A COMPREHENSIVE STUDY ON DIGITAL IMAGE MATTING

Haiyan Yang, Oscar C. Au, Jin Zeng, Mengqi Ji, Yuan Yuan, Sunil Prasad Jaiswal

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
{hyangac, eeu, jzengab, mj, yyuanad, spjaiswal}@ust.hk

ABSTRACT

Natural image matting refers to the problem of extracting the region of interest such as foreground object from an image based on user inputs like scribbles or trimap. Matting is an ill-posed problem inherently since we need to output three images out of only one input image. After a comprehensive survey and analysis of the existing matting literature, we observe that there are three key components in better estimating the alpha values, that is, the design of matting laplacian matrix, the definition of neighborhood and the choices of feature space. Based on this observation, we introduce a unified framework for digital image matting, which provides the possibility of obtaining a better understanding and direction of further improvement for image matting problem. The experimental results tested on different matting algorithms further prove the feasibility of our proposed framework.

Index Terms— digital image matting, local and nonlocal principle, feature space, matting laplacian, neighborhood construction

1. INTRODUCTION

In video/image editing and film production societies, extracting foreground objects from one texture image and then fusing it with a new scene (another texture image) is an attractive technique [1], which is called image matting and composition. As illustrated in Fig. 1, the upper row is the process of image matting where an “alpha image” (Fig. 1(c)) is outputted indicating the foreground object we are interested in, and the lower row is image composition where we blend the object (horses) with a new background.

Generally, we express the matting problem by formulating an image \( I_k \) as a convex combination of foreground image \( F_k \) and background image \( B_k \) as shown below:

\[
I_k = \alpha_k F_k + (1 - \alpha_k) B_k
\]

where \( I_k \) is the pixel value in each pixel location \( k \) for a given image. \( F_k \) and \( B_k \) stand for the foreground color and background color respectively, and \( \alpha_k \in [0, 1] \) is the so-called “alpha matte”. Equation (1) is also known as the composing equation, since we can obtain the composed image \( I_c \) by replacing original background \( B \) with a new \( B_c \).

Matting is a highly ill-posed and under-constrained problem inherently since three images, namely a foreground image \( F \), a background image \( B \) and the alpha matte, need to be obtained from only one image. To help combat such ill-posed problem, often an additional user-guided input called trimap is used in most matting literature. As shown in Fig. 1(b), pixels in the trimap are categorized into three classes, the foreground part, indicated by white pixels, the background part, indicated by black pixels, and uncertain, indicated by gray ones. Following the discussion in [2], matting methods can be roughly categorized into three classes: learning-based matting, sampling-based matting and propagation-based matting.


There has been much sampling-based work recently [6, 7, 8, 9, 10] after the introduction of the Bayesian matting framework [11] in 2001. Normally, these methods use different criteria to collect nearby known foreground and background pixels as samples to estimate alpha values for unknown pixels. However, these methods has a crucial drawback, i.e. the matting performance is highly related to the input images and trimaps. Inappropriate samples may be collected when the input image is complex and the trimap is coarse, leading to a poorly estimated result.

Propagation-based matting techniques also have a long tradition after the very insightful work proposed by Levin et al. in closed-form matting [12], where alpha matte is estimated by propagating smoothness constraint between neighboring pixels (which is modeled into an affinity matrix called matting laplacian) from known region to unknown region. In order to achieve a satisfactory result, various affinity matrices have been proposed [13, 14, 15, 16, 17] by improving or modifying closed-form matting from different aspects.

In this paper, we introduce a unified matting framework which belongs to the propagation-based matting category after a comprehensive survey on most of the existing propagation-based literature. Specifically, we summarized propagation-based matting methods us-

Fig. 1. Image matting and composition example. (a) is the original image and we want to extract two horses. (b) is the user marked trimap and (c) is the resulted alpha matte using our proposed method. (d) and (e) are the extracted object and a new scene (background), and (f) is the finally fused image.
ing a framework containing three components, and for each component, several different choices are briefly discussed. The remaining of this paper is organized as following. The details of unified framework will be discussed in section 2, followed by experimental results in section 3, and finally we conclude our paper in section 4.

2. A UNIFIED MATTING FRAMEWORK

We examine all the papers on propagation-based matting algorithms we can find and observe that a unified matting framework can be constructed to describe all such algorithms as shown in Fig. 2. In this section the unified matting framework will be described. We will also describe the choices of different framework components made by various matting algorithms.

2.1. Problem Formulation

All propagation-based matting algorithms start the problem by assuming some alpha-color model [12, 13, 15, 3] in which, for any pixel location $k$, a neighborhood or patch $N_k$ is defined around it such that, the alpha $\alpha_j$ at any location $j$ within the neighborhood $N_k$ is a linear function of the color components, $c_1, c_2$ and $c_3$ (which can be red, green, blue, or other color space), at $j$:

$$\alpha_j = a_k^1 I_k^1 + a_k^2 I_k^2 + a_k^3 I_k^3 + b, \quad j \in N_k,$$

where $I_k^1, I_k^2, I_k^3$ are the pixel value of the corresponding color components [13, 12, 18, 15, 3]. Note that the linear coefficients $a_k = \{a_k^1, a_k^2, a_k^3\}$, depend only on $k$ and are common for all $j \in N_k$.

With the alpha-color model, the matting problem is expressed as the following constrained optimization problem:

$$\begin{align*}
\text{minimize} & \quad \sum_{k \in I} \sum_{j \in N_k} (\alpha_j - \sum_{\epsilon \in \{c_1, c_2, c_3\}} a_k^\epsilon I_k^\epsilon - b)^2 \\
\text{subject to} & \quad \alpha_k = 1, \quad k \in \Omega_F, \\
& \quad \alpha_k = 0, \quad k \in \Omega_B.
\end{align*}$$

where $I$ is the whole image, $\alpha_k$ is the collection of all $\alpha$ in $I$, $\alpha_I$ is the collection of the linear coefficients $a_k$ of all local patches in $I$, $\Omega_F$ and $\Omega_B$ are the sets of pixels that are pre-labeled as definite foreground and background respectively by user inputs. In all existing algorithms, the linear coefficients in the $k^{th}$ patch are firstly solved by minimizing the objective function in (3). Since it is a $L_2$ norm minimization, a closed form expression for $a_k$ is obtained [12, 13].

Substituting $a_k$ back, the cost function is simplified to be a function of $\alpha_I$ only:

$$J(\alpha_I) = \alpha_I^T L \alpha_I$$

where $\alpha_I$ is arranged into an $N \times 1$ vector and $L$ is an $N \times N$ matrix commonly called the matting laplacian, and $N$ is the number of pixels in $I$. There are various forms of $L$ under different models and assumptions [12, 13, 15, 19, 16, 14] as shown below. The $i_{j}^{th}$ element $L_{ij}$ of $L$ has the following property:

$$L_{ij} = \begin{cases} 
W_{ii} : & \text{if } i = j, \\
-W_{ij} : & \text{otherwise},
\end{cases}$$

where $i$ and $j$ represent two different locations within $I$, $W$ is the “weight matrix” in [14, 16] or the “affinity matrix” in [12, 13, 15] such that $W_{ij}$ is the weight or affinity between pixel $i$ and pixel $j$. One property of $W$ is that $W_{ii} = \sum_j W_{ij}$, for $j \neq i$ such that the sum of each row of $L$ is zero. More details of $W$ will come in Section 2.2.

Then the constrained problem can be solved by employing its Lagrangian form:

$$J(\alpha) = \alpha^T L \alpha + \lambda(\alpha - b_S)^T D_S(\alpha - b_S)$$

where $D_S$ is an $N \times N$ diagonal matrix with diagonal elements being one for the user constrained pixels and zero otherwise, and $b_S$ is an $N \times 1$ vector containing user labeled alpha values. Finally the closed-form solution is given by the linear system below:

$$(L + \lambda D_S)\alpha = \lambda b_S$$

2.2. Matting Laplacian Matrix Design

It is pointed out in [16] that a key factor of the performance of a matting algorithm is the design of the affinity matrix $W$ ($W$ is related to $L$ in equation (5)). The $W$ has a nice property that each pair of “similar pixels” (or neighboring pixels in this paper) in $I$ would have high affinity values which has the effect that alpha values of labeled pixels are well propagated to its neighboring similar pixels that are unlabeled.

With any given neighborhood definition, there are several ways to design the Laplacian matrix $L$. For each pixel $k$ and its neighborhood $N_k$, one particular way to model the affinity matrix $W$ as in closed-form matting, color-clustering matting, learning-based matting, etc. [12, 13, 15] is

$$W_{ij} = \sum_{(k-i) \in N_k, j \in N_k} \frac{1}{|N_k|} \left[ 1 + (I_i - u_k)^T \Sigma_k^{-1} (I_j - u_k) \right]$$

where $I_i$ and $I_j$ are $3 \times 1$ vectors containing the three color components $c_1, c_2, c_3$ of pixel $i$ and pixel $j$ respectively, $\Sigma_k$ is the $3 \times 3$ color covariance matrix in $N_k$ and $u_k$ is the $3 \times 1$ mean color vector in $N_k$.

Another popular way is to define $W$ as a kernel function [16] as in nonlocal matting, in which the $(i, j)^{th}$ entry is the affinity value that pixel $i$ receives from its neighboring pixel $j$ with the effect that large penalties are assigned to dissimilar pixel pairs and vice versa.

$$W_{ij} = \exp\left(-\frac{1}{h^2} \left\| X(S(i)) - X(S(j)) \right\|_g^2 - \frac{1}{h^2} d^2(i, j) \right)$$

where $S(i)$ and $S(j)$ are patches around pixel $i$ and pixel $j$ respectively with $X(S(i))$ and $X(S(j))$ being the corresponding pixel values in the patches, $\| \cdot \|_g$ is the pixel distance and is basically the
Fig. 3. Image matting results tested on benchmark data sets [20]. (a) input image, (b) closed-form matting [12], (c) KNN matting [14], (d) groundtruth alpha, (e) learning based matting [3] and (f) color clustering matting [13]. It can be observed that (b) and (e) suffer from over-smoothing problem caused by collecting only local neighborhoods while (c) and (f) alleviate the problem by collecting nonlocal neighborhoods instead.

Frobenius norm $\| \cdot \|_F$ with each element weighted by a “point spread function” $g$, $d(i, j)$ is the spatial distance between pixel $i$ and pixel $j$. $h_1$ and $h_2$ are parameters to control the relative weights for the pixel distance and spatial distance.

There is a simple but effective kernel design in KNN matting [14] with the form

$$W_{ij}^3 = 1 - \frac{\|X(i) - X(j)\|}{C}$$  \hspace{1cm} (10)

where $X(i)$ and $X(j)$ are feature vectors of pixel $i$ and pixel $j$, and $C$ is a parameter. The 5 features used in KNN matting are: the 3 color components ($c_1, c_2, c_3$), and the two spatial coordinates ($x$ and $y$) such that the feature vector distance is effectively a combination of pixel distance and spatial distance.

Instead of using the kernel function only to assign affinity for each pixel, another design of matting laplacian is to use a combination of the kernel function and another smoothness term $H_{ij}$:

$$W_{ij}^4 = f(W_{ij}^3, H_{ij})$$  \hspace{1cm} (11)

where $H$ is a weight matrix obtained by solving the following minimization problem:

$$\sum_{i=1}^{N} \left\| X_i - \sum_{j=1}^{K} H_{ij} X_{ij} \right\|^2$$

2.3. Neighborhood Definition

In addition to the matting laplacian design, another key factor that would affect the performance of matting is the definition of neighborhood $N_k$. There are basically three classes of neighborhood definition in the propagation-based matting literature: local neighborhood, nonlocal neighborhood, and hybrid neighborhood.

2.3.1. Local Neighborhood

In general, local neighborhood is a collection of neighboring pixels that are “close” in terms of spatial distance, as in [12] and [3]. For example, in closed-form matting [12], the local neighborhood around pixel $k$ is defined simply as the pixels within a $3 \times 3$ window of $k$. By adopting local neighborhood, pixel pairs that are spatially nearby would always have high affinity values and vice versa [2].

While this is reasonable, local neighborhood can lead to severe over-smoothing problem in that small features (local features of $\alpha_1$ occupying very few pixels) tend to disappear as they are “cornered” by the surrounding large dominating areas. Regions with fast changing $\alpha$ can also have the fast-changing patterns smoothed. An example is shown in the highlighted region in Fig. 3(c) which tends to smoothed out or disappear after matting. Similarly, learning-based matting [3] also uses a $7 \times 7$ window as the local neighborhood, and has similar over-smoothing problem in Fig. 3(d).

The over-smoothing problem can be remedied, at least partially, in some cases by incorporating some smoothness term in the matting laplacian as in [18].

Note that different window size may play a key role in determining affinity values for pixels in different image regions. For example, an adaptive window is used in [21] to enhance matting performance somewhat.

2.3.2. Nonlocal Neighborhood

One way to alleviate the over-smoothness problem of local neighborhood. When local neighborhood is used, the local information is propagated without regard to distant regions which may have similar “behavior” in terms of similar $\alpha$ and similar local texture. With an implicit assumption that similar local texture (with similar local features) implies similar $\alpha$, nonlocal neighborhood is defined in general as a collection of pixels that are “close” in terms of some feature similarity measure or texture-based distance [16, 14, 13]. In general, there are two methods to define nonlocal neighborhood in propagation based matting literature: global and "semi-global".

Global nonlocal neighborhood is used in KNN matting [14], color-clustering matting [13] etc. in which the nonlocal neighborhood $N_k$ is defined as any pixel in the whole image with similar “appearance” in terms of some texture related features. Typically some global searching is needed to find such nonlocal neighbors. A commonly used search algorithm is the K-nearest neighbors algorithm (KNN) [17, 14, 13, 15] or its fast version, FLANN [22].

Semi-global nonlocal neighborhood is used in nonlocal matting [16], etc. [23] in which a relatively large window (e.g. $10 \times 10$) is specified and the nonlocal neighborhood $N_k$ is defined as any pixel within the large window that are similar in appearance. The similarity measure can be kernel functions such as $W_{ij}^2$ and $W_{ij}^3$ or other similarity measures such as $SIMI(i, j)$ [23] shown below:

$$SIMI(i, j) = C_i^T C_j / \| C_i \| / \| C_j \|$$  \hspace{1cm} (12)

where $C_i$ and $C_j$ are $3 \times 1$ vectors containing the three color components of pixel $i$ and pixel $j$.

The performances of nonlocal neighborhood based matting can be much better than local neighborhood-based methods [14] and [13] as shown in Fig. 3(e) and 3(f). However, as the kernel is often experimentally determined using training images, it is unclear whether the nonlocal neighborhood definition works equally well in all images. It may be desirable to have content-adaptive definition of the nonlocal neighborhood.

2.3.3. Hybrid Neighborhood

Very few existing propagation-based matting methods use hybrid neighborhood definition, but the one (LNSP matting, of which LNSP is “local and nonlocal smoothness prior”) that uses it performs the best in the alpha matting evaluation website [24].

As both local and nonlocal neighborhood have their own advantages and limitations for matting problem, LNSP matting [17] uses a graph model and solves the matting problem by combining both local and nonlocal neighborhoods. The matting laplacian in LNSP matting consists of two smoothness terms: (1) a local smoothness
2.4. Feature choices for nonlocal neighbor search

While local neighboring pixels are those that are within certain Euclidean distance, nonlocal neighboring pixels are identified by searching pixels that are “close” in terms of some local texture features. In some matting methods such as nonlocal matting [16], the features used to define nonlocal neighborhood are the three color components: \( F^3 = \{R, G, B\} \). In other methods such as [15, 17], the features used can be the three color components of a pixel plus its x-coordinate and y-coordinate: \( F^5 = \{R, G, B, x, y\} \). While in KNN matting [14], it is pointed out that different color space may also affect the final performance, and thus the features chosen are \( F^3 = \{\cos(h), \sin(h), s, v, x, y\} \), in which HSV color space is used instead of RGB. While these are good features, some method such as CCM use the features in \( F^2 \) plus additional features such as intensity gradients (\( dR, dG, dB \): gradients in red, green and blue components) and contrast ratio \( ctr \) which gives global structural information of an image: \( F^3 = \{R, G, B, x, y, dR, dG, dB, ctr\} \).

Other feature space is also possible and should probably be investigated, such as Scale Invariant Feature Transform (SIFT) which is one of the most popular feature extractors and descriptors in computer vision society. But the computation complexity can increase as more features are added.

3. EXPERIMENTAL RESULTS

We test various combination state-of-the-art matting methods on the benchmark datasets [20]. There are 27 training images with groundtruth in total, then we compute the peak signal-to-noise ratio (PSNR) of each method for all these images. Limited by the space here, we show the average PSNR result in Table. 1, as well as different choices these methods make based on our unified framework.

In Fig. 3, the second column are all local neighborhood based methods (closed-form matting and learning based matting) and the third column are all nonlocal neighborhood based methods (KNN matting and color-clustering matting). It shows that nonlocal neighborhood tends to capture more structural details and performs better than local neighborhood based methods. Fig. 4(e) and 4(f) are two different nonlocal neighborhood based methods, since they are different in terms of both matting laplacian matrix and feature choices, it is hard to tell which one is always better when looking at the results in Fig. 3(e) and 3(f). However, in Fig. 5, it can be observed that LNSP matting which use hybrid neighborhood definition performs much better than those only use nonlocal neighborhoods.

4. CONCLUSION AND FUTURE WORK

In this paper, we perform a comprehensive survey and analysis of propagation-based matting methods. Observing their common characteristics, we introduce a unified matting framework that describes all of them. The framework consists of three key components: matting laplacian matrix design, neighborhood definition, and feature choices for nonlocal neighbor search. And we describe how various matting methods fit into the framework. Experimental results show that the introduced framework allows us to analyze each component and make comparisons of various choices of individual components. The proposed framework can enable researchers to further design and test other possible alternatives for each component.

5. ACKNOWLEDGEMENT

This work is supported in part by the Hong Kong Government Research Grants Council (GRF project), and the Hong Kong Government Innovation and Technology Fund and the State Key Laboratory on Advanced Displays and Optoelectronics Technologies (Project No: ITC-PSKL12EG02).
6. REFERENCES


