Learning Based Digital Matting

Yuanjie Zheng
zheng.vision@gmail.com

Chandra Kambhamettu
chandra@cis.udel.edu

Video/Image Modeling and Synthesis (VIMS) Lab,
Department of Computer Science, University of Delaware, Newark, DE, USA

Abstract

We cast some new insights into solving the digital matting problem by treating it as a semi-supervised learning task in machine learning. A local learning based approach and a global learning based approach are then produced, to fit better the scribble based matting and the trimap based matting, respectively. Our approaches are easy to implement because only some simple matrix operations are needed. They are also extremely accurate because they can efficiently handle the nonlinear local color distributions by incorporating the kernel trick, that are beyond the ability of many previous works. Our approaches can outperform many recent matting methods, as shown by the theoretical analysis and comprehensive experiments. The new insights may also inspire several more works.

1. Introduction

Digital matting refers to the process of extracting a foreground object image F along with its opacity mask α (typically called "alpha matte") from a given digital image I, assuming that I is formed by linearly blending F and a background image B using α:

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

It is usually followed by a compositing process to create a new image by linearly blending the extracted foreground object image and a new background image with the extracted alpha matte, also using Eq. 1. Digital matting is an invaluable tool in image editing, film and video motion picture production, etc.

The digital matting problem is inherently under-constrained because it has more unknowns (F, B and α) than the constraints (Eq. 1). The ill-posed problem has been extensively studied by adding more information and constraints. The additional information is supplied by setting the scribbles [14] or trimaps [4], i.e. labeling some pixels which are definitely foreground or background, or providing multiple images or video [16]. The single image based matting problem (the topic of this paper), even with the known alpha value of the labeled pixels, is still ill-posed. Therefore, several works proposed additional constraints. For example, F and B can be well estimated with a series of nearest labeled foreground and background pixels [4, 15], the foreground and background colors are assumed to be locally smooth [12, 9, 8], and the alpha values are expected to be locally coherent [14, 6], etc. The first constraint is usually for the trimap based matting while the other two are generally for the scribble based matting. The constraints were proved to be very useful, and with them various techniques [16] have been proposed to efficiently extract high quality alpha mattes and foreground colors.

To apply the additional supplied information and constraints for solving the matting problem, one critical element in producing accurate results is how to model the relation between the alpha matte values and the image colors of a series of associated pixels. For simplicity, we call this relation the alpha-color relation. The associated pixels are some pixels with which the chosen alpha-color model is determined. The determined alpha-color model will then be used to predict α value of the unlabeled pixels. For the trimap based matting, usually a series of nearest labeled pixels are associated, while for the scribble based matting, usually some neighboring pixels of the pixel being estimated are associated.

The previous matting methods model the alpha-color relation in different ways, but for most of them the chosen model comes from the additional supplied information and constraints mentioned above. For example, both [15] and [9] assume the associated colors lie on a line in the RGB space, and based on this, some linear alpha-color models are deducted and employed. For [15], the associated pixels are a pair of labeled foreground and background pixels, and a linear alpha-color model is specified implicitly through computing the alpha value with the unlabeled pixel's distances to the pair of colors along the line. For [9], the associ-

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

ated pixels are the pixels in a local patch, and an explicit linear alpha-color model is then deducted. More assumptions and the resulting models will be explained in next section. A common problem of the previous matting techniques is that when their assumptions cannot be satisfied in practice, they may fail because the assumed alpha-color models (e.g. the linear models in [15] and [9]) are not sufficient for representing correctly the alpha-color relations.

In this paper, we attempt to cast some new insights into the single image based digital matting problem by treating it as a semi-supervised learning problem [18, 17, 21, 20] in machine learning, resulting in a local learning based matting approach and a global learning based approach. The local learning based approach learns the alpha-color model from the neighboring pixels of the pixel being estimated, and fits better the scribble based matting. The global learning based approach learns the model from some nearby labeled pixels, and suits better the trimap based matting.

Our approaches bear multiple advantages. First, they are easy to implement because only some simple matrix operations are required. Second, they are extremely accurate. They can efficiently handle the nonlinear local color distributions in the image with a more general alpha-color model by using the kernel trick, that are beyond the ability of many previous works like [9, 15]. They outperform many recent matting methods, as shown by the theoretical analysis and the comprehensive experiments. Third, our new insights could also inspire several more works following the line of using other learning methods and image features.

2. Previous Work

From the human intervention point of view, the single image based matting methods can be classified into three types: trimap based [4, 12, 6, 15, 13], scribble based [14, 9], and automatic [8]. Trimap can usually provide more labeled pixels but needs more labor than scribbles.

For modeling the alpha-color relation, there are basically three ways previously used to specify the associated pixels: choose a series of nearby labeled pixels, select the neighboring pixels, or use both.

The associated pixels are chosen as a series of nearby labeled pixels mostly in some trimap based matting methods. For example, Bayesian Matting [4], as a parametric technique, assumes and fits an oriented Gaussian distribution for each cluster of the image colors of the nearby labeled pixels and then uses it to estimate the α value of the unlabeled pixels with a maximum-likelihood criterion. The non-parametric technique in Robust Matting [15] instead samples some representative pairs of foreground color and background color from the labeled pixels by assuming that the color of the unlabeled pixel is on a line in the RGB space of the sampled pair of colors. The alpha value is then computed based on the unlabeled pixel's distances on the line to the pair of colors.

The associated pixels are selected from the pixel's neighborhood mostly in the scribble based matting methods, and also in some trimap based methods. For example, the Closed-Form Matting [9] and the Spectral Matting [8] assume a linear line in the RGB space for the colors of a local patch, resulting in a linear alpha-color model for solving the matting. Alternatively, the alpha-color relation is specified by assuming that the alpha matte's gradient is proportional to the image gradient in Poisson Matting [12] and that the absolute change of α value is encouraged to be consistent with the value of an exponential function of the absolute color changes (in the form of solving a matting Laplacian problem) in the Random-Walk Matting [6]. Recently, the alpha-color relation is specified by computing the alpha value based on the predefined distances' values to the scribbles for the geodesic distance based matting [1] and FuzzyMatte [19]. The distances are both computed based on the measurement of the neighboring pixels' color similarity. They assume that a shorter distance to the foreground scribbles or a larger distance to the background indicate a larger alpha matte value, and vice-versa.

The associated pixels are specified as both the nearby labeled pixels and the neighboring pixels in some trimap or scribble based matting methods. For example, Soft Scissors [13] specifies the alpha-color relation from some labeled pixels as in the robust matting and at the same time from a local patch as in the Closed-Form Matting. The Iterative BP Matting [14] assumes and fits a Gaussian Mixture Model for some nearest labeled pixels, enforces smoothness on the alpha value for the neighboring pixels, and solves the matte by solving a Markov Random Field (MRF).

We can see that the previous methods specify the alpha-color model from some assumptions on, for example, the associated pixels' color distribution, and the linearity of the model. In contrast, our approaches do not rely on these assumptions, and learn a more general alpha-color model which can be linear or nonlinear. Our approaches are more robust and can produce more accurate matte result.

Besides the single image based matting methods, there are also some other approaches using multiple images to make the matting problem over-constrained. They use a video or flash/no-flash pair of images, etc. References can be found in [16].

3. Estimating Alpha Matte with Learning

Matting consists of two main tasks: alpha matte's estimation, and foreground (and background) colors' computation. Given an image I for which the complete set of pixels is denoted by $\Omega = \{1, \cdots, n\}$ where n is the total number of pixels, and given a set of labeled pixels $\Omega_l \subset \Omega$ for which we know the α values, alpha matte estimation is defined as computing the α values of the set of unlabeled

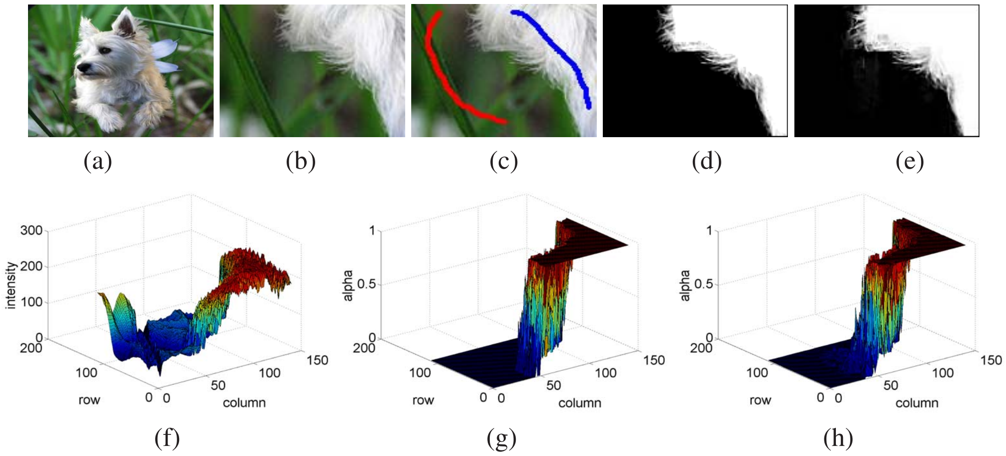


Figure 1. Local learning in alpha matte estimation learns the structure of the "image intensity surface". (a). An image from [15]. (b). Local patch of the input image. (c). Scribbles (blue for foreground and red for background). (d). Ground truth alpha matte. (e). Estimated alpha matte with local learning. (f). "Image intensity surface" of the corresponding gray image of (b). (g). Alpha surface of (d). (h). Alpha surface of (e). Note the similarities among the structures of the three surfaces.

pixels $\Omega_u = \Omega - \Omega_l$. Here the set of labeled pixels Ω_l is composed of two subsets: Ω_l^f labeled as definite foreground and for which we know α is 1, and Ω_l^b labeled as definite background and for which we know α is 0. We indicate Ω_l^f and Ω_l^b using blue and red colors respectively, as shown in Fig. 1. After an accurate alpha matte is obtained, we can simply use, for example, the method in [14] to determine a pair of foreground and background colors for each pixel.

We consider the alpha matte estimation as a learning problem. We treat each pixel $i \in \Omega$ as a data point denoted by $\mathbf{x}_i \in \mathbb{R}^d$. $\mathbf{x}_i$ can be set as I_i or other features extracted for pixel i. In this paper, we set $\mathbf{x}_i = I_i$ which is a scale value for a gray image ($d = 1$) or a vector composed of the RGB color components for a color image ($d = 3$). The learning problem can be then formulated as follows. Given a data point set $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{X} = \{\mathbf{x}_i\}_{i\in\Omega}$, and the alpha values $\{\alpha_i\}_{i\in\mathcal{X}_l}$ of the labeled data points $\mathcal{X}_l = \{\mathbf{x}_i\}_{i\in\Omega_l}$, our goal is to compute the accurate alpha values $\{\alpha_i\}_{i\in\mathcal{X}_u}$ of the unlabeled data points $\mathcal{X}_u = \mathcal{X} - \mathcal{X}_l$ through learning methods.

This learning problem apparently belongs to the semi-supervised learning problem [21, 20, 18, 17], i.e. learning (to specify α values of Ω_u) from partially labeled data (Ω_l). Semi-supervised learning addresses the learning process not only using the labeled data points but also the unlabeled data points. It can produce high accuracy even with little human effort for a general classification task.

3.1. Estimating Alpha Matte via Local Learning

Our local learning based matting technique trains a local alpha-color model for each pixel in the image only based on its neighboring pixels which are considered to be most related. As the local learning methods [21, 18] in machine learning can effectively use the manifold structure, the local learning in alpha matte estimation can exploit the structure of the "image intensity surface" formed by the colors of the image pixels on the regular lattice (called "image structure" in [6]), as shown in Fig. 1. Note that in the matting problem the neighboring pixels are determined according to the Euclidean distance between pixels on the regular lattice of the image, which is different from the general semi-supervised learning problems.

We next formulate the alpha matte estimation by first assuming each pixel's alpha value as a linear combination of its associated neighboring pixels. Then, we elaborate on determining the coefficients of the linear combination with a local learning process based on a linear alpha-color model, followed by relaxing the linear model to a more general one which can be nonlinear with the kernel trick [10].

3.1.1 Alpha Matte Estimation

In the estimation of alpha matte with local learning, for any pixel $i \in \Omega$, we assume that its alpha matte value α_i can be predicted by a linear combination of the alpha values $\{\alpha_j\}_{j\in\mathcal{N}_i}$ of its neighboring pixels $\mathcal{N}_i \subset \Omega$. We select the pixels in a 7×7 local path centered at i as the neighbors. We then estimate simultaneously the alpha values of all pixels through minimizing a quadratic cost.

We first denote $\mathcal{N}_i = \{\tau_1, \cdots, \tau_m\}$. If we use $\boldsymbol{\alpha}_i = [\alpha_{\tau_1}, \cdots, \alpha_{\tau_j}, \cdots \alpha_{\tau_m}]^T$ where $\tau_j \in \mathcal{N}_i$ to denote the vector of alpha values of $\mathcal{N}_i$, and $\mathbf{f}_i = [f_{i\tau_1}, \cdots, f_{i\tau_j}, \cdots, f_{i\tau_m}]^T$ to denote the vector of the linear combination coefficients, the combination for i can be represented by

$$\alpha_i = \mathbf{f}_i^T \boldsymbol{\alpha}_i. \tag{2}$$

We can also rewrite α_i in Eq. 2 in the form of the linear combination of alpha values of all the pixels. We denote the alpha values of all the pixels with the vector $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$ where the coefficients with $\boldsymbol{\xi}_i = [f_{i1}, \cdots, f_{in}]^T$. For $\boldsymbol{\xi}_i$, the values for the pixels in $\mathcal{N}_i$ are equal to the corresponding ones in $\mathbf{f}_i$, and the remaining are zero. We have

$$\alpha_i = \boldsymbol{\xi}_i^T \boldsymbol{\alpha}. \tag{3}$$

By introducing a new matrix $\mathbf{F}$ through stacking $\{\boldsymbol{\xi}_i\}_{i \in \Omega}$: $\mathbf{F} = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_n]$, we can rewrite Eq. 3 in a more concise format

$$\boldsymbol{\alpha} = \mathbf{F}^T \boldsymbol{\alpha}. \tag{4}$$

If we know $\mathbf{F}$, a classical way to estimate $\boldsymbol{\alpha}$ is to minimize the following quadratic cost

$$\arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha} - \mathbf{F}^T \boldsymbol{\alpha}\|^2 + c\|\boldsymbol{\alpha}_l - \boldsymbol{\alpha}_l^*\| \tag{5}$$

where $\boldsymbol{\alpha}_l$ denotes the vector of the alpha variables of the labeled pixels in Ω_l and $\boldsymbol{\alpha}_l^*$ denotes the vectors of the already known alpha values of the labeled pixels. As suggested in [18], different c values may lead to different algorithms for solving the Eq. 5. Here we set $c = \infty$, which forces the matte value to be 1 for the labeled foreground pixels and 0 for background. It can help in utilizing the additional provided information to the maximum extent [11].

In order to solve Eq. 5, we need to reformulate it. We introduce the diagonal matrix $\mathbf{C}$ of size $n \times n$, for which the jth diagonal element takes the constant value c if $j \in \Omega_l$, and other diagonal elements are zero. We also bring in the vector $\boldsymbol{\alpha}^*$ of length n, for which the jth element equals the already known alpha value of pixel j if $j \in \Omega_l$. Then, Eq. 5 is reformulated as

$$\arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \boldsymbol{\alpha}^T \left(\mathbf{I}_{(n)} - \mathbf{F}\right) \left(\mathbf{I}_{(n)} - \mathbf{F}\right)^T \boldsymbol{\alpha} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T \mathbf{C} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \tag{6}$$

where $\mathbf{I}_{(n)}$ is the $n \times n$ identity matrix. Eq. 6 is similar to the quadratic optimization problem proposed in [18].

By taking the first derivative of $\boldsymbol{\alpha}$ in Eq. 6 and setting it to zero, we get the solution

$$\boldsymbol{\alpha} = \left((\mathbf{I}_{(n)} - \mathbf{F})(\mathbf{I}_{(n)} - \mathbf{F})^T + \mathbf{C}\right)^{-1} \mathbf{C} \boldsymbol{\alpha}^*. \tag{7}$$

Eq. 7 can be computed if the linear combination coefficients in Eq. 2 are known. Their computation is accomplished by the local learning process explained next.

3.1.2 Local Learning

Local learning in alpha matte estimation tries to train a local alpha-color model for each pixel $i \in \Omega$ to describe the dependencies between $\{\mathbf{x}_j\}_{j \in \mathcal{N}_i}$ in $\mathbb{R}^d$ and $\{\alpha_j\}_{j \in \mathcal{N}_i}$ in $\mathbb{R}$, which can then be used to predict α_i from $\mathbf{x}_i$. This training process results in the estimation of $\mathbf{f}_i$ in Eq. 2, which is only based on the already known values $\mathbf{x}_i$ and $\{\mathbf{x}_j\}_{j \in \mathcal{N}_i}$, making Eq. 7 solvable.

For a data vector $\mathbf{x}$, we denote $\mathbf{x}' = [\mathbf{x}^T \; 1]^T$. We first choose a linear alpha-color local model for the local learning:

$$\alpha = \mathbf{x}^T \boldsymbol{\beta} + \beta_0 = \mathbf{x}'^T \begin{bmatrix} \boldsymbol{\beta} \\ \beta_0 \end{bmatrix} \tag{8}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_d]^T$ and β_0 are the model coefficients.

As before, for pixel $i \in \Omega$, we denote $\mathcal{N}_i = \{\tau_1, \cdots, \tau_m\}$. Note that $\boldsymbol{\alpha}_i = [\alpha_{\tau_1}, \cdots, \alpha_{\tau_m}]^T$; we denote the new notation: $\mathbf{X}_i = [\mathbf{x}'_{\tau_1} \cdots \mathbf{x}'_{\tau_m}]^T$ which is a matrix of size $m \times (d+1)$ and is stacked by the data values of the pixels in $\mathcal{N}_i$.

With the ridge regression technique [5], we can estimate $\boldsymbol{\beta}$ and β_0 by solving a quadratic optimization problem:

$$\arg\min_{\boldsymbol{\beta}, \beta_0} \left\| \boldsymbol{\alpha}_i - \mathbf{X}_i \begin{bmatrix} \boldsymbol{\beta} \\ \beta_0 \end{bmatrix} \right\|^2 + \lambda_r \begin{bmatrix} \boldsymbol{\beta} \\ \beta_0 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\beta} \\ \beta_0 \end{bmatrix} \tag{9}$$

where λ_r is a parameter for which we set 0.1.

The optimal solution of Eq. 9 can be easily derived as

$$\begin{aligned} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\beta}_0 \end{bmatrix} &= \left(\mathbf{X}_i^T \mathbf{X}_i + \lambda_r \mathbf{I}_{(d+1)}\right)^{-1} \mathbf{X}_i^T \boldsymbol{\alpha}_i \\ &= \mathbf{X}_i^T \left(\mathbf{X}_i \mathbf{X}_i^T + \lambda_r \mathbf{I}_{(m)}\right)^{-1} \boldsymbol{\alpha}_i. \end{aligned} \tag{10}$$

Substituting Eq. 10 into Eq. 8, we can finally get $\mathbf{f}_i$ of Eq. 2 as

$$\mathbf{f}_i = \left(\mathbf{X}_i \mathbf{X}_i^T + \lambda_r \mathbf{I}_{(m)}\right)^{-1} \mathbf{X}_i \mathbf{x}'_i, \tag{11}$$

which is free from $\{\alpha_j\}_{j \in \mathcal{N}_i}$ and only relates to $\mathbf{x}_i$ and $\{\mathbf{x}_j\}_{j \in \mathcal{N}_i}$.

The linear alpha-color model in Eq. 8 can be extended to being nonlinear with the kernel trick [10], by replacing $\mathbf{x} \in \mathbb{R}^d$ with a feature vector $\Phi(\mathbf{x}) \in \mathbb{R}^p$ where Φ is typically a nonlinear map function. Usually, $p > d$ and the nonlinear model in the low dimensional space can be represented by a linear model in the high dimensional space as shown in the following

$$\alpha = \Phi(\mathbf{x})^T \boldsymbol{\beta} + \beta_0 \tag{12}$$

where, unlike Eq. 8, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^T$ and $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \cdots, \phi_p(\mathbf{x})]^T$.

With the kernel trick, we only need to replace the inner product of any two data vectors $\mathbf{x}'_i$ and $\mathbf{x}'_j$ in Eq. 11 with their kernel function value $k(\mathbf{x}'_i, \mathbf{x}'_j)$. We first denote

$$\mathbf{k}_i = [k(\mathbf{x}'_{\tau_1}, \mathbf{x}'_i), \cdots, k(\mathbf{x}'_{\tau_m}, \mathbf{x}'_i)]^T \tag{13}$$

and

$$\mathbf{K}_i = \begin{bmatrix} k(\mathbf{x}'_{\tau_1}, \mathbf{x}'_{\tau_1}) & \cdots & k(\mathbf{x}'_{\tau_1}, \mathbf{x}'_{\tau_m}) \\ \vdots & \vdots & \vdots \\ k(\mathbf{x}'_{\tau_n}, \mathbf{x}'_{\tau_1}) & \cdots & k(\mathbf{x}'_{\tau_n}, \mathbf{x}'_{\tau_m}) \end{bmatrix}. \tag{14}$$

Then, $\mathbf{f}_i$ from the nonlinear local model is expressed as below

$$\mathbf{f}_i = (\mathbf{K}_i + \lambda_r \mathbf{I}_{(m)})^{-1}\mathbf{k}_i \tag{15}$$

which is also built only on $\mathbf{x}_i$ and $\{\mathbf{x}_j\}_{j\in\mathcal{N}_i}$. Note that the kernel function value can be computed as a preprocessing to the alpha estimation.

Compared to the previous matting methods, the kernel trick makes our alpha-color model more general considering its efficiency in representing nonlinear relations [10]. In this paper, we use the Gaussian kernel $k(\mathbf{x}'_i, \mathbf{x}'_j) = \exp\left(\frac{1}{\vartheta}\|\mathbf{x}'_i - \mathbf{x}'_j\|^2\right)$ where parameter ϑ is set to the variance value of the grayscale image of the given image.

3.1.3 Comparison to the Closed-Form Matting

The Closed-Form matting technique [9] computes matte by solving a matting Laplacian problem. With some of the symbols used in Eq. 6, its equation to calculate the optimal $\boldsymbol{\alpha}$ of an image can be rewritten as the below minimization:

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^T\mathcal{L}\boldsymbol{\alpha} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T\mathbf{C}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \tag{16}$$

where $\mathcal{L}$ is a $n \times n$ Laplacian matrix and $\mathbf{C}'$ is a diagonal matrix similar to $\mathbf{C}$ in Eq. 6. This minimization looks similar to Eq. 6 if we treat $\big(\mathbf{I}_{(n)} - \mathbf{F}\big)\big(\mathbf{I}_{(n)} - \mathbf{F}\big)$ in Eq. 6 as $\mathcal{L}$ in Eq. 16.

However, there are significant differences between the Closed-Form matting method and our local learning based matting approach, some of which bring in substantial improvements of performance to our approach. *First*, we derive our approach by treating the matting problem as a semi-supervised learning task. This is completely different from the Closed-Form method which was derived from the assumption of a linear local color distribution and uses a linear alpha-color model. *Second*, the Closed-Form method may fail when its assumption is not satisfied. In contrast, our method can learn a more general alpha-color model which can be nonlinear. This is a very important advantage considering that the nonlinear color distribution can frequently happen as shown in Fig. 2. Note that our more general model is very easy to implement by replacing the inner product of data vectors in Eq. 11 with the kernel function value. Moreover, in practice, to avoid the failure case, small window sizes (typically 3 × 3) for the local patch is used in [9] although a larger window size is more stable. In contrast, a larger window size (typically 7 × 7) can be used by our method. *Third*, the nonzero diagonal elements are set to a large number in $\mathbf{C}'$ in the Closed-Form method while to ∞ in $\mathbf{C}$ by us. As explained above, our approach can use the useful information in the labeled pixels to the maximum extent. *Fourth*, unlike the Closed-Form method, our approach can easily incorporate other features besides intensity/color.

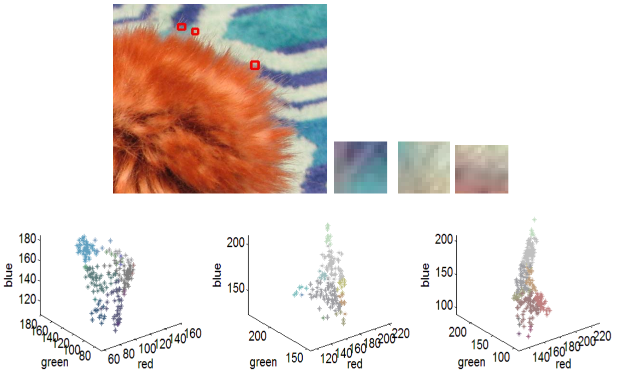


Figure 2. Top row: from left to right, a real image with three imposed red rectangles, and the three local patches specified by the red rectangles. Bottom row: from left to right, colors' distributions in the RGB space of the three patches.

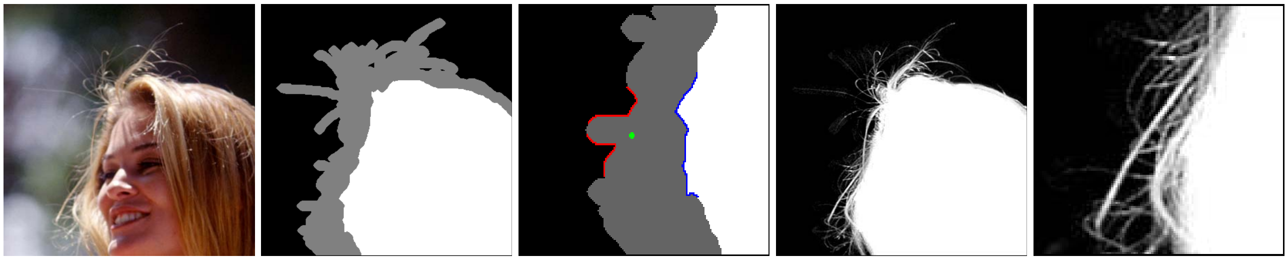
3.2. Estimating Alpha Matte via Global Learning

Global learning in alpha matte estimation is to estimate the alpha value of the unlabeled pixels with a global alpha-color model trained from some chosen labeled pixels. The chosen pixels closer to the unlabeled pixel can help more in the matte's estimation. Therefore it suits particularly to the case when a trimap is provided and the unknown region is slim. We choose for each unknown pixel two subsets from its nearby labeled foreground and background pixels respectively, and weight them to train the global alpha-color model using the weighted ridge regression technique [7].

To choose the subsets from the labeled pixels, we first select two subsets $Q_l^f \subseteq \Omega_l^f$ and $Q_l^b \subseteq \Omega_l^b$, in which for any pixel j we have $\mathcal{D}_j < \mathcal{D}_{th}$, where $\mathcal{D}_j$ means the shortest Euclidean distance of j to the pixels in Ω_u on the regular lattice, and $\mathcal{D}_{th}$ is a distance threshold. For each unknown pixel i, we then select two subsets $Q_l^{f'} \subset Q_l^f$ and $Q_l^{b'} \subset Q_l^b$, in which the pixels have the shortest distance to i. We set the two subsets having the same number (e.g. 80) of pixels. To compute $\mathcal{D}_j$, we use the algorithm in [3] that can finish in a linear time. $\mathcal{D}_{th}$ is determined by $\mathcal{D}_{th} = (\gamma_d|\Omega_u|)/|\Omega| + \sqrt{2}$ where γ_d is a constant and is empirically set to $\gamma_d = 1.2$. It is designed with the strategy that, when the unknown region is thicker, it is larger and more labeled pixels near the pixel being estimated can be selected, and otherwise, it is smaller and more distant labeled pixels can be chosen. Note that $\mathcal{D}_{th}$ is set with the distance to the unknown region instead of the pixel being estimated. An example is shown in Fig. 3.

For each pixel j in the subsets $Q_l^{f'}$ and $Q_l^{b'}$, we set a weight $w_j = 1/(\mathcal{D}_j)^{\gamma_w}$ where $\gamma_w = 0.25$ is an empirically determined constant. We further create a diagonal matrix $\mathbf{W}_{Q_i}$ with the w values of the pixels in $Q_l^{f'} \cup Q_l^{b'}$, whose size is $t \times t$ where $t = |(Q_l^{f'} \cup Q_l^{b'})|$.

The weighted ridge regression technique [7] is then used



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

Figure 3. Global learning in alpha matte estimation learns the alpha-color relation from some labeled pixels. (a). Input image. (b). Hand-drawn trimap. (c). Selected labeled foreground pixels (blue) and background pixels (red) for the estimation of the unknown pixel (green). (d). Alpha matte result. (e). Inset showing enlarged version of a patch of (d).

to train the global model. As in local learning, we first give the theoretical results for the linear alpha-color model and then extend it to the nonlinear case. We introduce $\boldsymbol{\alpha}_{Q_i}$ to denote the vector composed of the alpha values of the pixels in $Q_l^{f'} \cup Q_l^{b'}$ and $\mathbf{X}_{Q_i}$ to represent the matrix constructed in a similar way to $\mathbf{X}_i$ in section 3.1.2 but with the data values of the pixels in $Q_l^{f'} \cup Q_l^{b'}$ instead of $\mathcal{N}_i$. With a similar mathematical deduction process to local learning, for any pixel $i \in \Omega_u$, we have its alpha value's estimation with a linear model as below

$$\alpha_i = \mathbf{x}_i'^T \mathbf{X}_{Q_i}^T \mathbf{W}_{Q_i} \left(\mathbf{W}_{Q_i}\mathbf{X}_{Q_i}\mathbf{X}_{Q_i}^T\mathbf{W}_{Q_i} + \lambda_r \mathbf{I}_{(t)}\right)^{-1} \boldsymbol{\alpha}_{Q_i}. \tag{17}$$

Extending the linear model to a nonlinear model with the kernel trick [10], we get

$$\alpha_i = \mathbf{k}_{Q_i}^T \mathbf{W}_{Q_i}(\mathbf{W}_{Q_i}\mathbf{K}_{Q_i}\mathbf{W}_{Q_i} + \lambda_r \mathbf{I}_{(t)})^{-1} \boldsymbol{\alpha}_{Q_i}. \tag{18}$$

where $\mathbf{k}_{Qi}$ is created in a similar way as $\mathbf{k}_i$ in Eq. 13 and $\mathbf{K}_{Q_i}$ similar as $\mathbf{K}_i$ in Eq. 14 but with the data values of the pixels in $Q_l^{f'} \cup Q_l^{b'}$. Here, we use the same kernel function as the local learning.

Compared with the Bayesian Matting [4] and Robust Matting [15] etc., our global learning approach employs a different way to select some labeled pixels for computing the alpha-color model. Moreover, our approach can learn a more general alpha-color model with the kernel trick.

We can see that our local learning and global learning approaches are both easy to implement because only some simple matrix operations are required. It can be seen from Eqns. 7, 11, 17, and 18.

4. Results and Discussions

We provide both visual assessments and quantitative evaluations for comparing our learning based matting approaches with the previous methods. They are based on our own test images, and the public test sets of Wang-Cohen [15] and Levin *et al.* [8] for which the ground truth mattes are available. The set of Wang-Cohen consists of eight images named T1 to T8. For each test image, ten levels of fine-coarse trimaps are provided. The set of Levin *et al.* were captured on three dolls named "Monster", "Lion" and "Monkey". Each doll has a ground truth matte and six images captured with different backgrounds.

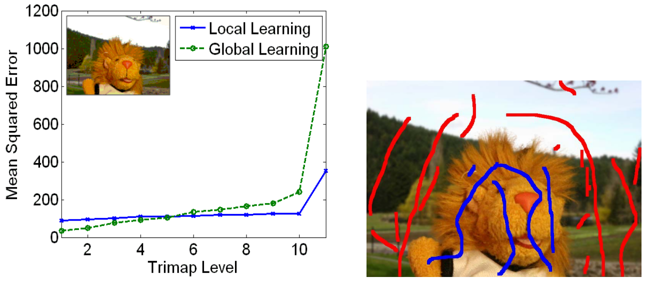
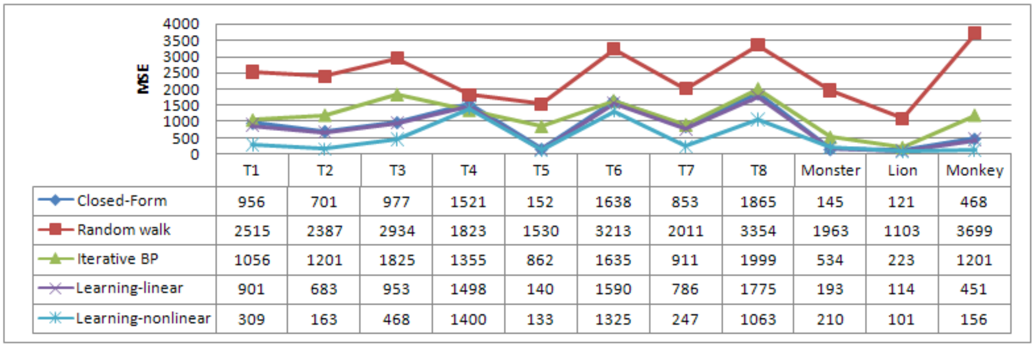


Figure 4. Local learning vs. global learning with a test image from [15]. Left: mean square error curves, for which the numbers $1 \sim 11$ on the x-axis correspond to the ten fine to coarse levels of trimap and the manually set scribbles, respectively. Right: the manually set scribbles.

4.1. Local Learning vs. Global Learning

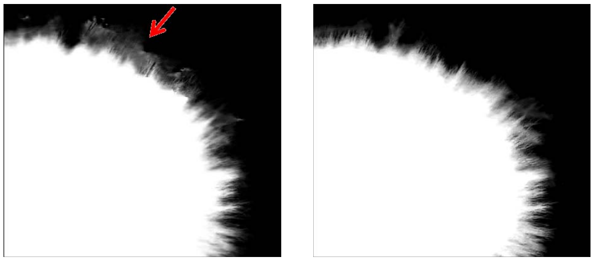
We ran our local learning and global learning approaches on the Wang-Cohen's eight test images. For each image, besides the provided eight trimaps, we manually set a series of sparse scribbles.

We found that when the unknown region in the trimap is very slim, our global learning approach outperforms the local learning, whereas when the trimap becomes coarser, it deteriorates very quickly. With the sparse scribbles, the errors produced by the global learning approach are very large, as shown by the results on one representative image in Fig. 4. Similar performance degradation may occur when the labeled pixels become sparser for some other trimap based matting techniques like the Bayesian Matting [4] and the Robust Matting [15], etc. In practice, the local learning approach fits better when sparse scribbles or a coarse trimap are provided while the global learning approach fits better when a fine trimap is offered.



	T1	T2	T3	T4	T5	T6	T7	T8	Monster	Lion	Monkey
Closed-Form	956	701	977	1521	152	1638	853	1865	145	121	468
Random walk	2515	2387	2934	1823	1530	3213	2011	3354	1963	1103	3699
Iterative BP	1056	1201	1825	1355	862	1635	911	1999	534	223	1201
Learning-linear	901	683	953	1498	140	1590	786	1775	193	114	451
Learning-nonlinear	309	163	468	1400	133	1325	247	1063	210	101	156

Figure 5. Mean Square Error (MSE) statistics of alpha matte computation on the test image sets.



Closed-Form Learning-nonlinear

Figure 6. Alpha matte results produced by the Closed-Form method and our nonlinear local learning approach for the original image in Fig. 2. The red arrow indicates the errors produced by the Closed-Form method.

4.2. Local Learning vs. Previous Scribble Based Methods

To evaluate the performance of our local learning based matting approaches (both the linear case with Eqns. 7 and 11 and the nonlinear case with Eqns. 7 and 15), we compare the Mean Squared Errors (MSE) of the matting results between our approaches and the Closed-Form Matting [9], Random-Walk Matting [6], Iterative BP Matting [14], with the two test image sets and some of our test images.

For the two public test sets, an operator who had no experience in any of the methods was asked to draw some foreground and background scribbles which pass through the main parts of the foreground and background objects. For each doll in the test images of Levin *et al*., the MSE values are averaged over the 6 images with different background. The MSE statistics are shown in Fig. 5.

We can see from the results, first, our approach with the linear alpha-color model produces better (for some degree) results than the Closed-Form Matting. As discussed in section 3.1, similar alpha-color models and similar solving methods are employed by the two methods. However, $\mathbf{C}$ in Eq. 6 and $\mathbf{C}'$ in Eq. 16 take different values. In addition, our approach employs a larger window size. Second, our approach with the nonlinear alpha-color model outperforms other methods for most of the images. It shows the efficiency of using the more general alpha-color model with the kernel trick in our matting scheme.

The strength of the more general alpha-color model with the kernel trick in our local learning approach can also be seen from Fig. 6. The nonlinear local color distributions as shown in Fig. 2 cause errors to the result of the Closed-Form method. In contrast, our approach produces high accurate results because the more general model can handle the complex local color distributions.

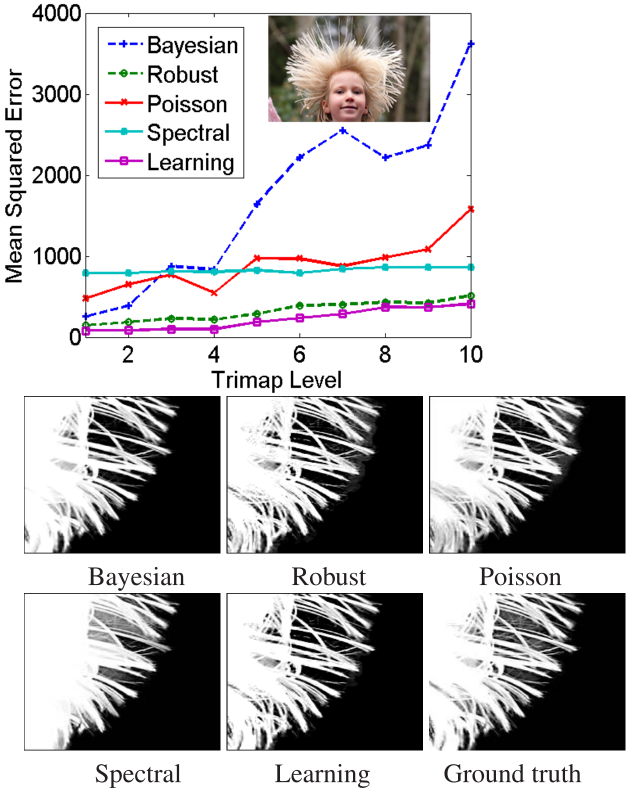
4.3. Global Learning vs. Previous Trimap Based Methods

To evaluate the performance of our global learning based matting approach, we compare the MSE of matting results with the Bayesian Matting [4], Robust Matting [15], Poisson Matting [12] and Spectral Matting [8], using the test set of Wang-Cohen.

From the results, we found that the comparisons on the 8 test images are similar. We show in this paper the results on a representative test image, as in Fig. 7.

From the comparisons, we have some findings: first, Robust Matting and our method produce most accurate results. Similar to the robust color sampling mechanism in Robust Matting, training a nonlinear alpha-color model with some nearest labeled pixels in our method is very efficient in predicting the unknown pixel's alpha values. Second, Poisson Matting and Bayesian Matting degrades more quickly when the trimap becomes coarser. Third, Spectral Matting's accuracy is lower compared to Robust Matting and our method. Note that the results can still show that our method outperforms the Closed-Form method because of the better performances of the Spectral Matting than the Closed-Form method reported in [8].

In the supplementary, more results are available including another example of MSE curve similar to Fig. 7, the alpha mattes and composition results by our method for some other popularly used test images, and the alpha mattes by our approach for all the test images used in this paper.



Bayesian Robust Poisson

Spectral Learning Ground truth

Figure 7. Results on a representative test image from [15], for which there are ten fine to coarse levels of trimap. Top: Mean square error curves. Bottom: Best mattes among the trimap levels for each method and the ground truth.

5. Conclusion and Future Work

By treating the digital matting problem as a semi-supervised learning task, we propose two new matting approaches: a local learning based method and a global learning based method. They are easy to implement because only some simple matrix operations are needed. They are extremely accurate because they can learn a more general alpha-color model which can be linear or nonlinear with the help of the kernel trick.

Our new insights casted into the matting problem could inspire several more works following the line of using other learning methods such as the support vector machine [2], other kernels [10], or other image features (e.g. texture features).

ACKNOWLEDGEMENT

This publication was made possible by Grant Number 2 P20 RR016472-08 under the INBRE program of the National Center for Research Resources (NCRR), a component of the National Institutes of Health (NIH).

References

[1] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV07*, 2007.

[2] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[3] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE TPAMI*, 17:529–533, 1995.

[4] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *CVPR01*, 2001.

[5] N. R. Draper and H. Smith. *Applied Regression Analysis*. John Wiley, New York, 2nd edition, 1981.

[6] L. Grady, T. Schiwietz, S. Aharon, and R. Westerman. Random walks for interactive alpha-matting. In *VIIP05*, 2005.

[7] P. W. Holland. Weighted ridge regression: Combining ridge and robust regression methods. *NBER Working Paper Series*, 1973.

[8] A. Levin, A. R. Acha, and D. Lischinski. Spectral matting. In *CVPR07*, 2007.

[9] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. In *IEEE TPAMI*, number 3, pages 228–242, 2008.

[10] B. Schlkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.

[11] D. Singaraju and R. Vidal. Interactive image matting for multiple layers. In *CVPR08*, 2008.

[12] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *ACM Trans. Graph*, volume 23, pages 315–321, 2004.

[13] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: An interactive tool for realtime high quality matting. *SIGGRAPH'07*, 2007.

[14] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV05*, 2005.

[15] J. Wang and M. Cohen. Optimized color sampling for robust matting. In *CVPR07*, 2007.

[16] J. Wang and M. Cohen. Image and video matting : A survey. *Foundations and Trends in Computer Graphics and Vision*, 3:97–175, 2008.

[17] M. Wu and B. Scholkopf. A local learning approach for clustering. In *Advances in Neural Information Processing Systems 19*, pages 1529–1536, Cambridge, Mass. USA, 2006. MIT Press.

[18] M. Wu and B. Scholkopf. Transductive classification via local learning regularization. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 628–635, Brookline, MA, USA, 2007. Microtome.

[19] Y. Zheng, C. Kambhamettu, J. Yu, T. Bauer, and K. Steiner. Fuzzymatte: A computationally efficient scheme for interactive matting. In *CVPR08*, 2008.

[20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA. USA, 2004. MIT Press.

[21] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison, 2005.