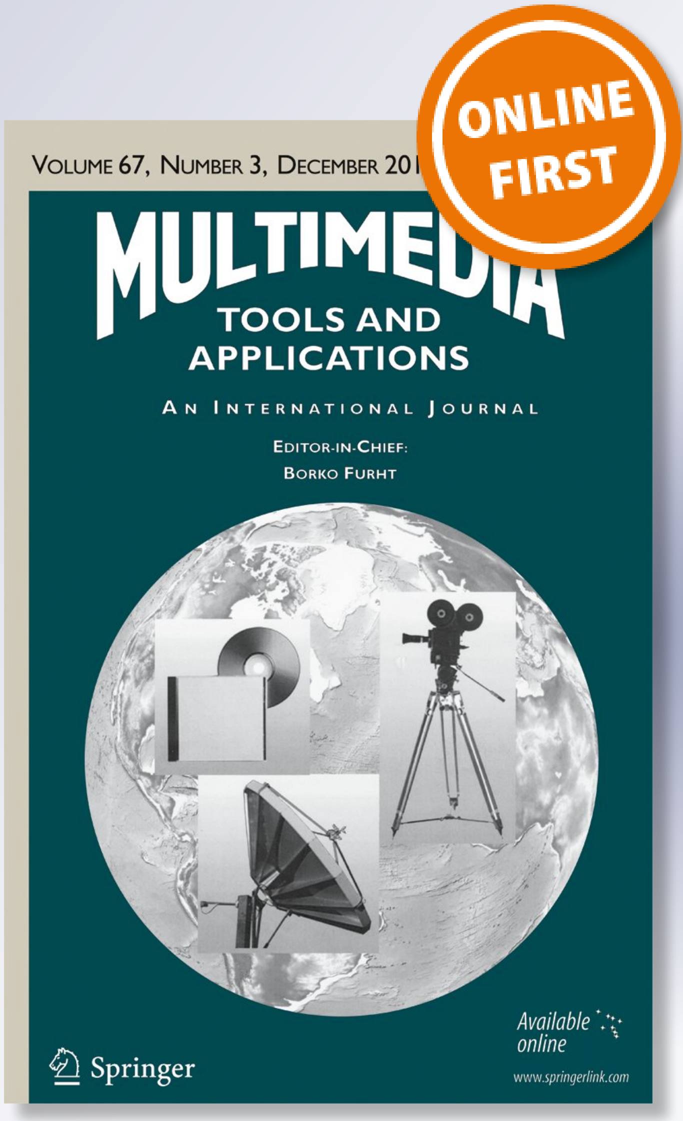
Adaptive propagation matting based on transparency of image

Xiangyu Zhu, Ping Wang & Zhenghai Huang

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-017-5357-7



Springer

Multimed Tools Appl
https://doi.org/10.1007/s11042-017-5357-7

Adaptive propagation matting based on transparency of image

Xiangyu Zhu[1] · Ping Wang[1] · Zhenghai Huang[1]

Received: 12 April 2017 / Revised: 10 September 2017 / Accepted: 25 October 2017

Abstract Image matting is an essential technique in many image and video editing applications. Although many matting methods have been proposed, it is still a challenge for most to obtain satisfactory matting results in the transparent foreground region of an image. To solve this problem, this paper proposes a novel matting algorithm, i.e. adaptive transparency-based propagation matting (ATPM) algorithm. ATPM algorithm considers image matting from a new slant. We pay attention to the transparencies of the input images and creatively assign them into three categories (highly transparent, strongly transparent and little transparent) according to the transparencies of the foreground objects in the images. Our matting model can make relevant adjustment in terms of the transparency types of the input images. Moreover, many current matting methods do not perform well when the foreground and background regions have similar color distributions. Our method adds texture as an additional feature to effectively discriminate the foreground and background regions. Experimental results on the benchmark dataset show that our method gets high-quality matting results for images of three transparency types, especially provides more accurate results for highly transparent images comparing with the state-of-the-art methods.

Keywords Matting · Transparency · Propagation · Texture

✉ Xiangyu Zhu
xyzhu@tju.edu.cn

Ping Wang
wang_ping@tju.edu.cn

Zhenghai Huang
huangzhenghai@tju.edu.cn

[1] School of Mathematics, Tianjin University, Tianjin 300354, China

1 Introduction

Matting aims to estimate the foreground and background layers of an image accurately. It is essential in many image and video editing applications. Mathematically, an observed image I is a convex combination of a foreground image F and a background image B as the following [25]:

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

where α represents the unknown alpha matte which defines the opacity of each pixel. And the value of α lies in [0, 1] with $\alpha = 1$ denoting a foreground pixel and $\alpha = 0$ indicating a background pixel. For a given input image, the foreground F, background B and α are all unknown. From the matting (1), we can easily see that all quantities on the right-hand side are unknown. Thus, for a three-channel color image, there are three equations and seven unknowns at each pixel, which makes matting a highly ill-posed problem. To simplify the problem and improve the results, it is necessary to consider a kind of prior knowledge about the foreground and background such as a trimap [39] or some user scribbles. In general, trimap is used in most matting methods and it roughly partitions an image into foreground, background, and unknown region.

Existing matting methods can be mainly categorized into color sampling-based approaches, propagation-based approaches and a combination of these two methods. Color sampling-based approaches collect enough color samples from known foreground and background regions for each unknown pixel firstly. Then the best sample pair is chosen from these samples to represent the foreground and background colors of the unknown pixel and estimate the alpha value of the unknown pixel. Early color sampling-based methods are parametric color-sampling methods [8, 30, 36]. These methods fit parametric models to color distributions of foreground and background regions. All recent color sampling-based methods are nonparametric methods [9, 10, 12, 13, 17, 26, 31, 34, 35, 37, 38, 43]. They collect the set of known F and B samples to estimate the alpha values of the unknown pixels. The main challenge is to prevent the true foreground and background samples from being missed when collecting the sample sets.

Propagation-based approaches [1, 4–6, 11, 14, 18–22, 33, 41, 42, 44, 45] do not need to estimate the foreground and background colors of the unknown pixels. They define the affinities representing the similarity between pixels and propagate alpha values of known regions toward unknown ones. Poisson matting [33] assumes that the foreground and background in the local window are smooth and the matte gradient is proportional to the image gradient. Random walk matting [11] describes the matting problem as a random walk problem. Closed-form matting [19] only needs a few black-and-white scribbles to show the input constraints instead of an accurate trimap. It assumes color line model [24] in local windows to obtain a matrix L called *matting Laplacian* and solves the alpha matte by minimizing a cost function. In general, the *matting Laplacian* is often added to other matting methods to strengthen the local smoothness of the matting results. The above three methods all belong to local propagation-based approaches. KNN matting [5] capitalizes on the nonlocal principle by using K nearest neighbors in matching nonlocal neighborhoods. This fast algorithm produces competitive results in transparent objects extraction. Manifold preserving editing propagation method [4] seeks to maintain the manifold structure formed by all pixels in a feature space and it is a novel nonlocal smooth prior on the alpha matte. Later, this nonlocal smooth prior is combined with the local Laplacian smooth term which comes from Closed-form matting to generate LNSP matting [6]. This combination generates some good results. Information-flow matting [1] relies on carefully defined pixel-to-pixel connections that enable effective use of information available in the image and the trimap. It achieves significant improvements on matte quality near challenging regions of the foreground object.

Combination of sampling and propagation matting usually puts the two ideas into an energy function. These combined methods [3, 15, 28] seek to make a good trade-off between the two approaches.

Recently, several deep learning works [7, 40] have been proposed for image matting. DCNN matting [7] takes the results of Closed-form matting [19], the results of KNN matting [5] and normalized RGB color images as inputs, and directly learns an end-to-end deep network to predict a new alpha matte. Given an input image and trimap, Deep matting [40] uses deep learning to directly compute the alpha matte. These methods achieve outstanding performance in image matting.

In this paper, we propose a new propagation framework for alpha matting. Unlike previous matting approaches, we consider the matting problem based on the transparency of the input image. A novel transparency detecting method is proposed and the input images are assigned into three categories according to the detecting results. Our method can adapt the matting model to fit nicely with different categories of images. Experimental results show that our framework outperforms previous propagation-based approaches and it shows considerable improvements when processing highly transparent images. In addition, the texture feature is used in our framework to improve matting by distinguishing between the foreground and background regions with similar color distributions. Experiments show that our method provides quantitatively better results after adding the texture information.

The main contributions of this paper are summarized as follows:

1) We see image matting from a new perspective. We propose an image transparency detecting strategy, which can be applied to any input image and does not need the ground truth matte of the input image. According to the detection results, the input images are divided into three categories: highly transparent, strongly transparent and little transparent.
2) We develop a novel propagation framework for alpha matting. According to the image's transparency type, our matting model can be adjusted adaptively for an excellent matting result. Our method provides both visually and quantitatively good results on a benchmark dataset [27]. Especially, it outperforms the state-of-the-art matting approaches in highly transparent image matting.

The paper is organized as follows: in Section 2, we present a brief introduction of two propagation-based methods related to our matting framework and compare their performances for images of different transparency types. In Section 3, we categorize the input images, extract the texture feature, and propose our matting algorithm. Experimental results are presented in Section 4. Finally the conclusion is presented in Section 5.

2 Related work

In the previous section, depending on the amount of transparencies in the respective ground truth matte, the original images are divided into three categories: highly transparent, strongly transparent and little transparent. The highly transparent images contain the most transparencies in its ground truth matte while the strongly transparent images take the second place and the little transparent images have the least transparencies. The specific classification strategy will be explained in the subsequent section. Our method involves two representative nonlocal propagation-based approaches, KNN matting [5] and manifold preserving editing propagation (MPEP) method [4]. In this section, we will analyze the different performances of these two approaches for images of different transparency types. Figure 1 shows the extracted mattes of images of three different transparency types. The insight originated from this section will motivate our novel propagation framework for alpha matting.

2.1 KNN matting

KNN matting [5] employs the nonlocal principle [2] to construct affinities for natural image matting. The nonlocal principle assumes that a denoised pixel i is a weighted sum of the pixels with similar characteristics and the weights are given by a kernel function $\mathcal{K}(i, j)$.

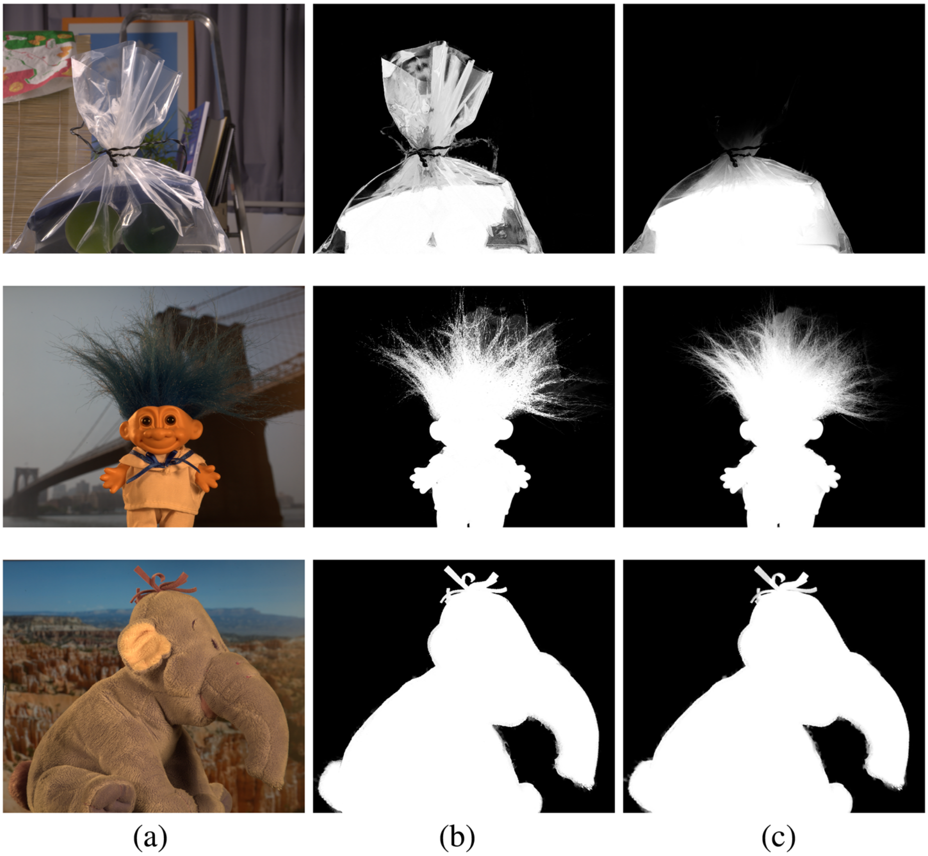


Fig. 1 Matting results of images of three different transparency types. The three examples from top to bottom represent the highly transparent image, the strongly transparent image and the little transparent image, respectively. **a** Input image. **b** KNN matting [5]. **c** Manifold preserving editing propagation [4]

Then, an $N \times N$ affinity matrix $\mathcal{A} = [\mathcal{K}(i, j)]$ can be obtained by using the kernel function $\mathcal{K}(i, j)$, where N is the total number of pixels in the input image. This matrix $\mathcal{A}$ provides the affinity of the image's α values. To compute $\mathcal{A}$, it is necessary to collect nonlocal neighborhoods j of pixel i in the feature space firstly. In KNN matting, a feature vector $X(i)$ at a given pixel i includes color and spatial information. For each pixel i, the K nearest neighbors (represented by j) in the feature space are collected through efficient KNN search. Then, the kernel function is computed as follows:

$$\mathcal{K}(i, j) = 1 - \frac{\|X(i) - X(j)\|}{C} \tag{2}$$

where C is set to make $\mathcal{K}(i, j) \in [0, 1]$. Further, the affinity matrix $\mathcal{A}$ can be computed. When trimap as the user input comes along, an optimization function can be derived and the optimal solution can be obtained.

In practice, KNN matting can achieve a high-quality matting result when processing the highly transparent image as shown in the first row of Fig. 1b. This thanks to the choice of kernel function $\mathcal{K}(i, j)$. In highly transparent images, most of the pixels in the trimap are unknown and a large proportion of the alpha values fall in between 0 and 1. Thus, highly transparent images pose a great challenge for matting. KNN matting computes the kernel function as (2) and obtains good results when extracting highly transparent objects. However, it cannot achieve satisfactory matting results in strongly transparent images. From the KNN matting result for the strongly transparent image, the second row of Fig. 1b, we can see that lots of the details are lost in the hair area. What's more, KNN matting has similar performance with manifold preserving editing propagation [4] in the little transparent image, as shown in the third row of Fig. 1b and c.

2.2 Manifold preserving editing propagation

Manifold preserving editing propagation (MPEP) [4] is a novel edit propagation algorithm that attempts to maintain the manifold structure constituted by all pixels in a feature space. It uses the locally linear embedding (LLE) [29] to represent each pixel as a linear combination of its nearest neighbors in a feature space. Firstly, for each pixel i with feature vector $\mathbf{X}_i$, MPEP method finds its K nearest neighbors in the feature space. Then, it computes a set of weights w_{ij} that best linearly reconstruct $\mathbf{X}_i$ from its neighbors and the specific computing method is introduced by Roweis and Saul [29]. All of the weights w_{ij} constitute a matrix W which captures the manifold structure of the data set in the feature space. At last, taking advantages of the user's input and the matrix W, an energy function about alpha value can be obtained and minimized by solving a sparse linear system.

In order to make the most of MPEP matting, we add *matting Laplacian* derived from Closed-form matting [19] into the energy function, which can strengthen the local smoothness of the matting results. In practice, MPEP matting can extract fine matting result in the strongly transparent image as shown in the second row of Fig. 1c. For the hair region in the second row of Fig. 1c, most of the details of the hair are retained for the reason that MPEP method preserves the overall manifold structure formed by all pixels in the feature space. MPEP matting, however, fails in extracting matting result of the highly transparent image, as shown in the first row of Fig. 1c. This is because there is insufficient feature for the method to differentiate the highly transparent object from the background region.

3 Adaptive transparency-based propagation matting algorithm

As discussed in Section 2, KNN matting [5] and MPEP matting [4] have different performances for images of different transparency types. For highly transparent images, KNN matting can extract high-quality matting results, but MPEP matting is weak for them. However, for strongly transparent images, KNN matting dose not perform well while MPEP matting can achieve high-quality matting results. For little transparent images, the matting results of these two methods are close to each other. Based on these observations, we try to construct a framework that incorporates the advantages of both KNN matting and MPEP matting effectively. To give full play to our framework, we can easily state that it is necessary to coordinate the proportion of KNN matting and MPEP matting in our framework according to the transparency types of the input images. For highly transparent images, a higher proportion of KNN matting is required. For strongly transparent images, a higher proportion of MPEP matting is effective. Finally, for little transparent images, a reasonable balance between these two methods must be found. In this section, we will describe how to construct our matting framework. It mainly consists of the following steps: 1) detecting the transparencies of the input images in order to divide them into three categories: highly transparent, strongly transparent and little transparent. 2) building the feature space. 3) constructing our matting framework by combining KNN matting and MPEP matting. 4) using a pre-processing step to expand known regions to unknown regions in the trimap. These steps combine to form our algorithm, i.e., adaptive transparency-based propagation matting (ATPM) algorithm. The framework of ATPM algorithm is shown in Fig. 2.

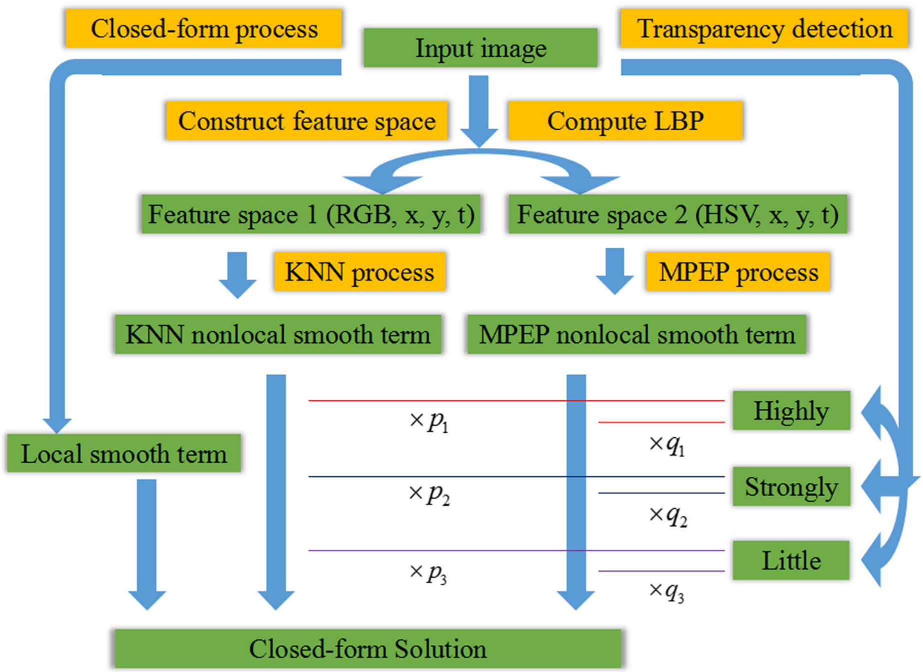


Fig. 2 Framework of ATPM algorithm. Here, the orange rectangles represent the intermediate processes, and the green rectangles represent the results of the corresponding processes. Here, p_i and q_i $(i = 1, 2, 3)$ are two parameters controlling the weights of these two nonlocal smooth terms

3.1 Detection of image transparency

The idea of categorizing images according to their transparencies is inspired by Rhemann et al. [27]. A ground truth matte is formed by the true alpha value of each pixel in an image. The alpha value lies in [0, 1] with $\alpha = 1$ denoting a foreground pixel and $\alpha = 0$ indicating a background pixel. If the alpha value is between 0 and 1, the pixel is called a mixed pixel. The amount of mixed pixels in an image determines its transparency. Thus, to categorize the input images, we should find the mixed pixels at first. In [27], the ground truth matte of each image in the benchmark dataset is known, so it is easy to find the mixed pixels and obtain the transparency of each image. But in reality, it is impossible to acquire the ground truth matte of an ordinary input image. So to find a substitute may be a solution. From Section 2, we know that KNN matting [5] performs well when processing highly transparent images. Hence, we try to use KNN matting to make a contribution. The process of detecting the transparency of the input image is shown in Fig. 3. Firstly, we use KNN matting to estimate the alpha matte of the input image. After obtaining the estimated alpha matte, we count up all the mixed pixels in it. It is important to note that some alpha values of the alpha matte estimated by KNN matting may be inaccurate. For example, the truth alpha values of several pixels are 0 essentially, but in the estimated alpha matte, the alpha values of these pixels may be very close to 0 but not equal to 0. Hence, in the estimated alpha matte, we define the pixel whose alpha value is between 0.1 and 0.9 as a mixed pixel. Then we find all the

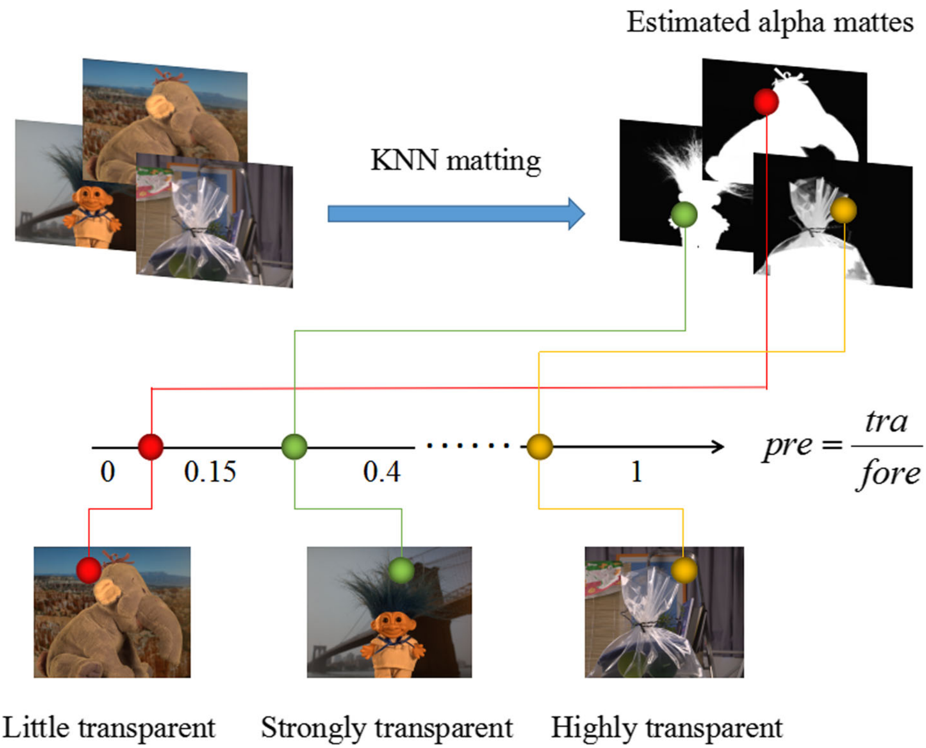


Fig. 3 Process of detecting the transparency of the input image. Here, tra is the amount of the mixed pixels in the estimated alpha matte. $fore$ is the total number of the foreground pixels and the mixed pixels in the estimated alpha matte. We use the ratio pre to classify the input images

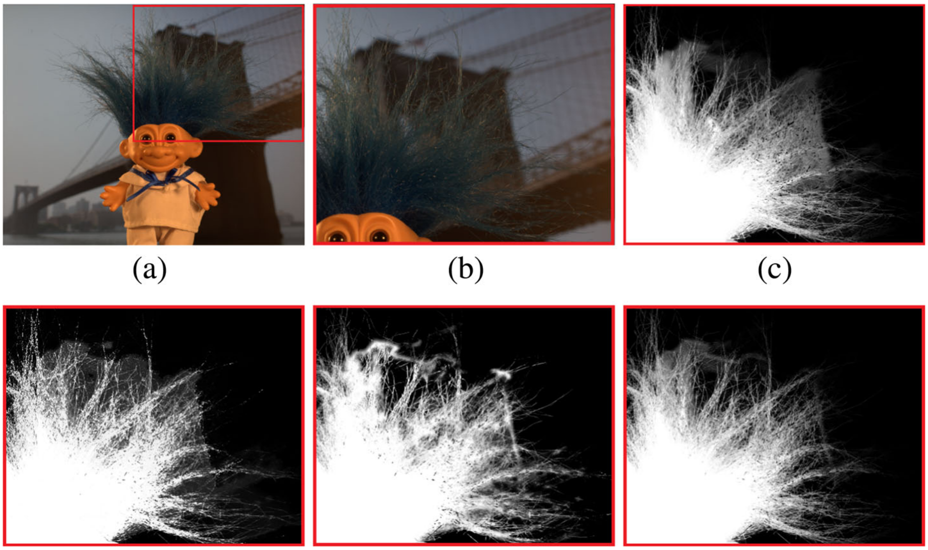
mixed pixels and define tra as the amount of the mixed pixels in the estimated alpha matte. We define $fore$ as the total number of the foreground pixels and the mixed pixels. Finally, a ratio pre is defined as follows:

$$pre = \frac{tra}{fore}. \tag{3}$$

Then, we can use pre to classify the input images. Through many experiments, we provide the classification strategy like this: for a given input image I, if $pre \in (0, 0.15]$, I belongs to little transparent images; if $pre \in (0.15, 0.40]$, I belongs to strongly transparent images; if $pre > 0.40$, I belongs to highly transparent images.

3.2 Constructing feature space

After determining the category of the input image, we will define a feature vector $X(i)$ at a given pixel i. In an original image, there are color and spatial information of the pixels. In general, many matting methods use these two features to construct the feature space. However, if the foreground and background regions of the input image have similar color distributions, these two features are insufficient for good matting results. The problem of overlapped color distributions is shown in Fig. 4. As shown in Fig. 4b, in the zoomed area, the color of the tower of the bridge is very similar to that of the hairs, so many matting methods fail in these regions. To solve this problem, we add texture as an additional feature for the matting task. So we use *Local Binary Patterns* (LBP) [23] to capture the texture feature of an image. LBP is an operator used to describe the local texture feature of an image. LBP works on a gray image. When getting an input image, we first convert it to a gray image. Then we compute the LBP value of each pixel in this gray image. All of the



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

Fig. 4 Illustration of overlap in color distributions of foreground and background regions. **a** Original image. **b** Zoomed-in region. **c** Closed-form matting [19]. **d** KNN matting [5]. **e** Shared matting [10]. **f** Proposed method computed LBP values form a matrix with the same size as the original image. This matrix contains the texture feature of the input image.

After acquiring the texture feature, we will combine it with the color and spatial information to construct the feature space. We construct two feature spaces based on HSV and RGB color spaces respectively. In this paper, we use RGB color space in KNN matting. Because when processing highly transparent images, RGB color space is better than HSV color space, an example is shown in Fig. 5. So, in KNN matting, a feature vector $X(i)$ at a given pixel i can be defined as

$$X(i) = (r, g, b, x, y, t)_i \tag{4}$$

where r, g, b are the respective RGB coordinates, (x, y) are the spatial coordinates of pixel i, t represents the LBP value of pixel i.

In this paper, we use HSV color space in MPEP matting, which is seldom used in previous matting methods. For some visually similar colors, their color values in RGB space are close while they can be distinguished in HSV space. Hence, in MPEP matting, a feature vector $X(i)$ at a given pixel i can be defined as

$$X(i) = (cos(h), sin(h), s, v, x, y, t)_i \tag{5}$$

where h, s, v represent the HSV coordinates, (x, y) are the spatial coordinates of pixel i, t is the LBP value of pixel i.

3.3 Proposed matting model

In Sections 3.1 and 3.2, we categorize the input images according to their transparencies and construct two different feature spaces. Based on these preparations, we will propose our matting model in this section. From KNN matting and MPEP matting, we can obtain two nonlocal smooth terms. In order to strengthen the local smoothness, we add *matting Laplacian* derived from Closed-form matting [19] as a local smooth term. By combining these two nonlocal smooth terms and the local smooth term in a matting model, we propose a new matting method which can adaptively adjust the weights of these two nonlocal smooth terms depending on the transparency type of the input image. Finally, the optimal result of the alpha matte can be obtained by solving a linear equation efficiently.

(a) (b) (c)

Fig. 5 KNN matting [5] results in the highly transparent image with different color spaces. **a** Input image. **b** Matting result with HSV color space. **c** Matting result with RGB color space

3.3.1 KNN nonlocal smooth term

As mentioned above, the key point of KNN matting is the nonlocal principle [2] which assumes that a denoised pixel i is a weighted sum of the pixels with similar appearance and the weights are given by a kernel function $\mathcal{K}(i, j)$. Firstly, for each pixel in the input image, we use KNN search to collect the K nearest neighbors in the feature space that is constructed for KNN matting in Section 3.2. We use j to represent the collected neighbors. According to the nonlocal principle, we can get:

$$E[X(i)] \approx \sum_j X(j)\mathcal{K}(i, j)\frac{1}{\mathcal{D}_i}, \tag{6}$$

$$\mathcal{D}_i = \sum_j \mathcal{K}(i, j) \tag{7}$$

where $X(i)$ is the feature vector of pixel i in the given feature space, $\mathcal{K}(i, j)$ is the kernel function computed as (2). By analogy of (6), we can obtain the expected value of the alpha matte

$$E[\alpha_i] \approx \sum_j \alpha_j \mathcal{K}(i, j)\frac{1}{\mathcal{D}_i}. \tag{8}$$

From (8), we can derive that

$$\mathcal{D}_i\alpha_i \approx \mathcal{K}(i, \cdot)^T \boldsymbol{\alpha} \tag{9}$$

where $\boldsymbol{\alpha}$ is a vector consisting of all α over the input image. Further, we can get the derivation:

$$\mathcal{D}\boldsymbol{\alpha} \approx \mathcal{A}\boldsymbol{\alpha} \tag{10}$$

where $\mathcal{A} = [\mathcal{K}(i, j)]$ is an $N \times N$ *affinity matrix* and $\mathcal{D} = diag(\mathcal{D}_i)$ is an $N \times N$ diagonal matrix. N is the total number of pixels in the image. So, $(\mathcal{D} - \mathcal{A})\boldsymbol{\alpha} \approx \mathbf{0}$, and we obtain the Laplacian $L_1 = \mathcal{D} - \mathcal{A}$.

3.3.2 MPEP nonlocal smooth term

As mentioned earlier, MPEP matting [4] uses the locally linear embedding (LLE) [29] to represent each pixel as a linear combination of its nearest neighbors in a feature space. It aims to maintain the manifold structure formed by all pixels in the given feature space. It propagates user edits by preserving the relationship in the result image. We use $\boldsymbol{X}_i$ to represent the feature vector of pixel i. All the pixels in the input image form a data set $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$. For each pixel $\boldsymbol{X}_i$, we find its K nearest neighbors in the feature space, namely $\boldsymbol{X}_{i1}, \ldots, \boldsymbol{X}_{ik}$. To compute the weights w_{ij} that make $\boldsymbol{X}_i$ be best reconstructed from these K nearest neighbors, we minimize

$$\sum_{i=1}^{N} \left\| \boldsymbol{X}_i - \sum_{j=1}^{K} w_{ij}\boldsymbol{X}_{ij} \right\|^2 \tag{11}$$

subject to the constraint $\sum_{j=1}^{K} w_{ij} = 1$. The specific computing method of w_{ij} is introduced by Roweis and Saul [29]. All the w_{ij} form a matrix W, which captures the manifold structure of the pixels in the feature space. Because it tries to maintain the manifold structure, in the result alpha matte, it requires $\alpha_i = \sum_{j=1}^{K} w_{ij}\alpha_{ij}$.

3.3.3 Local smooth term

Closed-form matting [19] is a representative local propagation-based approach based on the color line model [24]. It derives a *matting Laplacian matrix* to constrain the alpha matte in local windows. This *matting Laplacian* can enhance the local smoothness of the result alpha matte. Here, we define the *matting Laplacian* as L_2, which is an $N \times N$ matrix. The (i, j)th element of L_2 is

$$\sum_{k|(i,j)\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|}\left(1 + (I_i - \mu_k)\left(\Sigma_k + \frac{\epsilon}{|w_k|}I_3\right)^{-1}(I_j - \mu_k)\right)\right) \tag{12}$$

where δ_{ij} is the *Kronecker delta*, w_k is a 3×3 window, $|w_k|$ is the number of pixels in this window, μ_k is a 3×1 mean vector of the colors in the window w_k, Σ_k is a 3×3 covariance matrix, I_3 is a 3×3 identity matrix, and ϵ is a regularization coefficient.

Algorithm 1 ATPM algorithm

Input: Original image, trimap, λ, K, δ.
Output: $\boldsymbol{\alpha}$.

Pre-processing;
Compute *matting Laplacian matrix* L_2;
Compute texture feature matrix;
Compute Λ and G according to trimap;
Compute L_1 using KNN matting;
Compute pre according to (3);
if $pre > 0.4$ **then**
    $p = 1, q = 0.001$;
else if $0.15 < pre \leq 0.4$ **then**
    $p = 0.001, q = 1$;
else
    $p = 1, q = 1$;
end if
Compute W using MPEP matting;
$M = \Lambda + \delta L_2 + pL_1 + q(I - W)^T(I - W)$;
$\boldsymbol{\alpha} = M^{-1}(\Lambda G)$;
return $\boldsymbol{\alpha}$.

3.3.4 Closed-form solution

Firstly, we collect a subset of pixels $\mathcal{S}$, which represents the known foreground and background pixels from the trimap. To solve alpha value, we should minimize the energy function as follows:

$$\begin{aligned} E =& \lambda\sum_{i\in S}(\alpha_i - g_i)^2 + \delta\boldsymbol{\alpha}^T L_2\boldsymbol{\alpha} + p\boldsymbol{\alpha}^T L_1\boldsymbol{\alpha} \\ &+q\sum_{i=1}^{N}\left(\alpha_i - \sum_{j\in N_i} w_{ij}\alpha_j\right)^2 \end{aligned} \tag{13}$$

where λ is a constant which is set to 1000, g_i is set to 1 if i belongs to known foreground and 0 otherwise, the parameter δ controls the strength of the local smoothness and we set it to 1 here, N is the number of all pixels in the image, and $\boldsymbol{\alpha}$ is a vector formed by concatenating all α_i. The set N_i is the set of neighbors of pixel i. This energy function can be further written in a matrix form as

$$\begin{aligned} E = &(\boldsymbol{\alpha} - G)^T \Lambda (\boldsymbol{\alpha} - G) + \delta \boldsymbol{\alpha}^T L_2 \boldsymbol{\alpha} + p \boldsymbol{\alpha}^T L_1 \boldsymbol{\alpha} \\ &+ q \boldsymbol{\alpha}^T (I - W)^T (I - W) \boldsymbol{\alpha} \end{aligned} \tag{14}$$

where I is the identity matrix, Λ is a diagonal matrix, and G is a vector. Λ_{ii} is λ if $i \in \mathcal{S}$ and 0 otherwise. G_i is g_i if $i \in \mathcal{S}$ and 0 otherwise. p and q control the weights of the two nonlocal smooth terms. If the input image belongs to highly transparent images, we set $p = 1, q = 0.001$. If the input image belongs to strongly transparent images, we set $p = 0.001, q = 1$. If the input image belongs to little transparent images, we set $p = 1, q = 1$. The reason to set the values of p and q like this will be introduced in Section 4.1. Equation (14) is a quadratic function about $\boldsymbol{\alpha}$, which can be minimized by solving the linear equation in closed-form solution

$$(\Lambda + \delta L_2 + p L_1 + q (I - W)^T (I - W)) \boldsymbol{\alpha} = \Lambda G. \tag{15}$$

3.4 Pre-processing

In this section, to get more accurate matting results, we use a pre-processing method that comes from [32] to expand known regions to unknown regions in the trimap. According to this pre-processing method, an unknown pixel i is regarded as foreground if, for a known foreground pixel m,

$$(D(i, m) < E_{thr}) \wedge (\|I_i - I_m\| \leq (C_{thr} - D(i, m))) \tag{16}$$

where $D(i, m)$ is the Euclidean distance between pixels i and m in spatial domain, I_i is the color value of pixel i, E_{thr} and C_{thr} are the thresholds in spatial and color spaces. Similarly, we can compare the unknown pixels with a known background pixel by using the same method. After the pre-processing, we can obtain a more accurate trimap that has less unknown pixels.

All of the above processes are combined to form our ATPM algorithm.

4 Experimental results

In this section, we firstly discuss how to determine the combination weights p and q. Then we illustrate the effectiveness of LBP to discriminate between the regions that have similar color distributions. In Section 4.3, we give quantitative and visual comparisons of the proposed method with other matting methods over a benchmark dataset [27]. The benchmark dataset is composed of a test dataset and a training dataset. There are 27 images in the training dataset and their ground truth alpha mattes are available. The test dataset is formed by 8 images whose ground truth alpha mattes are hidden from the public. In Section 4.4, to further demonstrate the effectiveness of our method in extracting the alpha mattes of highly transparent images, we select some representative images that contain highly transparent objects to continue the experiment. Finally, failure cases are presented. All our experiments are executed on an Intel Xeon E5-2620 v3 running at 2.40 GHz with 32.0 GB memory and 64-bit Windows 7 operating system.

4.1 Combination weights *p* and *q*

To determine the weights p and q in (14), we do experiments on the training dataset. Experimental results are shown in Fig. 6. Firstly, all images in the training dataset are divided into three categories: highly transparent, strongly transparent and little transparent. Then we set 11 groups of values for weights p and q as shown in Fig. 6. For each combination

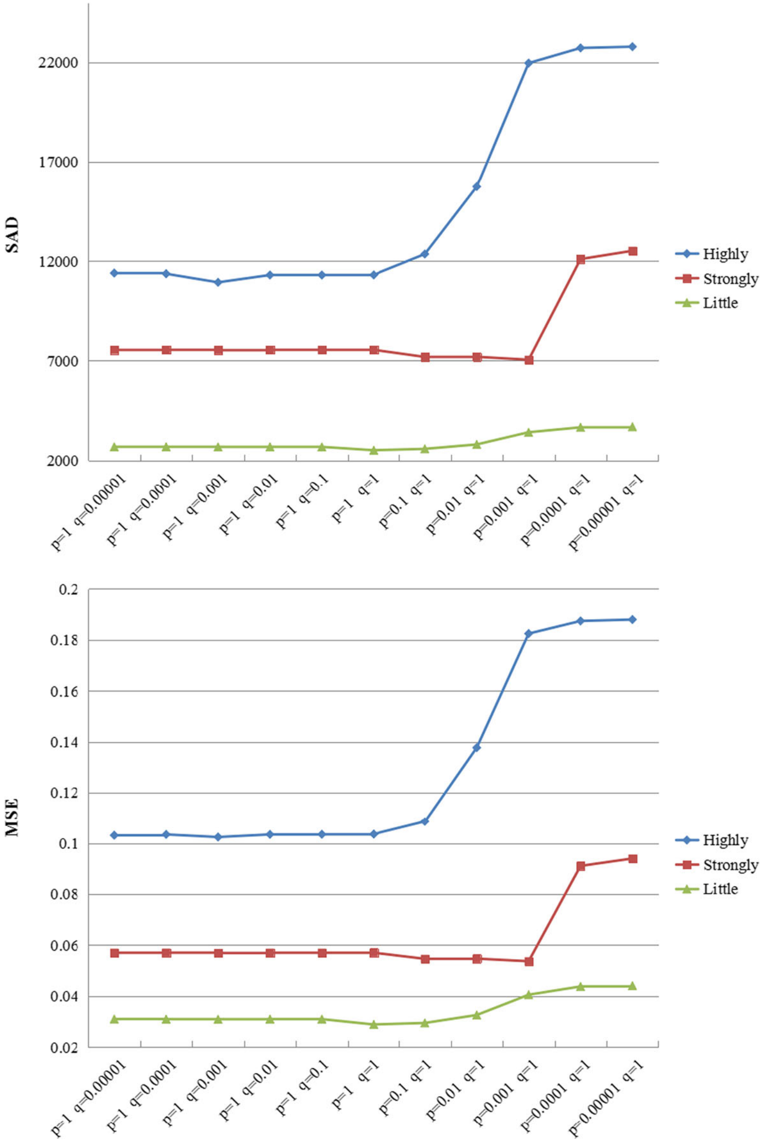
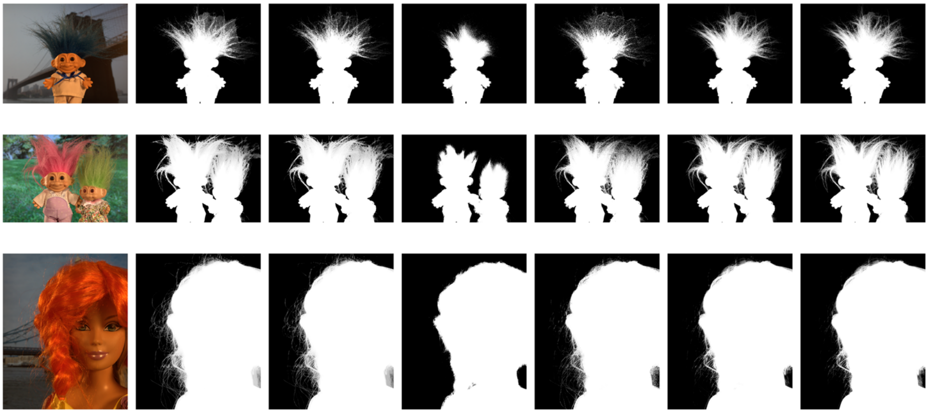


Fig. 6 Matting results of our method with different combinations of the weights p and q on the training dataset. All the images in the training dataset are divided into three categories (Highly, Strongly and Little) according to their transparencies. For each category, there are variations of average SAD (Sum of Absolute Differences) and MSE (Mean Squared Error)



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

Fig. 7 The effectiveness of LBP to discriminate between the regions that have similar color distributions. **a** Original image. **b** KNN matting [5] without LBP. **c** KNN matting with LBP. **d** MPEP matting [4] without LBP. **e** MPEP matting with LBP. **f** Proposed method without LBP. **g** Proposed method with LBP

of weights p and q, our method works on the whole training dataset. Finally, we compute the average SAD and MSE for each transparency category. In Fig. 6, the average SAD and MSE of the highly transparent images both get the lowest when $p = 1$ and $q = 0.001$. When $p = 0.001$ and $q = 1$, the average SAD and MSE of the strongly transparent images reach the minimum. For the little transparent images, the average SAD and MSE both get the minimum when $p = 1$ and $q = 1$. Therefore, we set the values of p and q as described in Section 3.3.4.

Table 1 Quantitative comparison to demonstrate the effectiveness of LBP

Method	GT04	GT03
Sum of absolute differences (10^3)		
KNN matting (without LBP)	15.540	9.272
KNN matting (with LBP)	15.483	9.056
MPEP matting (without LBP)	64.158	22.207
MPEP matting (with LBP)	21.526	9.502
Proposed method (without LBP)	7.117	7.921
Proposed method (with LBP)	7.016	6.553
Mean squared error (10^{-2})		
KNN matting (without LBP)	9.227	6.704
KNN matting (with LBP)	8.984	6.618
MPEP matting (without LBP)	31.828	15.921
MPEP matting (with LBP)	12.919	7.538
Proposed method (without LBP)	4.266	6.524
Proposed method(with LBP)	4.007	5.233

4.2 Effectiveness of LBP

In Section 3.2, we use LBP as the texture feature for accurate matte extraction. To demonstrate the effectiveness of LBP, in Fig. 7 we select 3 images (troll, GT04 and GT03) that contain significant overlaps in color distributions of foreground and background from the benchmark dataset. In our experiments, the parameters of LBP are set as follows: the radius size is 1 and the number of neighbors is 8. In our experiments, while the radius size we set is enough to produce satisfactory matting results, a larger radius cannot further improve the results. The results of KNN matting without and with LBP are shown in Fig. 7b and c respectively. By comparison, we find that KNN matting can remove the background from the foreground more effectively after LBP is added. Figure 7d and e are the comparisons of MPEP matting. Without LBP, the 3 images in Fig. 7d all lose lots of details of the hairs. In Fig. 7e, MPEP matting can preserve more details of the foreground with the help of LBP. Due to the use of *matting Laplacian* and pre-processing in our method, the performance improvement is not obvious enough in Fig. 7f and g. Thus, the quantitative comparison of 2 images (GT04 and GT03) from the benchmark training dataset is shown in Table 1. It is easy to see that SAD and MSE of our method both get lower after LBP is added. Besides that, SAD and MSE of KNN matting and MPEP matting are notably lower after LBP is added.

Table 2 Rank of matting methods with respect to SAD and MSE on the benchmark test dataset [27]

Method	Overall rank	Avg. small rank	Avg. large rank	Avg. user rank
Sum of absolute differences				
1. Information-flow matting	3.2	4.0	2.8	2.9
2. Deep matting	3.3	4.3	2.5	3.3
3. DCNN matting	4.6	6.3	2.8	4.8
4. Three-layer Graph matting	8.8	5.9	6.1	14.5
5. Proposed method	**11.7**	**14.9**	**13.1**	**7.1**
6. CSC matting	12.4	16.0	8.5	12.6
7. LNSP matting	13.1	9.4	12.8	17.1
8. GS matting	13.5	14.0	14.1	12.4
9. Patch-based matting	13.6	8.6	15.4	16.9
10. KL matting	14.1	13.3	13.3	15.8
Mean squared error				
1. Information-flow matting	4.8	6.5	3.6	4.1
2. DCNN matting	4.9	5.9	3.1	5.8
3. Deep matting	5.3	4.0	4.6	7.3
4. Three-layer Graph matting	9.3	6.6	6.8	14.6
5. LNSP matting	11.4	8.6	10.9	14.6
6. Patch-based matting	12.7	8.9	13.0	16.1
7. KL matting	13.9	13.3	12.8	15.8
8. CCM	14.0	17.3	14.5	10.4
9. Proposed method	**14.5**	**18.3**	**15.9**	**9.4**
10. GS matting	14.7	15.0	15.1	13.9

Small, large and user refer to the sizes of the trimaps

The results of our proposed method are highlighted in bold

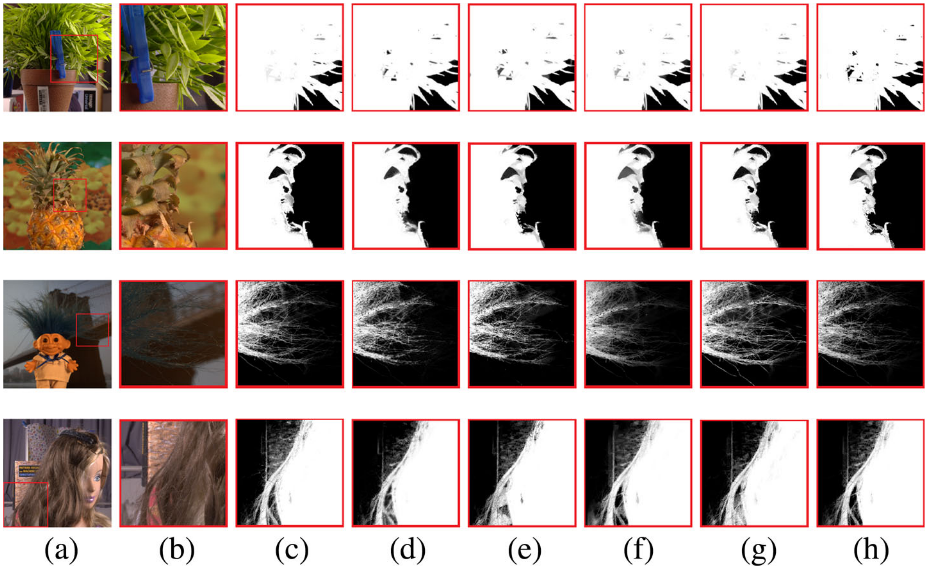


Fig. 8 Visual comparison of our method with other five matting methods in the top ten on the benchmark dataset [27]. **a** Original image. **b** Zoomed-in region. **c** Information-flow matting [1]. **d** DCNN matting [7]. **e** CSC matting [9]. **f** LNSP matting [6]. **g** GS matting [16]. **h** Proposed method

4.3 Evaluation on benchmark dataset

Table 2 shows the quantitative evaluation of our method when comparing with the top 9 matting approaches on the benchmark test dataset [27]. "Average small/large/user ranks" is the average ranks over all images in the test dataset for each of the three types of trimaps. The overall rank refers to the average rank over the whole test dataset and for all the trimaps. Our method performs well among the state-of-the-art methods. It ranks fifth for SAD and ranks ninth for MSE.

Visual comparisons of the proposed method with other five matting methods in the top ten on the benchmark test dataset [27] are shown in Figs. 8 and 9. These five matting methods are Information-flow matting [1], DCNN matting [7], CSC matting [9], LNSP matting [6] and GS matting [16]. Their matting results are from the benchmark dataset. We select five images from the benchmark test dataset to represent three different transparency types.

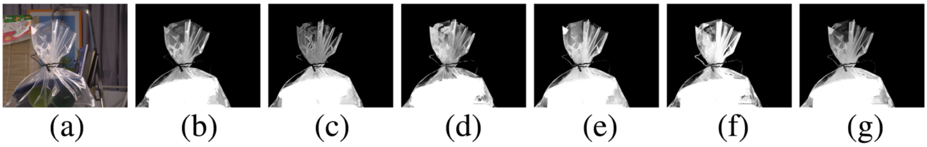


Fig. 9 Visual comparison of our method with other five matting methods on the image *plastic bag* from the benchmark dataset [27]. **a** Original image. **b** Information-flow matting [1]. **c** DCNN matting [7]. **d** CSC matting [9]. **e** LNSP matting [6]. **f** GS matting [16]. **g** Proposed method

The first and second rows of Fig. 8a represent the little transparent images, the third and fourth rows of Fig. 8a represent the strongly transparent images and Fig. 9a represents the highly transparent image. As shown in the first row of Fig. 8, in the zoomed region of the *plant*, it is hard for the other five methods to estimate the true background colors for the gaps between the leaves in the unknown region while our method performs well in this area. The similar situation appears in the second row, our method does better when estimating the true foreground color for *pineapple*'s leaves and the true background color for the holes in these leaves while the other methods cannot perform well on both points simultaneously. These two examples benefit from the balance of the two nonlocal smooth terms in our matting model when the input image belongs to little transparent images. In addition, the pre-processing contributes to the satisfactory results, too.

Both the third and fourth rows of Fig. 8 illustrate the problem about similar color distributions in foreground and background regions. In the third row of Fig. 8a, the color of the bridge in the background is very similar to that of the hairs of the *troll*. For the zoomed region of the *troll*, some parts of the bridge is considered as foreground as shown in the third row of Fig. 8g. The other four methods provide better results but miss some foreground details in the hairs as shown in the third row of Fig. 8c, d, e and f. Our method can discriminate between the foreground and background better and achieve a significantly better result. Similarly, in the fourth row of Fig. 8a, the color of the book in the background is similar to that of the hairs of the *doll*. Comparing to the other five methods, our method is able to remove the book from the foreground better and effectively restore the gaps between the hairs. Here, the LBP texture information helps to extract a more accurate alpha matte. It is more important that the MPEP nonlocal smooth term brings advantages when processing strongly transparent images.

In Fig. 9a, most region of the *plastic bag* is transparent, which is another great challenge for alpha matting. As shown in Fig. 9a, both foreground and background information is contained in the transparent region of the *plastic bag*, and there is a painting on the wall of the background. Due to the advantage of the KNN nonlocal smooth term, our method can avoid estimation bias and get the most accurate alpha matte. In Fig. 9b, c, d, e and f, these methods fail to differentiate the background pixels from the foreground. In contrast, our method

Table 3 Rank of the top 5 matting methods with respect to SAD and MSE on the image *plastic bag* from the benchmark dataset [27]

Method	Avg.	Small	Large	User
	Sum of absolute differences			
Proposed method	**16.8**	**17.2**	**17.6**	**15.7**
Information-flow matting	17.8	18.3	19.3	15.8
KNN matting	18.2	18.1	19.6	17.0
Deep matting	19.2	19.2	19.6	18.7
DCNN matting	19.4	19.9	19.2	19.1
	Mean squared error			
Proposed method	**1.0**	**1.1**	**1.1**	**0.8**
KNN matting	1.0	1.1	1.1	0.9
Information-flow matting	1.1	1.3	1.2	0.8
Deep matting	1.1	1.1	1.1	1.1
LNSP matting	1.1	1.4	1.2	0.8

Small, large, and user refer to different trimap sizes

The results of our proposed method are highlighted in bold obtains a considerably superior alpha matte. In Table 3, we select the top 5 matting methods on the *plastic bag* from the benchmark dataset, and our method ranks first with respect to both SAD and MSE. Here, we find that our method has better performance than KNN matting [5] when processing the highly transparent image. This demonstrates that our matting model not only leverages the advantage of the KNN nonlocal smooth prior, but also makes considerable improvements by combining it with the MPEP nonlocal smooth prior. The local smooth prior and the texture feature also make contributions to these improvements.

We also carry out experiments on the benchmark training dataset [27]. As shown in Fig. 10a, the four images from top to bottom are GT02, GT04, GT11, GT13 and they are chosen from the benchmark training dataset. Here, we compare our method with other five methods whose source codes are available. For GT02 and GT13, the difficulty of matting is to restore the tiny holes on the foreground objects. The results of Closed-form matting and LB matting lose lots of holes as shown in the first and fourth rows of Fig. 10b and d. Comparing to other algorithms, the proposed method retains the details of GT02 and GT13 better. The similarity of images GT04 and GT11 is that there are a lot of filiform objects in the foreground. In the second and third rows of Fig. 10, the results of our method are closest to the ground truth. Table 4 shows the quantitative comparison of our method with other five methods over the whole benchmark training dataset. For each method, there are average SAD and MSE over all images in the training dataset. In Table 4, our method gets the lowest SAD and MSE.

The proposed method is propagation-based which makes it have an advantage on running time when comparing to the sampling-based approaches. When the total number of pixels in the image is N, the complexity of our algorithm is $O(N)$. The complexities of KNN Matting and MPEP Matting are $O(1)$ and $O(N)$ respectively. So, our method is not more costly after combining the two approaches. Table 5 is the comparison of the running time of some matting algorithms. This experiment is carried out on 8 images from the benchmark test dataset. In Table 5, the first three methods are propagation-based approaches and the fourth, fifth and sixth methods are sampling-based approaches. The results presented in

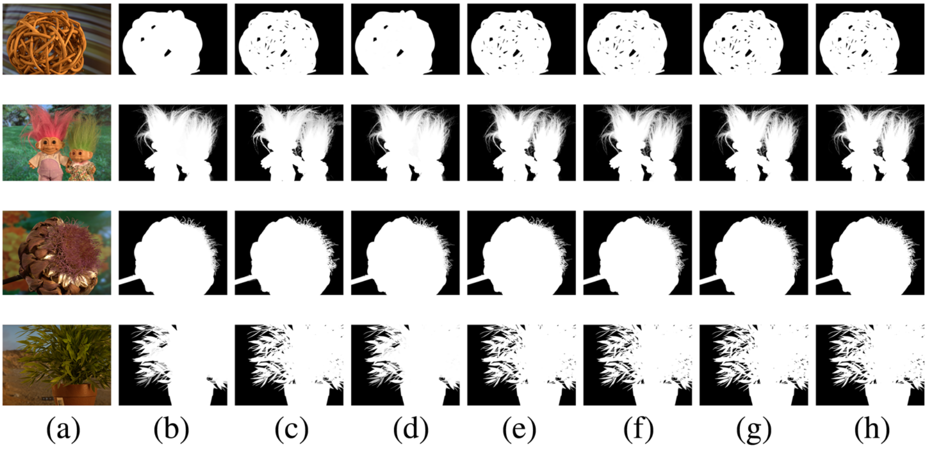


Fig. 10 Visual comparison of our method with other five matting methods on the benchmark training dataset [27]. **a** Original image. **b** Closed-form matting [19]. **c** KNN matting [5]. **d** LB matting [44]. **e** WCTM matting [31]. **f** KL matting [17]. **g** Proposed method. **h** Ground truth

Table 4 Quantitative comparison of our method with other five matting methods on the benchmark training dataset [27]

Method	SAD (10^3)	MSE (10^{-2})
Closed-form matting	5.325	5.348
KNN matting	6.358	5.572
LB matting	5.187	5.252
WCTM matting	4.920	4.514
KL matting	5.159	4.357
Proposed method	**4.127**	**4.021**

The results of our proposed method are highlighted in bold

Table 5 demonstrate that our algorithm has a significantly shorter running time than the three sampling-based approaches. However, our method does not have advantage on running time when comparing to the three propagation-based approaches. This may be caused by the computations of the weights w_{ij} in Section 3.3.2 and *matting Laplacian* in Section 3.3.3. The pre-processing in our method also takes some time.

4.4 Matte extraction in highly transparent images

From the experimental results in Section 4.3, we know that our method produces a remarkably superior result on the highly transparent image. In order to further illustrate the effectiveness of the proposed matting method in dealing with highly transparent images, we collect a set of images containing highly transparent objects. As shown in Fig. 11, we select four representative images from this set to compare our method with other five matting methods: Comprehensive Sampling [32], WCTM matting [31], KNN matting [5], LB matting [44] and Closed-form matting [19]. Because the images we select do not belong to the benchmark dataset, we can only compare our method with these five methods whose

Table 5 Comparison of the running time(s) of the matting algorithms

Method	Troll	Doll	Donkey	Elephant
Closed-form matting	21.3	11.0	7.2	7.4
KNN matting	10.1	8.0	7.4	9.4
LB matting	16.5	9.7	5.7	5.8
WCTM matting	348.8	216.1	167.5	159.3
KL matting	517.6	784.6	282.8	381.2
Comprehensive sampling	558.0	561.6	329.1	352.2
Proposed method	97.2	72.8	63.0	67.2
Method	Plant	Pineapple	Plastic bag	Net
Closed-form matting	13.9	8.7	16.4	28.5
KNN matting	7.4	8.2	10.9	10.7
LB matting	10.8	7.1	13.3	24.8
WCTM matting	256.2	196.2	336.6	668.3
KL matting	415.2	463.8	470.9	591.0
Comprehensive sampling	562.9	306.4	1106.1	1734.4
Proposed method	82.8	81.8	109.6	111.2

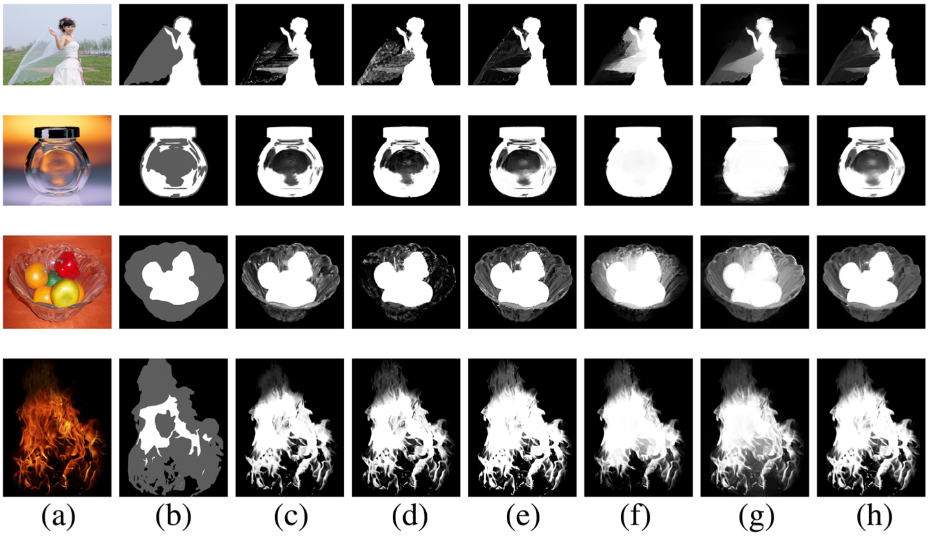


Fig. 11 Qualitative evaluations of matting methods on highly transparent images. **a** Original image. **b** Trimap. **c** Comprehensive Sampling [32]. **d** WCTM matting [31]. **e** KNN matting [5]. **f** LB matting [44]. **g** Closed-form matting [19]. **h** Proposed method

source codes are available. In the first row, the *bridal veil* produces a large transparent area and the background information is complicated. As shown in the first row of Fig. 11c, d, e, f and g, the estimated alpha values of the competitive methods for this transparent region are either too high or too low. However, our method shows less estimation bias in the first example. The examples in the second and third rows of Fig. 11 are both glasses, which are also common transparent objects in our lives. In the second row, for the first three methods, the estimated alpha values of the center area are all too low. LB matting and Closed-form matting both fail to differentiate the background pixels from the foreground. In contrast, our method extracts a more accurate result. In the third row, our method produces a better estimation of the alpha values in the glass bowl region than the other five methods. The last example is fire, which also contains some transparent pixels in the outer and inner. Our method performs better in some details of the fire while the other five methods cannot avoid estimation bias. The performance of our method in estimating high quality mattes for highly transparent images shows the potential advantage of the KNN nonlocal smooth term.

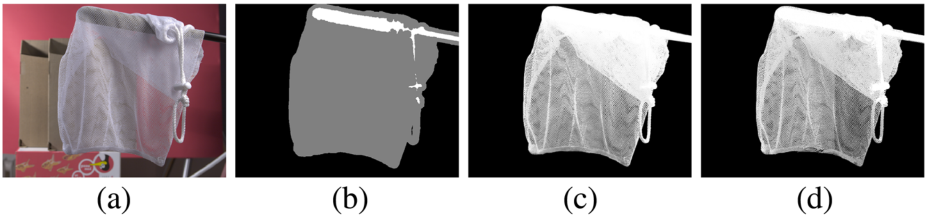


Fig. 12 Failure example of the proposed method. **a** Input image. **b** Trimap. **c** DCNN matting [7]. **d** Proposed method

4.5 Failure cases

There is an unsatisfactory example when dealing with an image from the benchmark test dataset [27], as shown in Fig. 12. The image *net* belongs to highly transparent images but it is different from the common highly transparent images which have smooth surfaces. In the image *net*, most region is composed of net structure which is very rough. Our method is derived from the traditional propagation-based methods. It exploits the relationships between pixels but the constrains constructed by it are not reliable on the image *net*. Another reason may be that the feature vector in our method cannot reflect the special construction of the image *net* well. These reasons may block the propagation of alpha and lead to the unsatisfactory result in the image *net*. For future work, the following directions will be considered to solve this problem. The first is to design some new features which are more adaptable. The second is to design a new framework which can find the reliable constrains in all circumstances.

5 Conclusion

In this paper, we proposed a novel propagation-based matting method, ATPM. What distinguishes this paper from previous matting methods is that our approach is adaptively based on the transparencies of the input images. Specifically, we firstly detect the transparencies of the input images. According to the detection results, we divide the input images into three categories: highly transparent, strongly transparent and little transparent. Our method adaptively coordinates the matting model based on the transparency type of the input image. Hence, the constraints in our framework can complement each other in different images to generate surprisingly good results. In addition, we add a local smooth term from Closed-form matting [19] to enhance the local smoothness of the result. We also use texture as a complementary information in the feature space to overcome the problem of overlapped color distributions. Our method can obtain fine matting results for images of three transparency types. Especially, our method produces remarkably superior alpha mattes for highly transparent images comparing with the state-of-the-art methods.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant No. 11431002).

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Informed Consent Informed consent was obtained from all individual participants included in the study.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Aksoy Y, Aydin TO, Pollefeys M (2017) Designing effective inter-pixel information flow for natural image matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 29–37

2. Buades A, Coll B, Morel JM (2008) Nonlocal image and movie denoising. Int J Comput Vis 76(2):123–139
3. Cao G, Li J, He Z, Chen X (2017) Divide and conquer: a self-adaptive approach for high-resolution image matting. In: International conference on virtual reality and visualization, pp 24–30
4. Chen X, Zou D, Zhao Q, Tan P (2012) Manifold preserving edit propagation. ACM Trans Graph (TOG) 31(6):1–7
5. Chen Q, Li D, Tang CK (2013) Knn matting. IEEE Trans Pattern Anal Mach Intell 35(9):2175–2188
6. Chen X, Zou D, Zhiying Zhou S, Zhao Q, Tan P (2013) Image matting with local and nonlocal smooth priors. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1902–1907
7. Cho D, Tai YW, Kweon I (2016) Natural image matting using deep convolutional neural networks. In: European conference on computer vision. Springer, pp 626–643
8. Chuang YY, Curless B, Salesin DH, Szeliski R (2001) A bayesian approach to digital matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), vol 2. IEEE, pp 264–271
9. Feng X, Liang X, Zhang Z (2016) A cluster sampling method for image matting via sparse coding. In: European conference on computer vision. Springer, pp 204–219
10. Gastal ES, Oliveira MM (2010) Shared sampling for real-time alpha matting. Computer Graphics Forum 29(2):575–584
11. Grady L, Schiwietz T, Aharon S, Westermann R (2005) Random walks for interactive alpha-matting. In: Proceedings of the IASTED international conference on visualization, imaging and image processing (VIIP), pp 423–429
12. He K, Rhemann C, Rother C, Tang X, Sun J (2011) A global sampling method for alpha matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2049–2056
13. He B, Wang G, Ruan Z, Yin X, Pei X, Lin X (2012) Local matting based on sample-pair propagation and iterative refinement. In: Proceedings of the IEEE international conference on image processing (ICIP). IEEE, pp 285–288
14. Hu WC, Chen WH, Huang DY, Yang CY (2016) Effective image forgery detection of tampered foreground or background image based on image watermarking and alpha mattes. Multimedia Tools and Applications 75(6):3495–3516
15. Jin M, Kim BK, Song WJ (2014) Adaptive propagation-based color-sampling for alpha matting. IEEE Transactions on Circuits and Systems for Video Technology 24(7):1101–1110
16. Johnson J, Varnousfaderani ES, Cholakkal H, Rajan D (2016) Sparse coding for alpha matting. IEEE Trans Image Process 25(7):3032–3043
17. Karacan L, Erdem A, Erdem E (2015) Image matting with kl-divergence based sparse sampling. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 424–432
18. Lee P, Wu Y (2011) Nonlocal matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2193–2200
19. Levin A, Lischinski D, Weiss Y (2008) A closed-form solution to natural image matting. IEEE Trans Pattern Anal Mach Intell 30(2):228–242
20. Levin A, Rav-Acha A, Lischinski D (2008) Spectral matting. IEEE Trans Pattern Anal Mach Intell 30(10):1699–1712
21. Li C, Wang P, Zhu X, Pi H (2017) Three-layer graph framework with the sumd feature for alpha matting. Comput Vis Image Underst. https://doi.org/10.1016/j.cviu.2017.06.011
22. Lin YC, Wang HA, Hsieh YF (2012) Image matting through a web browser. Multimedia Tools and Applications 61(3):551–570
23. Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59
24. Omer I, Werman M (2004) Color lines: image specific color representation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), vol 2. IEEE, pp 946–953
25. Porter T, Duff T (1984) Compositing digital images. In: Proceedings of the 11th annual conference on computer graphics and interactive techniques, pp 253–259
26. Rhemann C, Rother C, Gelautz M (2008) Improving color modeling for alpha matting. In: British machine vision conference 2008, Leeds, September, pp 1155–1164
27. Rhemann C, Rother C, Wang J, Gelautz M, Kohli P, Rott P (2009) A perceptually motivated online benchmark for image matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1826–1833

28. Rother C, Kolmogorov V, Blake A (2004) Grabcut: interactive foreground extraction using iterated graph cuts. ACM Trans Graph (TOG) 23(3):309–314
29. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
30. Ruzon MA, Tomasi C (2000) Alpha estimation in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), vol 1. IEEE, pp 18–25
31. Shahrian E, Rajan D (2012) Weighted color and texture sample selection for image matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 718–725
32. Shahrian E, Rajan D, Price B, Cohen S (2013) Improving image matting using comprehensive sampling sets. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 636–643
33. Sun J, Jia J, Tang CK, Shum HY (2004) Poisson matting. ACM Trans Graph (TOG) 23(3):315–321
34. Tan G, Chen H, Qi J (2016) A novel image matting method using sparse manual clicks. Multimedia Tools and Applications 75(17):10,213–10,225
35. Varnousfaderani ES, Rajan D (2013) Weighted color and texture sample selection for image matting. IEEE Trans Image Process 22(11):4260–4270
36. Wang J, Cohen MF (2005) An iterative optimization approach for unified image segmentation and matting. In: Proceedings of the IEEE international conference on computer vision (ICCV), vol 2. IEEE, pp 936–943
37. Wang J, Cohen MF (2007) Optimized color sampling for robust matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1–8
38. Wang J, Agrawala M, Cohen MF (2007) Soft scissors: an interactive tool for realtime high quality matting. ACM Trans Graph (TOG) 26(3):1–9
39. Wright S (2006) Digital compositing for film and video. Elsevier/Focal Press
40. Xu N, Price B, Cohen S, Huang T (2017) Deep image matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2970–2979
41. Xue B, Sapiro G (2007) A geodesic framework for fast interactive image and video segmentation and matting. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1–8
42. Yang CK, Kuo CN (2016) Automatic hair extraction from 2d images. Multimedia Tools and Applications 75(8):4441–4465
43. Zhang Z, Zhu Q, Xie Y (2012) Learning based alpha matting using support vector regression. In: Proceedings of the IEEE international conference on image processing (ICIP). IEEE, pp 2109–2112
44. Zheng Y, Kambhamettu C (2009) Learning based digital matting. In: Proceedings of the IEEE international conference on computer vision (ICCV). IEEE, pp 889–896
45. Zheng Y, Kambhamettu C, Yu J, Bauer T, Steiner K (2008) Fuzzymatte: a computationally efficient scheme for interactive matting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1–8

Xiangyu Zhu was born in 1994. He received his B.S. degree in applied mathematics from Tianjin University in 2016. He is currently a master candidate for computational mathematics in Tianjin University. His main research interests include image processing and machine learning.

Ping Wang was born in 1967. She received the B.S., M.S., and Ph.D. degrees from Tianjin University, Tianjin, China, in 1988, 1991, and 1998, respectively. She is a Professor, Master's Supervisor, and Ph.D. Supervisor with the School of Mathematics, Tianjin University. Her research interests include signal and information processing, pattern recognition, and image processing.

Zhenghai Huang received the B.S. degree from Central China Normal University, China, in 1988 and the M.S. degree from Huazhong University of Science and Technology, China, in 1996. In 1999, he received the Ph.D. degree from Fudan University, China. From 2002 to 2004, he was a Research Fellow with the Department of Mathematics, National University of Singapore. In 2004, he joined Tianjin University as a Professor of Mathematics. He has published more than 80 research papers in the areas of optimization theory and methods. Recently, his research interests include compressed sensing, low-rank matrix minimization, low-rank tensor minimization and pattern recognition etc.