

# A Survey on Pre-Processing in Image Matting

Gui-Lin Yao

*School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China*

E-mail: glyao@hrbcu.edu.cn

Received April 18, 2016; revised October 18, 2016.

**Abstract** Pre-processing is an important step in digital image matting, which aims to classify more accurate foreground and background pixels from the unknown region of the input three-region mask (Trimap). This step has no relation with the well-known matting equation and only compares color differences between the current unknown pixel and those known pixels. These newly classified pure pixels are then fed to the matting process as samples to improve the quality of the final matte. However, in the research field of image matting, the importance of pre-processing step is still blurry. Moreover, there are no corresponding review articles for this step, and the quantitative comparison of Trimap and alpha mattes after this step still remains unsolved. In this paper, the necessity and the importance of pre-processing step in image matting are firstly discussed in details. Next, current pre-processing methods are introduced by using the following two categories: static thresholding methods and dynamic thresholding methods. Analyses and experimental results show that static thresholding methods, especially the most popular iterative method, can make accurate pixel classifications in those general Trimaps with relatively fewer unknown pixels. However, in a much larger Trimap, these methods are limited by the conservative color and spatial thresholds. In contrast, dynamic thresholding methods can make much aggressive classifications on much difficult cases, but still strongly suffer from noises and false classifications. In addition, the sharp boundary detector is further discussed as a prior of pure pixels. Finally, summaries and a more effective approach are presented for pre-processing compared with the existing methods.

**Keywords** image matting, pixel classification, pre-processing, Trimap expansion

## 1 Introduction

### 1.1 Image Matting

Image matting is a key technique in digital image processing and editing<sup>[1-2]</sup>, which is commonly applied in image processing software, virtual studio, post production in movies, and so on. Given an input image, image matting mainly focuses on separating the foreground object from the background scene. The major difference between image matting and image segmentation is the introduction of  $\alpha$  channel<sup>[3]</sup>. In general, for each 2-dimension coordinate  $z = (x, y)$  in an image,  $\alpha_z \in [0, 1]$  is used to indicate the foreground transparency of the pixel located at position  $z$ , where  $\alpha_z = 1$  indicates a pure foreground (F) pixel and  $\alpha_z = 0$  indicates a pure background (B) pixel for  $z$ , while the case

$0 < \alpha_z < 1$  shows the pixel is a mixed one. The task of image matting is to obtain the exact  $\alpha$  from the input image. Specifically,  $\alpha_z$  is solved by the input color  $I_z$ , foreground color  $F_z$ , and background color  $B_z$  based on the following matting equation:

$$I_z = \alpha_z F_z + (1 - \alpha_z) B_z. \quad (1)$$

Commonly, a majority of pixels in most natural images belong to pure foreground or background, while only a small part of them are mixed ones, which would mostly happen at the “sharp boundary” or “soft boundary” of the foreground object such as hair, fur, transparent glass and plastic. Currently, most of the matting methods are assisted by a user interactive mask with three regions: known foreground region  $\Omega_F$ , known background region  $\Omega_B$ , and unknown region  $\Omega_U$ . Such

---

Survey

This work was supported by the Doctoral Scientific Research Start Fund Project of Harbin University of Commerce of China under Grant No. 15KJ06, the Youth Innovation Talent Support Program of Harbin University of Commerce under Grant No. 2016QN054, and the National Basic Research 973 Program of China under Grant No. 2015CB351804.

©2017 Springer Science + Business Media, LLC & Science Press, China

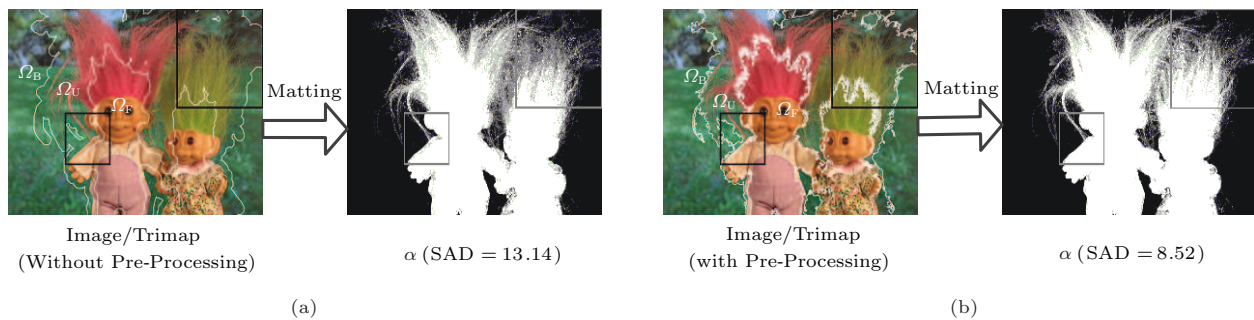


Fig.1. Example showing Trimap-based matting and the effect of pre-processing. SAD: sum of absolute difference.

an assistant mask is called Trimap shown in Fig.1. Here, the known regions  $\Omega_F$  and  $\Omega_B$  should contain most of the pure pixels with  $\alpha = 1$  and  $\alpha = 0$ , while the unknown region  $\Omega_U$  must contain all the mixed pixels with  $0 < \alpha < 1$  as well as the rest of pure pixels. Ideally,  $\Omega_U$  should only contain mixed pixels. However, this is a challenge and even an impossible task for human interactions.

The following two aspects are considered for the design of a Trimap. 1) A set of undisputed pure foreground or background pixels are pre-classified into  $\Omega_F$  and  $\Omega_B$  to reduce the unsolved set. 2) The pure regions of  $\Omega_F$  and  $\Omega_B$  are expected to be the guides to greatly reduce the solution space for these unsolved pixels in the unknown region  $\Omega_U$ . Generally, the true foreground color  $F$  and the background color  $B$  of each unknown pixel in  $\Omega_U$  can be simulated by those pixels in  $\Omega_F$  and  $\Omega_B$  respectively, and then the final  $\alpha$  can be solved.

Currently, the online evaluation system<sup>[4-5]</sup> provides a Trimap-based benchmark containing 27 training images with public ground truth  $\alpha^{\text{true}}$  and eight private test images without  $\alpha^{\text{true}}$ . For each input image, two basic types of Trimap with a large and a small size are provided respectively. The small-size Trimap indicates much fewer unknown pixels in this Trimap. According to the benchmark, the results of  $\alpha$  from a small Trimap are always superior to those from a large Trimap. Apparently, the size of  $\Omega_U$  is very important for Trimap-based matting.

Up to now, two main categories have appeared in the state-of-the-art Trimap-based matting algorithms: 1) sampling-based matting<sup>[6-23]</sup>, which assumes the true color of each unknown pixel could be approximated by samples from  $\Omega_F$  and  $\Omega_B$ , and solves the problem in a pixel-wise manner, and 2) affinity-based matting<sup>[24-29]</sup>, which solves the problem in a closed-form manner where  $\Omega_F$  and  $\Omega_B$  act as boundary conditions. Clearly, sampling-based matting is easy to debug for each pixel

after the matting process, while affinity-based matting could only change basic neighboring model and has to rerun the whole process after discovering mistakes. Besides, post-processing step is always realized by affinity-based matting to smooth the matte after sampling.

## 1.2 Pre-Processing

As discussed, an ideal  $\Omega_U$  in a Trimap should only include mixed pixels. However, a great number of pure pixels could still remain in  $\Omega_U$  from a user-drawn Trimap, especially from the “large-size” Trimap in the above benchmark. In practice, a requirement for a fine Trimap like the “small size” in the benchmark is tedious and usually unnecessary, especially for those input images with massive semi-transparent pixels or holes. Therefore, some of the recent sampling-based matting methods begin to apply a step called pre-processing before pure matting. This step, also called “reducing  $\Omega_U$ ”, actually employs some novel methods irrelevant to the well-known matting equation to pre-classify some of the potential pure pixels in  $\Omega_U$  into pure regions  $\Omega_F$  and  $\Omega_B$ . This could also be regarded as an extension for those pure pixels in  $\Omega_U$  which cannot be easily distinguished by a user-drawn Trimap. In general, by applying the same matting algorithm, the results of  $\alpha$  could be heavily raised with this pre-processing step compared with those without it. Fig.1 shows such an example. Besides, pre-processing could also raise the speed of pure matting, which will be discussed in Section 3.

The remainder of this paper is organized as follows. Section 2 presents a detailed description on the significance of pre-processing. Section 3 presents the experimental environment and simulates the improvements on matting with the help of pre-processing in both matte quality and processing speed. Section 4 proposes the classification of pre-processing methods and presents

the detailed analyses and experimental comparisons on this classification. Section 5 presents additional sharp boundary pure pixel priors. Section 6 makes summaries and presents a more effective approach. Section 7 draws conclusions.

## 2 Significance of Pre-Processing

### 2.1 Basic Functions

Obviously, one function of pre-processing is to classify some pixels in  $\Omega_U$  into pure ones, which may not be precisely solved based on a usual matting step. The arrows in Fig.2(a) show such pure foreground pixels. They are precisely classified into  $\Omega_F$  after pre-processing. In fact, the judgment and classification of pure pixels must rely on some accurate thresholds and one-side (unilateral) sample comparisons. For example, if a current pixel  $z$  is classified into foreground, the pre-processing step only has to compare  $z$  with foreground samples, i.e., no background samples are involved. This is superior to the sophisticated two-side (bilateral) computation of matting equation and will be discussed in details in Subsection 2.3.

Another function of pre-processing is to expect the classified pure pixels to provide sufficient and valid samples for those remaining unknown ones, especially for those mixed pixels with  $0 < \alpha < 1$ . On one hand, these pure pixels should be spatially close to the remainder part of  $\Omega_U$  to provide more accurate samples for the pure matting process. On the other hand, some of the newly classified pixels, which are actually more similar to the true F/B colors of these remainder pixels in  $\Omega_U$ , should be slightly different from these unknown pixels' regular samples in the initial  $\Omega_F$  or  $\Omega_B$ . The arrows in Fig.2(b) show some of the mixed pixels, whose  $\alpha$  values

can be more precisely computed based on these newly classified pure pixels as their samples.

Currently, pre-processing is always applied in sampling-based matting because of its pixel-wise manner. Hence, such a transition (from pre-processing to pure matting) looks natural. However, pre-processing could also be applied to affinity-based matting because both of them are independent and are consecutively applied.

In the reminder of this paper, a "Pre-" is marked in front of a pre-processing name.

### 2.2 Impact on Matting Benchmark

As discussed above, pre-processing has only appeared in sampling-based matting algorithms on the matting benchmark<sup>[4-5]</sup>. Hence, these algorithms are only employed here for comparisons. Table 1 shows the ranks of 20 sampling-based matting algorithms existing in the benchmark based on SAD (sum of absolute difference) and MSE (mean square error), where the nine algorithms in bold use pre-processing. Obviously, almost all the methods ranking on top of the benchmark employ pre-processing. Without it their ranks will intensively drop. A simulated rank comparison for some certain algorithms between with and without pre-processing will be performed in Section 3. Note that in Table 1, where the algorithms in bold use pre-processing, Shared (Real Time)<sup>[11]</sup> just eliminates the post-processing step of Matting Laplacian in Shared. SPS<sup>[18]</sup> is limited by the matting method itself. High-res<sup>[19]</sup> is just designed for high resolution images. Besides, the pre-processing in High-res is not a perfect one, which will be discussed later. Thus, the above three matting algorithms out of total nine applying pre-processing ones just show inferior results.

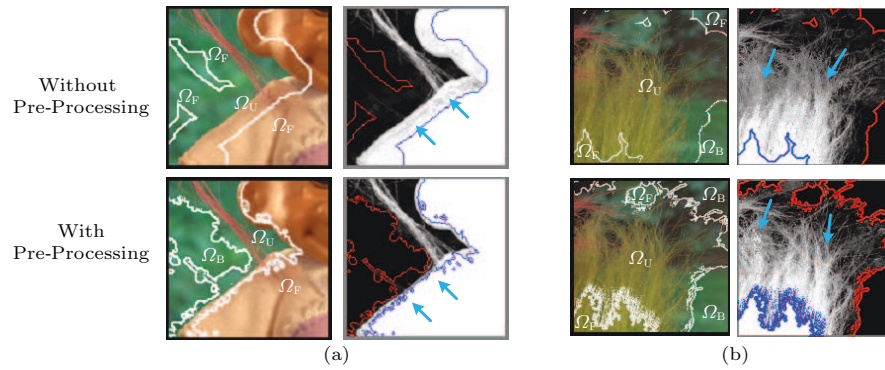


Fig.2. Functions of pre-processing in two local cases from Fig.1. (a) Local case where pre-processing could precisely compute some of the pure pixels that common matting step could not compute. (b) Local case where pre-processing could also bring in precise samples for the subsequent matting step.

**Table 1.** Ranks of 20 Sampling-Based Matting Algorithms on the Benchmark<sup>[4-5]</sup> Based on SAD and MSE

SAD		MSE ( $\times 10^{-2}$ )	
Algorithm	Overall Rank	Algorithm	Overall Rank
<b>KL-D-Sparse</b> <sup>[6]</sup>	<b>11.2</b>	<b>KL-D-Sparse</b>	<b>11.5</b>
<b>Comprehensive</b> <sup>[7]</sup>	<b>12.8</b>	<b>Comprehensive</b>	<b>12.8</b>
<b>CWCT</b> <sup>[8]</sup>	<b>13.7</b>	<b>CWCT</b>	<b>14.0</b>
<b>Sparse Coded</b> <sup>[9]</sup>	<b>14.1</b>	<b>WCT</b>	<b>16.6</b>
<b>WCT</b> <sup>[10]</sup>	<b>15.9</b>	<b>Sparse Coded</b>	<b>16.6</b>
<b>Shared</b> <sup>[11]</sup>	<b>16.7</b>	Global	17.3
Global <sup>[12]</sup>	18.6	<b>Shared</b>	<b>18.6</b>
Segmentation <sup>[13]</sup>	19.7	Segmentation	21.0
SRLO <sup>[14]</sup>	20.0	Improved Color	21.0
Improved Color <sup>[15]</sup>	20.6	Improving SC	22.5
Global (filter) <sup>[12]</sup>	21.9	SRLO	22.9
<b>Shared (Real Time)</b> <sup>[11]</sup>	<b>24.0</b>	<b>SPS</b>	<b>25.0</b>
Improving SC <sup>[16]</sup>	29.3	Global (filter)	25.7
Robust <sup>[17]</sup>	30.4	<b>Shared (RT)</b>	<b>27.2</b>
<b>SPS</b> <sup>[18]</sup>	<b>32.4</b>	Robust	30.1
<b>High-res</b> <sup>[19]</sup>	<b>33.6</b>	<b>High-res</b>	<b>32.2</b>
BP <sup>[20]</sup>	40.0	BP	38.5
Easy <sup>[21]</sup>	40.5	Improved Bayesian	41.4
Improved Bayesian <sup>[22]</sup>	41.0	Bayesian	42.1
Bayesian <sup>[23]</sup>	42.1	Easy	42.6

### 2.3 Necessity

Before the invention of pre-processing, matting equation (1) was applied to any pixel  $z \in \Omega_U$ , no matter whether  $z$  was a pure pixel or not. In fact, the application range of matting equation is limited to mixed pixels. While it is applied to pure pixels, some errors may happen.

Consider the 3-pixel wide area  $F_1$  and  $B_1$  inside  $\Omega_U$  spatially close to  $\Omega_F$  and  $\Omega_B$  respectively. Theoretically, the pixels in  $F_1$  and  $B_1$  must be pure pixels with  $\alpha = 1$  or  $\alpha = 0$ . However, some slight errors of  $\alpha$  always appear in these two areas from the matting algorithms in practice.

Table 2 shows the average errors of  $\alpha$  based on SAD in  $F_1$  and  $B_1$  and the corresponding ranks of the 20 algorithms from Table 1, where three basic types (large, small, user) of Trimap and eight test images are taken from the benchmark. Obviously, the  $\alpha$  in most of the algorithms applying pre-processing is extremely close to 0 or 1, while that of other algorithms is not. Note that the average error in High-res is also very high for reasons discussed in Subsection 2.2. Besides, Segmentation, Improved Bayesian, and SRLO have no pre-processing reported in their original texts. However, their errors are also small, which are perhaps caused by some slight pre-processing operations.

**Table 2.** Average Errors of  $\alpha$  with 20 Algorithms Shown in Table 1 Based on SAD in  $F_1$  and  $B_1$ 

Algorithm	Average Error
<b>WCT</b>	<b>0.002</b>
<b>KL-D-Sparse</b>	<b>0.005</b>
<b>Shared</b>	<b>0.007</b>
<b>Shared (Real Time)</b>	<b>0.007</b>
Segmentation	0.007
<b>Comprehensive</b>	<b>0.008</b>
<b>CWCT</b>	<b>0.008</b>
<b>Sparse Coded</b>	<b>0.009</b>
Improved Bayesian	0.012
SRLO	0.013
<b>SPS</b>	<b>0.057</b>
Improving SC	0.065
BP	0.089
Robust	0.319
Bayesian	0.450
Easy	0.450
Improved Color	0.451
Global	0.452
Global (filter)	0.452
<b>High-res</b>	<b>0.456</b>

The reason of these errors could be explained through the following example. Suppose  $F_z$  and  $B_z$  are samples generated from a sampling algorithm for an un-



known pixel  $z \in F_1$ . The approximate  $\alpha_z = 0.9$  is thus solved according to (1) as shown in Fig.3. However, the composed color  $I_z$  of pixel  $z$  is also close enough to its foreground sample  $F_z$  to make  $z$  be likely to be pure foreground. More importantly, the prior of  $z \in F_1$  actually tells  $\alpha_z = 1$ . Such an example has drawn a conclusion that it is not suitable for some pure pixels to apply the bilateral form of matting equation (1). Other unilateral criteria, such as color and spatial similarities to a unary known region  $\Omega_F$  or  $\Omega_B$ , must be introduced to classify them correctly into pure regions.

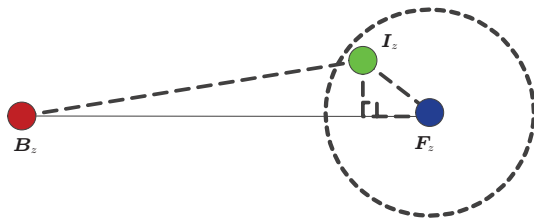


Fig.3. Illustration of the errors in pure sampling, where  $\alpha_z$  is likely to be fractional (near 0.9) according to matting equation. However, the similarity between its color  $I_z$  and foreground sample  $F_z$  indicates that  $z$  is more likely to be a pure foreground pixel.

Consequently, a precise matting process must include the following two steps. The first step is the pre-processing step, or the pure pixels classification step, which must apply some certain methods instead of matting equation. The second step is the execution of matting equation for those mixed pixels with  $0 < \alpha < 1$  and the remaining pure pixels. Thus, a good pre-processing step must classify enough pure pixels from  $\Omega_U$  as many as possible.

#### 2.4 Ideal Trimap and Pure Pixel Rates

According to the statistics of the 27 images from GT01 to GT27 in the training set, the average percentage of pure pixels in  $\Omega_U$  in a small-size Trimap is

60.5%, and in a large size it is 72.4%. Such massive pure pixels also indicate the importance of their initial classifications.

Because  $\alpha^{\text{true}}$  are all provided in the above 27 training images, all of them are thus segmented into three regions based on  $\alpha^{\text{true}}$ : true foreground region  $F = \{z | \alpha_z^{\text{true}} \geq 0.98\}$ , true background region  $B = \{z | \alpha_z^{\text{true}} \leq 0.02\}$ , and the remaining true unknown (mixed) region  $U$ . Such a three-region Trimap is called an ideal Trimap. Fig.4 shows four input large-size Trimaps and ideal Trimaps in four images, where blue and red lines show the initial known F/B edges  $\Phi_F$  and  $\Phi_B$  with unknown region  $\Omega_U$  between them, and the pixels in green show the true unknown region  $U$  in the ideal Trimap. It is clear that between  $\Omega_U$  and  $U$ , there still exist a great number of pure pixels, whose rate could also increase as the foreground edge becomes simpler.

### 3 Experimental Setup and Simulated Effect on Matting Algorithms

Although a rough rank comparison is presented from the benchmark in Table 1, it does not show these algorithms in bold when they do not apply pre-processing. In order to simulate the rank changes<sup>[4]</sup>, the methods of Shared, Sparse Coded, High-res, WCT, KL-D-Sparse and Comprehensive are implemented in this paper, which include results both without and with pre-processing.

Because the source codes of Shared, Sparse Coded and High-res are not publicly provided, the  $\alpha$  matting results without pre-processing under the same code condition for these three algorithms could not be obtained and uploaded to the benchmark. To simulate the rank change in this benchmark, the above three algorithms (without and with pre-processing) are implemented by ourselves in C++. Note that the real-time matting

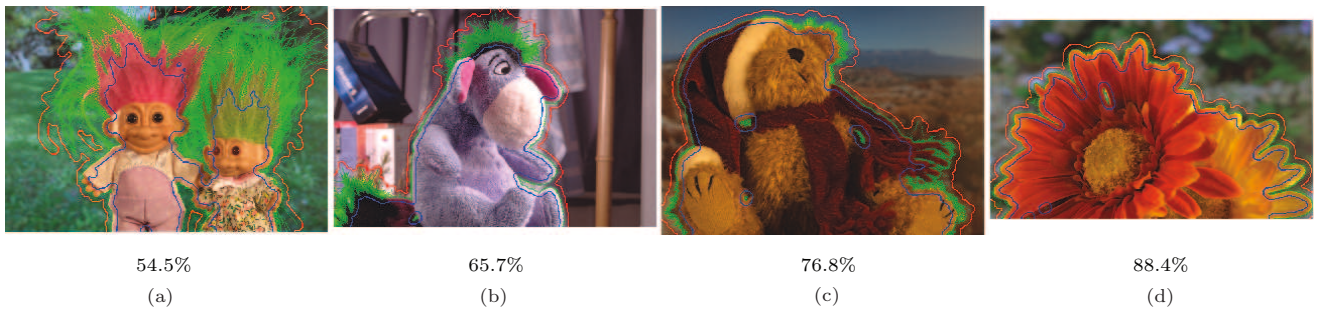


Fig.4. Comparison between initial Trimaps and ideal Trimaps, where the percentages indicate the rates between the amount of pure pixels and that of initial unknown region  $\Omega_U$ .

strategy in Shared similar to Subsection 2.2 and Subsection 2.3 is applied here for a clearer display of the rank change. Besides, WCT, KL-D-Sparse and Comprehensive are provided with public MATLAB source codes. For comparison, another six matting algorithms that do not employ pre-processing are also implemented in C++ to rank together with the above six algorithms.

Our experiments are performed on a PC with an Intel<sup>®</sup> Core i5 CPU with 3.3 GHz and 4 GB memory. The 27 training images in the  $\alpha$  evaluation benchmark system<sup>[4-5]</sup> are included. For each image, a ground truth  $\alpha^{\text{true}}$ , an ideal Trimap, and two types of Trimaps (large and small sizes) are also provided.

A simulated rank change similar to the benchmark<sup>[4]</sup> on SAD of the 27 training images with and without pre-processing for 12 matting algorithms is illustrated in Table 3, where bold texts indicate those applying pre-processing. It shows that the ranks could be greatly raised for relatively weak matting algorithms like WCT, Comprehensive and Sparse Coded after applying pre-processing. Besides, the rank of Shared (Real Time) is raised to the first out of 12 because it also has good matting performance without pre-processing. However, the high rate of false classification in Pre-High-

res makes High-res matting algorithm even worse. It could also conclude that, at least up to now, the conservative static thresholding methods like Pre-Shared-Init, Pre-WCT and Pre-Comprehensive could lead to an improved performance compared with the aggressive methods like Pre-High-res because of fewer false classifications, despite more missing classifications.

The average processing time on 27 training images with large- and small-size Trimaps (54 processing times in total) for each of the above six matting algorithms with and without pre-processing is also shown in Table 4. It is clear that all algorithms have notable speed accelerations in “pure matting” part, which is caused by the greatly reduced amount of unknown pixels. Besides, pre-processing only employs an easy and unilateral method instead of the pair-wise brute-force matting manner, and the total processing speed of these algorithms with pre-processing is thus mostly raised compared with that of those algorithms without it.

An exception is Sparse Coded whose total processing time is long. This is because it applies a special solution without both matting equation and the brute-force processing manner. Thus, the reduced unknown pixels do not quite influence the matting speed.

**Table 3.** Ranks of 12 Implemented Algorithms Based on SAD

Without Pre-Processing				With Pre-Processing			
Algorithm	Average Rank	Trimap Size		Algorithm	Average Rank	Trimap Size	
		Large	Small			Large	Small
KNN <sup>[28]</sup>	2.5	4.43 <sup>1</sup>	3.10 <sup>4</sup>	<b>Shared (Real Time)</b>	<b>1.0</b>	<b>3.21<sup>1</sup></b>	<b>2.36<sup>1</sup></b>
<b>KL-D-Sparse<sup>[6]</sup></b>	<b>3.0</b>	<b>4.72<sup>5</sup></b>	<b>2.83<sup>1</sup></b>	<b>KL-D-Sparse</b>	<b>2.0</b>	<b>3.48<sup>2</sup></b>	<b>2.45<sup>2</sup></b>
<b>Shared (Real Time)<sup>[11]</sup></b>	<b>3.5</b>	<b>4.71<sup>4</sup></b>	<b>3.03<sup>3</sup></b>	<b>Sparse Coded</b>	<b>3.0</b>	<b>3.60<sup>3</sup></b>	<b>2.53<sup>3</sup></b>
LNSP <sup>[29]</sup>	5.0	4.53 <sup>2</sup>	3.48 <sup>8</sup>	<b>WCT</b>	<b>4.5</b>	<b>3.84<sup>4</sup></b>	<b>2.69<sup>5</sup></b>
Global <sup>[12]</sup>	5.5	5.15 <sup>9</sup>	2.97 <sup>2</sup>	<b>Comprehensive</b>	<b>4.5</b>	<b>3.90<sup>5</sup></b>	<b>2.64<sup>4</sup></b>
Closed-Form <sup>[26]</sup>	5.5	4.91 <sup>6</sup>	3.12 <sup>5</sup>	KNN	7.0	4.43 <sup>7</sup>	3.10 <sup>7</sup>
<b>High-res<sup>[19]</sup></b>	<b>7.0</b>	<b>5.03<sup>8</sup></b>	<b>3.35<sup>6</sup></b>	Global	8.5	5.15 <sup>11</sup>	2.97 <sup>6</sup>
SVR <sup>[30]</sup>	7.0	4.70 <sup>3</sup>	3.63 <sup>11</sup>	LNSP	8.5	4.53 <sup>8</sup>	3.48 <sup>9</sup>
<b>WCT<sup>[10]</sup></b>	<b>8.5</b>	<b>5.15<sup>10</sup></b>	<b>3.40<sup>7</sup></b>	<b>High-res</b>	<b>9.0</b>	<b>4.20<sup>6</sup></b>	<b>3.77<sup>12</sup></b>
<b>Sparse Coded<sup>[9]</sup></b>	<b>8.5</b>	<b>4.93<sup>7</sup></b>	<b>3.59<sup>10</sup></b>	Closed-Form	9.0	4.91 <sup>10</sup>	3.12 <sup>8</sup>
Robust <sup>[17]</sup>	10.0	5.21 <sup>11</sup>	3.49 <sup>9</sup>	SVR	10.0	4.70 <sup>9</sup>	3.63 <sup>11</sup>
<b>Comprehensive<sup>[7]</sup></b>	<b>12.0</b>	<b>5.81<sup>12</sup></b>	<b>3.65<sup>12</sup></b>	Robust	11.0	5.21 <sup>12</sup>	3.49 <sup>10</sup>

**Table 4.** Processing Time of 6 Matting Algorithms (in Seconds)

Algorithm	No Pre-Processing	With Pre-Processing		
	Pure Matting	Pure Matting	Pre-Processing	Total
<b>Shared (RT)</b>	17.7	9.8	0.4	10.2
<b>Sparse Coded</b>	37.2	28.2	14.7	42.9
<b>High-res</b>	39.1	19.3	13.0	32.3
<b>WCT</b>	131.8	98.9	6.3	105.1
<b>KL-D-Sparse</b>	228.5	205.6	14.1	219.7
<b>Comprehensive</b>	347.9	284.8	14.1	298.9

All the details of the above pre-processing methods can be found in Section 4.

In summary, pre-processing must have a huge impact on the whole matting process. However in the current research field of image matting, pre-processing has not received enough attentions that it should have, and the corresponding analyses and summaries of various kinds of pre-processing are still lacking. Besides, many of the state-of-the-art matting articles only employ simple methods and spend limited scopes on pre-processing. A thorough interpretation and a plan of future improvement are thus carried out at the remainder of this paper.

#### 4 Analyses on Pre-Processing Methods

The state-of-the-art pre-processing methods can be classified into two basic categories: static thresholding methods and dynamic thresholding methods. The former can also be divided into one loop methods (Pre-WCT and Pre-Shared-Init) and iterative methods (Pre-Comprehensive), and the latter can also be divided into local pixel-wise unilateral learning based method (Pre-Shared-Mid) and global parametric bilateral ratio based method (Pre-High-res). The following subsections of Section 4 will give brief analyses on these four methods.

Denote an ideal Trimap as  $F \cup B \cup U$  and  $x_z = F/B/U$  as a certain classification for an unknown pixel  $z \in \Omega_U$ . The following two negative cases are introduced based on  $x_z$ : missing classification  $x_z = U \wedge z \in F \cup B$  and false classification  $x_z = F/B \wedge z \in U$ .

##### 4.1 Static Thresholding Method

The most popular pre-processing method in recent matting algorithms is the simple and basic static thresholding method, which defines an unchanged threshold for the color difference between unknown pixels and their samples. Besides, spatial distances between unknown pixels and known regions  $\Omega_F$  and  $\Omega_B$  are also employed as well as color difference in this method. Such spatial constraint could also have good effects because the continuity properties hold for both foreground object and background scene during the extensions from  $\Omega_F$  or  $\Omega_B$  to  $\Omega_U$ .

A major problem of this method is that the threshold is difficult to choose according to different local contexts and Trimap sizes.

##### 4.1.1 One Loop Method

Static thresholding methods were initially applied by Shared<sup>[11]</sup> and WCT<sup>[10]</sup> named one loop method based on pixel-wise manner denoted as Pre-Shared-Init and Pre-WCT respectively. Generally, for an unknown pixel  $z \in \Omega_U$ , if there exists a pixel  $F^i \in \Omega_F$ ,

$$\|I_z - F^i\| \leq C_{\text{thr}} \wedge D_s(z, F^i) \leq E_{\text{thr}}, \quad (2)$$

then  $z$  is classified into pure foreground as  $z \in \Omega_F$ , where  $D_s(\cdot, \cdot)$  is the spatial distance between two points, and  $C_{\text{thr}}$  and  $E_{\text{thr}}$  are color and spatial distance thresholds respectively. In Pre-Shared-Init,  $C_{\text{thr}} = 10$  and  $E_{\text{thr}} = 5$ , and in Pre-WCT,  $C_{\text{thr}} = 10$  and  $E_{\text{thr}} = 3$ . A similar formulation is applied to pure background classification. In fact, an additional texture space is also employed to complement the color of (2) in Pre-WCT. In practice, it only makes slight improvements on RGB color space. (2) is also called the basic constraint term for static thresholding methods. Moreover, such constraint could also be illustrated as a sampling square window centered at  $z$  with radius  $E_{\text{thr}}$  shown in Fig.5(a). Pixels inside both this square and  $\Omega_F/\Omega_B$  are named target F/B (foreground/background) samples.

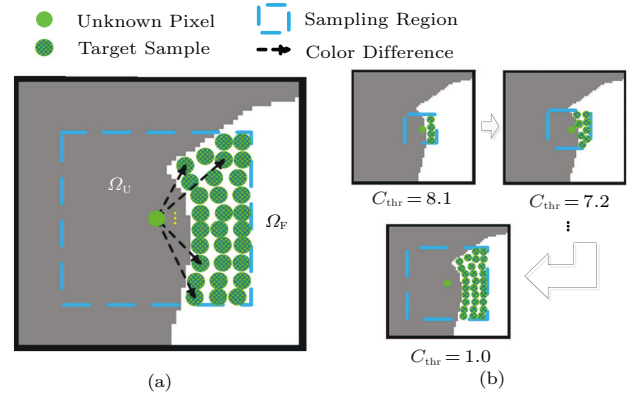


Fig.5. Illustration of the two static thresholding methods in pre-processing. (a) One loop method (Pre-WCT). (b) Iterative method (Pre-Comprehensive).

However, thresholds  $C_{\text{thr}}$  and  $E_{\text{thr}}$  are difficult to set in this case. Large and small thresholds will always lead to a large amount of false and missing classifications for different local images. In practice, the above settings on thresholds of  $C_{\text{thr}}$  and  $E_{\text{thr}}$  are extremely small and conservative, resulting in a great number of missing classifications.

##### 4.1.2 Iterative Method

The most popular method among the current pre-processing ones is the iterative static thresholding

method invented in Comprehensive<sup>[7]</sup>, which is denoted as Pre-Comprehensive in this paper and is further applied in CWCT<sup>[8]</sup>, SPS<sup>[18]</sup>, KL-D-Sparse<sup>[6]</sup>, and Sparse Coded<sup>[9]</sup>. Its key feature is the iterative utilization of (2), and the classification result of the current loop is also fed to the subsequent loops as a new input. The number of iterations on (2) is also  $E_{\text{thr}}$ . For the  $i$ -th loop with  $1 \leq i \leq E_{\text{thr}}$  in (2), the spatial threshold  $E_{\text{thr}}(i) = i$ , and the color threshold  $C_{\text{thr}}$  is calculated as

$$C_{\text{thr}}(i) = C_{\text{thr}} - (C_{\text{thr}} - \mu) \times (i/E_{\text{thr}}).$$

In CWCT, SPS and KL-D-Sparse,  $C_{\text{thr}} = E_{\text{thr}} = 9$ , and in Sparse Coded,  $C_{\text{thr}} = 4$  and  $E_{\text{thr}} = 15$ .  $\mu$  is a small value always calculated as  $\mu = C_{\text{thr}}/E_{\text{thr}}$ .

The above thresholds indicate that the spatial threshold (sampling radius)  $E_{\text{thr}}(i)$  increases and the color threshold  $C_{\text{thr}}(i)$  decreases as the iteration goes on. Fig.5(b) shows this iterative process. Theoretically, this method could be as many as  $1+2+\dots+E_{\text{thr}}$  pixels deep from  $\Omega_F$  and  $\Omega_B$  into  $\Omega_U$ , but it is much fewer in practice.

Obviously, such an iterative method is relatively more aggressive than one loop such that more pure pixels could be classified in common Trimaps and images with simple color distributions. Besides, this method can even lead to some false classifications in a small-size Trimap which is also tolerable for the subsequent matting process. However, similar to one loop, it cannot adjust the fixed thresholds to those Trimaps with much more unknown pixels and those images with quick color variances.

## 4.2 Dynamic Thresholding Method

Although the above static thresholding method is widely used in recent matting algorithms, it is still limited by color and spatial constraints, which could result in a high missing rate and could not make a breakthrough into the deep side of a large size  $\Omega_U$ , even for the much aggressive Pre-Comprehensive. In fact, the threshold must adapt to different images and local contexts.

The dynamic method is much more aggressive for the classification of  $\Omega_U$  compared with the relatively conservative static one. In fact, this method has not emerged for a long time because false classification can happen more frequently. However, in our opinion, such an aggressive manner must be encouraged because the exploration inside  $\Omega_U$  for more pure pixels is becoming more and more necessary in the future development of

image matting, especially for those pure pixels dissimilar to  $\Omega_F$  and  $\Omega_B$ , despite a higher risk of mistake.

Another change for dynamic method is discarding spatial constraints. Thus it could examine much deep inside of  $\Omega_U$ . However, the classification ability on the pixels in  $\Omega_U$  spatially close to  $\Omega_F$  and  $\Omega_B$  may not be as precise as the static method.

### 4.2.1 Local Learning Based Method

The ‘‘sample refinement’’ step of Shared<sup>[11]</sup> introduces a local pixel-wise unilateral learning based method denoted as Pre-Shared-Mid. Although such a method is in the middle part of Shared, its real effect is still the classification of pure pixels and can also be treated as pre-processing.

The basic conception of learning based method is that those pixels in  $\Omega_U$  spatially close enough to  $\Omega_F$  or  $\Omega_B$ , e.g., pixels in the  $5 \times 5$  region in this subsection, and  $F_1$  and  $B_1$  in Subsection 2.3, named learning samples, are most likely to be pure pixels and can simulate those unknown pixels deep inside  $\Omega_U$  far away from  $\Omega_F$  or  $\Omega_B$ . Generally, the color threshold is obtained from the average color difference between learning samples and the edges  $\Phi_F$  and  $\Phi_B$  in known regions. Therefore, the threshold could change in different local contexts and the classification performance is thus improved.

As shown in Fig.6(a), in the ‘‘sample refinement’’ step of Shared, three F/B target sample pairs are collected from the ‘‘sample gathering’’ step according to the least fitting errors (only three foreground samples are shown) for the unknown pixel  $z$ . We denote an arbitrary sample pair as  $\{\mathbf{F}_z, \mathbf{B}_z\}$ , and  $\sigma_f^2$  and  $\sigma_b^2$  are defined as

$$\begin{cases} \sigma_f^2 = \frac{1}{N} \sum_{q \in \Omega_f} \|\mathbf{I}_q - \mathbf{F}_z\|^2, \\ \sigma_b^2 = \frac{1}{N} \sum_{q \in \Omega_b} \|\mathbf{I}_q - \mathbf{B}_z\|^2, \end{cases} \quad (3)$$

where  $\Omega_f$  and  $\Omega_b$  are learning regions centered at  $\mathbf{F}_z$  and  $\mathbf{B}_z$  respectively and  $N = 25$ . Intuitively,  $\sigma_f^2$  and  $\sigma_b^2$  reflect the average local color variances. We denote the average  $\{\mathbf{F}_z, \mathbf{B}_z\}$  and  $\{\sigma_f^2, \sigma_b^2\}$  as  $\{\bar{\mathbf{F}}_z, \bar{\mathbf{B}}_z\}$  and  $\{\bar{\sigma}_f^2, \bar{\sigma}_b^2\}$  respectively. Then  $\{\bar{\sigma}_f, \bar{\sigma}_b\}$  is the dynamic threshold for  $z$  with respect to  $\{\bar{\mathbf{F}}_z, \bar{\mathbf{B}}_z\}$ . That is,

$$\begin{cases} z \in F, & \text{if } \|\mathbf{I}_z - \bar{\mathbf{F}}_z\| \leq \bar{\sigma}_f, \\ z \in B, & \text{if } \|\mathbf{I}_z - \bar{\mathbf{B}}_z\| \leq \bar{\sigma}_b. \end{cases}$$

Besides, if both of the above two conditions are met,  $z$  is not classified.

A major problem of such dynamic threshold is that if the  $5 \times 5$  neighbors of  $\mathbf{F}_z, \mathbf{B}_z$  have a large color variance, the thresholds  $\sigma_f^2$  and  $\sigma_b^2$  should become unstable.



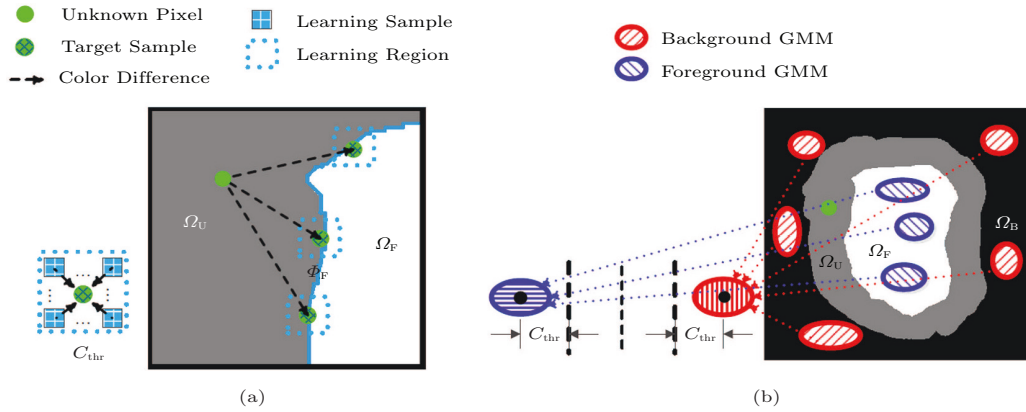


Fig.6. Illustration of the two dynamic thresholding methods in pre-processing. (a) Local pixel-wise unilateral learning based method (Pre-Shared-Mid). (b) Global parametric bilateral ratio based method (Pre-High-Res).

In other words, some of the thresholds are large and others are small. They could both cause missing classification and false classification. Therefore, too much noise could emerge in the final Trimap which will be shown in the consequent experiments.

In fact, the three target sample pairs and  $5 \times 5$  learning neighbors are definitely small sets. If they are too large to cover enough learning samples, such as large learning regions like  $F_1$  and  $B_1$  in Subsection 2.3, the noise could be greatly reduced on these thresholds. In addition, the initial sample pair  $\{F_z, B_z\}$  should also be selected by simple color difference instead of fitting errors, such that this type of dynamic thresholds could thus be placed before the sampling step of matting as a pre-processing step.

#### 4.2.2 Global Ratio Based Method

In the pre-processing step named ‘‘Trimap segmentation’’ of High-res<sup>[19]</sup>, a global parametric bilateral ratio based method denoted as Pre-High-res is employed. Because both sides of foreground and background should be simultaneously considered for the classification of any unknown pixel  $z$ , this type of classification is thus called ‘‘bilateral’’, which is also the only bilateral pre-processing method up to now.

This method firstly employs the binary segmentation results of GrabCut<sup>[31]</sup> denoted as  $F'$  and  $B'$ . Then the following energy function is minimized by means of Max-Flow<sup>[32-33]</sup> in MRF:

$$E(z, \theta) = \sum_i U_i^c(z_i) + U_i^p(z_i) + \theta_o U_i^o(z_i) + \theta_s \sum_{i,j \in \Omega} V_{ij}^s(z_i, z_j), \quad (4)$$

which is separated by data cost  $U_i(z_i)$  and smoothness

cost  $V_{ij}(z_i, z_j)$ . The unknown region  $U$  and the known region  $\bar{U}$  are thus solved. Finally, the final pure regions are obtained by  $\bar{U} \cap F' = F$  and  $\bar{U} \cap B' = B$ .

For smoothness cost  $V_{ij}^s(z_i, z_j)$  in (4),  $\Omega$  is the 8-pixel neighborhood and  $\theta_s = 0.1$  (50 in GrabCut), which means that smoothness cost is relatively weak compared with the data. In fact, the edge of the unknown region  $U$  is not necessarily smooth. For data cost  $U_i(z_i)$ , color similarity term  $U_i^c(z_i)$  is the most essential to the classification results and will be briefly discussed. Terms  $U_i^p(z_i)$  and  $U_i^o(z_i)$  will be discussed as other pure pixel priors in Section 5.

In practice, the above MRF could be separated into two types according to GrabCut result of  $F'/B'$ , and the color term  $U_i^c(z_i)$  could be defined as follows. If  $z_i \in F'$ , then  $z_i$  is classified into  $F$  or  $U$ , and  $U_i^c(z_i \in \bar{U}) = -\log P(\mathbf{I}_i | \theta_F)$ ; if  $z_i \in B'$ , then  $z_i$  is classified into  $B$  or  $U$ , and  $U_i^c(z_i \in \bar{U}) = -\log P(\mathbf{I}_i | \theta_B)$ ;  $U_i^c(z_i \in U) = -\log P(\mathbf{I}_i | \theta_U)$ .  $P(\mathbf{I} | \theta)$  is the probability of color  $\mathbf{I}$  under model  $\theta$ , and  $\theta_F$ ,  $\theta_B$  and  $\theta_U$  are Gaussian mixture models (GMMs) of global foreground, global background, and global unknown region, respectively, where  $\theta_U$  is obtained by the method of [34].

A much intuitive expression for color term  $U_i^c(z_i)$  is illustrated in Fig.6(b). The centroid of  $\theta_U$  could be roughly located at the midpoint between those of  $\theta_F$  and  $\theta_B$  in RGB space. Besides, the data cost part of the above MRF optimization step could also be treated as a binary classifier, such that the midpoints of  $\theta_U$  and  $\theta_F$ , and  $\theta_U$  and  $\theta_B$  are both classification boundaries shown in Fig.6(b). Thus, 1/4 and 3/4 positions of the centroids of  $\theta_F$  and  $\theta_B$  are F/B classification boundaries for  $z_i \in F'$  and  $z_i \in B'$ , respectively. And this method is thus named ‘‘ratio based’’ with 1:3 as color difference constraint. Note that only the color probability

$P(\mathbf{I}|\theta)$  is considered here and the negative logarithm in the above MRF is ignored.

The following drawbacks exist in Pre-High-res.

1) The ratio of F/B is opposite to the “absolute threshold” applied by all the previous methods. Empirically, absolute threshold is necessary in pre-processing, especially in complex regions. For instance, suppose the color difference between an unknown pixel  $z$  and the foreground color is 30, and the difference between  $z$  and the background color is 100. Thus the ratio is  $30/100 = 0.3 < 1/3$ . In fact,  $z$  is not likely to be classified into  $F$ , because the absolute threshold is about less than 10 for ordinary images and 30 is obviously too large. Moreover, it is still too large for the ratio 1:3 which could cause too many false classifications. In practice, 1:10 or less may be a more appropriate one.

2) In global method, when many overlapping colors exist in global foreground and background, plenty of false classifications in the final Trimap could appear. Two such cases are shown in Fig.7. In the local unknown region of the left bottom corner of GT24, the color of the foreground hair is similar to a piece of known background far away in green rectangle on top of this image. Besides, in the local unknown region of the right part of GT11, the color of a piece of the background is also similar to the several labeled known foreground pixels. Consequently, plenty of false classifications are caused in Pre-High-res.

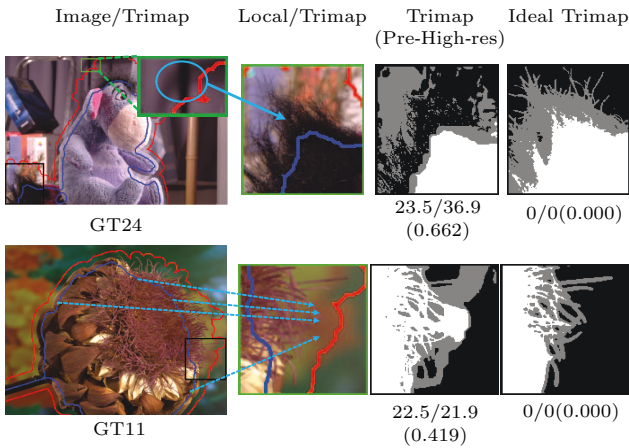


Fig.7. False classifications caused by global manner in Pre-High-res.  $x/y(z)$ : missing rate (%) / false rate (%) (false error).

### 4.3 Experimental Comparisons

The missing rate (MR) and the false rate (FR) are defined as follows:

$$\text{MR} \ \& \ \text{FR} = N_{\text{missing}} \ \& \ N_{\text{false}}/N_U,$$

where  $N_{\text{missing}}$  and  $N_{\text{false}}$  are the amounts of pixels with missing and false classification respectively, and  $N_U$  is the amounts of total unknown pixels. Note that these two rates are shown in a separate manner in this paper different from the error function of classification in High-res.

In practice, the false error (FE) should also be considered to further evaluate the degree of negative effect in false classification. For example, for a classification of  $x_z = F$  (i.e.,  $\alpha_z = 1$ ) and  $\alpha_z^{\text{true}} = 0.9$ , the false error for this classification of  $z$  is 0.1.

#### 4.3.1 Overall Comparison

Table 5 shows the average MR, FR, and FE in the 27 training images on the four pre-processing methods Pre-WCT, Pre-Comprehensive, Pre-Shared-Mid and Pre-High-res, as well as “no pre-processing” with original Trimap. Note that 72.4% and 62.5% are the same pure pixel rates in  $\Omega_U$  with those in Subsection 2.4.

The following aspects can be derived from Table 5.

1) Pre-WCT is the most conservative and has the highest MR (37.2% and 18.5%) and the lowest FR (0.1% and 2.8%) in both large and small Trimaps.

2) Pre-Comprehensive brings in least MR+FR (24.2% and 19.3%) and can thus be regarded as the best pre-processing method up to now. However, the FR in small-size Trimaps (8.0%) is relatively high, which reflects its weak ability on the adaptive adjustment according to different Trimap sizes.

3) The FRs in dynamic thresholding methods Pre-Shared-Mid and Pre-High-res are both high (6%~10%). Besides, MRs in these two methods are also not satisfactory (15%~20%) compared with static thresholding methods.

#### 4.3.2 Cases with Fewer Unknown Pixels

Four common and simple local cases are shown in Fig.8 to compare the resulting Trimaps for these four methods in both large (first two rows) and small Trimaps (last two rows). Obviously, the lower MR and the higher FR in Pre-Comprehensive compared with those in Pre-WCT show that Pre-Comprehensive is relatively more aggressive than Pre-WCT. However, due to the weak adjustment abilities of both methods, the FR can reach up to 15%~18% in small-size Trimaps with a narrow unknown region.

In contrast, because Pre-Shared-Mid highly relies on a small sample set of each target sample and is heavily influenced by noise, its results are thus unstable in

**Table 5.** Average Trimap Comparison of 4 Pre-Processing Methods on 27 Training Images

Pre-Processing Method	Large Trimap			Small Trimap		
	Missing Rate (%)	False Rate (%)	False Error	Missing Rate (%)	False Rate (%)	False Error
No pre-processing	72.4	0.0	0.000	60.5	0.0	0.000
Pre-WCT	37.2	0.1	0.083	18.5	2.8	0.099
Pre-Comprehensive	21.7	2.5	0.084	11.3	8.0	0.098
Pre-Shared-Mid	20.2	7.0	0.218	14.8	10.2	0.176
Pre-High-res	19.3	6.1	0.194	18.5	8.6	0.173

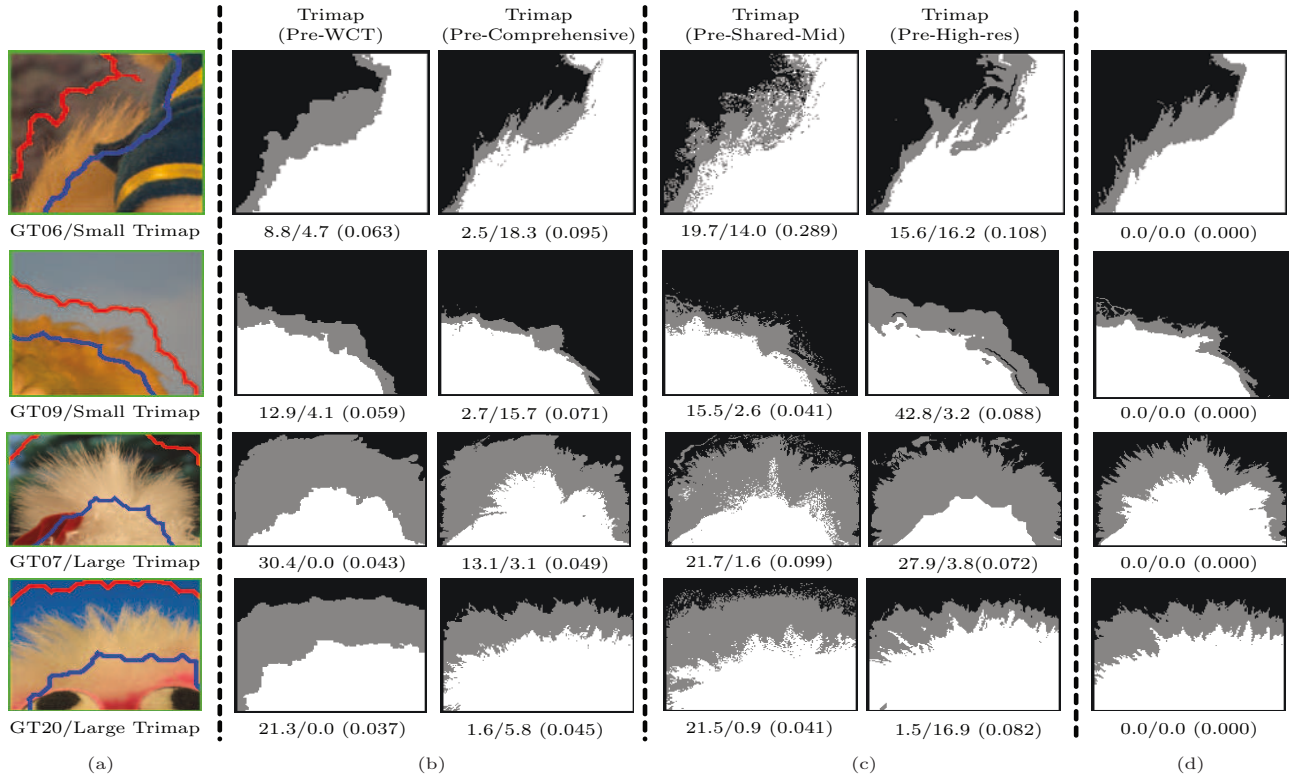


Fig. 8. Classification comparison of the Trimap results on 4 pre-processing methods in local images: missing rate (%) / false rate (%) (false error). (a) Image/Trimap. (b) Static threshold. (c) Dynamic threshold. (d) Ideal Trimap.

MR and FR. Besides, the results of Pre-High-res are even more unstable, which is mainly caused by the over global parametric sampling manner.

A conclusion can be drawn that in the cases with fewer unknown pixels and simple color distributions, the result of static methods is much better and more reliable than dynamic methods because both color and spatial static thresholding conditions can be easily met. Besides, these simple local cases are also common in the 27 training images, and the above results can thus be treated as explanations of Table 5.

#### 4.3.3 Cases with More Unknown Pixels

Fig.9 shows five local cases with a large unsolved area and relatively gradual color changes of unknown region, and plenty of missing classifications can be

found in static thresholding methods because of the conservative spatial constraints. In contrast, in the dynamic thresholding methods, the unsolved region is greatly reduced.

However in Pre-Shared-Mid and Pre-High-res, their FR and FE are both very high (also see yellow arrows in Fig.9). The reasons are discussed in Subsection 4.2. A difference between the two methods is that in Pre-Shared-Mid, a great deal of noise also appears like Fig.8 and the shape of final Trimap is rough due to the lack of samples. However in Pre-High-res, the shape of Trimap is much smooth because of the interference of smoothness cost in the energy function of MRF.

Fig.10 shows some more complex cases which have more intense color changes than Fig.9. Static thresholding methods are still conservative to solve unknown

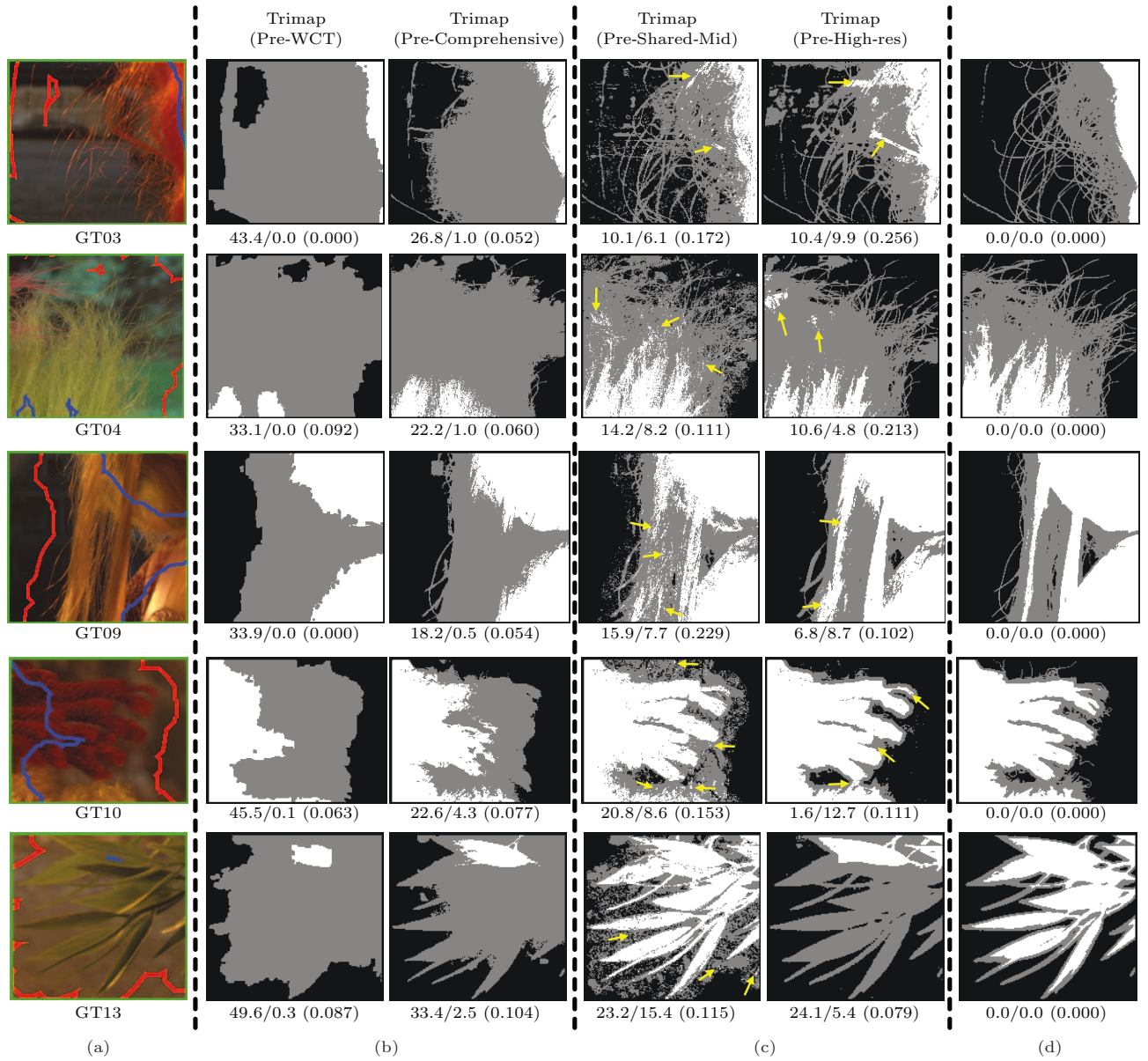


Fig.9. Comparisons on the 4 pre-processing methods in local cases with more unknown pixels:  $x/y(z)$ : missing rate (%) / false rate (%) (false error). (a) Local/Trimap. (b) Static threshold. (c) Dynamic threshold. (d) Ideal Trimap.

regions. Note that in Pre-High-res, the MR is much higher compared with that in Pre-Shared-Mid. This is mainly caused by the unstable global parametric manner that could weaken the effects of some key samples for some unknown pixels. Hence, these pixels could not be precisely classified by Pre-High-res due to large color differences.

In addition, in Pre-Shared-Mid of Fig.10, the main contours of foreground objects have already been roughly recognized compared with Pre-High-res despite too much noise and high FR. This is a breakthrough and is also realized in the fourth column of Fig.9. Thus,

it could be convinced that local learning based method is somewhat superior to global ratio based one in handling complex regions. Furthermore, this method could be further improved by increasing the amount of target samples and local size to eliminate noise and increase robustness. Such an idea will be described in details in Subsection 6.2.

## 5 Other Pure Pixel Priors

Apart from color similarities, there are some other methods to obtain pure pixel priors in the unknown re-



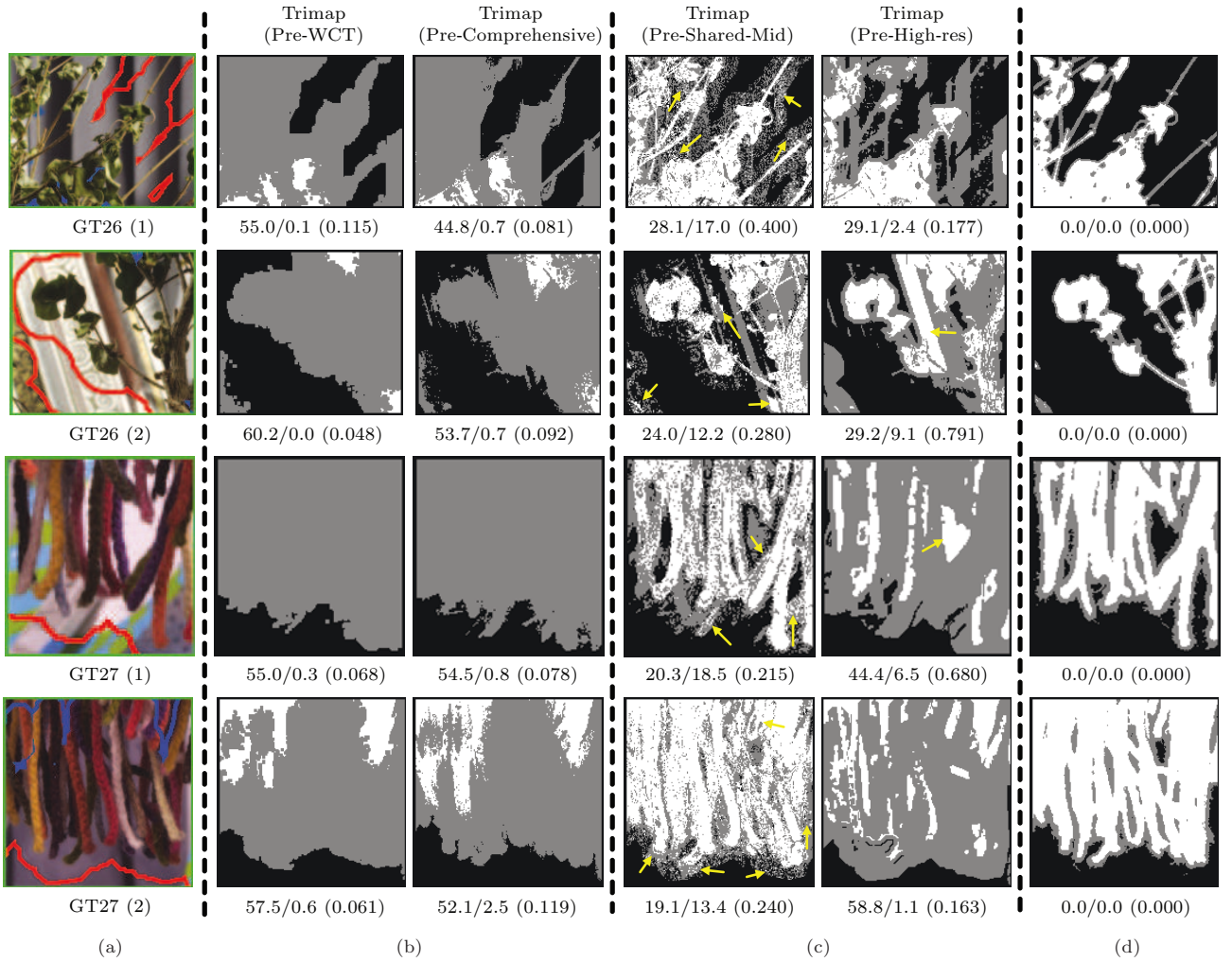


Fig.10. Comparisons in local cases with more unknown pixels and complex color distributions:  $x/y(z)$ : missing rate (%)/false rate (%) (false error). (a) Local/Trimap. (b) Static threshold. (c) Dynamic threshold. (d) Ideal Trimap.

gion. As discussed in Pre-High-res of Subsection 4.2.2, the following two additional terms are also defined as pure pixel priors in the energy function of (4) besides the color term  $U_i^c(z_i)$ .

*Known Region Prior  $U_i^p(z_i)$ .* For each unknown pixel  $z$ ,  $U_i^p(z_i \in \bar{U}) = \lambda$ , where  $\lambda$  is concerned with the ratio between the number of unknown pixels and that of total pixels. The size of the unknown region  $\Omega_U$  in different images can be adjusted by an adaptive  $\lambda$  predicted from a quadratic function. Besides, an optimal  $\lambda = 2.3$  could also be adopted independently of each image, which would lead to a relatively worse result.

However, the authors in High-res of [19] also admitted that  $\lambda$  is difficult to set and should be adjusted manually to generate a good Trimap. In practice, it has been widely acknowledged that an outstanding method must automatically handle these parameters by itself.

*Sharp Boundary Prior  $U_i^o(z_i)$ .* This indicates the solid foreground edge with quick and smooth transition to background, which is different from fuzzy boundary or semi-transparent inner. Generally, sharp boundary could exist at the segmentation boundary of GrabCut. The radius of unknown region in sharp boundary is likely to be the width of PSF (point spread function) of the camera and is set to 1 in this paper. The recovery of sharp boundary is helped by the color term  $U_i^c(z_i)$  in the data cost of (4) and by the  $\alpha$  result of Closed-Form<sup>[26]</sup>. Besides, sharp boundary could be further extended to obtain an additional known area denoted as “barrier bound”, which has the lowest energy in (4) (refer to the technical report version of [19] for details).

Fig.11 shows four examples of the detection of sharp boundary (pink pixels in the first column), on which the pre-processing results of Pre-High-res are based (bar-

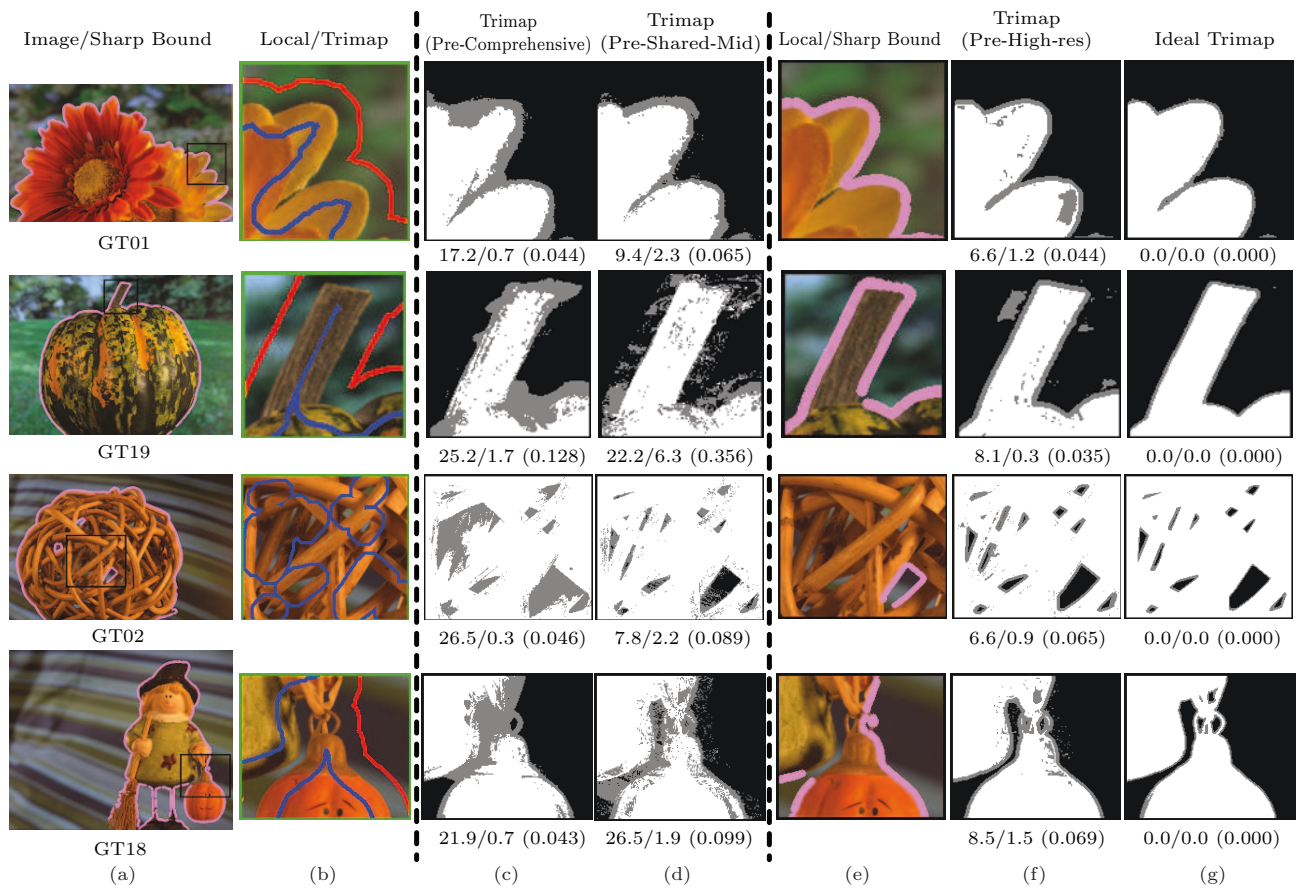


Fig.11. Local cases showing the advantages of sharp boundary detector in Pre-High-res.  $x/y(z)$ : missing rate (%)/false rate (%) (false error).

rier bound results are not shown). In the local cases of the last six columns, the missing and false rates of the Trimap results of Pre-High-res are much lower than those of Pre-Comprehensive and Pre-Shared-Mid with the help of sharp boundary and barrier bound. Note that in the last two rows of GT02 and GT18, the background samples are always spatially far away from known regions because of holes and narrow bounds in the image. The global manner in Pre-High-res could thus help to search far-away background samples, where the low overlapping colors between foreground and background in these cases are different from those in Fig.7.

However, it is risky for the direct application of the results of GrabCut (initial  $F'/B'$  segmentation) and Closed-Form (sharp boundary detection). In fact, the subsequent procedure should be highly influenced and could not be adjusted if the previous method fails. As also discussed in [35], the methods of GrabCut and Closed-Form are far away from perfect. Besides, this could also destroy the art of the whole process.

## 6 Summaries and Proposed Approach

### 6.1 Summaries

The main techniques of pre-processing methods can be summarized into the following aspects.

1) *Global and Local Target Sampling.* The fundamental sampling way is in local like Pre-WCT, Pre-Comprehensive and Pre-Shared-Mid. Besides, the global method in Pre-High-res should be employed to complement local method when the foreground object contains a lot of holes, or the size of unknown region is too large, where true samples are far away from the current unknown pixel. A main drawback for global method is the cause of false classification due to global overlapping colors. Meanwhile, global method can also weaken the effect of local colors and can bring in more missing classifications.

2) *Threshold Setting Methods:*

- *Static, Learning Based and Ratio Based.* Static thresholding methods like Pre-WCT and Pre-Comprehensive can always generate good Trimap re-

sults for Trimaps with fewer unknown pixels. However, they could not make any breakthrough for large-size Trimaps and for large fuzzy regions due to the conservative threshold and spatial constraint. The local learning based method in Pre-Shared-Mid is relatively successful with adaptive threshold according to different local contexts. However in Pre-High-res, false classification can easily happen due to large base threshold and large ratio.

- *Unilateral and Bilateral.* Take foreground pixel classification for example. Unilateral method is employed in Pre-Comprehensive and Pre-Shared-Mid, which compares the color of the current unknown pixel with only foreground samples (no background), and could only make a relatively fixed threshold for foreground. However, the bilateral method in Pre-High-res could also magnify the threshold based on the color comparison with background samples. Such an enlarged threshold could cover more pure foreground pixels which could hardly coincide with background ones.

3) *Priors.* The sharp boundary detector in Pre-High-res could bring in an effective prior of barrier bound for pure pixels near the sharp edge. Besides, some other priors, such as a defocused background scene, could also be introduced as prior pixels for complex defocused background edges. Predictably, such prior can classify those background pixels similar to foreground ones in color which cannot be routinely identified.

## 6.2 Proposed Approach

According to the above analyses, a relatively more effective approach is consequently formed for pre-processing, which is an improved version of Pre-Shared-Mid based on local pixel-wise learning bilateral based method shown in Fig.12. Note that parametric method like GMM could also be introduced based on the simplicity of local color distributions.

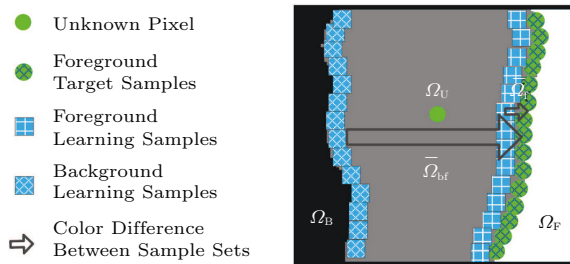


Fig.12. Proposed local approach for pre-processing in matting.

Foreground pixel classification is also shown here as an example. A local window is firstly generated including certain unknown pixels. Then a foreground threshold  $\bar{\sigma}_f$  is obtained similar to (3) from the average color difference between target samples and learning samples whose amounts are both greatly increased. The increased sizes could result in a more stable threshold and could greatly reduce noise. Besides, the foreground learning sample set is similar to that of the area  $F_1$  in  $\Omega_U$  defined in Subsection 2.3.

Meanwhile, as discussed in Subsection 6.1, the bilateral method like Pre-High-res is also employed by introducing a set of background learning samples, which is spatially close to known background region  $\Omega_B$  similar to the area  $B_1$ . The average foreground and background color difference  $\bar{\sigma}_{bf}$  is obtained according to the average difference between the background learning set and the foreground target set. Predictably,  $\bar{\sigma}_{bf}$  is much larger than  $\bar{\sigma}_f$ . Thus, threshold  $\bar{\sigma}_f$  could be enlarged into a new  $\hat{\sigma}_f$  according to the following equation:

$$\hat{\sigma}_f = (\bar{\sigma}_{bf} - \bar{\sigma}_f) \times \theta, \quad (5)$$

where  $\theta$  is small value like 0.01. For instance, suppose the average color difference between the learning and the target sets of foreground  $\bar{\sigma}_f$  is 2. In practice, the ideal threshold  $\hat{\sigma}_f$  for foreground classification is always slightly greater than 2, and is decided here by  $\bar{\sigma}_{bf} - \bar{\sigma}_f$ . When it is 50,  $\hat{\sigma}_f$  is enlarged to about 2.5 according to (5). And when it is 100,  $\hat{\sigma}_f$  could be up to 3.

Note that the static thresholding method like Pre-WCT or Pre-Comprehensive could also be employed for an initial rough classification with some conservative thresholds.

## 7 Conclusions

In digital image matting, the specific characteristic for pure foreground and background pixels makes it important for pre-processing to be an isolated step from the pure matting which employs the traditional matting equation. The pre-processing step could not only precisely classify those pure pixels which cannot be easily classified by most of pure matting methods, but also bring in effective samples for the subsequent matting process. However, pre-processing step has not been paid enough attention in the current research field of image matting, and is only treated as a small supplement to pure matting in the state-of-the-art algorithms. Moreover, its complexity is also neglected.



This survey paper firstly classified the pre-processing step as static thresholding methods and dynamic thresholding methods, and then made thorough analyses on the advantages and disadvantages of the two basic categories. Experimental results showed that pre-processing could greatly improve the matting results, but the missing and false classifications are also popular. Besides, theoretical analyses and experimental results also indicated that static thresholding methods are good at initial conservative classifications for the unknown region, and dynamic thresholding methods tend to make aggressive classifications in complex cases. In addition, some other pure pixel priors such as sharp boundary detector were also raised and discussed.

Finally, in order to further overcome the problems of the state-of-the-art pre-processing methods, a design thinking of a more effective approach of pre-processing is finally presented to inspire new work on this field that can bring in more accurate pure prior samples for the performance of matting.

**Acknowledgment** The author would like to thank Shao-Hui Liu and anonymous reviewers for their constructive and helpful comments which definitely improve the quality of the paper.

## References

- [1] Wang J, Agrawala M, Cohen M F. Soft scissors: An interactive tool for realtime high quality matting. *ACM Transactions on Graphics*, 2007, 26(3): Article No. 9.
- [2] Liu K, Li X, Dong Y. Superpixel fats for fast foreground extraction. In *Proc. IEEE China Summit and International Conference on Signal and Information Processing*, Jul. 2015, pp.132-136.
- [3] Porter T, Duff T. Compositing digital images. In *Proc. the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Jul. 1984, pp.253-259.
- [4] Rhemann C, Rother C, Wang J, Gelautz M, Kohli P, Rott P. Alpha matting evaluation website, 2009. <http://www.alphamatting.com>, Jun. 2016.
- [5] Rhemann C, Rother C, Wang J, Gelautz M, Kohli P, Rott P. A perceptually motivated online benchmark for image matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp.1826-1833.
- [6] Karacan L, Erdem A, Erdem E. Image matting with KL-divergence based sparse sampling. In *Proc. IEEE International Conference on Computer Vision*, Dec. 2015, pp.424-432.
- [7] Shahrian E, Rajan D, Price B, Cohen S. Improving image matting using comprehensive sampling sets. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp.636-643.
- [8] Varnousfaderani E, Rajan D. Weighted color and texture sample selection for image matting. *IEEE Transactions on Image Processing*, 2013, 22(11): 4260-4270.
- [9] Johnson J, Rajan D, Cholakkal H. Sparse codes as alpha matte. In *Proc. the British Machine Vision Conference*, Sept. 2014, pp.245-253.
- [10] Shahrian E, Rajan D. Weighted color and texture sample selection for image matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp.718-725.
- [11] Gastal E S L, Oliveira M M. Shared sampling for real-time alpha matting. *Computer Graphics Forum*, 2010, 29(2): 575-584.
- [12] He K, Rhemann C, Rother C, Tang X, Sun J. A global sampling method for alpha matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2011, pp.2049-2056.
- [13] Rhemann C, Rother C, Kohli P, Gelautz M. A spatially varying PSF-based prior for alpha matting. In *Proc. the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp.2149-2156.
- [14] He B, Wang G, Shi C *et al.* High-accuracy and quick matting based on sample-pair refinement and local optimization. *IEICE Transactions on Information and Systems*, 2013, E96-D(9): 2096-2106.
- [15] Rhemann C, Rother C, Gelautz M. Improving color modeling for alpha matting. In *Proc. the British Machine Vision Conference*, Sept. 2008, pp.1155-1164.
- [16] Cheng J, Miao Z. Improving sampling criterion for alpha matting. In *Proc. the 2nd IAPR Asian Conference on Pattern Recognition*, Nov. 2013, pp.803-807.
- [17] Wang J, Cohen M F. Optimized color sampling for robust matting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007.
- [18] AL-Kabbany A, Dubois E. Improved global-sampling matting using sequential pair-selection strategy. In *Proc. Visual Information Processing and Communication*, Feb. 2014.
- [19] Rhemann C, Rother C, Rav-Acha A, Sharp T. High resolution matting via interactive Trimap segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2008.
- [20] Wang J, Cohen M F. An iterative optimization approach for unified image segmentation and matting. In *Proc. the 10th IEEE International Conference on Computer Vision*, Oct. 2005, pp.936-943.
- [21] Guan Y, Chen W, Liang X, Ding Z, Peng Q. Easy matting: A stroke based approach for continuous image matting. *Computer Graphics Forum*, 2006, 29(2): 567-576.
- [22] Tan W, Fan T, Chen X, Ouyang Y, Wang D, Li G. Automatic matting of identification photos. In *Proc. International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, Nov. 2013, pp.387-388.
- [23] Chuang Y Y, Curless B, Salesin D H, Szeliski R. A Bayesian approach to digital matting. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2001, pp.264-271.
- [24] Sun J, Jia J Y, Tang C K, Shum H Y. Poisson matting. *ACM Transactions on Graphics*, 2004, 23(3): 315-321.
- [25] Grady L, Schiwietz T, Aharon S, Westermann R. Random walks for interactive alpha matting. In *Proc. International Conference on Visualization, Imaging, and Image Processing*, Sept. 2005, pp.423-429.



- [26] Levin A, Lischinski D, Weiss Y. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, 30(2): 228-242.
- [27] Lee P, Wu Y. Nonlocal matting. In *Proc. the 24th IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2011, pp.2193-2200.
- [28] Chen Q, Li D, Tang C K. KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(9): 2175-2188.
- [29] Chen X, Zou D, Zhou S Z, Zhao Q, Tan P. Image matting with local and nonlocal smooth priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp.1902-1907.
- [30] Zhang Z, Zhu Q, Xie Y. Learning based alpha matting using support vector regression. In *Proc. the 19th IEEE International Conference on Image Processing*, Sept. 30-Oct. 3, 2012, pp.2109-2112.
- [31] Rother C, Kolmogorov V, Blake A. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004, 23(3): 309-314.
- [32] Boykov Y, Kolmogorov V. An experimental comparison of Min-Cut/Max-Flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, 26(9): 1124-1137.
- [33] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, 23(11): 1222-1239.
- [34] Juan O, Keriven R. Trimap segmentation for fast and user-friendly alpha matting. In *Proc. the 3rd Variational, Geometric, and Level Set Methods in Computer Vision*, Oct. 2005, pp.186-197.
- [35] Wang J, Cohen M F. Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision*, 2007, 3(2): 97-175.



**Gui-Lin Yao** received his B.E., M.E. and Ph.D. degrees in computer science and technology from Harbin Institute of Technology, Harbin, in 2003, 2005 and 2013, respectively. He is currently an associate professor in the School of Computer and Information Engineering, Harbin University of Commerce, Harbin. His research interests include image processing, computer vision, and video surveillance.