New Appearance Models for Natural Image Matting

Dheeraj Singaraju
Johns Hopkins University
Baltimore, MD, USA.
dheeraj@cis.jhu.edu

Carsten Rother
Microsoft Research
Cambridge, UK.
carrot@microsoft.com

Christoph Rhemann *
Vienna University of Technology
Vienna, Austria.
rhemann@ims.tuwien.ac.at

Abstract

Image matting is the task of estimating a fore- and background layer from a single image. To solve this ill posed problem, an accurate modeling of the scene's appearance is necessary. Existing methods that provide a closed form solution to this problem, assume that the colors of the foreground and background layers are locally linear. In this paper, we show that such models can be an overfit when the colors of the two layers are locally constant. We derive new closed form expressions in such cases, and show that our models are more compact than existing ones. In particular, the null space of our cost function is a subset of the null space constructed by existing approaches. We discuss the bias towards specific solutions for each formulation. Experiments on synthetic and real data confirm that our compact models estimate alpha mattes more accurately than existing techniques, without the need of additional user interaction.

1. Introduction

Image matting addresses the problem of estimating the partial opacity of each pixel in a given image. In particular, one assumes that the intensity I_i of the i^{th} pixel can be written as the convex combination of a foreground intensity F_i and a background intensity B_i, as

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i, \tag{1}$$

where α_i is referred to as the pixel's partial opacity value or *alpha matte*. By definition, this value is constrained to take values in $[0, 1]$. We note that for each pixel in color images, (1) gives us 3 equations in 7 unknowns. Consequently, the image matting problem is highly under-constrained. To this effect, the user is required to provide some additional information in order to make the problem well posed. Such information is typically provided in the form of a *trimap* by marking different regions in the image as (a) foreground; $\alpha = 1$, (b) background; $\alpha = 0$, and (c) unknown; $\alpha \in [0, 1]$.

The goal of image matting algorithms is therefore to estimate the alpha mattes of the pixels in the unknown region. Incipient methods such as [12] use the trimap to construct basic color models for the foreground and background, which are subsequently used to estimate the alpha mattes in the unknown region as per (1). Due to their naive modeling schemes, such algorithms fail on images with complex intensity variations. Methods such as [2, 15] solve such issues by using local propagation techniques to estimate the alpha mattes. However, their good performance is subject to the use of a tight trimap.

Subsequent research in image matting witnessed the use of a number of algorithms originally intended for image segmentation [11, 17, 3, 4, 1, 20]. In most cases, these algorithms use a sparse trimap to estimate a binary segmentation, which is then used to generate a tight trimap for the image. The alpha mattes are estimated using this tight trimap and can then potentially be refined by alternating between re-estimation of the alpha mattes and the trimap. It is important to appreciate the fact that matting and segmentation are different problems. In order to estimate the alpha mattes of an image, one needs to develop extremely accurate models for the scene's appearance. This is not so for the case of segmentation, where it suffices to define image features that help to distinguish the object from the background.

Consequently, recent work in image matting has seen a surge of research towards developing algorithms that exploit various features specific to the matting problem [6, 19, 16, 7, 10, 9]. It was shown in [6] that if one assumes the intensities of the foreground and background layers to vary linearly in small image patches, then the alpha mattes could be estimated in a closed form fashion. It was later demonstrated that the performance of local propagation based methods such as [6] could be improved by additionally learning global color models [19, 16, 9]. Recent work has also focused on enforcing sparsity of the alpha mattes [7, 10]. For a more detailed review of image matting algorithms, we refer the reader to [18].

In this paper, we propose to improve the state of the art for image matting by developing accurate models for a scene's appearance, and hence fundamentally improve the building blocks of matting algorithms. In particular, we

*This work was supported in part by Microsoft Research Cambridge through its PhD Scholarship Programme and a travel sponsorship.

focus on the *Matting Laplacian* proposed by [6], which is a matrix characterizing a cost function for image matting. This cost function is derived under the assumption that the foreground and background layers of each image patch exhibit linear variation in the intensities. As we show, this assumption can be an overfit for the image data, if the colors of either layer are locally constant. Specifically, we show that for small perturbations in the image data, [6] might construct a null space of possible solutions, which is larger than desired, thereby making the problem more ambiguous. We then show how one can construct more compact models for the alpha mattes, which have a null space of provably smaller dimension than that of [6]. Furthermore, the different formulations have a different bias towards specific solutions, which we will discuss in detail. Compelling experiments on synthetic and real images validate our claims. Consequently, we present a new framework for closed form solutions to image matting, which is theoretically principled and yields high quality alpha mattes.

Note that since the Matting Laplacian has been used in [6, 19, 16, 7, 10, 9] for regularization of the alpha mattes, our framework can be used to improve the performance of these algorithms. Furthermore, our framework can be applied to alternative applications such as light mixture estimation [5], since they use variants of the Matting Laplacian.

2. The Matting Laplacian: A Review

Omer and Werman [8] empirically showed that the distribution of colors in real images is locally linear in RGB space. Inspired by this work, Levin *et al*. [6] state that given any small patch in the query composite image, the intensities of the corresponding foreground and background layers can be assumed to lie on lines in RGB space. In particular, for a small patch $\mathcal{W}_i$ centered around pixel i, there exist colors $(F_{i1}, F_{i2}, B_{i1}, B_{i2})$ such that the foreground and background colors (F_j, B_j) of each pixel $j \in \mathcal{W}_i$ can be expressed as

$$\begin{aligned} F_j &= \beta_j^F F_{i1} + (1-\beta_j^F) F_{i2}, \text{ and} \\ B_j &= \beta_j^B B_{i1} + (1-\beta_j^B) B_{i2}. \end{aligned} \tag{2}$$

Under this assumption, [6] showed that there exist affine functions $v_i = (a_i^R, a_i^G, a_i^B, b_i)$ characteristic to the patch $\mathcal{W}_i$, such that the alpha matte α_j of each pixel $j \in \mathcal{W}_i$ can be written as

$$\alpha_j = a_i^R I_j^R + a_i^B I_j^B + a_i^G I_j^G + b_i, \tag{3}$$

where I_j^R, I_j^G and I_j^B refer to the RGB values of pixel j.

The problem of estimating the mattes $\boldsymbol{\alpha}$ in the image can consequently be posed as one of finding the minimizer of

$$J(\boldsymbol{\alpha}, v) = \sum_{i \in \mathcal{V}} \left[\sum_{j \in \mathcal{W}_i} \left(\alpha_j - a_i^R I_j^R - a_i^B I_j^B - a_i^G I_j^G - b_i \right)^2 \right], \tag{4}$$

where $\mathcal{V}$ is the set of all pixels in the image. Essentially, this corresponds to minimizing the residual of the affine model v_i defined in (3) for every small patch $\mathcal{W}_i$. Note that [6] actually uses a modification of the cost function $J(\boldsymbol{\alpha}, v)$ by introducing an additional regularization term, as

$$J_\epsilon(\boldsymbol{\alpha}, v) = J(\boldsymbol{\alpha}, v) + \epsilon \sum_{i \in \mathcal{V}} \left({a_i^R}^2 + {a_i^G}^2 + {a_i^B}^2 \right). \tag{5}$$

The regularization term is introduced in order to enforce the affine function $(a_i^R, a_i^G, a_i^B, b_i) = (0, 0, 0, c), c \in [0, 1]$, or in other words, enforce constancy of alpha mattes over the patch $\mathcal{W}_i$. The motivation for this is twofold. Firstly, the user provided scribbles are typically sparse and constrain far fewer pixels than the perfectly tight trimap. Hence, for many pixels in the image, an α of 0 or 1 is desired, independent of the appearance model. Secondly, real images often have highly textured patches that do not satisfy the color line model, but nonetheless may have uniform alpha mattes across the patch. The alpha mattes of such patches can be explained by an affine model of the form $v = (0, 0, 0, c), c \in [0, 1]$. Therefore, the model allows for certain complex cases beyond the color line model of (2).

Note that the constructed cost function $J_\epsilon(\boldsymbol{\alpha}, v)$ depends on two unknown quantities; the alpha mattes $\boldsymbol{\alpha}$ and the affine functions v. However, [6] showed that this can be reduced to a cost function that depends solely on the alpha mattes. For the sake of simplicity, let us define matrices $G_i \in \mathbb{R}^{(|\mathcal{W}_i|+3)\times 4}$ and $\bar{\alpha}_i \in \mathbb{R}^{|\mathcal{W}_i|+3}$. The first $|\mathcal{W}_i|$ rows of G_i are given by $\begin{bmatrix} I_j^R & I_j^G & I_j^B & 1 \end{bmatrix}, j \in \mathcal{W}_i$ and the last three rows are given by $\begin{bmatrix} \sqrt{(\epsilon)} \boldsymbol{I}_3 & \mathbf{0} \end{bmatrix}$, where $\boldsymbol{I}_n$ is an identity matrix of size $n \times n$. The first $|W_i|$ entries of $\bar{\alpha}_i$ are given by $\alpha_j, j \in \mathcal{W}_i$ and the last three entries are equal to 0. Given this notation, $J_\epsilon(\boldsymbol{\alpha}, v)$ can be rewritten as

$$J_\epsilon(\boldsymbol{\alpha}, v) = \sum_{i \in \mathcal{V}} \|G_i v_i - \bar{\alpha}_i\|^2. \tag{6}$$

Now, we see that we can estimate the affine function v_i for each patch $\mathcal{W}_i$, as

$$v_i = \text{argmin}_v \|G_i v - \bar{\alpha}_i\|^2 = (G_i^\top G_i)^{-1} G_i^\top \bar{\alpha}_i. \tag{7}$$

Therefore, using the expression for v_i from (7), we see that the cost function $J_\epsilon(\boldsymbol{\alpha}, v)$ can be reduced to a cost function dependent on the alpha mattes only, as

$$\begin{aligned} J_\epsilon(\boldsymbol{\alpha}) &= \sum_{i \in \mathcal{V}} \left[\bar{\alpha}_i^\top (\boldsymbol{I}_{|\mathcal{W}_i|+3} - G_i (G_i^\top G_i)^{-1} G_i^\top) \bar{\alpha}_i \right] \\ &= \boldsymbol{\alpha}^\top L \boldsymbol{\alpha}. \end{aligned} \tag{8}$$

The matrix L is referred to as the *Matting Laplacian* and we refer the reader to [6] for a detailed derivation of its entries. Note that the constructed cost function is quadratic in the alpha mattes. Therefore, the minimizer of this cost function

can be estimated by solving a linear system. Hence, we have a closed form solution for the alpha mattes.

What is the null space? It is of interest to inspect the nature of the solutions to this system for a small patch $\mathcal{W}_i$ in the image. Let us define a matrix $M_i = \boldsymbol{I}_{|\mathcal{W}_i|+3} - G_i(G_i^\top G_i)^{-1}G_i^\top$. Note that $M_i G_i = G_i - G_i = \mathbf{0}$. Therefore, by construction, the columns of the matrix G_i and their linear combinations are null vectors of the matrix M_i. If the foreground and background layers truly satisfy the color line model, we know that the vector of alpha mattes $\bar{\alpha}_i$ is given by $G_i v_i$. In other words, the vector of alpha mattes is given by a linear combination of the columns of G_i and hence lies in the null space of M_i. Also, in such cases G_i is of rank 4, and therefore M_i has a null space of dimension 4. As a result, the vector of alpha mattes is only one of the potential minimizers of the constructed cost function. For instance, we have seen from our earlier discussion that the constant solution is also part of the null space. It is precisely for this reason that the user is required to mark scribbles in the image and embed constraints, so that the algorithm can recover the true alpha mattes as the minimizers of $J_\epsilon(\boldsymbol{\alpha})$.

2.1. Limitations of the Matting Laplacian

In practice, it can be observed that [6] does not always recover the ground truth alpha mattes. This is due to several reasons, a few of which are outlined below.

1. Violation of the color-line model: In natural images, it often happens that the image data does not satisfy the color line model. In the case of complex intensity variations, it is obvious that the color line model is too simple to explain the image intensities. In such scenarios, one would have to resort to data driven schemes such as [19, 9] for generating candidate foreground and background colors. However, we are particularly interested in the case when the intensity variations are much simpler than the color line, such as being locally constant. As we shall show later, the true dimension of possible solutions for such image patches is less than 4 and hence the algorithm of [6] provides an overfit. Since such patches occur commonly in natural images, it is of interest to construct more compact models for the alpha mattes in such patches.
2. Insufficient user interaction: Recall that the system constructed by [6] has a 4-dimensional null space for each local image patch considered by the algorithm. Hence, high level user interaction is required to resolve any ambiguities. One could potentially incorporate prior knowledge in order to bias the system towards certain family of solutions. [6] biases the mattes to be locally constant, which can prove to be unsatisfactory in practice. As shown in Figure 2(c,e) when the user marks pixels corresponding to one layer only, [6] assigns constant alpha mattes equal to 0 or 1 to all unmarked pixels, based on whether the scribbles correspond to background or foreground respectively. We will address this problem in detail later. Alternatively, [7] biases the mattes towards 0 or 1 using non-linear priors. This resulting system is however prohibitively slow in practice. Moreover, the mattes estimated at the scribbled pixels do not necessarily match the values specified by the user. Note that one could also predict the alpha value for each pixel with a certain confidence value, as in [19, 9]. However, such frameworks are beyond the scope of this paper.

3. Appearance Models Beyond Color Lines

In this work, we consider the cases when the color line models are violated, such that at least one of the foreground or background layers lie on a point rather than a line in color space. Specifically, we inspect the rank of the matrix G_i introduced in Section 2 and analyze the cases when the rank of the matrix is less than 4. We show that for these cases, the intensities of the composite image lie on linear/affine spaces of dimension less than or equal to 4. Since in general, the Matting Laplacian of [6] has a 4 dimensional null space, we demonstrate that our method is more robust to noisy data.

Further, we show that the solution space of our formulation includes the constant alpha solution, which is important for highly textured areas. We will also show that the model of [6] has a natural bias towards constant solutions, while our model has a natural bias for a constant 0 (or 1) solution. Both biases, our and [6] are not optimal, since the ideal bias is towards 0 *and* 1 simultaneously. Unfortunately, the ideal bias leads to a non-linear system, *e.g.* as shown in [7], which is very challenging to optimize and hence [7] is not ranked very well in recent evaluations [10]. We will see in section 4, that our method performs on average favourably, which suggests that robustness to noise overweighs the influences of the different bias. Also, we will show that for the special case where the user specified unknown region is bounded by constraints of one type only, e.g. only foreground, the bias of our formulation towards 0 is clearly preferable.

3.1. Line-Point Color Models

We first consider the case when the colors of exactly one layer satisfy the color line model, while the colors of the other layer are constant and hence satisfy a *color point* model. Without loss of generality, assume that the foreground intensities are constant and that the background intensities lie on a color line. It is easy to check that if the type of models were interchanged, our following analysis would result in the same cost function. Now, by the hypothesis, $\forall j \in \mathcal{W}_i, F_j = F$ and $\forall j \in \mathcal{W}_i, B_j = \beta_j B_1 + (1-\beta_j)B_2$. Therefore, the composite intensity I_j of a pixel $j \in \mathcal{W}_i$ can be expressed as

$$\begin{aligned} \forall j \in \mathcal{W}_i : I_j &= \alpha_j F + (1-\alpha_j)[\beta_j B_1 + (1-\beta_j)B_2] \\ &= \alpha_j(F-B_2) + (1-\alpha_j)\beta_j(B_1-B_2) + B_2. \end{aligned} \tag{9}$$

For this scenario, we derive two important results as given by Theorem 1.

Theorem 1 *Consider an image patch $\mathcal{W}_i$ around a pixel $i \in \mathcal{V}$, such that the RGB intensities of the pixels in the patch, satisfy the line-point color models. Define a matrix $G_i^{LP} \in \mathbb{R}^{|\mathcal{W}_i| \times 3}$, whose rows are given as $\begin{bmatrix} I_j^R & I_j^G & I_j^B \end{bmatrix}, j \in \mathcal{W}_i$. Also define a matrix $\bar{\alpha}_i \in \mathbb{R}^{|\mathcal{W}_i|}$, whose entries are given by $\alpha_j, j \in \mathcal{W}_i$*

1. *If the foreground color point does not lie on the background color line and there are at least three pixels $a, b, c \in \mathcal{W}_i$ such that $\alpha_a \neq \alpha_b \neq \alpha_c$ and $(1-\alpha_a)\beta_a \neq (1-\alpha_b)\beta_b \neq (1-\alpha_c)\beta_c$, then $Rk(G_i^{LP}) = 3$.*
2. *If $Rk(G_i^{LP}) = 3$, the alpha matte α_j of each pixel $j \in \mathcal{W}_i$, can be expressed as a linear function of the pixel's intensities, via unique coefficients $v_i = (a_i^R, a_i^B, a_i^G) \in \mathbb{R}^3$ characteristic to the patch $\mathcal{W}_i$, as*

$$\begin{aligned} &\forall j \in \mathcal{W}_i : \alpha_j = a_i^R I_j^R + a_i^B I_j^B + a_i^G I_j^G \\ &\implies \bar{\alpha}_i = G_i^{LP} v_i. \end{aligned} \tag{10}$$

Proof.

1. Note that by definition, we have

$$G_i^{LP} = \underbrace{\begin{bmatrix} \vdots & \vdots & \vdots \\ \alpha_j & (1-\alpha_j)\beta_j & 1 \\ \vdots & \vdots & \vdots \end{bmatrix}}_{\Gamma_i} \underbrace{\begin{bmatrix} F - B_2 & B_1 - B_2 & B_2 \end{bmatrix}^\top}_{H_i}. \tag{11}$$

   Since by hypothesis, F does not lie on the line spanned by B_1 and B_2, the matrix H_i is full rank, and hence of rank 3. If the alpha mattes do not lie in any critical configuration, then the matrix Γ_i in (11) is also full rank and hence of rank 3. Consequently, the matrix G_i^{LP} is also of rank 3.

2. If G_i^{LP} is rank 3, we know that H_i also must be rank 3 and hence invertible. Hence, we can rewrite (11) as

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ \alpha_j & (1-\alpha_j)\beta_j & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} = G_i^{LP} H_i^{-1}. \tag{12}$$

   Therefore, we see that there exists a unique linear function v_i given by the first column of H_i^{-1}, that relates the alpha matte of each pixel to its RGB intensities.

■

Observe that this is different from (3) derived under the color line assumption, where the alpha mattes were affine functions of the intensities. Now, there is no constant term present in the expression for the alpha mattes. As earlier, we can estimate the unknowns by minimizing the cost function

$$J_3(\boldsymbol{\alpha}, v) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{W}_i} (\alpha_j - v_i^\top I_j)^2 = \sum_{i \in \mathcal{V}} \|\bar{\alpha}_i - G_i^{LP} v_i\|^2. \tag{13}$$

If we assume that the alpha mattes of the pixels in the patch $\mathcal{W}_i$ are known, the coefficients v_i for each patch $\mathcal{W}_i$ can be estimated by minimizing the function $J_3(\boldsymbol{\alpha}, v)$, as

$$v_i = \underset{v}{\operatorname{argmin}} \|G_i^{LP} v - \bar{\alpha}_i\|^2 = (G_i^{LP^\top} G_i^{LP})^{-1} G_i^{LP^\top} \bar{\alpha}_i. \tag{14}$$

Substituting the expression for v_i from (14), we see that the cost function $J_3(\boldsymbol{\alpha}, v)$ can be reduced to a cost function dependent on the alpha mattes only, as

$$\begin{aligned} J_3(\boldsymbol{\alpha}) &= \sum_{i \in \mathcal{V}} \left[\bar{\alpha}_i^\top (\boldsymbol{I}_{|\mathcal{W}_i|} - G_i^{LP} (G_i^{LP^\top} G_i^{LP})^{-1} G_i^{LP^\top}) \bar{\alpha}_i \right] \\ &= \boldsymbol{\alpha}^\top L_3 \boldsymbol{\alpha}. \end{aligned} \tag{15}$$

As earlier, the constructed cost function is quadratic in the alpha mattes. Therefore, the alpha mattes can be estimated in closed form by solving a linear system.

What is the null space? We now repeat the exercise of inspecting the nature of the solutions to this system for a small patch $\mathcal{W}_i$ in the image. Let us define a matrix $M_i^{LP} = \boldsymbol{I}_{|\mathcal{W}_i|} - G_i^{LP} (G_i^{LP^\top} G_i^{LP})^{-1} G_i^{LP^\top}$. By an earlier argument, we know that the columns of the matrix G_i^{LP} and their linear combinations, are null vectors of the matrix M_i^{LP}. Recall from Theorem 1 that $\bar{\alpha}_i = G_i^{LP} v_i$. Therefore, we can conclude that the vector of true alpha mattes lies in the null space of M_i^{LP}.

Since G_i^{LP} is rank 3, we have that the null space of M_i^{LP} is of dimension 3. When the image data exactly obeys the line-point model, the RGB intensities lie on a plane spanned by the model parameters F, B_1 and B_2. Since the locus of any point x on the plane can be expressed in terms of the perpendicular to the plane $d \in \mathbb{R}^3$ as $d^\top x = 1$, we note that there exists a linear function d such that $\forall j \in \mathcal{W}_i : d^\top I_j = 1$. Consequently, the matrix G_i constructed by Levin *et al.* [6] is also rank 3, because the last column comprising of all 1s can be expressed as a linear combination of the first 3 columns that contain the image intensities. Hence, the null space of the Matting Laplacian is also 3 dimensional.

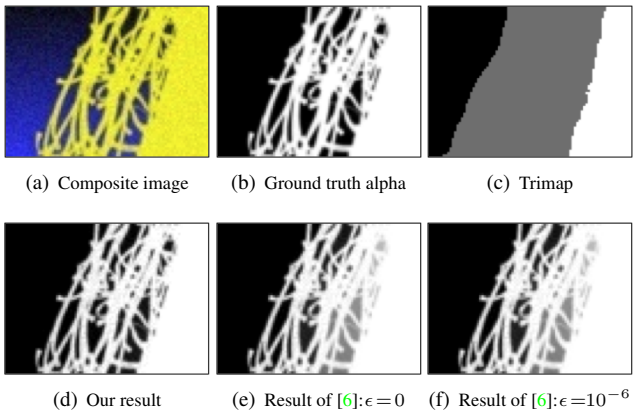
Note that in the statement of Theorem 1, we have mentioned that there must be at least three pixels in the window $a, b, c \in \mathcal{W}_i$ such that $\alpha_a \neq \alpha_b \neq \alpha_c$ and $(1-\alpha_a)\beta_a \neq (1-\alpha_b)\beta_b \neq (1-\alpha_c)\beta_c$. This corresponds to the condition that the composite intensities completely span the plane defined by the foreground color point and the background color line. However, when the alpha mattes of all the pixels in a window are constant, *i.e.* $\forall j \in \mathcal{W}_i, \alpha_j = k \in [0, 1]$,

the intensities span only a subset of the plane defined by the foreground color point and the background color line. Specifically, they span a line on this plane and hence the rank of G_i^{LP} is 2 in this case. Since this line is a part of the plane discussed above, all the composite intensities do still satisfy the locus $\forall j \in \mathcal{W}_i : d^\top I_j = 1$. Hence, *we see that our constructed cost function naturally allows for locally constant alpha mattes*, because there exists a linear function $kd \in \mathbb{R}^3$ such that $\forall j \in \mathcal{W}_i, \alpha_j = (kd)^\top I_j = k(d^\top I_j) = k$. This is an important property when dealing with trimaps that are not tight.

What happens on real, noisy images? Unfortunately, real data is always corrupted by some noise. In case the image data has a slight perturbation from the exact color-line model, the intensities do no longer lie on a plane. Hence, there is no $d \in \mathbb{R}^3$ such that $\forall j \in \mathcal{W}_i : d^\top I_j = 1$, and the matrix G_i constructed by [6] is rank 4. As a result, the null space of the Matting Laplacian is rank 4. The null space obtained using our framework, however, still has a null space of dimension 3, by construction. Since the first 3 columns of G_i are exactly the same as G_i^{LP}, the column span of G_i^{LP} is a subset of the column span of G_i. Therefore, the null space obtained using our framework is a strict subset of the null space constructed by [6]. This implies that our proposed model is more compact than that of [6]. Hence, the *key* observation is that the extra degree of freedom of the Matting Laplacian is used to explain image noise.

What is the bias? The space of solutions for the alpha matte given by our model or the model of [6] is typically quite large (see [7]). However, there is an implicit bias towards the result given by the linear solver. In fact, this bias is enforced *naturally by the structure of the different cost functions*. While [6] naturally biases the mattes to be locally constant, our new cost function pushes the alpha mattes towards 0. By construction, the Matting Laplacian L has the vector with all equal entries, as its trivial null vector. Therefore, [6] is biased towards estimating locally smooth alpha mattes. On the other hand, our cost function L_3 is a positive semi-definite matrix and not necessarily a Laplacian matrix. It has a trivial null vector which has all entries equal to 0, and consequently our algorithm estimates alpha mattes with a bias towards 0. Note that by solving for $1 - \alpha$ rather than α, we can also bias the alpha mattes towards 1.

Result on toy data. Figure 1 illustrates the advantage of our proposed model. The yellow foreground is a point in RGB space, and the background lies on a color line, varying from light to dark blue. Hence, we have a perfect line-point color model. We add some noise to the composite image in order to slightly perturb it from this model. Notice that [6] produces erroneous alpha mattes due to its larger null space, and our method recovers a much better alpha matte. As expected [6] has a bias towards locally smooth mattes, and careful inspection shows that our result has a tiny shift towards 0. Furthermore, note that the trimap is not very tight, and our method correctly recovers those pixels which should be truly 0 or 1.



(a) Composite image (b) Ground truth alpha (c) Trimap

(d) Our result (e) Result of [6]: $\epsilon = 0$ (f) Result of [6]: $\epsilon = 10^{-6}$

Figure 1. Comparison of our proposed framework with that of [6] for the line-point case.

Results for insufficient user input. We now demonstrate that our algorithm can recover the alpha mattes even when the user provides scribbles for only one of the layers. Recall that since [6] prefers locally constant mattes, it will produce a result with all pixels having $\alpha = 0$ or $\alpha = 1$. Therefore, for a fair comparison, we propose a new version of [6], in order to bias the alpha mattes towards 0. In particular, we estimate the alpha mattes by minimizing the cost function

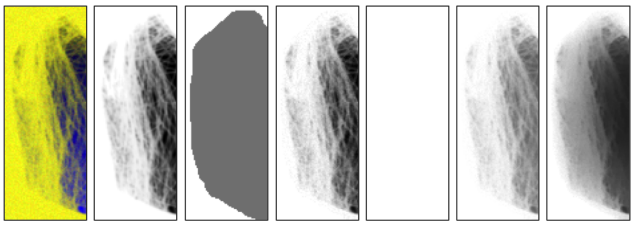
$$\tilde{J}_\epsilon(\boldsymbol{\alpha}, v) = J(\boldsymbol{\alpha}, v) + \epsilon \sum_{i \in \mathcal{V}} \left(a_i^{R^2} + a_i^{G^2} + a_i^{B^2} + b^2 \right). \quad (16)$$

In this modification, we are biasing the affine models of (3) towards $(0, 0, 0, 0)$. It can easily be checked that we can eliminate the unknown affine models in a similar fashion as described earlier, and obtain a closed form solution for the alpha mattes. We hence have a formulation which can potentially estimate the alpha mattes even when the user provides scribbles for one of the layers only.

Figure 2 shows a toy example for the line-point color model where the user marks scribbles for the foreground only. We are able to recover visually pleasing alpha mattes, since our natural bias towards 0 is the desired bias in this case. However, we do not get good results with our proposed modification of [6] even when we increase ϵ in (16). As discussed, these resulting alpha mattes are biased to be locally smooth due to the nature of the Matting Laplacian.

3.2. Point-Point Color Models

We now consider the case when the colors of both the layers are constant and hence satisfy the color point model. By the hypothesis, $\forall j \in \mathcal{W}_i, F_j = F$ and $\forall j \in \mathcal{W}_i, B_j = B$. Therefore, the composite intensity I_j of a pixel $j \in \mathcal{W}_i$ can be expressed as $\forall j \in \mathcal{W}_i : I_j = \alpha_j F + (1 - \alpha_j) B$.



(a) (b) (c) (d) (e) (f) (g)

Figure 2. Comparison of our proposed framework with that of [6], when the user provides scribbles for one layer only. (a) Composite image (b) Ground truth alpha (c) Trimap (d) Our result (e) – (g) Result of [6]: $\epsilon = 0$, $\epsilon = 10^{-4}$ and $\epsilon = 10^{-2}$ respectively

For this scenario, we derive two important results as given by Theorem 2.

Theorem 2 *Consider an image patch $\mathcal{W}_i$ around a pixel $i \in \mathcal{V}$, such that the RGB intensities of the pixels in the patch, satisfy the point-point color models. Define a matrix $G_i^{PP} \in \mathbb{R}^{|\mathcal{W}_i| \times 3}$, whose rows are given as $\begin{bmatrix} I_j^R & I_j^G & I_j^B \end{bmatrix}, j \in \mathcal{W}_i$.*

1. *If the alpha mattes of all the pixels in the patch are not equal and the color points of the two layers are distinct, then $Rk(G_i^{LP}) = 2$.*
2. *If $Rk(G_i^{PP}) = 2$, there exists a projection $\Pi : I \in \mathbb{R}^3 \rightarrow \tilde{I} \in \mathbb{R}^2$, such that the alpha matte α_j of each pixel $j \in \mathcal{W}_i$ can be expressed as a linear function of the projected intensities $\tilde{I}_i \in \mathbb{R}^2$, via unique coefficients $v_i = (a_i^1, a_i^2) \in \mathbb{R}^2$ characteristic to $\mathcal{W}_i$, as*

$$\forall j \in \mathcal{W}_i : \alpha_j = a_i^1 \tilde{I}_i^1 + a_i^2 \tilde{I}_i^2. \tag{17}$$

Proof.

1. Note that by definition, we have

$$G_i^{PP} = \underbrace{\begin{bmatrix} \vdots & \vdots \\ \alpha_j & 1 \\ \vdots & \vdots \end{bmatrix}}_{\Gamma_i} \underbrace{\begin{bmatrix} F\text{–}B & B \end{bmatrix}^\top}_{H_i}. \tag{18}$$

Since by hypothesis, $F \neq B$, the matrix H_i in (18) is full rank, and hence of rank 2. If the alpha mattes of all the pixels are not equal, then matrix Γ_i in (18) is also of rank 2. Hence, the matrix G_i^{PP} has rank 2.

2. Let the color line passing through the foreground and background intensities F and B be given by d_{fb}. Therefore, there exist parameters $I_0 \in \mathbb{R}^3$ and $\lambda_f, \lambda_b \in \mathbb{R}$ such that $F = I_0 + \lambda_f d_{fb}$ and $B = I_0 + \lambda_b d_{fb}$. Now the compositing equation (1) states that for each pixel j in the patch $\mathcal{W}_i$,

$$\begin{aligned} I_j &= \begin{bmatrix} F\text{–}B & B \end{bmatrix} \begin{bmatrix} \alpha_j \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} I_0 & d_{fb} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 \\ \lambda_f - \lambda_b & \lambda_b \end{bmatrix}}_{\Lambda_i} \begin{bmatrix} \alpha_j \\ 1 \end{bmatrix}. \end{aligned} \tag{19}$$

Define the projection function as $\Pi(I_j) = \tilde{I}_j = \begin{bmatrix} \tilde{I}_i^1 & \tilde{I}_i^2 \end{bmatrix}^\top = \begin{bmatrix} I_0 & d_{fb} \end{bmatrix}^\dagger I_j$, where $A^\dagger$ denotes the pseudo-inverse of a matrix A. We then have

$$\forall j \in \mathcal{W}_i : \tilde{I}_j = \Lambda_i \begin{bmatrix} \alpha_j \\ 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} \alpha_j \\ 1 \end{bmatrix} = \Lambda_i^{-1} \tilde{I}_j. \tag{20}$$

Hence, we conclude that the alpha matte of each pixel is given by a linear combination of the pixels' projected intensities, the coefficients of combination being uniquely given by the first row of Λ_i^{-1}.

■

As earlier, we define a matrix $\tilde{G}_i \in \mathbb{R}^{|\mathcal{W}_i| \times 2}$, the rows of which are given by $\tilde{I}_j, j \in \mathcal{W}_i$, and also a matrix $\bar{\alpha}_i \in \mathbb{R}^{|\mathcal{W}_i|}$, whose entries are given by $\alpha_j, j \in \mathcal{W}_i$. The problem of finding the unknown alpha mattes can then be posed as one of minimizing the cost function

$$J_2(\boldsymbol{\alpha}, v) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{W}_i} (\alpha_j - v_i^\top \tilde{I}_j)^2 = \sum_{i \in \mathcal{V}} \|\bar{\alpha}_i - \tilde{G}_i v_i\|^2. \tag{21}$$

Recall that the coefficients v_i for each patch $\mathcal{W}_i$ can be estimated in closed form as

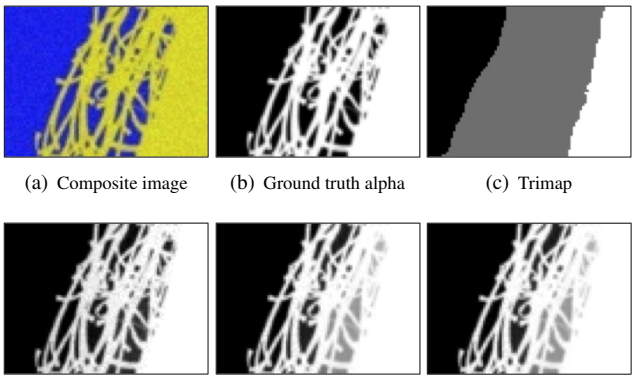
$$v_i = \underset{v}{\operatorname{argmin}} \|\tilde{G}_i v - \bar{\alpha}_i\|^2 = (\tilde{G}_i^\top \tilde{G}_i)^{-1} \tilde{G}_i^\top \bar{\alpha}_i. \tag{22}$$

Substituting the expression for v_i from (22), we see that the cost function $J_2(\boldsymbol{\alpha}, v)$ can be reduced to a cost function dependent on the alpha mattes only, as

$$\begin{aligned} J_2(\boldsymbol{\alpha}) &= \sum_{i \in \mathcal{V}} \left[\bar{\alpha}_i^\top (\boldsymbol{I}_{|\mathcal{W}_i|} - \tilde{G}_i (\tilde{G}_i^\top \tilde{G}_i)^{-1} \tilde{G}_i^\top) \bar{\alpha}_i \right] \\ &= \boldsymbol{\alpha}^\top L_2 \boldsymbol{\alpha}. \end{aligned} \tag{23}$$

Since the constructed cost function is quadratic in the alpha mattes, the alpha mattes can be estimated in closed form by solving a linear system.

What is the null space? We know that for each small image patch $\mathcal{W}_i$, since $\tilde{G}_i$ is of rank 2, the null space of the matrix $M_i = \boldsymbol{I}_{|\mathcal{W}_i|} - \tilde{G}_i (\tilde{G}_i^\top \tilde{G}_i)^{-1} \tilde{G}_i^\top$ is of rank 2. Now, we can proceed as we did in the rank 3 case, and verify that the column span of $\tilde{G}_i$ is a subset of the column span of the matrix G_i employed by [6]. Consequently, the 2 dimensional null space constructed by our framework is a subset of the



(a) Composite image (b) Ground truth alpha (c) Trimap

(d) Our result (e) Result of [6]:$\epsilon = 0$ (f) Result of [6]:$\epsilon = 10^{-6}$

Figure 3. Comparison of our proposed framework with that of [6] for the point-point case.

4 dimensional null space constructed by [6], and our proposed model for the alpha mattes is more compact than that of [6]. Also, since the locus of a point (x, y) on a line can be represented as $c_1 x + c_2 y = 1$, we can always find models (a_i^1, a_i^2) such that our framework admits locally constant solutions. It is also easy to show that, as for the rank 3 case, the Laplacian of [6] is rank 2 for noise free data.

Results on toy data. Figure 3 gives a toy example to illustrate the advantage of our proposed model. The yellow foreground and blue background constitute distinct points in RGB space. Therefore, this scenario corresponds to the point-point color model. We add some noise to the composite image. The conclusions are the same as in Figure 1, *i.e.* [6] produces a solution which is worse than ours. Again, observe that [6] has biased towards locally smooth mattes, while ours has a small bias towards 0.

4. Experiments

In this section, we present a quantitative and qualitative comparison of our proposed framework with that of [6], and show that our formulation helps to estimate better mattes. First, we give the details of our numerical implementation and then present an analysis of our tests.

4.1. Numerical Implementation

Like in [6], we consider image patches of size 3×3. Note that we need to estimate the rank of each patch $\mathcal{W}_i$, and for this, we first construct a matrix G_i, the rows of which are given by $\begin{bmatrix} I_j^R & I_j^G & I_j^B & 1 \end{bmatrix}, j \in \mathcal{W}_i$. We then estimate the singular value decomposition of this matrix as $G = U\Sigma V^\top$, where the diagonal entries of Σ are given by $\sigma_1 > \sigma_2 > \sigma_3 > \sigma_4$. Now, we normalize the singular values as $\tilde{\Sigma} = \lambda\Sigma$, where $\lambda = (\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2)^{-0.5}$. This normalization ensures that the rank estimation is not sensitive to scale variations in the image. We inspect the normalized singular values $\tilde{\sigma}_i = \lambda\sigma_i$, and estimate the rank as rank($\mathcal{W}_i$)= $\text{argmax}_k[\tilde{\sigma_k} > \delta]$, where δ is a pre-defined tolerance value. We use $\delta = 0.0025$ in all our experiments.

Given the rank of a patch $\mathcal{W}_i$, we choose the appropriate cost function C_i for the patch as discussed in Section 3. In particular, if the rank of $\mathcal{W}_i$ is 2, 3 or 4, we construct the matrix C_i of size $|\mathcal{W}_i| \times |\mathcal{W}_i|$ as $C_i = L_2$ in (21), $C_i = L_3$ in (13), or $C = L$ in (4) respectively, by restricting $\mathcal{V} = i$. In the case of rank 1, we construct C_i using the cost function of (4), which essentially forces the alpha mattes in the patch to be equal. We then define a vector $\bar{\alpha}_i \in \mathbb{R}^{|\mathcal{W}_i|}$, the entries of which are given by $\{\alpha_j\}, j \in \mathcal{W}_i$. We therefore need to estimate the alpha mattes of the image by minimizing $\sum_i \bar{\alpha}_i^\top C_i \bar{\alpha} = \boldsymbol{\alpha}^\top \tilde{L} \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the vector containing the alpha mattes of all the pixels in the image. Note that $\tilde{L}$ is a positive semi-definite matrix of size $|\boldsymbol{\alpha}| \times |\boldsymbol{\alpha}|$ obtained by aggregating the matrices C_i. Now, we need to minimize this cost function subject to the constraints that the set of pixels scribbled as foreground (say $\mathcal{F}$) have $\alpha = 1$ and the set of pixels scribbled as background (say $\mathcal{B}$) have $\alpha = 0$. As shown in [14], the solution to this problem

$$\begin{aligned} \boldsymbol{\alpha} &= \underset{\boldsymbol{\alpha}}{\text{argmin}}\ \boldsymbol{\alpha}^\top \tilde{L} \boldsymbol{\alpha} \\ &\text{s.t. } \alpha_i = 0 \text{ if } i \in \mathcal{F}, \text{ and } \alpha_i = 1 \text{ if } i \in \mathcal{B}. \end{aligned} \tag{24}$$

is equivalent to solving a linear system. Hence, we have a new closed form solution for the alpha mattes of an image.

4.2. Results

We perform our evaluation on the database used in [10], which contains 27 high quality images. For the purpose of testing, we dilate the perfect trimap by 22 pixels. Figures 4–6 are typical examples. In what follows, we compare the performance of the following 4 algorithms: (a) `Rank-adaptive`: our proposed algorithm in which we construct the cost function by analyzing the rank of each image patch; (b) `Rank-adaptive-mod`: a modification of our algorithm, where we treat all rank 2 patches as rank 3; (c) `Levin`: the algorithm of [6]; and (d) `Levin-mod`: the algorithm of [6] with our proposed modification of biasing the mattes towards 0 as in (16), but *only* for those connected regions in the trimap which have only 1 as boundary conditions. Note, for `Levin-mod` we tried different values for ϵ in eqn. (16) and selected the best as $\epsilon = 10^{-3}$.

The performance of each algorithm is evaluated using the following three different metrics. Given the computed α matte and the ground truth α^*, we compute the metrics SAD: $\sum_i |\alpha_i - \alpha_i^*|$, MSE: $\sum_i (\alpha_i - \alpha_i^*)^2$, and gradient error: $\sum_i (\nabla\alpha_i - \nabla\alpha_i^*)^2$.

Table 1 shows these errors for different methods (averaged over all test cases). The best result for each metric is highlighted in bold. We note that `Levin-mod` obviously outperforms `Levin`. Interestingly, `Rank-adaptive-mod` performs better than `Rank-adaptive`. Visual inspection shows that the bias towards 0 is more pronounced for the rank 2 case than for

rank 3. This points towards the fact that for real images, our rank 3 formulation can account for the rank 2 cases also and has much more stable performance. Note that this is not a drawback of our formulation, since we can have a black box for rank estimation, which always gives values of 3 or 4.

Importantly, the modification of our algorithm `Rank-adaptive-mod` performs better than `Levin` as well as `Levin-mod` in 2 out of the 3 metrics. These improvements can also be observed visually, since these three error metrics may not be representative of the error observed by a human.

Table 1. Mean errors for the estimation of alpha mattes

Method	SAD	MSE	Gradient
`Levin`	4733	0.3276	679.3
`Levin-mod`	**4727**	0.3173	678.3
`Rank-adaptive`	5583	0.3328	690.9
`Rank-adaptive-mod`	5171	**0.2788**	**633.9**

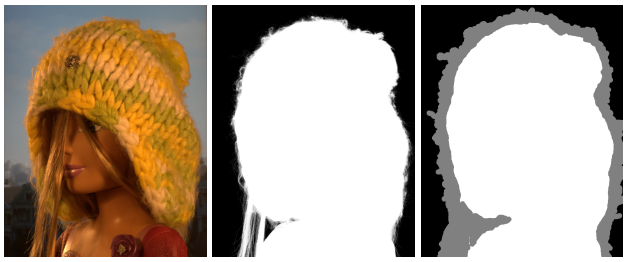
Figures 4–6 show typical results obtained in the above error analysis. We show the results of `Levin-mod` and `Rank-adaptive-mod` since these algorithms rank the best in the error metrics. In the images displaying the rank estimated by us, we use the following color-coding: dark blue - marked pixels, light blue - rank 1, green - rank 2, orange - rank 3, and red - rank 4. It is important to note that most of the unmarked pixels in the trimap have rank less than 4. In spite of introducing a bias towards 0 alpha mattes, `Levin-mod` cannot deal with *holes* in the trimap. This is due to its inherent bias to estimate locally smooth alpha mattes. Our method, however, has no such problem and is able to recover visually pleasing alpha mattes. Moreover, in spite of an inherent bias of our method towards 0 alpha mattes, our algorithm is able to accurately estimate 1 alpha mattes in several regions of the trimap, such as the boundary of the ball, the girl and the leaves in Figures 4–6.

We now address the issue of using scribbles vs. trimap as user interaction. In Figure 7, we see that when we use the same scribbles as used in [6], our framework is able to capture finer details of the alpha matte of the dandelion, as compared to [6]. On the other hand, in Figure 8, we see that when we use the same scribbles used in [6], our method gives suboptimal performance. Specifically, due to the inherent bias, our method tries to fit fractional alpha even though the true alpha matte is 1 in large portions of the bear. However, this can be easily fixed by connecting the scribbles and flood filling them to create a trimap, as shown in Figure 8(d). Our result (Fig. 8(f)) with this trimap is comparable to that of [6] (Fig. 8(e)). In general, our method outperforms [6] when the user input is a trimap, which as exhibited in our experiments, need not be tight. Note that this is not a limitation, since recent methods such as [1, 10, 9, 20] also use the given scribbles to generate a trimap, and then estimate the alpha mattes using this trimap.

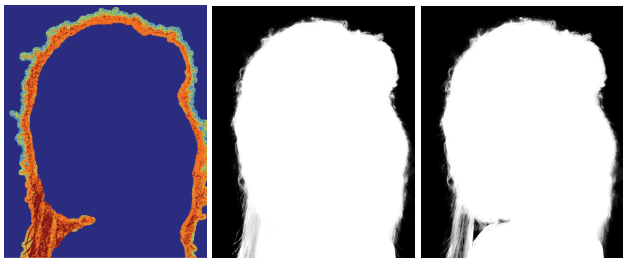
(a) Image (b) Ground truth alpha (c) Trimap

(d) Estimated rank (e) Results of `Levin-mod` (f) Results of `Rank-adaptive-mod`

Figure 4. Comparison of our framework vs. [6] on an image of a ball



(a) Image (b) Ground truth alpha (c) Trimap



(d) Estimated rank (e) Results of `Levin-mod` (f) Results of `Rank-adaptive-mod`

Figure 5. Comparison of our framework vs. [6] on an image of a girl

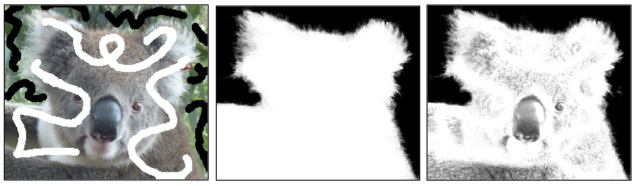
(a) Image (b) Ground truth alpha (c) Trimap

(d) Estimated rank (e) Results of `Levin-mod` (f) Results of `Rank-adaptive-mod`

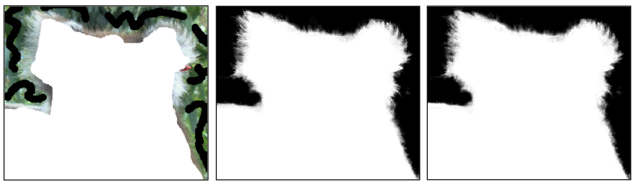
Figure 6. Comparison of our framework vs. [6] on an image of a plant

(a) Image + scribbles (b) Result of `Levin` (c) Result of `Rank-adaptive-mod`

Figure 7. Comparison of our framework vs. [6] on a dandelion's image.



(a) Image + scribbles originally used in [6] (b) Result of `Levin` (c) Result of `Rank-adaptive-mod`



(d) Trimap via connecting scribbles+floodfill (e) Result of `Levin` (f) Result of `Rank-adaptive-mod`

Figure 8. Comparison of our framework vs. [6] on an image of a bear.

5. Conclusions

In this work, we have presented new appearance models for the problem of image matting. By construction, these appearance models are more compact than that proposed by [6], and as shown in our analysis, outperform the traditional color line model of [6], without the need of any additional user interaction. Future work entails the need of closed form solvers for the mattes of image patches that have complex intensity variation and hence do not satisfy the color line model or the color point model.

References

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

[2] Y.-Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *CVPR (2)*, pages 264–271, 2001. 1

[3] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. In *Proceedings of the Fifth IASTED International Conference on Visualization, Imaging and Image Processing*, pages 423–429, 2005. 1

[4] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting: A stroke based approach for continuous image matting. *Eurographics*, 25(3):567–576, 2006. 1

[5] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. *ACM Trans. Graph.*, 27(3), 2008. 2

[6] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):228–242, 2008. 1, 2, 3, 4, 5, 6, 7, 8, 9

[7] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1699–1712, 2008. 1, 2, 3, 5

[8] I. Omer and M. Werman. Color lines: Image specific color representation. In *CVPR (2)*, pages 946–953, 2004. 2

[9] C. Rhemann, C. Rother, and M. Gelautz. Improving color modeling for alpha matting. In *BMVC*, 2008. 1, 2, 3, 8

[10] C. Rhemann, C. Rother, A. Rav-Acha, and T. Sharp. High resolution matting via interactive trimap segmentation. In *CVPR*, 2008. 1, 2, 3, 7, 8

[11] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated Graph Cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 1

[12] M. A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *CVPR*, pages 1018–1025, 2000. 1

[13] D. Singaraju, C. Rhemann, and C. Rother. New appearance models for natural image matting. In *Technical Report, Microsoft Research, Cambridge*, 2008.

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

[15] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Trans. Graph.*, 23(3):315–321, 2004. 1

[16] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: An interactive tool for realtime high quality matting. *ACM Trans. Graph.*, 26(3):9, 2007. 1, 2

[17] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, 2005. 1

[18] J. Wang and M. F. Cohen. Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(2), 2007. 1

[19] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007. 1, 2, 3

[20] Y. Zheng, C. Kambhamettu, J. Yu, T. Bauer, and K. Steiner. Fuzzymatte: A computationally efficient scheme for interactive matting. In *CVPR*, 2008. 1, 8