Designing Effective Inter-Pixel Information Flow for Natural Image Matting

Yağız Aksoy[1,2], Tunç Ozan Aydın[2], Marc Pollefeys[1]

[1] ETH Zürich [2] Disney Research Zürich

Abstract

We present a novel, purely affinity-based natural image matting algorithm. Our method relies on carefully defined pixel-to-pixel connections that enable effective use of information available in the image and the trimap. We control the information flow from the known-opacity regions into the unknown region, as well as within the unknown region itself, by utilizing multiple definitions of pixel affinities. This way we achieve significant improvements on matte quality near challenging regions of the foreground object. Among other forms of information flow, we introduce color-mixture flow, which builds upon local linear embedding and effectively encapsulates the relation between different pixel opacities. Our resulting novel linear system formulation can be solved in closed-form and is robust against several fundamental challenges in natural matting such as holes and remote intricate structures. While our method is primarily designed as a standalone natural matting tool, we show that it can also be used for regularizing mattes obtained by various sampling-based methods. Our evaluation using the public alpha matting benchmark suggests a significant performance improvement over the state-of-the-art.

1. Introduction

Extracting the opacity information of foreground objects from an image is known as natural image matting. Natural image matting has received great interest from the research community through the last decade and can nowadays be considered as one of the classical research problems in visual computing. Mathematically, image matting requires expressing pixel colors in the transition regions from foreground to background as a convex combination of their underlying foreground and background colors. The weight, or the *opacity*, of the foreground color is referred to as the alpha value of that pixel. Since neither the foreground and background colors nor the opacities are known, estimating the opacity values is a highly ill-posed problem. To alleviate the difficulty of this problem, typically a *trimap* is provided in addition to the original image. The trimap is a rough segmentation of the input image into foreground, background, and regions with unknown opacity.

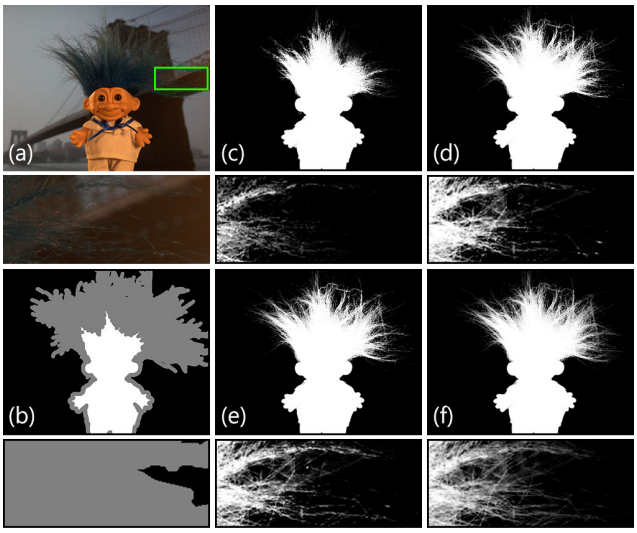

Figure 1. For an input image (a) and a trimap (b), we define several forms of information flow inside the image. We begin with color-mixture flow (c), then add direct channels of information flow from known to unknown regions (d), and let effective share of information inside the unknown region (e) to increase the matte quality in challenging regions. We finally add local information flow to get our spatially smooth result (f).

Affinity-based methods [4, 5, 11] constitute one of the prominent natural matting approaches in literature. These methods make use of pixel similarities to propagate the alpha values from the known-alpha regions to the unknown region. They provide a clear mathematical formulation, can be solved in closed-form, are easy to implement, and typically produce spatially consistent mattes. However, current methods fail to effectively handle alpha gradients spanning large areas and spatially disconnected unknown regions (i.e. *holes*) even in simple cases as demonstrated in Figure 2. This is because a straightforward formulation using the pixel-to-pixel affinity definitions can not effectively represent the complex structures that are commonly seen in real-life objects.

In order to alleviate these shortcomings, we rely on a careful, case-by-case design of how alpha values should propagate inside the image. We refer to this propagation as *information flow*. The key idea of our paper is a novel strat-

egy for controlling information flow both from the known trimap regions to the unknown region, as well as within the unknown region itself. We formulate this strategy through the use of a variety of affinity definitions including the *color-mixture flow*, which is based on local linear embedding and tailored for image matting. Step-by-step improvements on the matte quality as we gradually add new building blocks of our information flow strategy are illustrated in Figure 1. Our final linear system can be solved in closed-form and results in a significant quality improvement over the state-of-the-art. We demonstrate the matting quality improvement quantitatively, as well as through a visual inspection of challenging image regions. We also show that our energy function can be reformulated as a post-processing step for regularizing the spatially inconsistent mattes estimated by sampling-based natural matting algorithms.

2. Related work

Opacity estimation in images is an active research topic with a diverse set of applications, such as green-screen keying [1], soft color segmentation [2, 17], reflection removal [16], and deblurring [12]. In this paper, we aim to estimate the opacity channel of objects in front of a complex background, a problem referred to as natural image matting.

The numerous natural matting methods in the literature can be mainly categorized as either sampling-based or affinity-based. In this section, we briefly review methods that are the most relevant to our work and refer the reader to a comprehensive survey [19] for further information.

Sampling-based methods [8, 9, 10, 15] typically seek to gather numerous samples from the background and foreground regions defined by the trimap and select the best-fitting pair according to their individually defined criteria for representing an unknown pixel as a mixture of foreground and background. While they perform well especially around remote and challenging structures, they require affinity-based regularization to produce spatially consistent mattes. Also, our experience with publicly available matting code suggests that implementing sampling-based methods can be challenging at times.

Affinity-based matting methods mainly make use of pixel similarity metrics that rely on color similarity or spatial proximity and propagate the alpha values from regions with known opacity. Local affinity definitions, prominently the matting affinity [11], operate on a local patch around the pixel location to determine the amount of local information flow and propagate alpha values accordingly. The matting affinity is also widely adopted as a post-processing step in sampling-based methods [8, 10, 15] as proposed by Gastal and Oliveira [9].

Methods utilizing nonlocal affinities similarly use color similarity and spatial proximity for determining how the alpha values of different pixels should relate to each other. KNN matting [4] determines several neighbors for every unknown pixel and enforces them to have similar alpha values relative to their distance in a feature space. The manifold-preserving edit propagation algorithm [5] also determines a set of neighbors for every pixel, but represents each pixel as a linear combination of its neighbors in their feature space.

Chen *et al*. [6] proposed a hybrid approach that uses the sampling-based robust matting [18] as a starting point and refines its outcome through a graph-based technique where they combine a nonlocal affinity [5] and the matting affinity. Cho *et al*. [7] combined the results of closed-form matting [11] and KNN matting [4], as well as the sampling-based method comprehensive sampling [15], by feeding them into a convolutional neural network.

In this work, we propose color-mixture flow and discuss its advantages over the affinity definition utilized by Chen *et al*. [5]. We also define three other forms of information flow, which we use to carefully distribute the alpha information inside the unknown region. Our approach differs from Chen *et al*. [6] in that our overall information flow strategy goes beyond combining various pixel affinities, as we discuss further in Section 3, while requiring much less memory to solve the final system. Instead of using the results of other affinity-based methods directly as done by Cho *et al*. [7], we formulate an elegant formulation that has a closed-form solution. To summarize, we present a novel, purely affinity-based matting algorithm that generates high-quality alpha mattes without making use of a sampling-based method or a learning step.

3. Method

Trimaps are typically given as user input in natural matting, and they consist of three regions: fully opaque (foreground), fully transparent (background) and of unknown opacity. $\mathcal{F}$, $\mathcal{B}$ and $\mathcal{U}$ will respectively denote these regions, and $\mathcal{K}$ will represent the union of $\mathcal{F}$ and $\mathcal{B}$. Affinity-based methods operate by propagating opacity information from $\mathcal{K}$ into $\mathcal{U}$ using a variety of affinity definitions. We define this flow of information in multiple ways so that all the pixels inside $\mathcal{U}$ receives information effectively from different regions in the image.

The opacity transitions in a matte occur as a result of the original colors in the image getting mixed with each other due to transparency or intricate parts of an object. We make use of this fact by representing each pixel in $\mathcal{U}$ as a mixture of similarly-colored pixels and defining a form of information flow that we call *color-mixture flow* (Section 3.1). We also add connections from every pixel in $\mathcal{U}$ to both $\mathcal{F}$ and $\mathcal{B}$ to facilitate direct information flow from known-opacity regions to even the most remote opacity-transition regions in the image (Section 3.2). In order to distribute the information from the color-mixture and $\mathcal{K}$-to-$\mathcal{U}$ flows, we define intra-$\mathcal{U}$ flow of information, where pixels with simi-

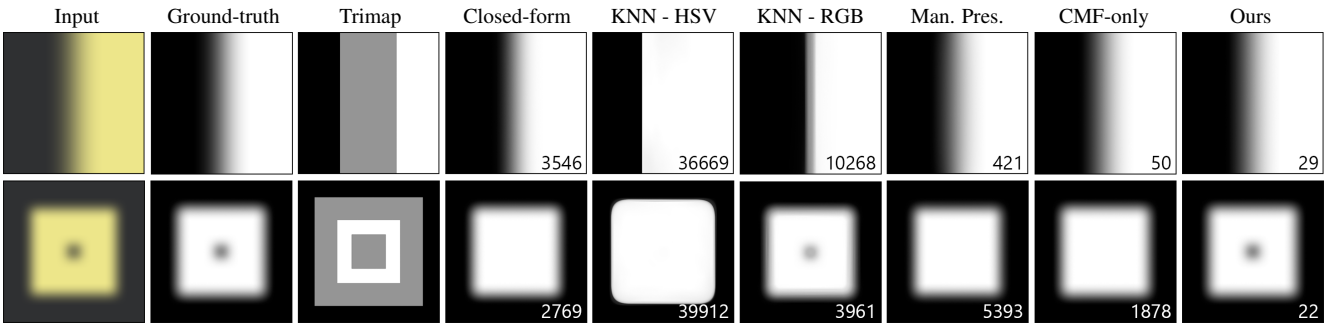

Figure 2. We created two duotone 500x500 images and blurred them to get soft transitions between regions. The numbers show the sum of absolute differences between the estimated alpha mattes and the ground truth. Closed-form matting [11] uses local information flow, KNN Matting [4] uses HSV- or RGB-based similarity measure, and manifold-preserving edit propagation [5] uses LLE weights [14]. We observe a performance improvement in large opacity gradients even when only the color-mixture flow (CMF) is used (Section 3.1). Notice also that both large gradients and holes are recovered with high precision using our final formulation. See text for further discussion.

lar colors inside $\mathcal{U}$ share information on their opacity with each other (Section 3.3). Finally, we add local information flow, a pixel affecting the opacity of its immediate spatial neighbors, which ensures spatially coherent end results (Section 3.4). We formulate the individual forms of information flow as energy functions and aggregate them in a global optimization formulation (Section 3.5).

3.1. Color-mixture information flow

Due to transparent objects as well as fine structures and sharp edges of an object that cannot be fully captured due to the finite-resolution of the imaging sensors, certain pixels of an image inevitably contain a mixture of corresponding foreground and background colors. By investigating these color mixtures, we can derive an important clue on how to propagate alpha values between pixels. The amount of the original foreground color in a particular mixture determines the opacity of the pixel. Following this fact, if we represent the color of a pixel as a weighted combination of the colors of several others, those weights should also represent the opacity relation between the pixels.

In order to make use of this relation, for every pixel in $\mathcal{U}$, we find $K_{CM} = 20$ similar pixels in a feature space by an approximate K nearest neighbors search in the whole image. We define the feature vector for this search as $[r, g, b, \tilde{x}, \tilde{y}]^T$, where $\tilde{x}$ and $\tilde{y}$ are the image coordinates normalized by image width and height, and the rest are the RGB values of the pixel. This set of neighbors, selected as similar-colored pixels that are also close-by, is denoted by $\mathcal{N}_p^{CM}$.

We then find the weights of the combination $w_{p,q}^{CM}$ that will determine the amount of information flow between the pixel p and $q \in \mathcal{N}_p^{CM}$. The weights are defined such that the colors of the neighbors of a pixel gives the original pixel color when combined:

$$\arg\min_{w_{p,q}^{CM}} \left\| \boldsymbol{c}_p - \sum_{q \in \mathcal{N}_p^{CM}} w_{p,q}^{CM} \boldsymbol{c}_q \right\|^2, \qquad (1)$$

where $\boldsymbol{c}_p$ represents the 3x1 vector of RGB values. We minimize this energy using the method by Roweis and Saul [14]. Note that since we are only using RGB values, the neighborhood correlation matrix computed during the minimization has a high chance of being singular as there could easily be two neighbors with identical colors. So, we condition the neighborhood correlation matrix by adding $10^{-3} I_{K_{CM} \times K_{CM}}$ to it before inversion, where $I_{K_{CM} \times K_{CM}}$ is the identity matrix.

Note that while we use the method by Roweis and Saul [14] to minimize the energy in (1), we do not fully adopt their local linear embedding (LLE) method. LLE finds a set of neighbors in a feature space and uses all the variables in the feature space to compute the weights in order to reduce the dimentionality of input data. Manifold-preserving edit propagation [5] and LNSP matting [6] algorithms make use of the LLE weights directly in their formulation for image matting. However, since we are only interested in the weighted combination of colors and not the spatial coordinates, we exclude the spatial coordinates in the energy minimization step. This increases the validity of the estimated weights, effects of which can be observed even in the simplest cases such as in Figure 2, where manifold-preserving weight propagation and CMF-only results only differ in the weight computation step.

We define the energy term representing the color-mixture flow as:

$$E_{CM} = \sum_{p \in \mathcal{U}} \left(\alpha_p - \sum_{q \in \mathcal{N}_p^{CM}} w_{p,q}^{CM} \alpha_q \right)^2. \qquad (2)$$

3.2. $\mathcal{K}$-to-$\mathcal{U}$ information flow

The color-mixture flow already provides useful information on how the mixed-color pixels are formed. However, many pixels in $\mathcal{U}$ receive information present in the trimap indirectly through their neighbors, all of which can possibly

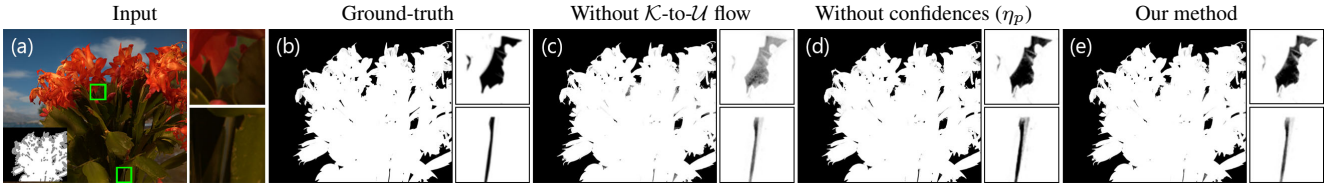

Figure 3. Direct information flow from both $\mathcal{F}$ and $\mathcal{B}$ to even the most remote regions in $\mathcal{U}$ increases our performance around holes significantly (top inset). Using confidences further increases the performance, especially around regions where foreground and background colors are similar (bottom inset).

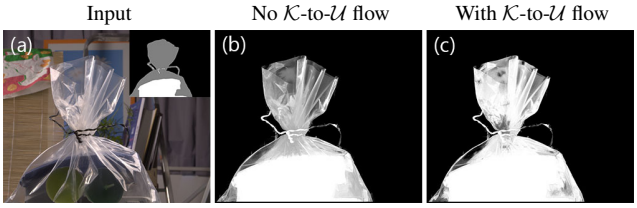

Figure 4. $\mathcal{K}$-to-$\mathcal{U}$ flow does not perform well when the foreground object is highly-transparent. See text for discussion.

be in $\mathcal{U}$. This indirect information flow might not be enough especially for remote regions that are far away from $\mathcal{K}$.

In order to facilitate the flow of information from both $\mathcal{F}$ and $\mathcal{B}$ directly into every region in $\mathcal{U}$, we add connections from every pixel in $\mathcal{U}$ to several pixels in $\mathcal{K}$. For each pixel in $\mathcal{U}$, we find $K_{\mathcal{KU}} = 7$ similar pixels in both $\mathcal{F}$ and $\mathcal{B}$ separately to form the sets of pixels $\mathcal{N}_p^{\mathcal{F}}$ and $\mathcal{N}_p^{\mathcal{B}}$ with K nearest neighbors search using the feature space $[r, g, b, 10 * \tilde{x}, 10 * \tilde{y}]^T$ to favor close-by pixels. We use the pixels in $\mathcal{N}_p^{\mathcal{F}}$ and $\mathcal{N}_p^{\mathcal{B}}$ together to represent the pixel color $\boldsymbol{c}_p$ by minimizing the energy in (1). Using the resulting weights $w_{p,q}^{\mathcal{F}}$ and $w_{p,q}^{\mathcal{B}}$, we define an energy function to represent the $\mathcal{K}$-to-$\mathcal{U}$ flow:

$$E_{\mathcal{KU}} = \sum_{p \in \mathcal{U}} \left(\alpha_p - \sum_{q \in \mathcal{N}_p^{\mathcal{F}}} w_{p,q}^{\mathcal{F}} \alpha_q - \sum_{q \in \mathcal{N}_p^{\mathcal{B}}} w_{p,q}^{\mathcal{B}} \alpha_q \right)^2 \quad (3)$$

Note that $\alpha_q = 1$ for $q \in \mathcal{F}$ and $\alpha_q = 0$ for $q \in \mathcal{B}$. This fact allows us to define two combined weights, one connecting a pixel to $\mathcal{F}$ and another to $\mathcal{B}$, as:

$$w_p^{\mathcal{F}} = \sum_{q \in \mathcal{N}_p^{\mathcal{F}}} w_{p,q}^{\mathcal{F}} \quad \text{and} \quad w_p^{\mathcal{B}} = \sum_{q \in \mathcal{N}_p^{\mathcal{B}}} w_{p,q}^{\mathcal{B}} \quad (4)$$

such that $w_p^{\mathcal{F}} + w_p^{\mathcal{B}} = 1$, and rewrite (3) as:

$$E_{\mathcal{KU}} = \sum_{p \in \mathcal{U}} \left(\alpha_p - w_p^{\mathcal{F}} \right)^2. \quad (5)$$

The energy minimization in (1) gives us similar weights for all q when $\boldsymbol{c}_q$ are similar to each other. As a result, if $\mathcal{N}_p^{\mathcal{F}}$ and $\mathcal{N}_p^{\mathcal{B}}$ have pixels with similar colors, the estimated weights $w_p^{\mathcal{F}}$ and $w_p^{\mathcal{B}}$ become unreliable. We account for this fact by augmenting the energy function in (5) with confidence values.

We can determine the colors contributing to the mixture estimated by (1) using the weights $w_{p,q}^{\mathcal{F}}$ and $w_{p,q}^{\mathcal{B}}$:

$$\boldsymbol{c}_p^{\mathcal{F}} = \frac{\sum_{q \in \mathcal{N}_p^{\mathcal{F}}} w_{p,q}^{\mathcal{F}} \boldsymbol{c}_q}{w_p^{\mathcal{F}}}, \quad \boldsymbol{c}_p^{\mathcal{B}} = \frac{\sum_{q \in \mathcal{N}_p^{\mathcal{B}}} w_{p,q}^{\mathcal{B}} \boldsymbol{c}_q}{w_p^{\mathcal{B}}}, \quad (6)$$

and define a confidence metric according to how similar the estimated foreground color $\boldsymbol{c}_p^{\mathcal{F}}$ and background color $\boldsymbol{c}_p^{\mathcal{B}}$ are:

$$\eta_p = \left\| \boldsymbol{c}_p^{\mathcal{F}} - \boldsymbol{c}_p^{\mathcal{B}} \right\|^2 / 3. \quad (7)$$

The division by 3 is to get the confidence values between $[0, 1]$. We update the new energy term to reflect our confidence in the estimation:

$$\tilde{E}_{\mathcal{KU}} = \sum_{p \in \mathcal{U}} \eta_p \left(\alpha_p - w_p^{\mathcal{F}} \right)^2. \quad (8)$$

This update to the energy term increases the matting quality in regions with similar foreground and background colors, as seen in Figure 3.

It should be noted that the $\mathcal{K}$-to-$\mathcal{U}$ information flow is not reliable when the foreground object is highly transparent, as seen in Figure 4. This is mainly due to the low representational power of $\mathcal{N}_p^{\mathcal{F}}$ and $\mathcal{N}_p^{\mathcal{B}}$ for $\boldsymbol{c}_p$ around large highly-transparent regions as the nearest neighbors search does not give us well-fitting pixels for $w_{p,q}^{\mathcal{F}}$ estimation. We construct our final linear system accordingly as we discuss further in Section 3.5.

3.2.1 Pre-processing the trimap

Prior to determining $\mathcal{N}_p^{\mathcal{F}}$ and $\mathcal{N}_p^{\mathcal{B}}$, we pre-process the input trimap in order to facilitate finding more reliable neighbors, which in turn increases the effectiveness of the $\mathcal{K}$-to-$\mathcal{U}$ flow. Trimaps usually have regions marked as $\mathcal{U}$ despite being fully opaque or transparent, as drawing a very detailed trimap is a very cumbersome and error-prone job. Several methods [8, 10] refine the trimap as a pre-processing step by expanding $\mathcal{F}$ and $\mathcal{B}$ starting from their boundaries with $\mathcal{U}$ as proposed by Shahrian *et al.* [15]. Incorporating this technique improves our results as shown in Figure 5(d). We also apply this extended $\mathcal{F}$ and $\mathcal{B}$ regions after the matte estimation as a post-processing. Since the trimap trimming method by Shahrian *et al.* [15] propagates known regions

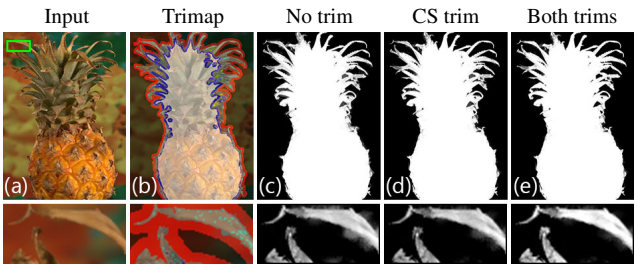

Figure 5. The trimap is shown overlayed on the original image (b) where the extended foreground regions are shown with blue (CS trimming [15]) and cyan (patch-search) and the extended background regions with red (CS trimming) and yellow (patch-search). CS trimming makes the fully opaque / transparent regions cleaner, while our trimming improves the results around remote structures.

only to nearby pixels, in addition to this edge-based trimming, we also make use of a patch-based trimming step.

To this end, we extend the transparent and opaque regions by relying on patch statistics. We fit a 3D RGB normal distribution N_p to the 3×3 window around each pixel p. In order to determine the most similar distribution in $\mathcal{F}$ for a pixel $p \in \mathcal{U}$, we first find the 20 distributions with closest mean vectors. We define the foreground match score $b_p^{\mathcal{F}} = \min_{q \in \mathcal{F}} B(N_p, N_q)$, where $B(\cdot,\cdot)$ represents the Bhattacharyya distance between two normal distributions. We find the match score for background $b_p^{\mathcal{B}}$ the same way. We then select a region for pixel p according to the following rule:

$$p \in \begin{cases} \hat{\mathcal{F}} & \text{if } b_p^{\mathcal{F}} < \tau_c \text{ and } b_p^{\mathcal{B}} > \tau_f \\ \hat{\mathcal{B}} & \text{if } b_p^{\mathcal{B}} < \tau_c \text{ and } b_p^{\mathcal{F}} > \tau_f \\ \hat{\mathcal{U}} & \text{otherwise} \end{cases} \tag{9}$$

Simply put, an unknown pixel is marked as $\hat{\mathcal{F}}$, *i.e.* in foreground after trimming, if it has a strong match in $\mathcal{F}$ and no match in $\mathcal{B}$, which is determined by constants $\tau_c = 0.25$ and $\tau_f = 0.9$. By inserting known-alpha pixels in regions far away from $\mathcal{U}$-$\mathcal{K}$ boundaries, we further increase the matting performance in challenging remote regions (Figure 5(e)).

3.3. Intra-$\mathcal{U}$ information flow

Each individual pixel in $\mathcal{U}$ receives information through the color-mixture and $\mathcal{K}$-to-$\mathcal{U}$ flows. In addition to these, we would like to distribute the information inside $\mathcal{U}$ effectively. We achieve this by encouraging pixels with similar colors inside $\mathcal{U}$ to have similar opacity.

For each pixel in $\mathcal{U}$, we find $K_{\mathcal{U}} = 5$ nearest neighbors only inside $\mathcal{U}$ to determine $\hat{\mathcal{N}}_p^{\mathcal{U}}$ using the feature vector defined as $\boldsymbol{v} = [r, g, b, \tilde{x}/20, \tilde{y}/20]^T$. Notice that we scale the coordinate members of the feature vector we used in Section 3.1 to decrease their effect on the nearest neighbor selection. This lets $\hat{\mathcal{N}}_p^{\mathcal{U}}$ have pixels inside $\mathcal{U}$ that is far away, so that the information moves more freely inside the unknown region. We use the neighborhood $\mathcal{N}_p^{\mathcal{U}} = \hat{\mathcal{N}}_p^{\mathcal{U}} \cup \{q \mid p \in \hat{\mathcal{N}}_q^{\mathcal{U}}\}$ to make sure that information flows both ways between p to $q \in \hat{\mathcal{N}}_p^{\mathcal{U}}$. We then determine the amount of information flow using the L^1 distance between feature vectors:

$$w_{p,q}^{\mathcal{U}} = \max\left(1 - \|\boldsymbol{v}_p - \boldsymbol{v}_q\|_1,\ 0\right) \quad \forall q \in \mathcal{N}_p^{\mathcal{U}}. \tag{10}$$

The energy term for intra-$\mathcal{U}$ information flow then can be defined as:

$$E_{\mathcal{U}\mathcal{U}} = \sum_{p \in \mathcal{U}} \sum_{q \in \mathcal{N}_p^{\mathcal{U}}} w_{p,q}^{\mathcal{U}} \left(\alpha_p - \alpha_q\right)^2. \tag{11}$$

The information sharing between the unknown pixels increases the matte quality around intricate structures as demonstrated in Figure 1(e).

KNN matting [4] uses a similar affinity definition to make similar-color pixels have similar opacities. However, relying only on this form of information flow alone for the whole image creates some typical artifacts in the resulting alpha mattes. Depending on the feature vector definition and the image colors, the resulting alpha values may erroneously underrepresent the smooth transitions (KNN - HSV case in Figure 2) when the neighbors of the pixels in $\mathcal{U}$ happen to be mostly in only $\mathcal{F}$ or $\mathcal{B}$, or create flat, constant alpha regions instead of subtle gradients (KNN - RGB case in Figure 2). Restricting information flow to be applied solely based on color similarity fails to represent the complex alpha transitions or wide regions with an alpha gradient.

3.4. Local information flow

Spatial connectivity is one of the main cues for information flow. We connect each pixel in $\mathcal{U}$ to its 8 immediate neighbors denoted by $\mathcal{N}_p^L$ to ensure spatially smooth mattes. The amount of local information flow should also adapt to strong edges in the image.

To determine the amount of local flow, we rely on the matting affinity definition proposed by Levin *et al.* [11]. The matting affinity utilizes the local patch statistics to determine the weights $w_{p,q}^L$, $q \in \mathcal{N}_p^L$. We define our related energy term as follows:

$$E_L = \sum_{p \in \mathcal{U}} \sum_{q \in \mathcal{N}_p^L} w_{p,q}^L \left(\alpha_p - \alpha_q\right)^2. \tag{12}$$

Despite representing local information flow well, matting affinity by itself fails to represent large transition regions (Figure 2 top), or isolated regions that have weak or no spatial connection to $\mathcal{F}$ or $\mathcal{B}$ (Figure 2 bottom).

3.5. Linear system and energy minimization

Our final energy function is a combination of the four energy definitions representing each form of information flow:

$$E_1 = E_{CM} + \sigma_{\mathcal{K}\mathcal{U}} E_{\mathcal{K}\mathcal{U}} + \sigma_{\mathcal{U}\mathcal{U}} E_{\mathcal{U}\mathcal{U}} + \sigma_L E_L + \lambda E_{\mathcal{T}}, \tag{13}$$

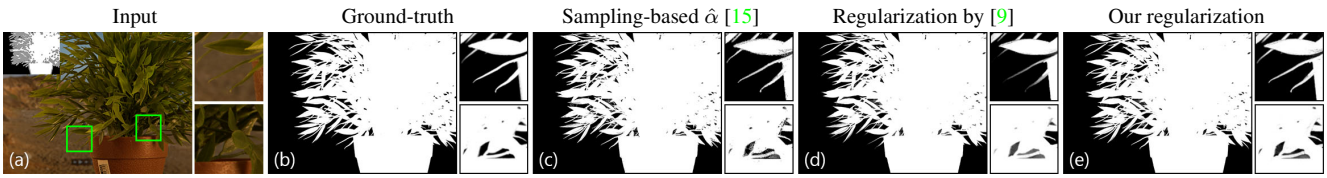

Figure 6. The matte regularization method by Gastal and Oliveira [9] loses remote details (top inset) or fills in holes (bottom inset) while our regularization method is able to preserve these details caught by the sampling-based method.

where $\sigma_{\mathcal{KU}} = 0.05$, $\sigma_{\mathcal{UU}} = 0.01$, $\sigma_L = 1$ and $\lambda = 100$ are algorithmic constants determining the strength of corresponding information flows, and

$$E_{\mathcal{T}} = \sum_{p \in \mathcal{F}} (\alpha_p - 1)^2 + \sum_{p \in \mathcal{B}} (\alpha_p - 0)^2$$

is the energy term to keep the known opacity values constant. For an image with N pixels, by defining $N \times N$ sparse matrices W_{CM}, $W_{\mathcal{UU}}$ and W_L that have non-zero elements for the pixel pairs with corresponding information flows and the vector $\boldsymbol{w}^{\mathcal{F}}$ that has elements $w_p^{\mathcal{F}}$ for $p \in \mathcal{U}$, 1 for $p \in \mathcal{F}$ and 0 for $p \in \mathcal{B}$, we can rewrite (13) in matrix form as:

$$\begin{aligned} E_1 =& \boldsymbol{\alpha}^T \mathcal{L}_{IFM} \boldsymbol{\alpha} + (\boldsymbol{\alpha} - \boldsymbol{w}^{\mathcal{F}})^T \sigma_{\mathcal{KU}} \mathcal{H} (\boldsymbol{\alpha} - \boldsymbol{w}^{\mathcal{F}}) + \\ & (\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{K}})^T \lambda \mathcal{T} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{K}}), \end{aligned} \tag{14}$$

where $\mathcal{T}$ is an $N \times N$ diagonal matrix with diagonal entry (p, p) 1 if $p \in \mathcal{K}$ and 0 otherwise, $\mathcal{H}$ is a sparse matrix with diagonal entries η_p as defined in (7), $\boldsymbol{\alpha}_{\mathcal{K}}$ is a row vector with p^{th} entry being 1 if $p \in \mathcal{F}$ and 0 otherwise, $\boldsymbol{\alpha}$ is a row-vector of the alpha values to be estimated, and $\mathcal{L}_{IFM}$ is defined as:

$$\begin{aligned} \mathcal{L}_{IFM} =& (D_{CM} - W_{CM})^T (D_{CM} - W_{CM}) + \\ & \sigma_{\mathcal{UU}} (D_{\mathcal{UU}} - W_{\mathcal{UU}}) + \sigma_L (D_L - W_L), \end{aligned} \tag{15}$$

where the diagonal matrix $D_{(\cdot)}(i,i) = \sum_j W_{(\cdot)}(i,j)$.

The energy in (14) can be minimized by solving

$$(\mathcal{L}_{IFM} + \lambda \mathcal{T} + \sigma_{\mathcal{KU}} \mathcal{H})\, \boldsymbol{\alpha} = (\lambda \mathcal{T} + \sigma_{\mathcal{KU}} \mathcal{H})\, \boldsymbol{w}^{\mathcal{F}}. \tag{16}$$

We define a second energy function that excludes the $\mathcal{K}$-to-$\mathcal{U}$ information flow:

$$E_2 = E_{CM} + \sigma_{\mathcal{UU}} E_{\mathcal{UU}} + \sigma_L E_L + \lambda E_{\mathcal{T}}, \tag{17}$$

which can be written in matrix form as:

$$E_2 = \boldsymbol{\alpha}^T \mathcal{L}_{IFM} \boldsymbol{\alpha} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{K}})^T \lambda \mathcal{T} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{K}}), \tag{18}$$

and can be minimized by solving:

$$(\mathcal{L}_{IFM} + \lambda \mathcal{T})\, \boldsymbol{\alpha} = \lambda \mathcal{T} \boldsymbol{\alpha}_{\mathcal{K}}. \tag{19}$$

We solve the linear systems of equations in (16) and (19) using the preconditioned conjugate gradients method [3].

As mentioned before, the $\mathcal{K}$-to-$\mathcal{U}$ information flow is not effective for highly transparent objects. To determine whether to include the $\mathcal{K}$-to-$\mathcal{U}$ information flow and solve for E_1, or to exclude it and solve for E_2 for a given image, we use a simple histogram-based classifier to determine if we expect a highly transparent result.

If the matte is highly transparent, the pixels in $\mathcal{U}$ are expected to mostly have colors that are a mixture of $\mathcal{F}$ and $\mathcal{B}$ colors. On the other hand, if the true alpha values are mostly 0 or 1 except for soft transitions, the histogram of $\mathcal{U}$ will likely be a linear combination of the histograms of $\mathcal{F}$ and $\mathcal{B}$ as $\mathcal{U}$ will mostly include very similar colors to that of $\mathcal{K}$. Following this observation, we attempt to express the histogram of the pixels in $\mathcal{U}$, $\mathcal{D}_{\mathcal{U}}$, as a linear combination of $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{B}}$. The histograms are computed from the 20 pixel-wide region around $\mathcal{U}$ in $\mathcal{F}$ and $\mathcal{B}$, respectively. We define the error e, the metric of how well the linear combination represents the true histogram, as:

$$e = \min_{a,b} \|a\mathcal{D}_{\mathcal{F}} + b\mathcal{D}_{\mathcal{B}} - \mathcal{D}_{\mathcal{U}}\|^2. \tag{20}$$

Higher e values indicate a highly-transparent matte, in which case we prefer E_2 over E_1.

4. Matte regularization for sampling-based matting methods

Sampling-based natural matting methods usually select samples for each pixel in $\mathcal{U}$ either independently or by paying little attention to spatial coherency. In order to obtain a spatially coherent matte, the common practice is to combine their initial guesses for alpha values with a smoothness measure. Multiple methods [8, 9, 10, 15] adopt the post-processing method proposed by Gastal and Oliveira [9] which combines the matting affinity [11] with the sampling-based alpha values and corresponding confidences. This post-processing technique leads to improved mattes, but since it involves only local smoothness, the results can still be suboptimal as seen in Figure 6(d).

Our approach with multiple forms of information flow can also be used for post-processing in a way similar to that of Gastal and Oliveira [9]. Given the initial alpha values $\hat{\alpha}_p$ and confidences $\hat{\eta}_p$ found by a sampling-based method, we define the matte regularization energy:

$$E_R = E_2 + \sigma_R \sum_{p \in \mathcal{U}} \hat{\eta}_p (\alpha_p - \hat{\alpha}_p)^2, \tag{21}$$

Table 1. Our scores in the alpha matting benchmark [13] together with the top-performing published methods at the time of submission. *S*, *L* and *U* denote the three trimap types, small, large and user, included in the benchmark. Bold and blue numbers represent the best scores obtained among all the published methods in the benchmark*.

	Average Rank**				Troll			Doll			Donkey			Elephant			Plant			Pineapple			Plastic bag			Net		
	Overall	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*	*S*	*L*	*U*
Sum of Absolute Differences																												
Ours	2.1	2.8	1.6	2.0	**10.3**	**11.2**	12.5	5.6	7.3	7.3	**3.8**	**4.1**	**3**	**1.4**	**2.3**	**2.0**	5.9	7.1	**8.6**	**3.6**	**5.7**	**4.6**	18.3	19.3	**15.8**	20.2	22.2	22.3
DCNN [7]	3.2	4.6	1.6	3.4	12.0	14.1	14.5	**5.3**	**6.4**	6.8	3.9	4.5	3.4	1.6	2.5	2.2	6.0	**6.9**	9.1	4.0	6.0	5.3	19.9	**19.2**	19.1	**19.4**	**20.0**	**21.2**
CSC [8]	10	13.5	6.4	10.3	13.6	15.6	14.5	6.2	7.5	8.1	4.6	4.8	4.2	1.8	2.7	2.5	5.5	7.3	9.7	4.6	7.6	6.9	23.7	23.0	21.0	26.3	27.2	25.2
LNSP [6]	10.7	7.3	10.3	14.6	12.2	22.5	19.5	5.6	8.1	8.8	4.6	5.9	3.6	1.5	3.5	3.1	6.2	8.1	10.7	4.0	7.1	6.4	21.5	20.8	16.3	22.5	24.4	27.8
Mean Squared Error																												
Ours	3.5	5.0	2.1	3.4	**0.3**	**0.4**	0.5	0.3	0.4	0.5	0.3	0.3	0.2	0.1	0.1	0.1	0.4	0.4	**0.6**	0.2	**0.3**	0.3	1.3	1.2	**0.8**	0.8	0.8	0.9
DCNN [7]	3.7	4.5	1.9	4.6	0.4	0.5	0.7	**0.2**	**0.3**	0.4	**0.2**	0.3	0.2	0.1	**0.1**	**0.1**	0.4	**0.4**	0.8	0.2	0.4	0.3	1.3	1.2	1.0	**0.7**	**0.7**	0.9
LNSP [6]	9.2	6.6	8.6	12.4	0.5	1.9	1.2	0.2	0.4	0.5	0.3	0.4	0.2	**0.0**	0.1	0.2	0.4	0.5	0.8	**0.2**	0.3	0.4	1.4	1.2	0.8	1.0	1.1	1.5
KL-D [10]	11.6	11.0	10.5	13.3	0.4	0.9	0.7	0.3	0.5	0.5	0.3	0.4	0.3	0.1	0.2	0.1	0.4	0.4	1.2	0.4	0.6	0.6	1.7	2.0	2.1	0.8	0.8	0.9

* Some columns do not have a bold number when the best-scoring algorithm for that particular image-trimap pair is not among the top-ranking methods included here.

** The ranks presented here only take the already-published methods at the time of the submission into account, hence could differ from the online version of the benchmark.

where $\sigma_R = 0.05$ determines how much loyalty should be given to the initial values. This energy can be written in the matrix form as

$$E_R = \boldsymbol{\alpha}^T \mathcal{L}_{IFM} \boldsymbol{\alpha} + (\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}})^T \sigma_R \hat{\mathcal{H}} (\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{K}})^T \lambda \mathcal{T} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{K}}) \tag{22}$$

and minimized by solving

$$(\mathcal{L}_{IFM} + \lambda \mathcal{T} + \sigma_R \hat{\mathcal{H}}) \boldsymbol{\alpha} = (\lambda \mathcal{T} + \sigma_R \hat{\mathcal{H}}) \hat{\boldsymbol{\alpha}}. \tag{23}$$

Figure 6 shows that this non-local regularization of mattes is more effective especially around challenging foreground structures such as long leaves or holes as seen in the insets. In the next section, we will numerically explore the improvement we achieve by replacing the matte regularization step with ours in several sampling-based methods.

5. Results and discussion

We quantitatively evaluate the proposed algorithm using the public alpha matting benchmark [13]. At the time of submission, our method ranks in the first place according to the sum-of-absolute-differences (SAD) and mean-squared error (MSE) metrics. The results can be seen in Table 1. Our unoptimized research code written in Matlab requires on average 50 seconds to process a benchmark image.

We also compare our results qualitatively with the closely related methods in Figure 7. We use the results that are available on the matting benchmark for all except manifold-preserving matting [5] which we implemented ourselves. Figure 7(c,d,e) show that using only one form of information flow is not effective in a number of scenarios such as wide unknown regions or holes in the foreground object. The strategy DCNN matting [7] follows is using the results of closed-form and KNN matting directly rather than formulating a combined energy using their affinity definitions. When both methods fail, the resulting combination also suffers from the errors as it is apparent in the pineapple and troll examples. The neural network they propose also seems to produce mattes that appear slightly blurred. LNSP matting [6], on the other hand, has issues around regions with holes (pineapple example) or when the foreground and background colors are similar (donkey and troll examples). It can also oversmooth some regions if the true foreground colors are missing in the trimap (plastic bag example). Our method performs well in these challenging scenarios mostly because, as detailed in Section 3, we carefully define intra-unknown region and unknown-to-known region connections which results in a more robust linear system.

Table 2. Performance improvement achieved when our matte regularization method replaces the method by Gastal and Oliveira [9] in the post-processing steps of 3 sampling-based methods. The training dataset [13] of 27 images and 2 trimaps per image (S and L) was used for this comparison.

	Sum of Absolute Differences			Mean Squared Error		
	Overall	S	L	Overall	S	L
KL-D [10]	24.4 %	22.4 %	26.5 %	28.5 %	25.9 %	31.0 %
SM [9]	6.0 %	3.7 %	8.4 %	13.6 %	8.5 %	18.8 %
CS [15]	4.9 %	10.0 %	-0.1 %	18.7 %	25.5 %	11.8 %

We also compare the proposed post-processing method detailed in Section 4 with the state-of-the-art method by Gastal and Oliveira [9] on the training dataset provided by Rhemann *et al.* [13]. We computed the non-smooth alpha values and confidences using the publicly available source code for comprehensive sampling [15], KL-divergence sampling [10] and shared matting [9]. Table 2 shows the percentage improvement we achieve over Gastal and Oliveira [9] for each algorithm using SAD and MSE as error measures. Figure 8 shows an example for regularizing all three sampling-based methods. As the information coming from alpha values and their confidences found by the sampling-based method is distributed more effectively by the proposed method, the challenging regions such as fine structures or holes detected by the sampling-based method are preserved when our method is used for post-processing.

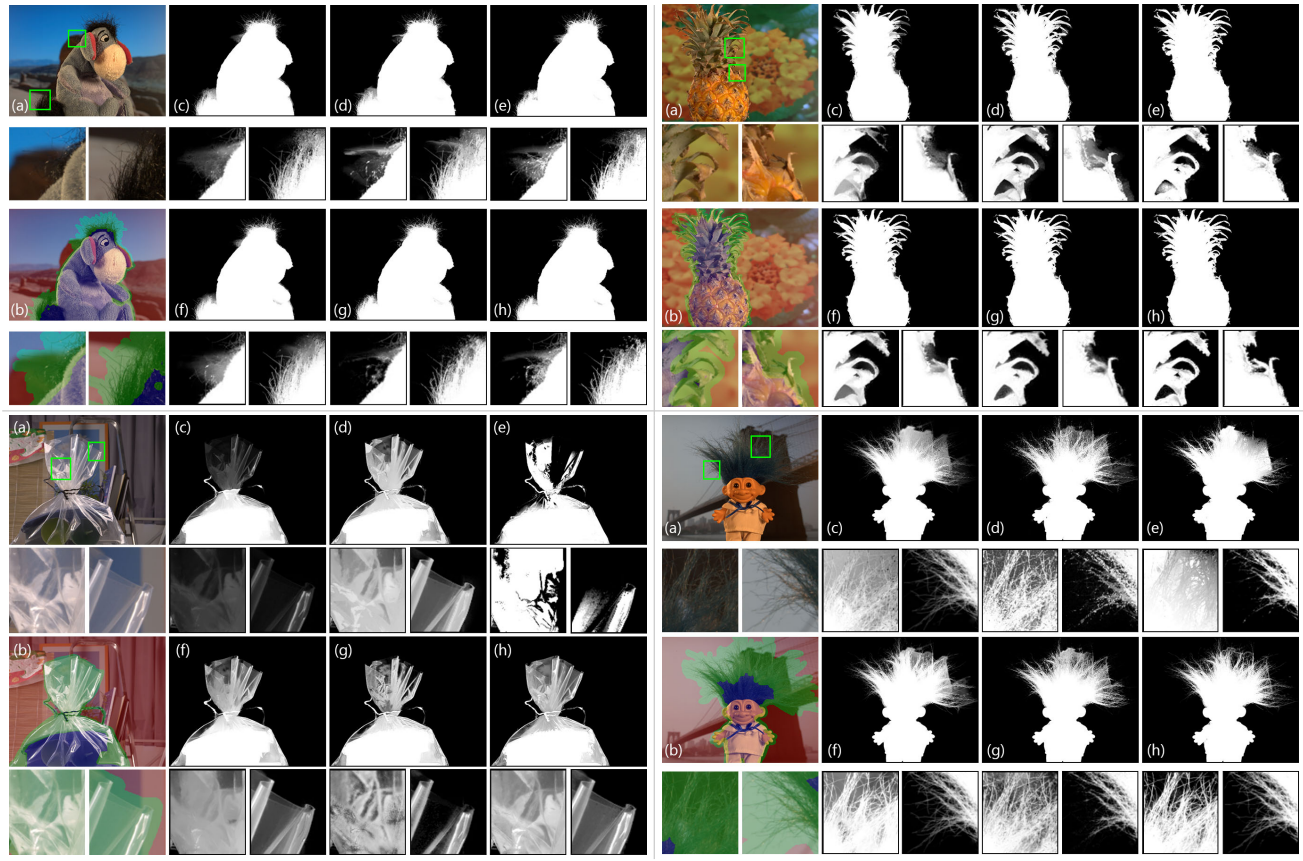

Figure 7. Several examples from the alpha matting benchmark [13] are shown (a) with trimaps overlayed onto the images (b). The mattes are computed by closed-form matting [11] (c), KNN matting [4] (d), manifold-preserving edit propagation [5] (e), LNSP matting [6] (f), DCNN matting [7] (g) and the proposed method (h). See text for discussion.

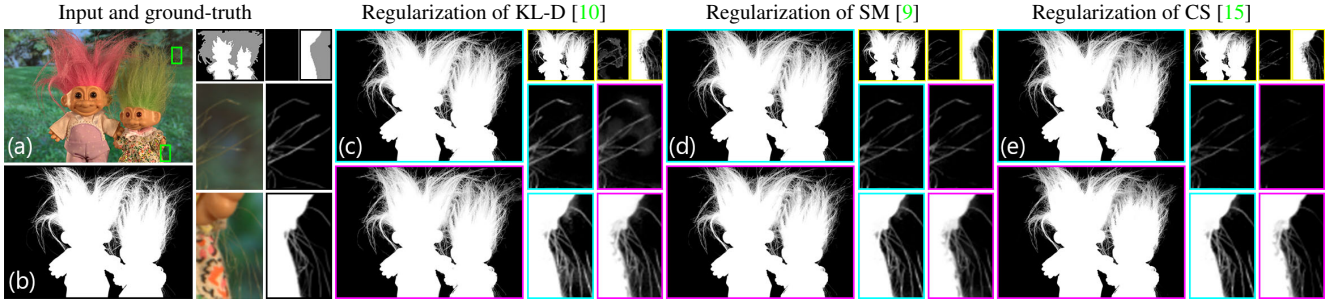

Figure 8. Matte regularization using the proposed method (cyan) or [9] (magenta) for three sampling-based methods (yellow). Our method is able to preserve remote details while producing a clean matte (top inset) and preserve sharpness even around textured areas (bottom).

6. Conclusion

In this paper, we proposed a purely affinity-based natural image matting method. We introduced color-mixture flow, a specifically tailored form of LLE weights for natural image matting. By carefully designing flow of information from the known region to the unknown region, as well as distributing the information inside the unknown region, we addressed several challenges that are common in natural matting. We showed that the linear system we formulate outperforms the state-of-the-art in the alpha matting benchmark. We also showed that our formulation can be used to replace the commonly used matte refinement step in sampling-based matting methods to achieve an increase in the final matte quality.

Acknowledgements: We would like thank Aljoša Smolić for our discussions in the early stages of this work, Simone Meyer, Jean-Charles Bazin and Kaan Yücer for their feedback on the text, and Simone Croci for his help in result generation.

References

[1] Y. Aksoy, T. O. Aydın, M. Pollefeys, and A. Smolić. Interactive high-quality green-screen keying via color unmixing. *ACM Trans. Graph.*, 35(5):152:1–152:12, 2016. 2

[2] Y. Aksoy, T. O. Aydın, A. Smolić, and M. Pollefeys. Unmixing-based soft color segmentation for image manipulation. *ACM Trans. Graph.*, 36(2):19:1–19:19, 2017. 2

[3] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1994. 6

[4] Q. Chen, D. Li, and C.-K. Tang. KNN matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9):2175–2188, 2013. 1, 2, 3, 5, 8

[5] X. Chen, D. Zou, Q. Zhao, and P. Tan. Manifold preserving edit propagation. *ACM Trans. Graph.*, 31(6):132:1–132:7, 2012. 1, 2, 3, 7, 8

[6] X. Chen, D. Zou, S. Zhou, Q. Zhao, and P. Tan. Image matting with local and nonlocal smooth priors. In *Proc. CVPR*, 2013. 2, 3, 7, 8

[7] D. Cho, Y.-W. Tai, and I. S. Kweon. Natural image matting using deep convolutional neural networks. In *Proc. ECCV*, 2016. 2, 7, 8

[8] X. Feng, X. Liang, and Z. Zhang. A cluster sampling method for image matting via sparse coding. In *Proc. ECCV*, 2016. 2, 4, 6, 7

[9] E. S. L. Gastal and M. M. Oliveira. Shared sampling for real-time alpha matting. *Comput. Graph. Forum*, 29(2):575–584, 2010. 2, 6, 7, 8

[10] L. Karacan, A. Erdem, and E. Erdem. Image matting with KL-divergence based sparse sampling. In *Proc. ICCV*, 2015. 2, 4, 6, 7, 8

[11] 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, 5, 6, 8

[12] J. Pan, Z. Hu, Z. Su, H.-Y. Lee, and M.-H. Yang. Soft-segmentation guided object motion deblurring. In *Proc. CVPR*, 2016. 2

[13] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *Proc. CVPR*, 2009. 7, 8

[14] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 3

[15] E. Shahrian, D. Rajan, B. Price, and S. Cohen. Improving image matting using comprehensive sampling sets. In *Proc. CVPR*, 2013. 2, 4, 5, 6, 7, 8

[16] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *Proc. CVPR*, 2015. 2

[17] J. Tan, J.-M. Lien, and Y. Gingold. Decomposing images into layers via RGB-space geometry. *ACM Trans. Graph.*, 36(1):7:1–7:14, 2016. 2

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

[19] Q. Zhu, L. Shao, X. Li, and L. Wang. Targeting accurate object extraction from an image: A comprehensive study of natural image matting. *IEEE Trans. Neural Netw. Learn. Syst*, 26(2):185–207, 2015. 2