error (MSE), and gradient error (which detects errors in the gradient and is correlated to what humans visually perceive as correct as explained in [11]) averaged over all images in the dataset are shown. Our algorithm outperformed Closed-Form Matting and Robust Matting on all three measures. We also achieved a lower gradient error than both Closed-Form Matting and Robust Matting on all examples individually.

Tables 2 and 3 show our results on the private dataset for the sum of absolute difference and the mean squared error respectively as computed by submitting our results to www.alphamatting.com. For size reasons, only a subset of the tables are shown. On this more difficult dataset, we perform comparable to the current best algorithms in generating an alpha matte, while only one of these algorithms, [6], computes foreground and background colors (and does so as a postprocess). Note the magnitudes of the values in Tables 2 and 3 differ from those in Table 1 because they are scaled at www.alphamatting.com for presentation purposes.

A comparison of our technique to several others is shown in Figure 1 on the "Plant" example from the private dataset. Our algorithm is able to cleanly matte the leaves while maintaining the "holes" through which the background shows. Additional examples of alpha mattes generated by our algorithm are shown in Figure 3.

4.2. Foreground/Background Accuracy

A particular focus of our algorithm is achieving high accuracy for the foreground and background at each unknown pixel. To determine accuracy of our foreground estimation, we compare our method to several others using the true foreground colors of the public dataset in [11]. Because the foreground colors were only provided in a raw 16-bit format without gamma correction and white balance, we manually set the gamma and white balance of the foreground images and their corresponding raw 16-bit input images to produce input images similar in color appearance to the standard input images of the public dataset.

Algorithm	MAD	MSE
Our Method	0.063	0.005
Closed-Form Matting [6]	0.082	0.016
Bayesian Matting [3]	0.11	0.02
Robust Matting [17]	0.16	0.031

Table 4. Foreground alpha product (αF) error over public dataset from [11]. The mean absolute difference (MAD) and mean squared error (MSE) are shown.

We compare our foreground colors to those generated simultaneously with alpha by Bayesian Matting [3], those generated as a postprocess by Closed-Form Matting [6], and those generated as an initialization to Robust Matting [17]. The error is computed by taking the difference of the foreground color multiplied by the ground truth α, since pixels with a high α are more important visually when compositing.

The results are shown in Table 4. The MAD and MSE averaged over all 27 images are shown for each algorithm. Our algorithm outperformed all other methods on 18 of the 27 images using the MAD as a metric, and 20 of the 27 using MSE.

Several examples of foregrounds and backgrounds generated by our algorithm are shown in Figure 2 in comparison to those generated by Closed-Form Matting [6]. The regions being shown in each close-up are indicated by red and blue boxes corresponding to foreground and background respectively. In each example, the foreground is multiplied by α and the background by $(1-\alpha)$ using the α computed by the respective algorithm. The foreground and background colors produced by our method appear more plausible, confirming the quantitative results in Table 4. For example, this is easily seen in the top example of a plant, where the leaves of the plant appear blue and the spaces in between the leaves appears green in the Closed-Form Matting results, but appear more correct in our results. Additional examples are shown in Figure 3.

Unfortunately, only the true foreground colors from the public dataset of [11] are available, so a quantitative evaluation of this dataset cannot be performed for the background colors. Based on the visual appearance, we predict that the accuracy of the background would be similar to that of the foreground for each of these algorithms. For example, note how in Figure 2 our method appears to better reconstruct the occluded backgrounds.

Algorithm	Rank (of 11)	Troll small	Troll large	Doll small	Doll large	Donkey small	Donkey large	Elephant small	Elephant large	Plant small	Plant large	Pineapple small	Pineapple large	Bag small	Bag large	Net small	Net large
Our Method	3.8	16.9	27.1	10.4	14.2	6.5	8.0	3.4	8.3	6.4	8.9	6.8	11.0	25.9	28.3	40.7	51.2
Rhemann [9]	1.9	14.9	24.5	6.7	9.5	4.6	6.1	2.6	5.4	7.5	9.9	6	10.1	26.1	26.7	23.8	25.6
Closed-Form [6]	2.8	12.7	21.9	5.9	8.5	4.6	6.1	2.2	4.6	9.3	12.1	8.3	14.9	34.2	32.4	26.5	25.7
Robust [17]	3.8	17.3	28.4	10.1	16.9	4.8	6.5	2.8	7.3	7.3	14	6.8	14.6	22.7	26.1	34.4	37
High-Res [10]	4.5	18.6	25.8	8.6	14.1	5.0	6.2	2.5	8.3	7.8	14	8.5	18.1	35.3	38.1	38.7	54.6
Random Walk [4]	6.5	17.9	20.3	11.3	15.6	5.8	7	3.4	6.7	13.1	22.1	12.3	18	44.1	43.5	75.1	81.8
Geodesic [1]	7.3	26.9	38.5	14.2	16.5	11.7	14	7.6	15.1	12.8	16.7	7.3	12.1	37.3	37.4	48.6	50

Table 2. Sum of absolute difference on private dataset from [11]. The table includes the top seven of eleven algorithms and data from two of the three trimaps. Iterative BP [18], Easy Matting [5], Bayesian Matting [3], and Poisson Matting [15] finish out the rankings.

Algorithm	Rank (of 11)	Troll small	Troll large	Doll small	Doll large	Donkey small	Donkey large	Elephant small	Elephant large	Plant small	Plant large	Pineapple small	Pineapple large	Bag small	Bag large	Net small	Net large
Our Method	4.1	0.9	2.6	0.8	1.2	0.5	0.7	0.2	0.7	0.5	0.7	0.6	1.0	1.9	2.1	3.3	4.5
Rhemann[9]	1.9	0.8	2.4	0.3	0.5	0.3	0.4	0.1	0.3	0.7	0.7	0.4	0.7	2.0	1.9	1.3	1.5
Closed-Form[6]	3.1	0.5	1.8	0.3	0.4	0.3	0.4	0.1	0.3	1.2	1.4	0.8	1.6	3.0	2.7	1.3	1.2
Robust[17]	3.5	1.1	2.8	0.7	1.5	0.3	0.4	0.1	0.5	0.5	1.2	0.5	1.5	1.5	1.8	2.4	2.3
High-Res[10]	4.2	1.2	2.2	0.5	1.1	0.3	0.4	0.1	0.7	0.6	1.2	0.8	2.0	3.2	3.4	2.6	4.3
Iterative BP [18]	6.4	1.7	2.6	1.5	2.6	0.5	0.7	0.2	0.8	1.1	2	1	2	2.8	3.3	3	3.8
Random Walk [4]	6.7	1	1.1	1	1.7	0.5	0.6	0.2	0.4	2	3.4	1.6	2.3	4.6	4.4	8.3	9.4

Table 3. Mean squared error on private dataset from [11]. The table includes the top seven of eleven algorithms and data from two of the three trimaps. Geodesic Matting [1], Bayesian Matting [3], Easy Matting [5], and Poisson Matting [15] finish out the rankings.

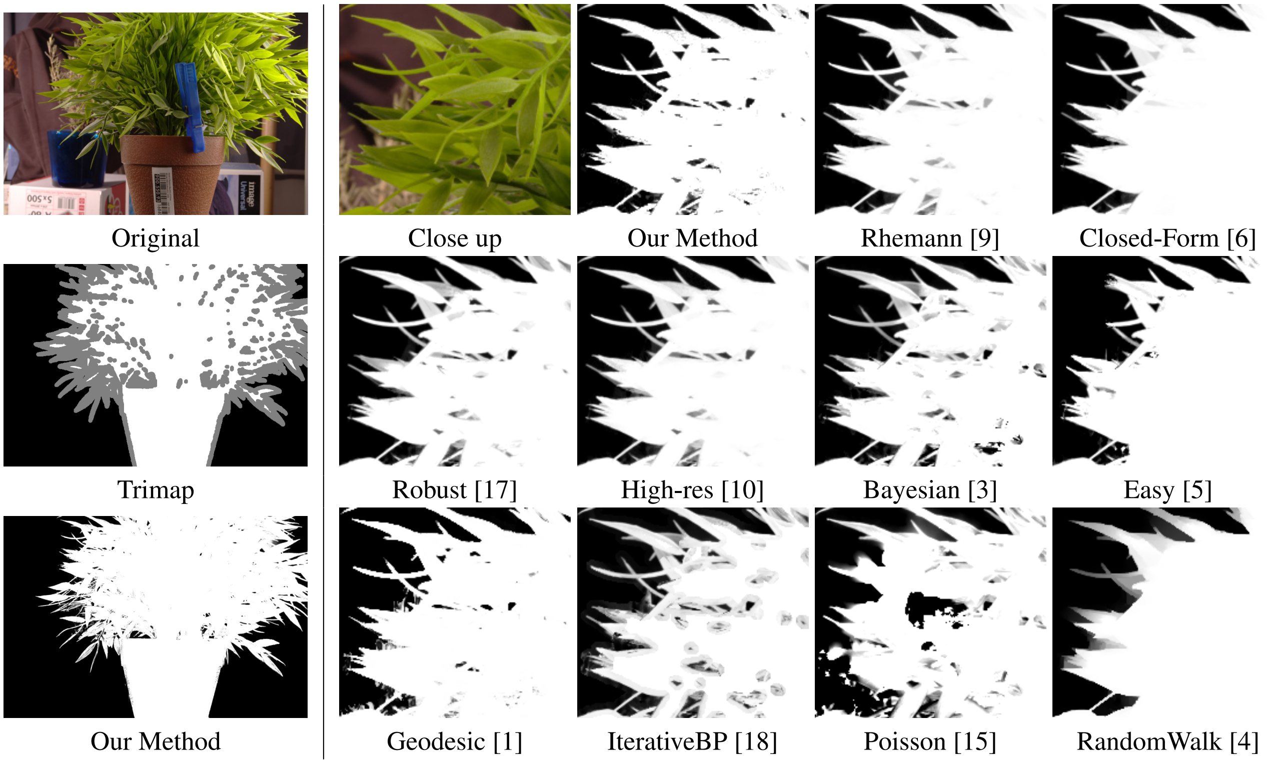
Original
Close up
Our Method
Rhemann [9]
Closed-Form [6]
Trimap
Robust [17]
High-res [10]
Bayesian [3]
Easy [5]
Our Method
Geodesic [1]
IterativeBP [18]
Poisson [15]
RandomWalk [4]

Figure 1. The left column shows our results on the "Plant" example from [11]. The remaining columns show a comparison of our technique to several others on a region of this image.

5. Conclusion

We have introduced a new method for computing a solution to the full matting equation for a given image by simultaneously estimating the foreground, background, and alpha at each unknown pixel. We do so by including additional terms to affect the foreground, background, and alpha values, formulating the terms into a single energy equation,

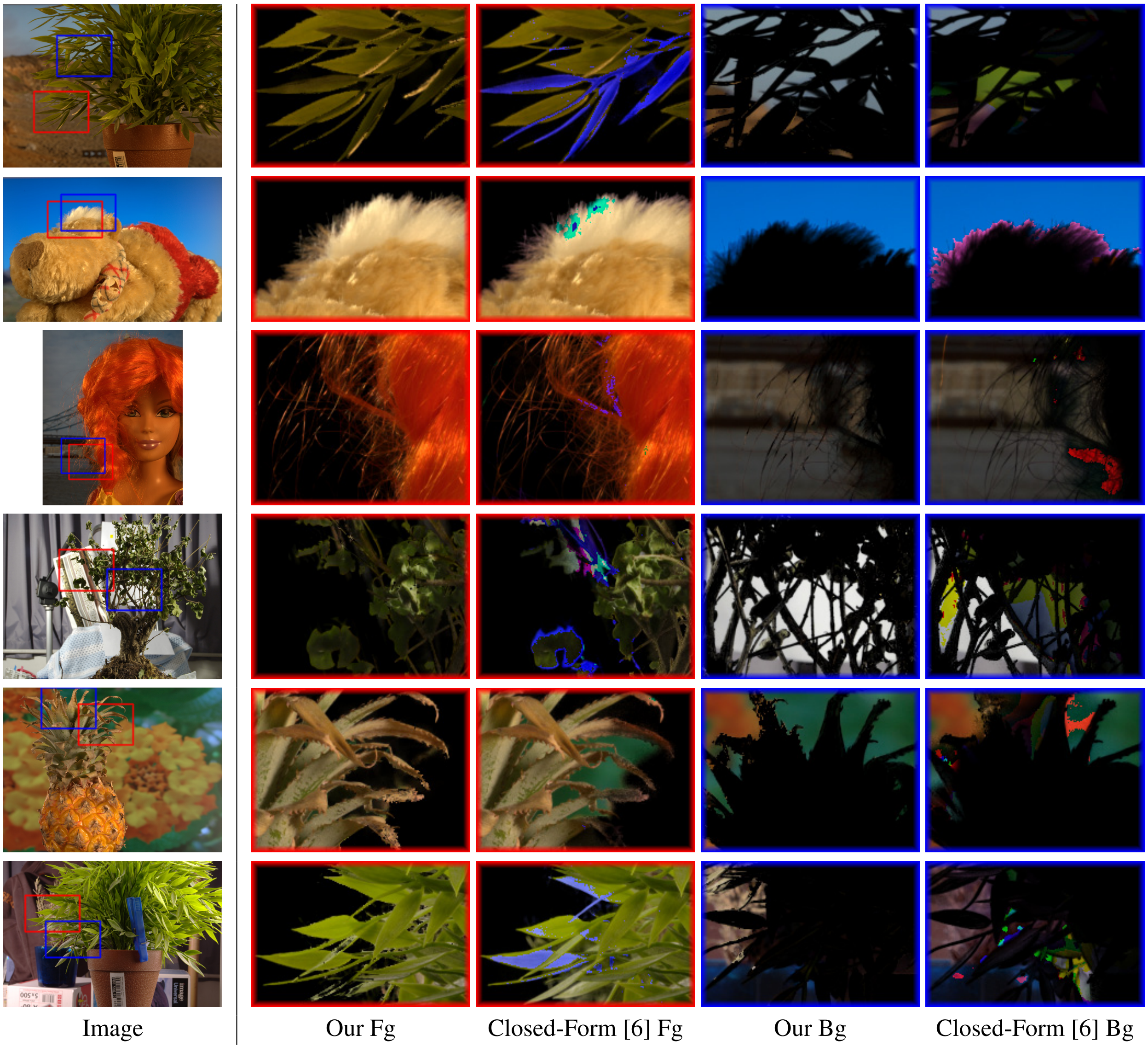

Figure 2. Comparison of the foreground and background generated by our method and Closed-Form Matting [6]. The foreground is multiplied by α and the background by $(1 - \alpha)$ using the α computed by the respective method. The red and blue boxes in the original images indicate the close-up areas for the foreground and background respectively.

and minimizing that equation using gradient descent with momentum and a partial form of graduated non-convexity. This focus on estimating not just alpha but also the foreground and background allows us to optimize over explicit constraints on the foreground and background, which most current algorithms that optimize only over alpha cannot do. Our results are comparable to the best current algorithms for computing the alpha matte and show superior performance in computing the foreground colors in all tests performed.

While this work improves foreground and background estimation for mattes and provides competitive results in computing alpha, our energy function is complex and difficult to optimize. Future work could improve this by incorporating more advanced optimization techniques or by simplifying the terms to allow for easier minimization. Additional or improved terms could also be added to potentially improve results.

References

[1] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. *IEEE ICCV 2007*, pages 1–8, 2007.

[2] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[3] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. *IEEE CVPR*, 2:264–

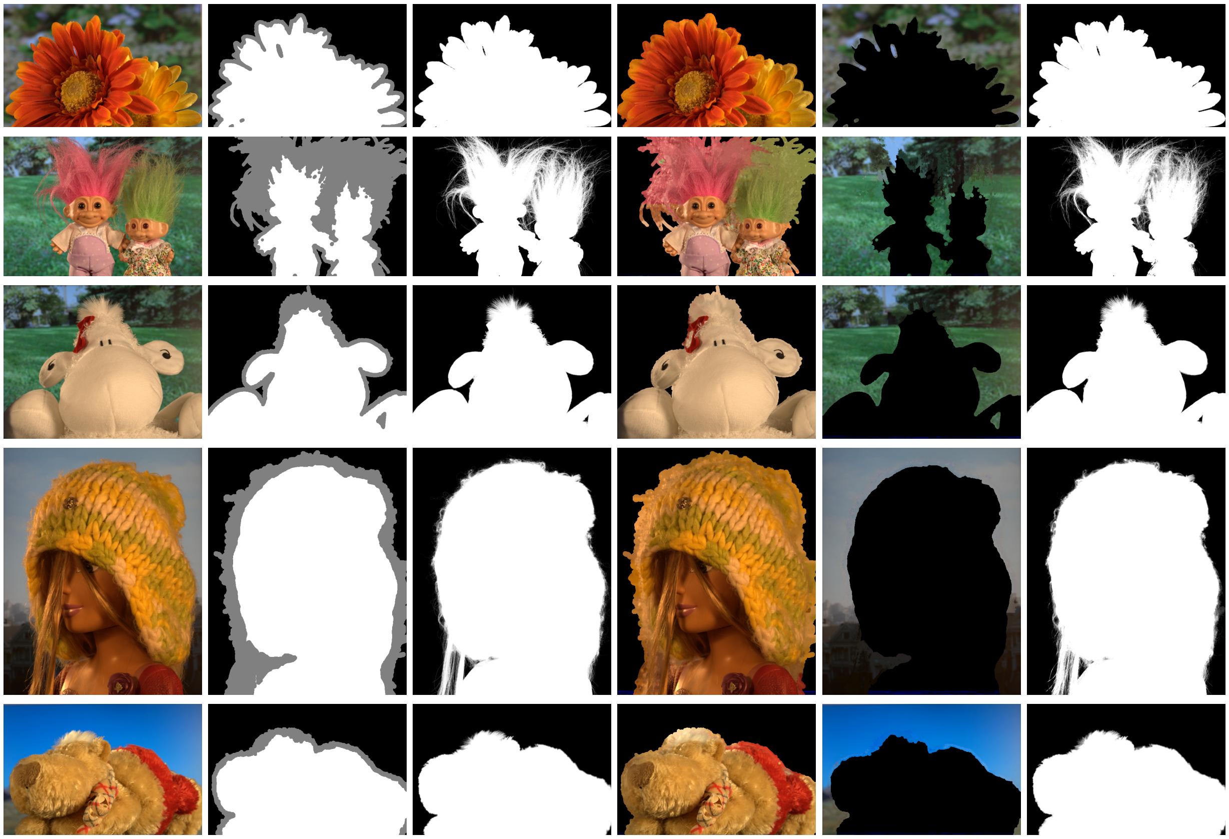

Figure 3. The foreground, background, and alpha generated by our algorithm.

271, 2001.

[4] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. *VIIP*, pages 423–429, 2005.

[5] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting: A stroke based approach for continuous image matting. *Proceedings of Eurographics 2006*, pages 567–576, 2006.

[6] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE PAMI*, 30(2):228–242, 2008.

[7] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *IEEE PAMI*, 30(10):1699–1712, 2008.

[8] C. R. Maurer, Jr., R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE PAMI*, 25(2):265–270, 2003.

[9] C. Rhemann, C. Rother, and M. Gelautz. Improving color modeling for alpha matting. *BMVC*, pages 1155–1164, 2008.

[10] C. Rhemann, C. Rother, A. Rav-Acha, M. Gelautz, and T. Sharp. High resolution matting via interactive trimap segmentation. *CVPR*, 2008.

[11] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. *IEEE CVPR*, June 2009.

[12] M. Ruzon and C. Tomasi. Alpha estimation in natural images. In *IEEE CVPR*, volume 1, pages 18–25, 2000.

[13] D. Singaraju, C. Rother, and C. Rhemann. New appearance models for natural image matting. *IEEE CVPR*, 2009.

[14] D. Singaraju and R. Vidal. Interactive image matting for multiple layers. *IEEE CVPR*, pages 1–7, 2008.

[15] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *ACM SIGGRAPH*, pages 315–321, 2004.

[16] J. Wang, M. Agrawala, and M. Cohen. Soft scissors : An interactive tool for realtime high quality matting. *ACM SIGGRAPH*, pages 9:1–9:6, 2007.

[17] J. Wang and M. Cohen. Optimized color sampling for robust matting. *IEEE CVPR*, pages 1–8, 2007.

[18] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *IEEE ICCV*, pages 936–943, 2005.

[19] Y. Zheng and C. Kambhamettu. Learning based digital matting. *ICCV*, 2009.