University of Alberta

SINGLE COMPLEX IMAGE MATTING

by

Yufeng Shen

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Spring 2010
Edmonton, Alberta

Examining Committee

Herbert Yang, Computing Science

Joerg Sander, Computing Science

Benoit Rivard, Earth and Atomospheric Science

Abstract

Single image matting refers to the problem of accurately estimating the foreground object given only one input image. It is a fundamental technique in many image editing applications and has been extensively studied in the literature. Various matting techniques and systems have been proposed and impressive advances have been achieved in efficiently extracting high quality mattes. However, existing matting methods usually perform well for relatively uniform and smooth images only but generate noisy alpha mattes for complex images. The main motivation of this thesis is to develop a new matting approach that can handle complex images. We examine the color sampling and alpha propagation techniques in detail, which are two popular techniques employed by many state-of-the-art matting methods, to understand the reasons why the performance of these methods degrade significantly for complex images. The main contribution of this thesis is the development of two novel matting algorithms that can handle images with complex texture patterns. The first proposed matting method is aimed at complex images with homogeneous texture pattern background. A novel texture synthesis scheme is developed to utilize the known texture information to infer the texture information in the unknown region and thus alleviate the problems introduced by textured background. The second proposed matting algorithm is for complex images with heterogeneous texture patterns. A new foreground and background pixels identification algorithm is used to identify the pure foreground and background pixels in the unknown region and thus effectively handle the challenges of large color variation introduced by complex images. Our experimental results show that the proposed matting methods can effectively handle images with complex background and generate cleaner alpha mattes than existing matting methods.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Single image matting refers to the problem of accurately estimating the foreground object given only one input image. It is a fundamental technique in many image editing applications and has been extensively studied for more than two decades. The first formal introduction of the matting problem was by Porter and Duff in 1984 [1]. The original purpose of their work is to introduce the alpha channel as the way to blend the foreground and background images. Mathematically, a given image I is considered to be a linear combination of a background image *B* and a foreground image *F* using the *compositing equation:*

$$I = \alpha F + (1 - \alpha)B \tag{1.1}$$

where the alpha matte α takes on values in the range [0,1]. The pixel with corresponding $\alpha = 1 \; or \; 0$ is said to be a pure foreground or definite background pixel, respectively. Otherwise, it is a mixed pixel. The task of single image matting is to accurately estimate α, *F* and *B,* given only the input image I. For an image with three color channels, there are three equations and seven unknowns at each pixel. Hence, single image matting is inherently an under-constrained problem. Most existing matting approaches require the user to provide some prior knowledge about the image foreground and background and also make different assumptions about the image statistics to constrain the ill-posed problem to be tractable. Once the alpha matte is estimated, the foreground image *F* can then be reconstructed and composited with a new background using Equation 1.1. A matting example is shown in Figure 1-1. The alpha matte is estimated using the robust matting algorithm [2]. We can see that with the accurately estimated alpha matte, the girl is well extracted from the input image and blended with a different background image.

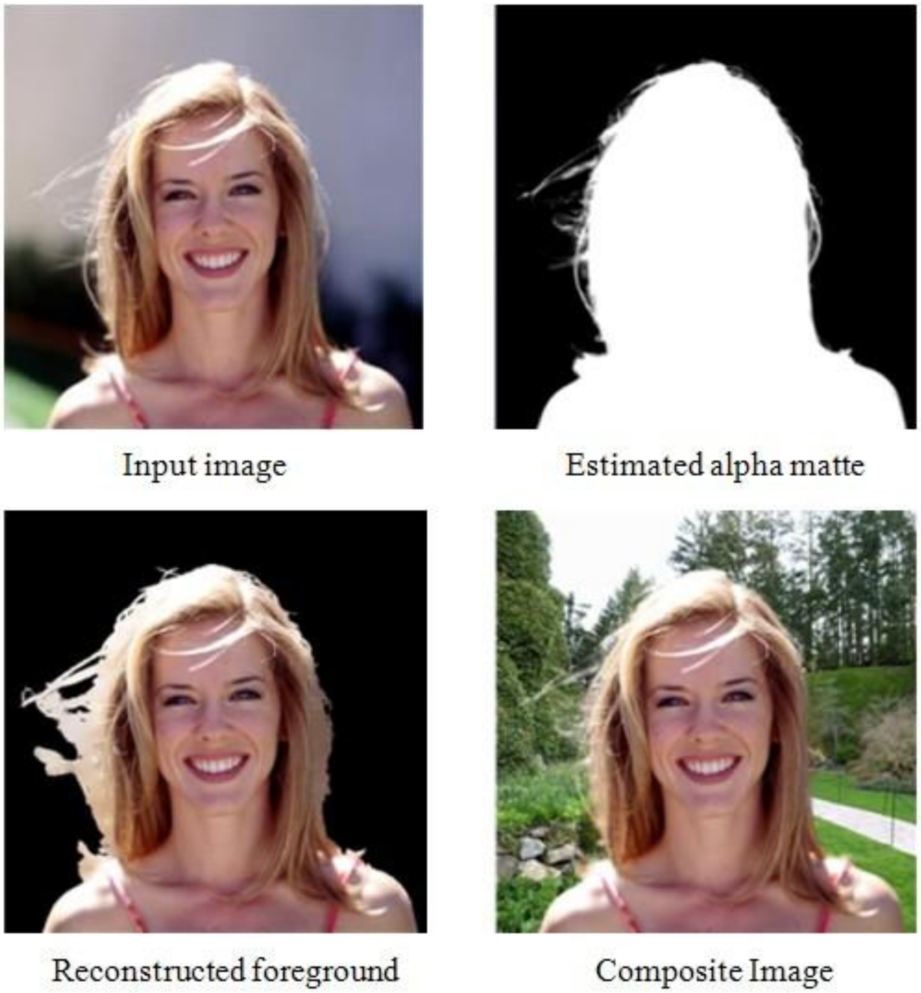
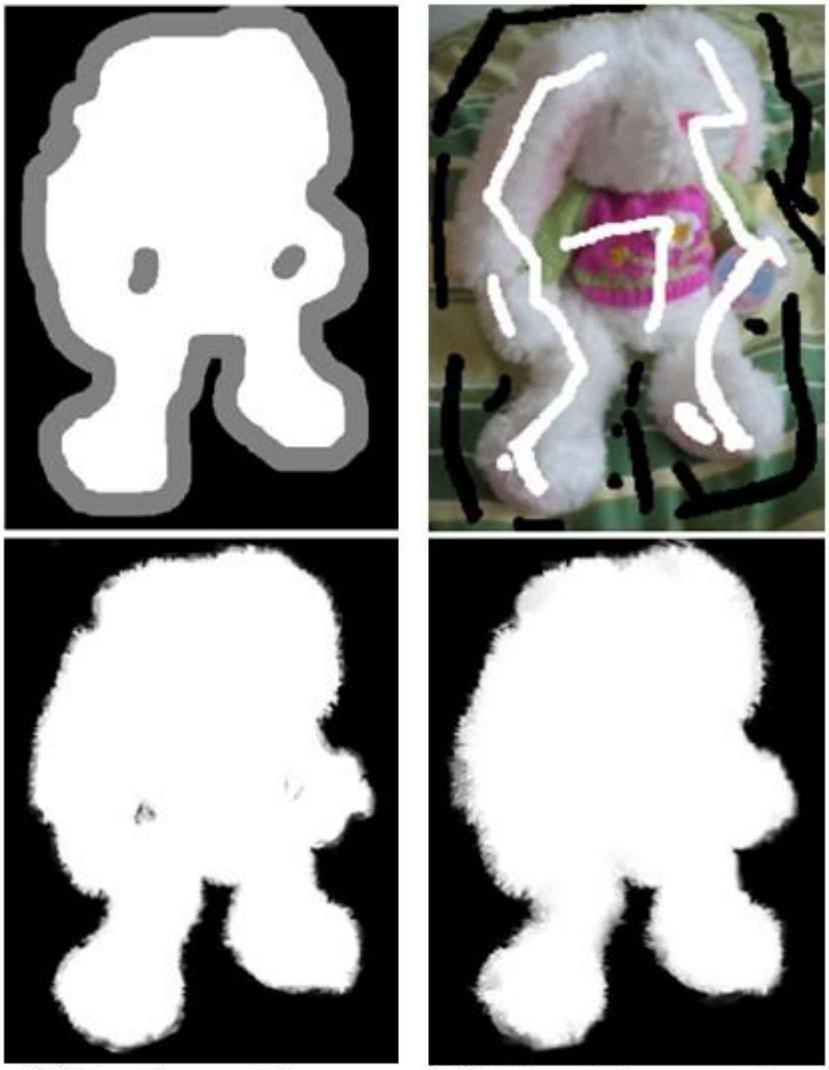


Figure 1-1: A matting example, generated by robust matting [2].

User input plays an important role in almost all of the existing matting methods. If a given image is interpreted according to the compositing Equation 1.1 only, then infinite solutions exist. An obvious solution is $\alpha = 1$, that is $I = F$ which means that whole image is foreground or $\alpha = 0$, that is $I = B$ which means that the whole image is background. Using user input is to take advantage of the user's perception of what is background and foreground and hence, can reduce the solution space. The most common way of user input is to let the user divide the input image into three regions: definite foreground region, definite background region and unknown region. This user specified three-level pixel map is called the *trimap.* One example of a trimap is shown in Figure 1-2. The trimap for the input image in the first row of Figure 1-2(b) is shown in the first row of Figure 1-2(a), where the black region denotes the definite background region, the white region the definite foreground region and the grey region the unknown region. Starting with the trimap, the matting problem is simplified to estimating α, F and B for pixels only in the unknown region based on the information of known foreground and background pixels. The matting result shown in the second row of Figure 1-2(a) is generated by an iterative matting algorithm [3].

Generally speaking, better matting results can be achieved with more accurate trimap since less unknown pixels are needed to be estimated and more known information about the background and foreground are available. A desirable trimap is thus the one that covers as many as possible the definite foreground and background pixels. On the other hand, providing an accurate trimap is a time-consuming process especially when the foreground object has a large number of semi-transparent regions or holes. So a good and practical matting algorithm should take into account the balance between the accuracy of the matte result and the amount of user effort required. Some recently proposed matting methods allow the user to specify only a few foreground and background scribbles which can be considered as a very coarse trimap. Closed-form matting [4] is such an example. As shown in the first row of Figure 1-2(b), the white scribbles specify the definite foreground and the black scribbles specify the definite background and the matting result of closed-form matting is shown in the second row of Figure 1-2(b).



(a) Iterative matting with trimap

(b) Closed-form matting with scribbles

Figure 1-2: Different matting algorithms with different user input.

Starting with the user specified trimap, different matting methods have different ways of utilizing the known foreground and background information and make different assumptions of the image statistics to solve the matting problem. In general, existing matting methods can be roughly divided into three categories: the color sampling based methods [2, 5-8], the alpha propagation based methods [4, 9-12] and the optimization based methods [2-3, 13] which combine both the alpha propagation and color sampling methods. The color sampling based matting methods, which assume color smoothness in a small neighbourhood and sample nearby definite foreground and background pixels to interpolate the alpha value of an unknown pixel. Alpha propagation based matting methods assume that the alpha values of neighbouring pixels are correlated with some local image statistics and propagate the known alpha values into the unknown region. There are also some recently proposed optimization based matting methods which combine both color sampling and alpha propagation techniques.

Current matting methods can achieve good results on relatively uniform and smooth images. However, for complex images, for example if the background contains highly textured regions, the accuracy of these methods degrades significantly. The reason is because the large color variance in the complex background violates the underlying assumptions of both the color sampling and alpha propagation based methods. For the color sampling based methods, due to the large variation of pixel color in the background texture, a small number of background samples from a nearby definite background region may not be sufficient to capture the true background color of the unknown pixels. For the alpha propagation based methods, strong edges in the complex background, some of which may be even stronger than the edges separating the foreground and background will block the alpha values to propagate to the unknown region.

The focus of this thesis is to address the difficulty introduced by complex images in single image matting. We give a detailed analysis of why existing matting methods cannot handle complex images and then propose two new algorithms that have better performance than existing matting methods in handling images with complex background, especially with textured background. To tackle the problem of complex single image matting, we first start with the case where the image background contains homogeneous texture patterns and develop a novel texture synthesis based matting method which can utilize the known texture information to infer the texture information in the unknown region and thus alleviate the problems introduced by textured background. Then for matting with complex image which has heterogeneous texture patterns, we develop a new foreground and background pixels identification algorithm which can identify the pure foreground and background pixels in the unknown region and thus effectively handle the challenges of large color variation introduced by complex images.

Though this thesis is focused on single image matting, it is worth mentioning that there are some extensions of single image matting which use extra information other than the single input image. An example of an early matting system is called blue screen matting [14] which deals with input images that are shot against a constant-colored background. It is easy to infer from the compositing Equation 1.1 that knowing the true constant background color B simplifies the matting problem a lot. More recently, some special imaging systems are designed to use extra information in extracting an accurate matte, such as flash matting [15] in which the input is a pair of flash and non-flash images, matting using a camera array [16] to capture input images from different view-points, video matting [17] which pulls a high-quality alpha matte and foreground from a video sequence, defocus video matting [18] which is a specialized video matting technique using multiple synchronized video streams that share the same point of view but differ in their plane of focus. While utilizing additional information can lead to noticeable improvement of the matte quality, single image matting remains the foundation of all the matting systems. So the contributions made by this thesis to single complex image matting will benefit a broad range of existing matting applications.

Chapter 2:

Background and Related Works

As introduced in the previous chapter, single image matting itself is an ill-posed problem. Even with the user specified trimap, additional assumptions for the image are needed for estimating a high quality matte. Generally, existing matting methods can be divided into three categories according to the assumption they make for an image: color sampling based methods, alpha propagation based methods and the optimization based methods which combine both alpha propagation and color sampling methods. Color sampling based matting methods, which assume color smoothness in a small neighbourhood and sample nearby definite foreground/background pixels to interpolate the alpha value of the unknown pixel, are discussed in section 2.1. Alpha propagation based matting methods assume that the alpha values of neighbouring pixels are correlated with some local image statistics and use such correlation to propagate the known alpha values into the unknown region. Existing alpha propagation based matting methods are reviewed in section 2.2. While pure color sampling based methods and pure alpha propagation based methods each have their own weaknesses, combining these two approaches is expected to improve the matting result than using individual approach alone. Some recently proposed alpha optimization based methods which combine both color sampling and alpha propagation are briefly reviewed in section 2.3.

2.1 Color Sampling Based Matting Methods

Color sampling based matting methods make the local smoothness assumption on image statistics that there exists local correlation between an unknown pixel I_z and its nearby known foreground and background pixels. Usually for each unknown pixel I_z, a set of nearby known foreground and background pixels are sampled and the colors of these samples are assumed to be close to the true foreground and background colors (F_z and B_z) of I_z. Hence, these color samples can be used to build color models to estimate F_z and B_z and compute the alpha value of $\mathrm{I_z}$.

The general idea of color sampling and color model building is quite intuitive; however, implementing a practical matting algorithm that works for general images requires several important questions to be answered. For example, how to define the “neighbourhood” of a pixel? How many samples should be collected for each unknown pixels? How to build reliable color models that can accurately estimate F_z and B_z? Existing color sampling based methods deal with these questions in different ways and we give the details of them in the following sections.

2.1.1 Mishima’s Method

Mishima [7] developed a blue screen matting technique based on representative foreground and background samples. As shown Figure 2-1(a), since a blue screen is used as the controlled background, all the background pixels are assumed to form only one color cluster and are approximated in the color space by a polyhedron (triangular mesh). All the foreground pixels form another polyhedron outside the background one. Then for an unknown pixel, its alpha value is estimated by calculating its relative position to the two polyhedras as shown in Figure 2-1(e).

2.1.2 Knockout

Unlike Mishima's method, the Knockout system [5] works with unconstrained foreground and background. For an unknown pixel I, the system computes its foreground color F_z by extrapolating nearby known foreground colors. As shown in Figure 2-1(b) & (f), F_z is calculated as the weighted sum of nearby foreground color samples, and the weights are proportional to their spatial distances to I. The background color B_z is first calculated in the same way and then refined by considering its relative position to I and F_z. The alpha value for each color channel is estimated individually using the corresponding channel of F_z and B_z. Finally, α_z is estimated as the weighted sum of the three alpha values, where the weight is proportional to the foreground and background difference in the corresponding color channel.

2.1.3 Ruzon and Tomasi's Method

While Mishima's method and the Knockout system use non-parametric sampling scheme, Ruzon and Tomasi proposed a parametric sampling algorithm in 2000 [8]. In their approach, the alpha values are measured along a manifold connecting the "frontiers" of each object's color distribution. The way the "frontiers" is defined and the color model is built are summarized as follows:

(1) A narrow band around the foreground boundary is considered as the skeleton of the unknown region. Some anchor points are selected along the skeleton to divide the unknown band into intervals, as shown in Figure 2-1(c).

(2) A local spatial window is defined for each anchor point which covers a local unknown region, a local foreground region and a local background region.

(3) Non-oriented Gaussian distribution is used as the local foreground and background color models and estimated based on the local foreground and background pixels for each window in the CIE-Lab color space [19].

(4) For each local window, nearby foreground and background Gaussians candidates are connected while rejecting some connections according to certain “intersection” and “angle” criteria.

(5) The observed color of an unknown pixel is assumed to be drawn from a Gaussian distribution, which is modeled as an intermediate distribution of a pair of foreground and background Gaussians. The mean and covariance of the intermediate distribution is linearly interpolated by a pair of foreground and background Gaussian distributions and the alpha value is estimated according to the linear weight, as shown in Figure 2-1(g). The best alpha value is defined as the one that corresponds to the intermediate distribution for which the observed color has the maximum probability.

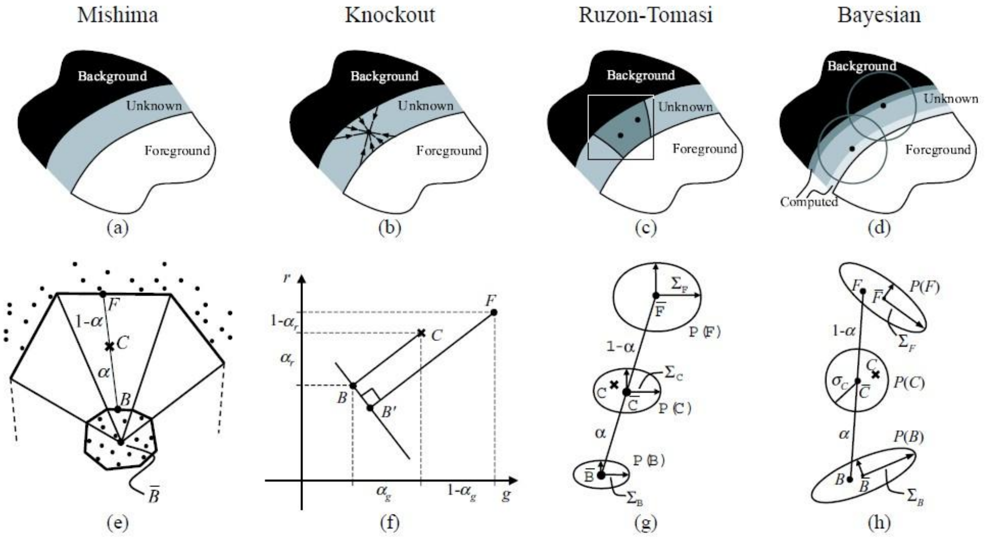


Figure 2-1: Illustration of the color models of different matting methods.

The main drawback of Ruzon and Tomasi’s method is that using non-oriented Gaussians as a local color model works well only when the local region has small color variance. Local regions with large color variance will result in large fitting errors. Also, the alpha value for each unknown pixel is estimated independently, which could make the final alpha matte not smooth enough.

2.1.4 Bayesian Matting

Bayesian matting proposed by Chuang et al. in 2001 [6] uses a similar idea of using the Gaussian distribution as local color models as Ruzon and Tomasi's algorithm, but also includes some improvements on the main drawbacks of Ruzon and Tomasi's method discussed above. First, a sliding window is used to collect neighbouring pixels for each unknown pixel as shown in Figure 2-1(d). Second, not only the pixels in the local foreground and background windows are used to estimate the Gaussian distributions, but also the estimated values of Fs, Bs and αs of the nearby pixels are used in the estimation of the current Gaussian distribution. Finally, the alpha estimation is formulated as a maximum a posterior (MAP) problem which is a well studied technique under the Bayesian framework in statistics. Mathematically, for an unknown pixel I_z, α_z, F_z and B_z are estimated by

$$\begin{aligned} & arg \max_{F_z,B_z,\alpha_z} P(F_z, B_z, \alpha_z | I_z) \\ & = arg \max_{F_z,B_z,\alpha_z} L(I_z|F_z, B_z, \alpha_z) + L(F_z) + L(B_z) + L(\alpha_z) \end{aligned} \tag{2.1}$$

where $L(\cdot)$ is the log likelihood $L(\cdot) = logP(\cdot)$. The first term is measured as

$$L(I_z|F_z, B_z, \alpha_z) = -\|I_z - \alpha_z F_z - (1 - \alpha_z B_z)\|^2 / \sigma_z^2 \tag{2.2}$$

where the color variance σ_z is computed in the local window. This term is simply the estimation residual according to the compositing equation and regularized with the local color variance. $L(F_z)$ is estimated as the probability of being drawn from a local foreground Gaussian distribution. The foreground pixels in the nearby definite foreground region are collected to estimate an oriented Gaussian with mean $\bar{F}$ and covariance $\sum F$. $L(F_z)$ is then defined as

$$- (F_z - \bar{F})^T \Sigma_F^{-1} (F_z - \bar{F})/2 \tag{2.3}$$

$L(B_z)$ is calculated in the same way by using background samples as shown in Figure 2-1 (h). $L(\alpha_z)$ is treated as a constant. Equation 2.1 is solved by iteratively estimating F_z, B_z and α_z using the following steps:

(1) Fix α_z to solve for F_z and B_z as

$$\begin{bmatrix} \Sigma_F^{-1} + I\alpha_z^2/\sigma_z^2 & I\alpha_z(1-\alpha_z)/\sigma_z^2 \\ I\alpha_z(1-\alpha_z)/\sigma_z^2 & \Sigma_B^{-1} + I(1-\alpha_z)^2/\sigma_z^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \Sigma_F^{-1}\bar{F} + I_z\alpha_z/\sigma_z^2 \\ \Sigma_B^{-1}\bar{B} + I_z(1-\alpha_z)/\sigma_z^2 \end{bmatrix} \tag{2.4}$$

where I is a 3*3 identity matrix.

(2) Fix F_z and B_z to solve for α_z as

$$\alpha_z = \frac{(I_z - B_z)(F_z - B_z)}{\|F_z - B_z\|^2} \tag{2.5}$$

When there are multiple foreground or background clusters, this optimization is performed for each pair of foreground and background clusters and the pair which gives the maximum likelihood is chosen.

When the input image has a uniform background and a fine trimap is provided, Bayesian matting can generate quite accurate alpha mattes. However, for a complex image, the underlying assumption of Bayesian matting is violated and the results tend to be very noisy. For instance, if the input image has a highly textured background, using Gaussians distribution to model the local colors which has a large color variance is not suitable. Also, if the trimap is coarse, for the unknown region which is far away from the definite foreground and definite background regions, the correlations between the unknown pixels and the foreground and background samples are weak. Hence, large estimation errors will result for terms $L(F_z)$ and $L(B_z)$ in Equation 2.1. More detailed analysis is presented in section 3.1.

2.1.6 Optimized Color Sampling in Robust Matting

All the color sampling methods discussed so far use all the pixels in a local window as color samples to build a local color model. However, when the image has complex foreground and/or background patterns, some local windows will have large color variance and it is more often the case than not that only a small number of pixels in the local window have strong relation with the unknown pixels. Selecting “good” samples to estimate the alpha values of the unknown pixels is then vital in getting high quality matte, which inspires an optimized color sampling procedure called robust matting [2].

The main idea of robust matting is to select “good” sample pairs which can explain the color of the unknown pixels as a linear combination of themselves. Specifically, as shown in Figure 2-2(b), for a foreground and background pixel pair (F^i,B^j), a *distance ratio* $R_d(F^i, B^j)$ is defined as the ratio of between (1) the interpolation residue (that is the distance between the unknown pixel color, I_z , and the color it would have, $\hat{I}$, predicted by the linear model in Equation 1.1), and (2) the distance between the foreground/background pair:

$$R_d\left(F^i, B^j\right) = \frac{\left\| I_z - (\widehat{\alpha_z} F^i + (1 - \widehat{\alpha_z}) B^j) \right\|}{\| F^i - B^j \|} \tag{2.6}$$

where $\widehat{\alpha_z}$ is computed as in Equation 2.5. In the example shown in Figure 2-2(b), the distance ratio will be much higher for pair (F_1, B_1) than for pair (F_2, B_2) indicating that the latter is a better choice for estimating the alpha value for I_z. The distance ratio alone will favor sample pairs that are widely separated in the color space, in which case the denominator $\left\| F^i - B^j \right\|$ will be large. Since most pixels are expected to

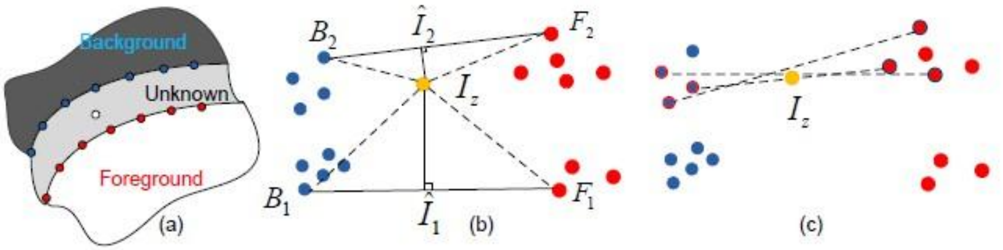


Figure 2-2: Color sampling scheme for robust matting.

be fully foreground or background, pixels with colors that lie nearby in color space to the foreground and background samples are more likely to be fully foreground or background themselves. Thus, for each individual sample two more weights $w(F^i)$ and $w(B^j)$ are defined as $w\left(F^i\right) = 1.0 - \exp\{-\left\|F^i - I_z\right\|^2 / D_F^2\}$ and $w\left(B^j\right) = 1.0 - \exp\{-\left\|B^j - I_z\right\|^2 / D_B^2\}$, where D_F^2 and D_B^2 are, respectively, the minimum distances between foreground/background sample and the current pixel, i.e., $\min_i \left\|F^i - I_z\right\|$ and $\min_i \left\|B^j - I_z\right\|$.

Combining these factors, the final *confidence* value $f(F^i, B^j)$ for a sample pair is defined as

$$f\left(F^i, B^j\right) = \exp\left\{-\frac{R_d(F^i, B^j)^2 \cdot w\left(F^i\right) \cdot w\left(B^j\right)}{\sigma^2}\right\} \qquad (2.7)$$

where σ is fixed to be 0.1.

For each unknown pixel, the confidence value for all the foreground and background sample pairs are computed as in Equation 2.7. The estimated alpha values of the three pairs with the highest confidence are selected and the average alpha value is used as the initial guess of the alpha value for the unknown pixel, which will be further adjusted by an optimization process. To effectively accommodate for the high color variances in the background and foreground regions, a relatively large number of foreground and background pixels are sparsely selected along the boundary of the unknown region to form the sample set, as shown in Figure 2-2(a).

2.1.5 Global Color Models

All the color sampling schemes introduced above can be considered as local color sampling in the sense that they assume that the unknown region is a narrow band between the definite foreground and definite background boundaries and collect nearby samples to approximate local color models. This assumption will not hold if the trimap provided by the user is very coarse or consists of just a few paint strokes, where the majority of unknown pixels are very far away from the known foreground and background samples and have very weak correlations with them. To tackle this problem, some global color sampling methods are proposed recently which can estimate relatively good mattes with roughly specified trimaps. For example, the iterative matting approach [3] estimates the global Gaussian Mixture Models (GMMs) for definite foreground and background colors, and for an unknown pixel, samples are drawn from all the Gaussians in the hope to cover all the possibilities that its foreground/background color could have. The geodesic matting approach [9] also uses a mixture of Gaussians to model the global foreground and background color distributions in the CIE-Luv space, and fast kernel density estimation methods [20] are used to reduce the computational complexity of constructing PDFs. However, using the GMMs as the global color models for complex foreground and background suffers from the problems of insufficient samples and large estimation errors and the performance of these global color sampling methods are not significantly better than the local color sampling methods.

2.1.7 Summary

Exploiting the local smoothness characteristic of natural images by sampling nearby definite foreground and background pixels to estimate the alpha values of unknown pixels is intuitive and proves to be quite effective for solving the otherwise ill-posed matting problem. For those color sampling based methods introduced in this chapter to work well, their assumption on the local smoothness must hold and the trimap provided should be of sufficient details. However, as is discussed in Chapter 3, when the input image becomes complex, the large color variance presented in the image violates the smoothness assumptions of these methods and their performance will degrade significantly.

2.2 Alpha Propagation Based Matting Methods

As discussed in the previous section, in color sampling based methods, the information provided by nearby foreground and background samples becomes less reliable in a complex scene where the color variance is large. To avoid this problem some recently proposed matting approaches have taken a different way than estimating the true background and foreground of unknown pixels. Instead, these methods directly model the affinities between the alpha values of neighbouring unknown pixels by using some intrinsic local image statistics. The alpha values of the unknown pixels are then solved by propagating the known alpha values into the unknown regions using these alpha affinities.

The alpha propagation based approaches have two major advantages when compared with color sampling-based approaches. First, affinities are always defined in a small neighbourhood, usually between immediately connected pixels or pixels in a 3 × 3 window. In such a small window, the correlations among pixels are usually strong and thus the local smoothness assumption typically holds, even for moderately complex images. On the contrary, when the input trimap is coarse, in sampling-based approaches, samples which are collected far from the target pixel, may or may not be useful at all. Also, the defined affinities enforce the resulting matte to be locally smooth, and thus fundamentally can avoid matte discontinuities from which sampling-based approaches suffer.

2.2.1 Poisson Matting

With the assumption that the intensity of image foreground and background are locally smooth, Poisson matting [12] employs a simple linear relation between the change of image intensity and the change of alpha value. The derivation is briefly described as following. By taking the partial derivatives on both sides of the matting equation 1.1, one gets

$$\nabla I_z = (F_z - B_z)\nabla\alpha_z + \alpha_z\nabla F_z + (1-\alpha_z)\,\nabla B_z \qquad (2.8)$$

where $\nabla = \left(\frac{\partial}{\partial \mathrm{x}}, \frac{\partial}{\partial \mathrm{y}}\right)$ is the gradient operator. By assuming that F_z and B_z are locally smooth, $\alpha_z\nabla F_z + (1-\alpha_z)\,\nabla B_z$ is relatively small compared with $(F_z - B_z)\nabla\alpha_z$, and can be omitted in Equation 2.8. Then the matte gradient can be approximated as

$$\nabla\alpha_z = \frac{1}{F_z - B_z}\nabla I_z \qquad (2.9)$$

It simply means that the matte gradient is proportional to the gradient of image intensity. To estimate the absolute gradient value, $F_z - B_z$ needs to be estimated first. In the system, F_z and B_z are simply chosen as the nearest foreground and background colors for the unknown pixel.

The final matte is then constructed by solving Poisson equations on the image lattice as:

$$\alpha^* = arg\min_{\alpha} \int\int_{z\in\Omega} \left\| \nabla\alpha_z - \frac{1}{F_z - B_z}\nabla I_z \right\|^2 dz \qquad (2.10)$$

with the Dirichlet boundary condition which is consistent with the user provided trimap. Ω is the unknown region in the trimap. Solving the Poisson equation is a well studied problem and in the original paper, a solver using Gauss-Seidel iteration with over-relaxation is used.

If local smoothness assumption made for the image holds, then transforming and simplifying the matting problem into solving the Poisson equation is technically sound. However, once the image consists of complex scenes and the local smoothness assumption is violated, the neglected term $\alpha_z \nabla F_z + (1 - \alpha_z)\,\nabla B_z$ is no longer negligible and sampling nearest known foreground and background pixels to estimate F_z and B_z will also introduce large errors in the whole derivation and hence, noticeable noise and errors in the final matte result are commonly observed. To alleviate the above mentioned problems a set of local filters and operations are defined in the proposed system, which enables the user to manually correct the final matte by solving the local Poisson equation. Although good results can be achieved in this way, it is often a time-consuming process for the user and the desired results are not always easy to achieve.

2.2.2 Random Walk Matting

A common smoothness term, also known as affinity, used in many spectral image segmentation approaches [21] is defined as:

$$\mathrm{w_{ij}} = \exp(-\frac{\left\| I_i - I_j \right\|^2}{\sigma^2}) \qquad (2.11)$$

where σ is a free parameter which is set by the user according to experience or is adjusted automatically based on some local image statistics. The random walk matting system [10] uses a generalized form of the affinity defined in Equation

2.11 with the modification that color distances are not measured in the original RGB space, but in the channels created by using Local Preserving Projections (LPP) techniques [22]. The projections defined by the LPP algorithm are given by solving the following generalized eigenvector problem:

$$\mathrm{ZL}Z^T x = \lambda Z D Z^T x \tag{2.12}$$

where Z is the $3\times N$ matrix with each I_i as a column, D is the diagonal matrix defined by $D_{ii} = d_i$ and L is the graph Laplacian matrix given by

$$L_{ij} = \begin{cases} d_i & : \quad if \;\; i = j \\ -W_{ij} & : \quad if\ i\ and\ j\ are\ neighbors \\ 0 & : \quad otherwise \end{cases} \tag{2.13}$$

The solution of the generalized eigenvector problem of (2.12) is denoted as Q, where each eigenvector is a row of Q. The final alpha affinity then is defined as

$$w_{ij}^{*} = \exp\left(- \frac{\left(I_i - I_j\right)^T Q^T Q (I_i - I_j)}{\sigma^2} \right) \tag{2.14}$$

It is shown in the original paper that the alpha propagation works better in the LPP-projected space than in the original RGB color space in discriminating the boundaries between foreground and background regions. This work is the first to bring up the question of which color space is more suitable for the matting problem, since most of the existing approaches apply their analysis in the RGB space.

With the alpha affinity defined in Equation 2.14 and the specified alpha value for the definite foreground and background pixels according to the trimap, the final alpha values of unknown pixels can then be solved by a random walk algorithm. Given an unknown pixel, its alpha value is defined to be the probability that a random walker starting from this location will reach a pixel in the foreground before reaching the background, when biased to avoid crossing the foreground boundary. It is shown that these probabilities can be calculated exactly by solving a system of linear equations.

2.2.3 Geodesic Matting

Instead of calculating the probability that a random walker will reach the foreground first starting from an unknown pixel, the geodesic matting approach [9] measures the weighted geodesic distance that a random walker will travel from its origin to reach the foreground. In this approach the geodesic distance computation is linear in time and has minimal memory requirements. Hence, it allows the system to achieve fast and high-quality segmentation and matting using a few user scribbles, when the affinities between pixels are assigned properly.

Mathematically, the geodesic distance $d(i, z)$ is simply the smallest integral of a weighted function over all the paths on the image lattice from pixel I_i to pixel I_z, defined as

$$d(i,z) = \min_{C_{i,z}} \int_0^1 |W \cdot \dot{C}_{i,z}(p)| dp \tag{2.15}$$

where $\dot{C}_{i,z}(p)$ is a path connecting the pixels i, z (for p = 0 and p = 1 respectively). The weight W is set to be the gradient of the likelihood that a pixel belongs to the foreground i.e., $W = \nabla P_F(x)$. To compute this likelihood, the user-specified foreground and background pixels are used to train a mixture of Gaussians using fast kernel density estimation methods [20], resulting in the foreground probability distribution function (PDF) $P(x/F)$ and background PDF $P(x/B)$, and $P_F(x)$ is set to be $\frac{P(x|F)}{P(x|F)+P(x|B)}$.

The geodesic distance from an unknown pixel I_z to the foreground is defined as $D_F(z) = \min_{i\in\Omega_\mathrm{F}} d(i,z)$, and its distance to the background is defined in a similar way.

Finally, the alpha value is estimated as:

$$\alpha_z = \frac{W_F(z)}{W_F(z)+W_B(z)} \tag{2.16}$$

where $W_F(z) = D_F(z)^{-r} \cdot P_F(z)$, is the locally adjusted foreground weight. The idea is to combine the geodesic distance $D_F(z)$ with the locally recomputed foreground probability. The parameter *r* controls the smoothness of the edges. The background weight $W_B(z)$ is computed in a similar way.

The major advantage of this approach is that the formulation is based on weighted distance functions (geodesics), which can be solved as a first order geometric Hamilton-Jacobi equation in computationally optimal linear time. This is particularly favorable for video matting where computational complexity remains to be a serious issue.

The disadvantage of the proposed system is that the weight *W* is set in a rather simple way, and will not work well when the foreground and background color distributions have large overlaps, where the PDFs *P*(*x/F*) and *P*(*x/B*) cannot be estimated properly. However, the proposed geodesic-distance-based matting framework is quite general. Hence, the PDF estimation step could be potentially improved by using more sophisticated discriminate models when dealing with complex scenes.

2.2.4 Closed-form Matting

As shown in the previous sections, Poisson matting uses nearest known foreground background pixels to estimate $F_z - B_z$ in Equation 2.9, and geodesic matting uses user-specified foreground and background pixels to train global color models, both of which are similar in the spirit to the color sampling scheme and thus have the same inherent problem of color sampling based methods that the sampling based estimation may not be accurate and robust. The recently proposed closed-form matting approach [4] can be considered to be a "pure" alpha propagation method that avoids the sampling and estimation step by explicitly deriving a cost function from local smoothness assumptions on foreground and background colors *F* and *B*, and show that in the resulting expression it is possible to analytically eliminate *F* and *B*, yielding a quadratic cost function only in α, which can be easily solved as a sparse linear system of equations.

The underlying assumption made in closed-form matting is the color line model [23], which assumes that the true foreground and background colors of each pixel is a linear mixture of two constant foreground and background colors *F* and *B* over a small window (typically 3 ×3 or 5× 5) centered at that pixel. Under this assumption it is then shown that the alpha values in a small window w can be expressed as

$$\alpha_i = \sum_c a^c I_i^c + b, \forall i \in w \tag{2.17}$$

where c refers to the color channels, and a^c and b are constants in the window. The matting cost function is then defined as

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - \sum_c a_j^c I_i^c - b_j)^2 + \epsilon \sum_c {a_j^c}^2 \right) \tag{2.18}$$

which is the overall prediction error of the color line model plus a regularization term. Furthermore, a^c and b can be eliminated from the cost function, yielding a quadratic cost in α alone:

$$J(\alpha) = \alpha^T L\alpha \qquad (2.19)$$

where L is an $N \times N$ matrix, whose (i, j)-th element is:

$$\sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + (I_i - \mu_k)(\Sigma_\mathrm{k} + \frac{\epsilon}{|w_k|} I_3)^{-1}(I_j - \mu_k))) \qquad (2.20)$$

where Σ_k is a 3×3 covariance matrix, μ_k is a 3×1 mean vector of the colors in a window w_k, and I_3 is the 3 × 3 identity matrix.

The matrix L, which is called the *matting Laplacian*, is the most important analytic result from this approach. The optimal alpha values are then computed as

$$\alpha = arg\ min\ \alpha^T L\alpha, \qquad s.t.\ \alpha_i = 1\ or\ 0, \qquad \forall i \in \partial\Omega \qquad (2.21)$$

which is essentially a problem of minimizing a quadratic error score, and thus can be obtained by solving a linear system.

By comparing the affinity defined in Equation 2.20 with the one defined in Equation 2.11 we can see that they both have the same property that nearby pixels with similar colors tend to have higher affinity values, while nearby pixels with larger color differences tend to have smaller affinity values. However, the matting affinity in Equation 2.20 is more powerful than the one defined in Equation 2.11 because in Equation 2.11 the scaling parameter σ is a global constant while in Equation 2.20 more localized parameters (the covariance Σ_k and mean μ_k) are used. As a result, this localized adaptive setting leads to a significant improvement in performance, as demonstrated in [4].

2.2.5 Spectral Matting

Further analysis has been conducted on the proposed matting Laplacian in Equation 2.20, resulting in an automatic matting approach called *spectral matting*

[11]. This is the only approach that tries to pull out a foreground matte in a completely automatic fashion.

In this approach the input image is modeled as a convex combination of K image layers as

$$I_z = \sum_{k=1}^{K} \alpha_z^k F_z^k \tag{2.22}$$

where F_z^k is the kth matting component of the image. The most important conclusion from this approach is that the smallest eigenvectors of the matting Laplacian L span the individual matting components of the image, thus recovering the matting components of an image is equivalent to finding a linear transformation of the eigenvectors. Detailed steps are as follows:

(1) Compute the eigenvectors of L as $E = [e^1, \ldots, e^K]$, so E is a $N \times K$ matrix (N is the total number of pixels);

(2) Initialize α^k by applying a k-means algorithm on the smallest eigenvectors, and project the indicator vectors of the resulting clusters onto the span of the eigenvectors E:

$$\alpha^k = EE^T m^{c^k} \tag{2.23}$$

(3) Compute the matting components by minimizing an energy function defined as

$$\sum_{z,k} |\alpha_z^k|^r + |1 - \alpha_z^k|^r, where\ \alpha^k = Ey^k \tag{2.24}$$

subject to $\sum_k \alpha_z^k = 1$. γ is chosen to be 0.9 for a robust measure.

(4) Group components into the final foreground matte by testing various combinations of matting components and computing the corresponding cost as $J(\alpha) = \alpha^T L\alpha$. To do this more efficiently the correlations between the matting components via L are pre-computed as

$$\Phi(k, l) = (\alpha^k)^T L\alpha^l \qquad (2.25)$$

and the matting cost can be computed as $J(\alpha) = b^T \Phi b$, where b is a K-dimensional binary vector indicating the selected components.

(5) When user's input is provided, the grouping process can take advantage of it by solving a graph-labeling problem using the min-cut algorithm. Details can be found in [11].

The spectral matting approach derives an analogy between hard spectral image segmentation and image matting, and thus provides a very interesting theoretical result. This work is a milestone in theoretic matting research. However, in practice, this approach is limited to images that consist of a modest number of visually distinct components, as pointed out by the authors. For these images, many other approaches can be used to generate higher quality mattes, although more user inputs are required. Furthermore, given the fact that the size of E is $N \times K$, the memory consumption of this approach is very high, and hence, the practical application range of this approach is limited.

2.2.6 Summary

Alpha propagation based methods generally produce smoother alpha matte and are more robust compared to color sampling based methods because the local smoothness assumption is usually satisfied in a small window, though the performance of different alpha propagation methods vary with their specific derivation of the alpha affinities. Poisson matting makes strong assumptions on the smoothness of foreground and background colors, and will introduce significant errors when dealing with complex scenes. The closed-form matting derives the affinity by conducting very insightful analysis on the theoretical aspects of the matting problem, and has significantly better performance than other alpha propagation based approaches.

There are also some drawbacks of alpha propagation based methods. First, unlike sampling-based approaches, most approaches focus on first estimating the alpha values, and only then estimate true foreground colors for unknown pixels based on pre-computed alphas, rather than estimating them jointly for an optimal solution. Secondly, the alpha matte is estimated in a propagation fashion, from known pixels to unknown ones, thus small errors could be propagated and accumulated to produce bigger errors. And when there are strong edges in the unmarked background or foreground region, the propagation will be blocked because of the local smoothness assumption is violated, which is demonstrated in Chapter 3.

2.3 Alpha Optimization by Combining Color Sampling and Alpha Propagation

The Markov Random Fields (MRFs) [24-25] style energy function are quite popular in many recently proposed optimization-based computer vision and graphics systems [21, 26-28], which is defined by two terms:

$$E = E_d + \lambda E_s \tag{2.26}$$

The first term on the right side E_d is usually called the data term, which represents the semantic goal of the optimization problem. For example, in the object segmentation problem, this term might enforce that pixels whose colors are closer to the known object colors and further away from background colors should be classified as foreground. The data term is usually the sum of a set of per-pixel data costs $d_p(l)$, $E_d = \sum_p d_p(l)$, where l is a possible label that the pixel p can be assigned to. The second term E_s is called the smoothness term, which encourages the preservation of smoothness of some image statistics between neighbouring pixels. Usually E_s is defined as the sum of spatially varying horizontal and vertical nearest-neighbour smoothness costs, $V_{pq}(l_p, l_q)$, $E_s = \sum_{\{p,q\}\in\varkappa} V_{pq}(l_p, l_q)$, where $\varkappa$ denote all the neighbouring pixels pair according to a predefined neighbouring system. Once the energy function is defined, a variety of optimization tools can be employed to minimize it in closed-from or approximately.

For the matting task, the form of the MRFs energy provides a natural way of unifying the color sampling approaches and the alpha-propagation approaches into an optimization framework. Intuitively, the sampling techniques discussed in Chapter 2.1 are capable of analyzing the distances between an unknown pixel and the known foreground and background colors. Hence, they can be used to assign data costs to pixels as the semantic constraint. The affinities defined in Chapter 2.2 represent the relationships between nearby pixels, which can be employed to set smoothness costs for neighbouring pixel pairs. By combining different sampling methods and affinities in a single optimization process, more accurate and robust matting solutions are expected.

2.3.1 Iterative Matting

The iterative matting approach [3] is the first to formulate the matting task as a MRFs style optimization problem. Starting with sparse user inputs such as a few foreground and background paint scribbles, the alpha values of those unknown pixels in a narrow band around the frontier of the definite foreground region is estimated by solving an optimization problem and by iteratively extending the foreground frontier the alpha values are gradually propagated from the known pixels to the unknown ones. In each iteration, the energy function to be minimized is defined as

$$E = \sum_{z \in \psi} E_d(\alpha_z) + \lambda \cdot \sum_{z,v \in \psi} E_s(\alpha_z, \alpha_v) \tag{2.27}$$

The data cost $E_d(\alpha_z)$ is constructed using the color sampling method. The alpha values are discretized into K levels and the likelihood $L_k(z)$, which is the probability of pixel z with alpha level k, is estimated based on nearby known and previously estimated alphas. The data cost is then defined for each of the possible states α_z^k *as*

$$E_d(\alpha_z^k) = 1 - \frac{L_k(z)}{\sum_{k=1}^{K} L_k(z)} \tag{2.28}$$

The neighbourhood cost $E_s(\alpha_z, \alpha_v)$ is defined using the classic affinity in Equation 2.11, as

$$E_s(\alpha_z, \alpha_v) = 1 - \exp(-(\alpha_z - \alpha_v)/\sigma_s^2) \tag{2.29}$$

where σ_s is set to be 0.2 empirically.

With the MRF defined above, finding a labeling, which means computing the α level for each pixel with minimum energy corresponds to the MAP estimation problem, is solved by using the loopy belief propagation (BP) algorithm [29].

This approach opens a new avenue for matting research to exploring ways to significantly reduce user’s efforts that are traditionally required for creating accurate trimaps, especially for images with large portions of semi-transparent foreground where the trimap is difficult, if not impossible, to create accurately.

However, it also presents two major limitations. The global color sampling scheme is used to guide the matte propagation, which requires the foreground and background to have distinct, well separable color distributions. Furthermore, the expensive non-linear belief propagation optimization process is employed multiple times to create a matte, which could converge to undesirable local minima. The required processing time of this approach is usually very long, which is undesirable in an interactive setting.

2.3.2 Easy Matting

In the easy matting system [13], the energy function to be minimized is defined as:

$$E = \sum_{z \in \psi} \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\left\| I_z - \widehat{I_z} \right\|^2}{\sigma_z^2} + \lambda \cdot \sum_{z \in \mathbb{N}(z)} \frac{(\alpha_z - \alpha_v)^2}{\| I_z - I_v \|} \right) \qquad (2.30)$$

where N is the number of pixels, $\mathbb{N}(z)$ defines the neighbourhood of z, and $\left\| I_z - \widehat{I_z} \right\|^2$ is the estimation residue for pixel z according to the compositing equation. Both the data term and the neighbourhood term are designed in a similar way as in the iterative matting approach [3], except that no exponential mappings are employed. This greatly simplifies the optimization process because Equation 2.30 is a quadratic function, and the energy function can be easily minimized by solving a set of linear equations using the conjugate gradient method.

Another improvement made in this approach over previous ones is that the weight λ in Equation 2.30 is dynamically adjusted rather than manually fixed, as

$$\lambda = e^{-(k-\beta)^3} \qquad (2.31)$$

where k is the iteration count and β is an pre-defined constant which is set to be 3.4 in the system. In early iteration, the value of λ is large, which emphasizes more on the neighbourhood term, and hence encourages the alpha values in the foreground and background regions to smoothly spread out. Later on when the propagation of alpha encounter the object boundary where discontinuity exists, the iteration count has increased so that the value of λ is decreased, in which case the data term plays a more important role in estimating the alpha values. This dynamic weight setting helps the iterative algorithm avoid stepping into bad local minima in early stages.

2.3.3 Robust Matting

The robust matting method [2] uses the optimized color sampling scheme described in Section 2.1.6 as the data term and the matting Laplacian derived in closed-form matting as the smoothness term, resulting in a well-balanced system capable of generating high quality results while maintaining a reasonable degree of robustness against different user inputs. The energy function to be minimized in this approach is defined as

$$E = \sum_{z \in \psi} \left[\widehat{f_z} (\alpha_z - \widehat{\alpha_z})^2 + (1 - \widehat{f_z})(\alpha_z - \delta(\widehat{\alpha_z} > 0.5))^2 \right] + \lambda \cdot J(\alpha, a, b) \quad (2.32)$$

where $\widehat{\alpha_z}$ and $\widehat{f_z}$ are, respectively, the estimated alpha and the confidence value in the color sampling step as described in Section 2.1.6, and $J(\alpha, a, b)$ is the neighbourhood energy defined in Equation 2.18, where the parameters a and b can be analytically eliminated in the optimization process.

In [2], minimizing the energy function defined in Equation 2.32 is interpreted as solving a corresponding graph labeling problem as shown in Figure 2-3, where Ω_F and Ω_B are virtual nodes representing pure foreground and pure background. The white nodes represent unknown pixels on the image lattice. The light red nodes represent the foreground pixels and light blue nodes the background pixels marked by the user. A data weight is defined between each pixel and a virtual node to enforce the data constraint, and an edge weight is defined between two neighbouring pixels to enforce the neighbourhood constraint.

Numerically, similar to the closed-form matting approach [4], the energy function to be minimized is defined as a quadratic function in α_z, and can be solved using a linear system solver. The Laplacian matrix in the linear system is defined as

$$L_{ij} = \begin{cases} W_{ii} : & if\ \ i = j \\ -W_{ij} : & if\ i\ and\ j\ are\ neighbors \\ 0\ \ : & otherwise \end{cases} \tag{2.33}$$

where $W_{ii} = \sum_j W_{ij}$. L is a sparse, symmetric, positive-definite matrix with dimension $N \times N$, where N is the number of nodes in the graph, including all the pixels in the image plus two virtual nodes Ω_B and Ω_F . W_{ij} is exactly the same as the one defined in Equation 2.20 if i and j are neighbouring pixels; otherwise W_{ij} is equal to the data cost $W_{i,F}$ or $W_{i,B}$ if j is a virtual node.

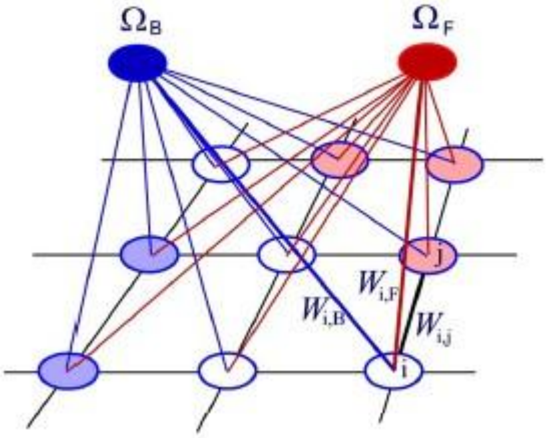


Figure 2-3: Matting is formulated as solving a graph labeling problem in robust matting.

Note that the confidence value $\widehat{f_z}$ plays an important role in balancing the data cost and the neighbourhood cost in Equation 2.32. This is motivated by the fact that color sampling will not be always reliable for every pixel and bad estimations are typically associated with low confidence values. Hence, using the confidence value to tone down incorrect data costs and let neighbourhood costs take over for those pixels will produce better mattes with less noise.

2.3.4 Summary

Both color sampling based methods and alpha propagation based methods have their own advantages and disadvantages. Color sampling based methods work better when the foreground and background color distribution have less overlap in the color space and tightly specified trimaps are provided, but generate noisy mattes if the underlying assumption is violated. On the other hand, alpha propagation based methods are more robust to coarsely specified trimaps and generate smoother mattes, but may not be as accurate as color sampling based methods when the local color variation is large even with distinct foreground and background color distributions. By combining these two methodologies into a unified optimization framework, the advantages of these two methodologies are utilized and a good balance between accuracy and robustness is achieved. However, when the image is complex, even the state-of-the-art combined matting methods give poor quality results because the underlying assumptions are severely compromised due to the large color variation of the complex foreground and background, which are discussed in the next chapter.

Chapter 3:
Complex Image Matting

3.1 Motivation

Though current matting methods can get good results on relatively uniform and smooth images, the accuracy of these methods degrades significantly when the images are more complex. Figure 3-1 shows one such example. The input image is a composite image with textured background. Three powerful matting methods are tested and none of them can generate a satisfactory alpha matte. Figure 3-2 shows another example which is taken from [30]. In the image the doll is taken against a real background scene which contains complex texture patterns. The matting results of four top ranking matting methods in [30] are shown in Figure 3-2 and we can see that the alpha mattes generated are very noisy.

The reason for the poor matting result shown in Figure 3-1 and Figure 3-2 is because the highly textured background violates the underlying assumptions of both the color sampling and alpha propagation based methods. For the color sampling based methods, due to the large variation of pixel color in the background texture, a small number of background samples from a nearby definite background region may not be sufficient to capture the true background color of the unknown pixels. For the alpha propagation based methods, strong edges in the complex background, some of which may be even stronger than the edges separating the foreground and background, will block the alpha values to propagate to the unknown region.

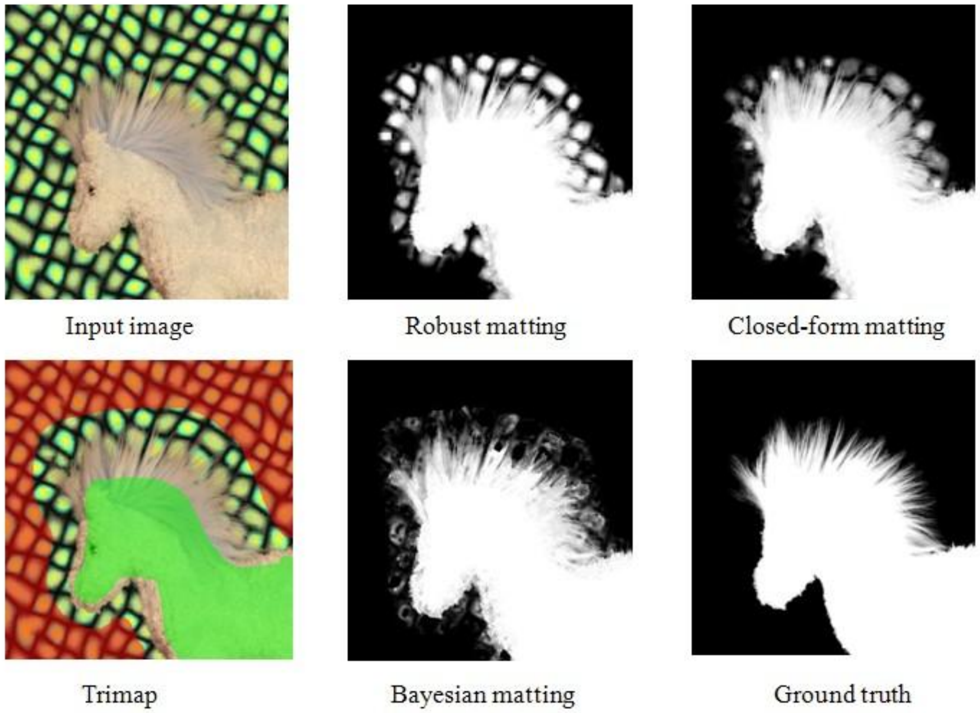


Figure 3-1: Poor alpha estimated due to textured background.

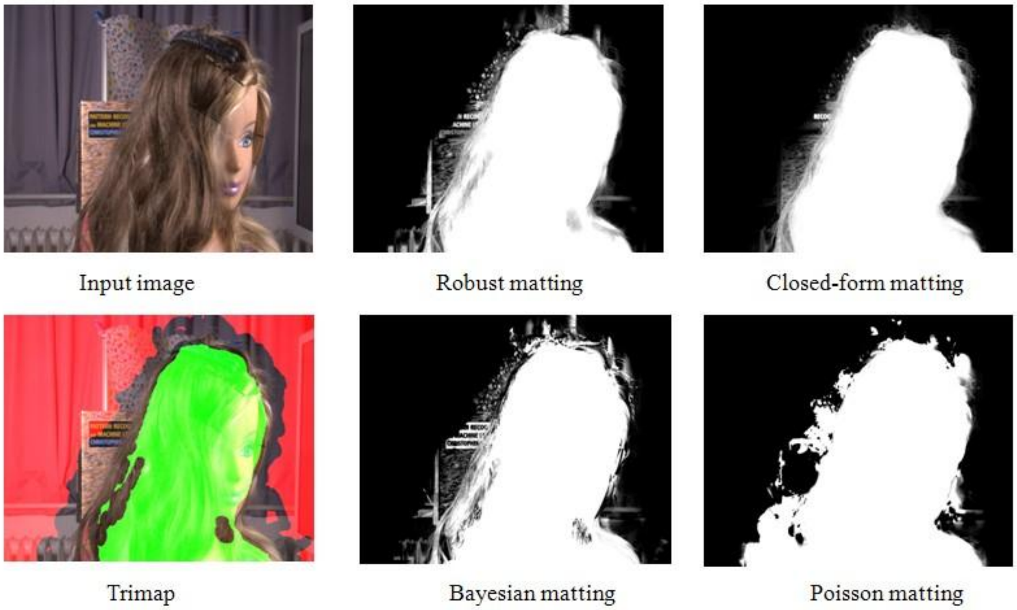


Figure 3-2: Poor alpha estimated due to complex background.

In order to gain better understanding of the challenges posed by complex background, we present in this section a detailed analysis of the failure cases of some existing matting methods for complex images as shown in Figure 3-1 and Figure 3-2. We use two popular matting methods as illustration in our analysis: robust matting [2] and closed-form matting [4], which are, respectively, the representatives of the color sampling based matting methods and alpha propagation based matting methods discussed in Chapter 2, and are recognized in the research community to produce high quality matting results. Hence, we believe that our analysis is general enough to reveal the fundamental difficulties of most, if not all, matting methods when dealing with complex background.

Color sampling based matting approaches collect nearby foreground and background sample pixels to interpolate pixels in the unknown region. When the background is uniform and smooth, a small number of samples are sufficient to capture the variation of the background color. If the background is complex and, in particular, has large color variation, there is a higher chance that the background samples may not include the desired true background samples for the unknown pixels. The “color interpolation” part of Figure 3-3 shows an example of such failure of robust matting due to insufficient number of samples. For an

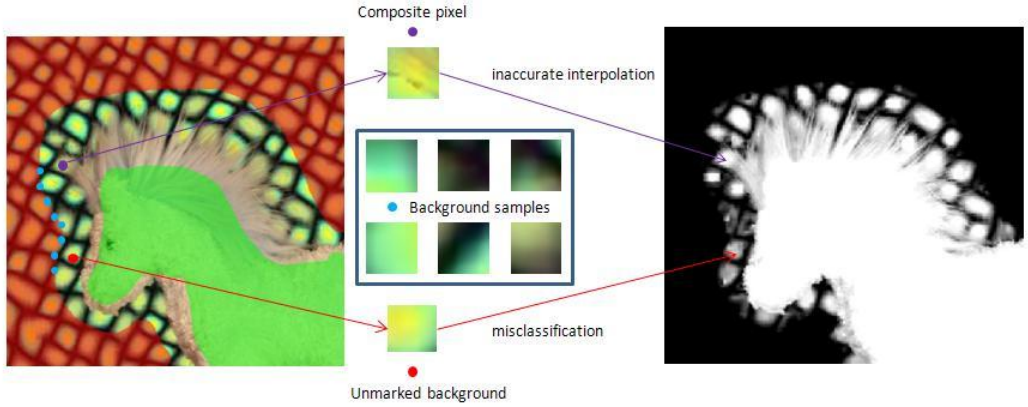


Figure 3-3: Demonstration of the failure of robust matting due to insufficient background samples.

unmarked background pixel in the unknown region, the background samples are collected from nearby pixels along the boundaries of definite background regions. We can see that in this example none of the background samples can serve as a good estimation of the background for the unknown pixel (in this case the correct result is to identify the unknown pixel as a pure background pixel). The consequence of inaccurate background estimation is two-fold: 1) Unmarked background pixels in the unknown region may be misclassified as composite pixels and incorrect alpha values will be computed; 2) Inaccurate alpha values are estimated for composite pixels in the unknown region due to the lack of proper background samples. Simply increasing the number of samples is not a feasible or a good solution for robust matting. In robust matting, every background sample is combined with every foreground sample to find the foreground-background pair which has the minimum interpolation residue. A large number of background samples will incur an unaffordable computation cost. Furthermore, blindly increasing the number of samples does not guarantee the inclusion of appropriate background samples.

Alpha propagation based matting approaches also fail with textured background. Generally, the alpha values are supposed to propagate within the foreground and the background regions and stopped by the strong edges at the boundary between the foreground and background. The challenge of textured background is that the background itself may contain many strong edges which will prevent the alpha values from propagating even among the unmarked background pixels in the unknown region. Figure 3-4 shows an example of how alpha propagation is blocked in closed-form matting. The multiplication signs indicate the strong edges where the alpha propagation into the diamond shaped region is blocked, though the whole region is actually unmarked background. Using a color model to eliminate such unwanted blockage of alpha propagation is suggested by the authors of closed-form matting. However, building an accurate color model in the presence of texture is also a difficult problem.

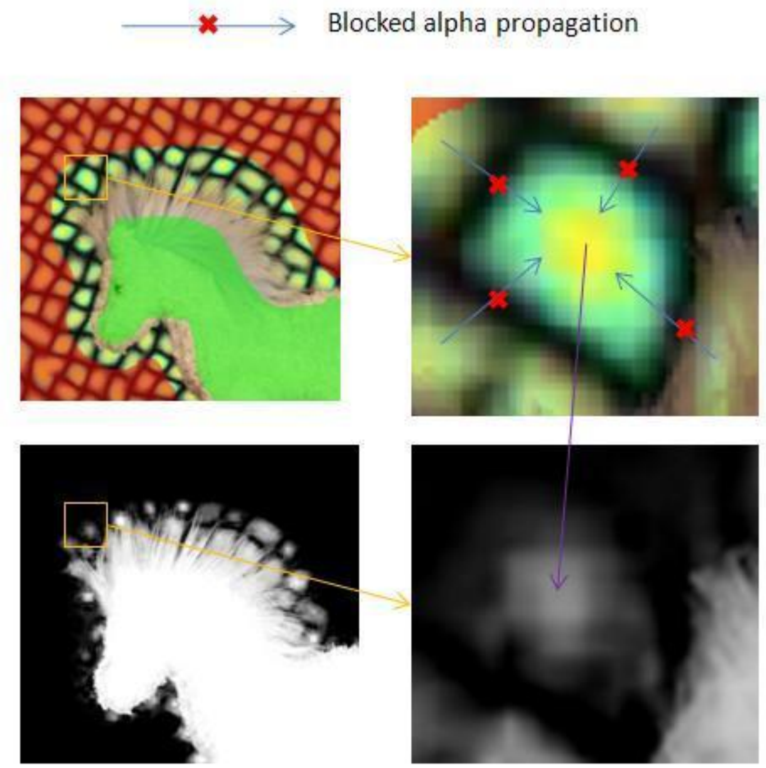


Figure 3-4: Demonstration of the blockage of alpha propagation using closed-form matting.

The same analysis also applies to the doll example in Figure 3-2. While the texture pattern in the horse example is regular and homogeneous, the background in the doll example is more complex and contains heterogeneous texture patterns. From the matting results of existing methods we can see that the main difficulty arises from the background flag and book regions, where the strong edges of the texture patterns prevent the effectiveness of alpha propagation and the large color variation makes it very difficult for the color sampling scheme to get good samples as illustrated in Figure 3-5.

The motivation of this thesis is to address the above mentioned problems introduced by complex image. We first start with the case where the image background contains homogeneous texture patterns as the one shown in Figure 3-1 and develop a novel texture synthesis based matting method which can utilize the known texture information to infer the texture information in the unknown region and thus alleviate the problems introduced by textured background. Then for matting with complex image which has heterogeneous texture patterns as shown in Figure 3-2, we develop a new foreground/background pixels identification algorithm which can identify the pure foreground/background pixels in the unknown region and thus effectively handle the challenges of large color variation introduced by complex images.

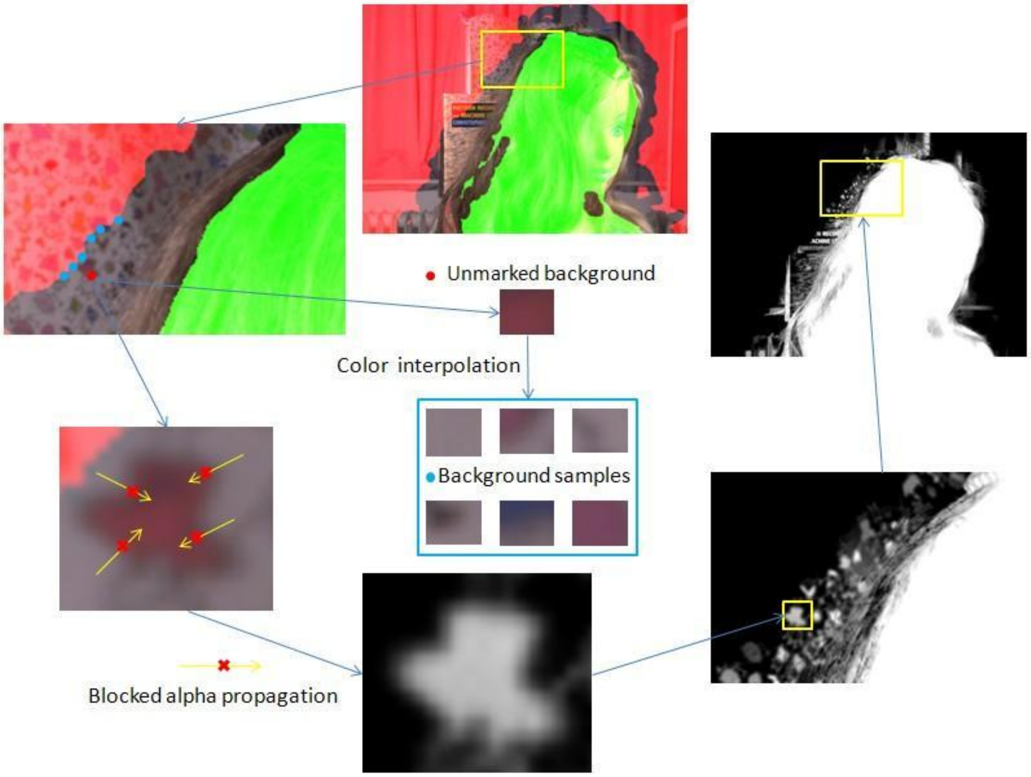


Figure 3-5: Demonstration of the failure case of robust matting and closed-form matting due complex background for the doll example used in Figure 3-2.

3.2 Complex Image Matting by Texture Synthesis

In this section we propose a new texture synthesis based color sampling scheme which can effectively handle the challenges introduced by textured background.

The basic idea of our method is to leverage the texture information of the definite background region to provide good background samples for the unknown pixels. Our method works by first applying texture synthesis in the unknown region, with the definite background as the source texture and the unknown region as the target texture. Notice that recently there is a similar inpainting scheme proposed in [31] to solve matting problem, their method is designed only for non-texture images. After synthesis, for those unmarked background pixels in the unknown region, the synthesized background pixels will serve as good background samples for them since the synthesized pixels are controlled to have similar values as the observed values of those unmarked background pixels. For the composite pixels in the unknown region, though the corresponding synthesized background pixels are not suitable background samples for them, their coherent relation with neighbouring synthesized pixels can be utilized to find appropriate background samples in the definite background texture.

In the following, first a brief review of the optimization based controllable texture synthesis technique is given in section 3.2.1. Then in section 3.2.2, we demonstrate how the controllable texture synthesis technique can be used to solve the textured background matting problem. Adapted alpha estimation and optimization framework for textured background is discussed in section 3.2.3. Finally the implementation details are discussed in section 3.2.4.

3.2.1 Controllable texture synthesis

Given an input texture example, the goal of texture synthesis is to create a new output texture which looks similar but not identical to the input texture. In [32], the texture synthesis procedure is modelled as the optimization of an energy function which measures the similarity between the input and output textures. Given an input texture Z, the texture energy of the output texture X is formulated as:

$$E_t(\mathrm{x};\{\mathrm{z}_p\}) = \sum_{p \in X^{\varphi}} \|\mathrm{x}_p - \mathrm{z}_p\|^2 \tag{3.1}$$

where X^{φ} is a subset of X over which we want to accumulate the texture energy, x_p is a vector formed by concatenating the pixels in the neighbourhood centered on pixel p with a predefined neighbourhood width w and z_p is a similarly constructed neighbourhood in Z whose appearance is most similar to x_p using the Euclidean norm. Informally, the texture synthesis procedure works as follows: for each neighbourhood x_p in the output texture, sampling in the source texture the most similar neighbourhood z_p to replace x_p. When two or more sampled neighbourhoods overlap, the synthesized value of the overlapped pixel is computed to be the average value of all the sampled pixels. The sum of square differences between the synthesized pixel value and the sampled pixel value is defined as the texture energy of the synthesized pixel. Iteratively minimizing the accumulated texture energy over a set of designated pixels in the output texture will gradually evolve the output texture to become similar to the input texture.

Additional desired properties of the output texture can be fulfilled by augmenting the texture energy function with constraints, e.g.,

$$E(\mathrm{x}) = E_t(\mathrm{x};\{\mathrm{z}_p\}) + \lambda E_c(\mathrm{x};\mathrm{u}) \tag{3.2}$$

where u is the control variable and λ is the relative weighting coefficient. This is called controllable texture synthesis. The optimization of this energy function can be done through an EM-like algorithm which alternates the optimization over one of x and $\{\mathrm{z}_p\}$ while fixing the other. More details can be found in [32].

3.2.2 Texture synthesis based color sampling

Our goal is to extend the above texture synthesis technique to provide good background samples for the unknown pixels. Using the definite background region as source texture to perform unconstrained texture synthesis in the unknown region works well only when the background texture is regular. To handle background with near-regular or irregular texture, we need to constrain the texture synthesis process in the unknown region. Usually a large portion of the unknown pixels are just unmarked background and foreground pixels with relatively few boundary pixels that are true composite pixels. In fact, only those unmarked background pixels are what we want to synthesize. Hence, we can apply the controllable texture synthesis method in the unknown region with the additional control of using the *observed* pixel values of the unknown region as the target values for the synthesized background pixels. So we define the controlled texture energy as:

$$\begin{aligned} E(\mathrm{x}) &= E_t\left(\mathrm{x};\{\mathrm{z}_p\}\right) + \lambda E_c(\mathrm{x};\mathrm{u}) \qquad (3.3) \\ &= \sum_{p \in X} \left\| \mathrm{x}_p - \mathrm{z}_p \right\|^2 + \lambda \sum_{p \in X} (\mathrm{x}(p) - \mathrm{u}(p))^2 \end{aligned}$$

where $\mathrm{x}(p)$ is the synthesized colour of pixel p, $\mathrm{u}(p)$ is the observed colour of pixel p and other terms are defined before.

The function of the controlled energy term is to match the texture pattern of the unknown region using texture samples from the definite background region. So after synthesis, there will be a high chance for the unmarked background pixels to be consistent with or similar to their corresponding synthesized background pixels. Then for the unmarked background pixels, the corresponding synthesized pixels will serve as good background samples. To find good background samples for the composite pixels in the unknown region is a more challenging problem. Since composite pixels do not have the same texture pattern as the background texture, usually we cannot find suitable texture samples in the definite background region to match the texture pattern of composite pixels. In this case, we use the coherence property of texture to find good background samples for the composite pixels.

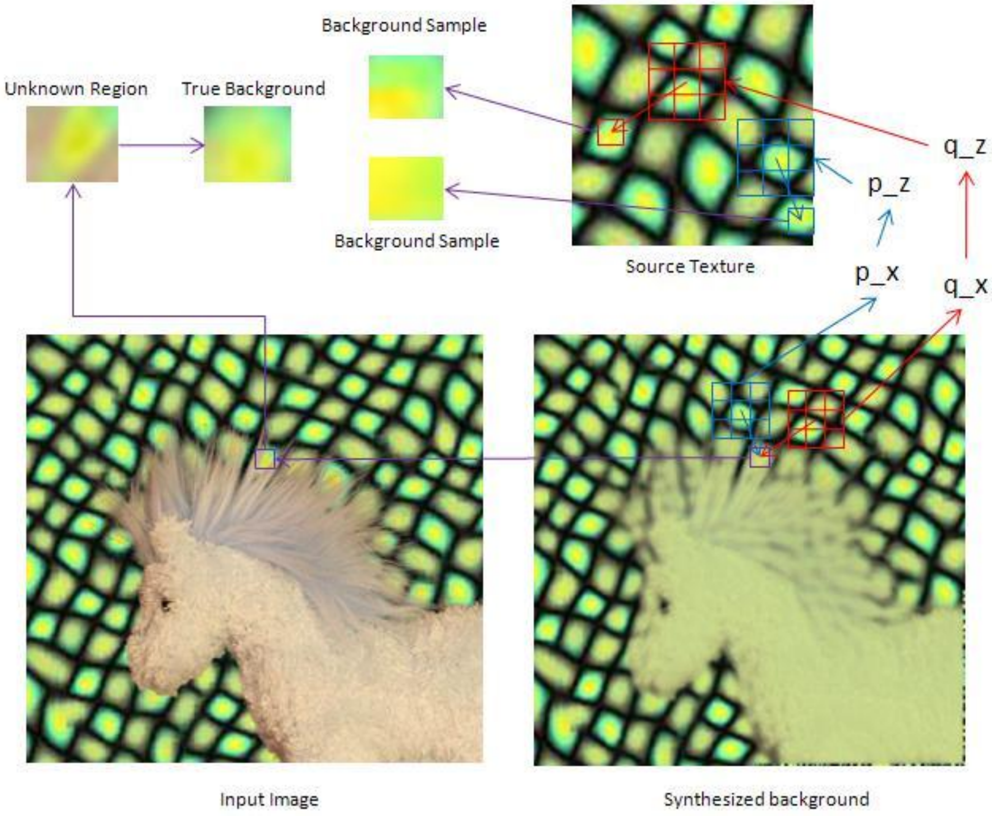


Figure 3-6: Using coherence property to find background samples.

We use the example in Figure 3-6 to illustrate the concept of texture coherence. In the synthesized background image *X*, the neighbourhoods centered at pixels p_x and q_x are sampled from neighbourhoods centered at pixels p_z and q_z, respectively, in the source texture *Z*. The purple square in the input image is a composite region and we want to find good background samples for it. In the synthesized background, we denote the offsets between the purple square region and pixels p_x and q_x as o_p and o_q, respectively. Then in the source texture image if we shift the pixels p_z and q_z by the same offset o_p and o_q, respectively, we can get two background samples for the purple square region (shown as small red square and blue square in the source texture image), which are very close to the true background of the purple square region because of the texture coherence property.

In summary, to find good background samples for pixels in the unknown region, we first do the controlled texture synthesis in the unknown region using the definite background as the source texture and the observed color of the unknown pixels as the target color for synthesizing background pixels. After synthesis, for each synthesized pixel p_x, we will have a corresponding pixel p_z in the source texture where p_x is sampled from. Then with a pre-defined set of offsets $O = \{\mathbf{0}, o_{1,} o_{2,} o_{3\dots} o_N\}$, pixel p_z $+ o_i$ in the source texture becomes a background sample for pixel p_x $+ o_i$ in the unknown region. Notice that the zero offset $\mathbf{0}$ means using the corresponding synthesized background pixel as the background sample which is designated for the unmarked background pixels.

Though we have only discussed how to find background samples so far, the same texture synthesis based scheme can be used to find foreground samples as well by synthesizing foreground texture in the unknown region. So in the following discussion, we assume that both background and foreground samples are obtained using the texture synthesis based scheme.

3.2.3 Alpha estimation and optimization

After the background and foreground samples for the unknown pixels are collected, the next step is alpha estimation and optimization. In this section we show how to adapt the alpha estimation and optimization framework used in robust matting to handle textured background.

Given a pixel in the unknown region with observed value C and its background samples $B = \{B^1, B^2 \dots B^m\}$and foreground samples $F = \{F^1, F^2 \dots F^n\}$, we want to select a pair of foreground sample and background sample $\{F^i, B^j\}$ that can best explain the value of C. In [2] a *distance ratio* $R_d(F^i, B^j)$ is defined to measure the fitness of using a linear combination of foreground and background samples to interpolate the unknown pixel C:

$$R_d\left(F^i, B^j\right) = \frac{\left\| C - (\alpha' F^i + (1 - \alpha')B^j) \right\|}{\| F^i - B^j \|} \tag{3.4}$$

where $\alpha^{'}$ is the estimated alpha:

$$\alpha^{'} = \frac{\left(C - B^j\right)(F^i - B^j)}{\| F^i - B^j \|^2} \tag{3.5}$$

To incorporate the fact that most unknown pixels are just unmarked foreground or background pixels, each individual sample is associated with additional weights $\omega(F^i)$ and $\omega(B^j)$ defined as

$$\omega\left(F^i\right) = \exp\left\{- \left\| F^i - C \right\|^2 / min_k(\| F^k - C \|^2)\right\} \tag{3.6}$$

$$\omega\left(B^j\right) = \exp\left\{- \left\| B^j - C \right\|^2 / min_k(\| B^k - C \|^2)\right\} \tag{3.7}$$

Finally, a confidence value $f(F^i, B^j)$ for the sample pair $\{F^i, B^j\}$ is defined as

$$f(F^i, B^j) = exp\left\{- \frac{R_d(F^i, B^j)^2 \cdot \omega\left(F^i\right) \cdot \omega\left(B^j\right)}{\sigma^2}\right\} \tag{3.8}$$

where σ is fixed to be 0.1.

In robust matting, the highest confidence value $f'(F^i, B^j)$ for an unknown pixel C is used as its final confidence value and serves as an indicator of how to compute the alpha value for the unknown pixel: a high confidence value means that the estimated alpha value using colour samples is more reliable, while a low confidence value means that no good sample pair is found to interpolate the unknown pixel or the unknown pixel is a potential unmarked background or foreground pixel, in which case the method resorts to using alpha propagation to compute the alpha value for the unknown pixel.

As discussed in section 3.1, textured background tends to block the alpha propagation. The way that robust matting defines the confidence value cannot resolve this issue: potential unmarked background and foreground pixels are assigned low confidence so that their alpha values are obtained through alpha propagation, which suffers from the problem of blocked alpha propagation when the background and/or foreground are textured.

When the potential unmarked background and foreground pixels are identified, they can serve as the "alpha seeds" to facilitate the alpha propagation through unmarked background and foreground regions. Assigning high confidence value to potential unmarked background and foreground pixels can achieve this goal. We follow [33] to define new weights $\omega'(F^i)$ and $\omega'(B^j)$ as (though the original motivation in [33] is not to handle textured background/foreground)

$$\omega'\left(F^i\right) = \exp\left\{- max_k(\|F^k - C\|^2)/\left\|F^i - C\right\|^2\right\} \qquad (3.9)$$

$$\omega'\left(B^j\right) = \exp\left\{-max_k(\|B^k - C\|^2)/\left\|B^j - C\right\|^2\right\} \qquad (3.10)$$

which will result in assigning high confidence values for potential unmarked background and foreground pixels.

For each unknown pixel C, the highest confidence value $\hat{f}$ of all the sample pairs $\{F^i, B^j\}$ is used as the final confidence value and the corresponding estimated alpha is denoted as $\hat{\alpha}$. We follow [4] to do alpha optimization using the estimated alpha values as soft constraints and using the matting Laplacian L as the smoothness term. The optimization problem is formulated as

$$\alpha = argmin\ \alpha^T L\, \alpha + (\alpha - \hat{\alpha})^T \widehat{\Gamma}(\alpha - \hat{\alpha}) \qquad (3.11)$$

where the diagonal matrix $\widehat{\Gamma}$ is defined as in [33] with each diagonal element $\hat{\gamma}_z$ regularized with $\hat{\alpha}_z$'s corresponding confidence value $\hat{f}_z$:

$$\hat{\gamma}_z = \gamma \cdot \hat{f}_z \tag{3.12}$$

where γ is set to be 10^{-3}. Optimal alpha values can be computed by solving a sparse linear system obtained by differentiating Equation 3.11 and setting the derivatives to zero.

3.2.4 Implementation details

The width w of the sampling neighbourhood in texture synthesis is an important parameter that should be large enough to capture the structure information of the source texture. The setting of w is usually *ad hoc* for unconstrained texture synthesis. For our controlled texture synthesis, however, a small neighbourhood size is enough since we have the target texture to guide the synthesis. In our implementation w is set to be 5.

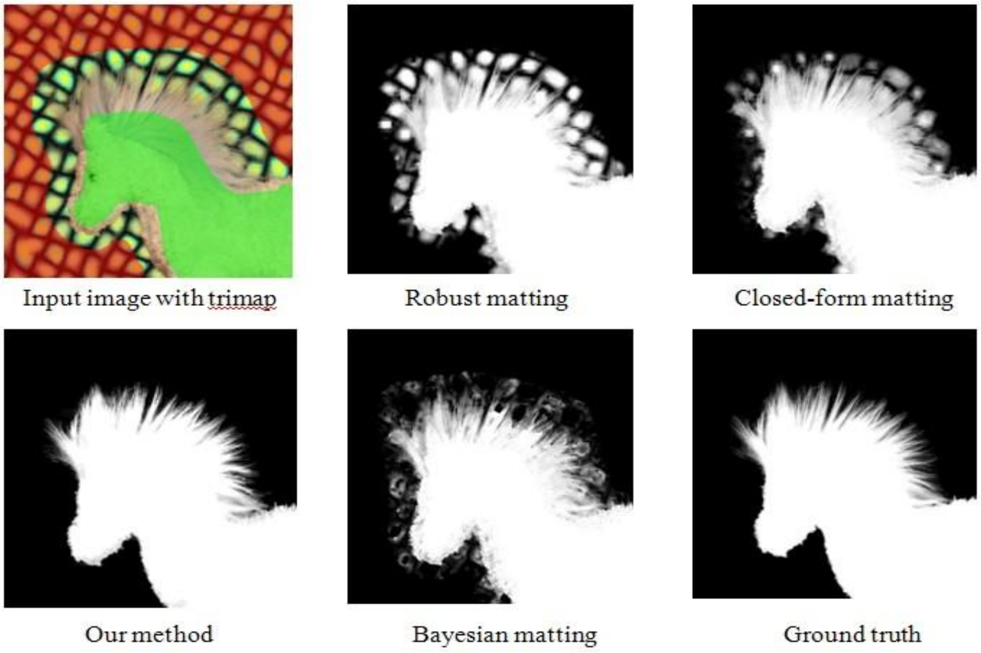
The weighting coefficient λ in Equation 3.3 is set to be the number of neighbourhoods that can overlap at the same pixel, which in our case $w \cdot w = 25$, so that the sampled neighbourhoods and the observed pixel value will have equal influence on the synthesized value.

The dominant computation cost in the texture synthesis is the nearest neighbour search in the source texture for each neighbourhood in the output texture. When the definite background region (the source texture) is large, the time cost for direct searching becomes unaffordable even with a small neighbourhood size. We follow [34-35] to use the parallel k-coherence search to do the nearest neighbour search. The accelerated texture synthesis poses negligible overhead on the whole matting process.

The design of offset set O decides the number of samples in the foreground and background regions to be collected for an unknown pixel. While more foreground and background samples give more robust alpha estimation for the unknown pixel, the time complexity for alpha estimation is $O(n^2)$ when the number of foreground and background samples is $O(n)$. In our implementation we set that each unknown pixel C gets background samples from all the synthesized background pixels within a 5 by 5 neighbourhood centered at C and the same for foreground samples. So for each unknown pixel C a total of 25 background pixels and 25 foreground pixels are collected.

3.2.5 Experimental results and discussion

Figure 3-7 shows the matting result of our method and the comparison with other matting methods on the input image and trimap shown in Figure 3-1. The matte result for robust matting is generated by "EZ Mask" [36]. For closed-form matting we use its authors' original implementation and for Bayesian matting we use our own implementation. We can see that our method achieves significant improvement over the three matting methods and comparison with the ground truth image shows that our matting result is quite accurate qualitatively.



Input image with trimap

Robust matting

Closed-form matting

Our method

Bayesian matting

Ground truth

Figure 3-7: Comparison of different matting methods for input image in Fig. 3-1.

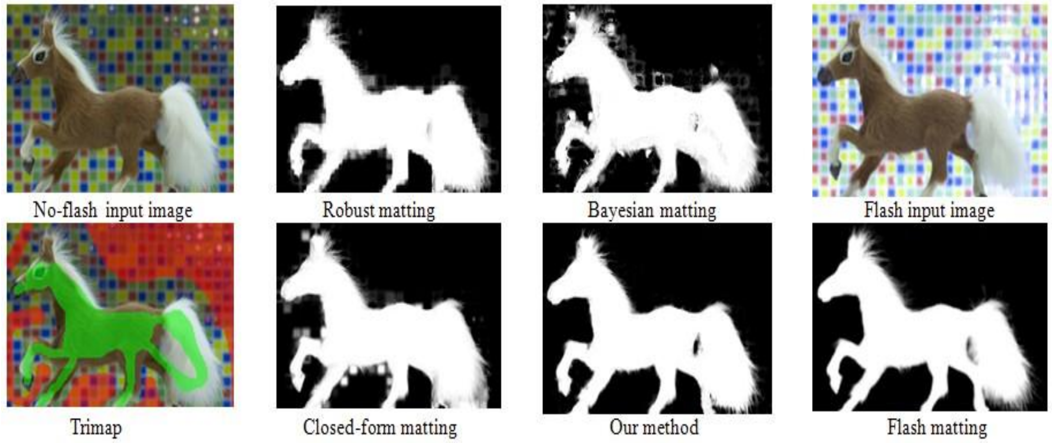


Figure 3-8: Comparison with flash matting.

The example in Figure 3-8 is used in flash matting [15] to demonstrate that how no-flash and flash image pair can be combined to generate more accurate alpha matte for challenging matting problem. The texture background in the no-flash input image confuses most existing matting methods. With an additional flash input image, flash matting can generate quite impressive matting result, while our method using only one no-flash input image can generate matting result that is comparable to that of flash matting.

To quantitatively measure the performance of our proposed method, we run the experiments using data set in [37] along with the other three matting methods to do the comparison. The same experimental methodology as in [37] is used. There are a total of 6 test images with known ground truth as shown in Figure 3-9. Each input image is tested using different matting algorithms on 10 different trimaps from coarse to fine in accuracy. The Mean Squared Error (MSE) with respect to the ground truth is used as a quantitative measurement. Table 1 shows the experimental results. T1-T6 are six test images. X:Y in each cell represents minimum MSE : maximum MSE on the corresponding test image. We can see that the results of our method are comparable with the best results of the other three methods. Notice that for image T5, which has complex texture background, our result is significantly better than other matting methods, validating the effectiveness of our method in handling with complex texture background.

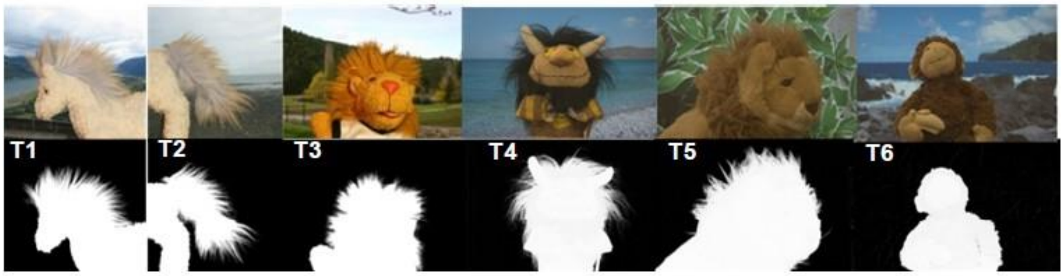


Figure 3-9: Test images with ground truth taken from [37].

Table 1: MSE of different matting algorithms.

	Bayesian matting	Closed-form matting	Robust matting	Our method
T1	356: 873	86:102	53:88	50:83
T2	380:1374	110:264	48:132	54:105
T3	150:820	58:82	40:73	45:65
T4	312:786	95:321	102:250	98:202
T5	460:940	176:426	203:452	34:82
T6	50:97	32:62	30:53	35:48

Though we have achieved satisfactory matting results on images with homogeneous textured background, the challenge of natural image matting is that the background can be arbitrarily complex and can exhibit inhomogeneous, globally varying texture. Our observation is that if the texture patterns in the background are relatively regular and sufficient representative texture samples are specified as definite background, e.g. T5, then our method can generate much better matte results than existing methods. In other cases, our method generates results that are comparable to that by other methods. The input image in Figure 3-10 shows an example of such complex background with different but relatively regular textures. The blurred trees, grass and halos in the background all show some texture patterns. By utilizing the texture information, such as the halo texture pattern in the red rectangle region and the tree texture pattern in the yellow rectangle region, our method creates cleaner alpha matte than existing methods.

However, when the complex image background contains relatively irregular or random texture patterns, it is more difficult for the texture synthesis process to collect sufficient texture samples for synthesising the unknown region. As shown in Figure 3-11, the background region of the image contains irregular texture patterns like the flag and the cover of the book. Upon close observation of the input image and the trimap, we can see that the definite background region actually cannot provide enough texture samples to synthesis the texture patterns in the unknown region. As a result, the matting result of our texture synthesis based method is quite noisy. So to handle the case of complex image with arbitrary texture patterns, especially random and irregular texture patterns with insufficient samples, we develop another matting method which is based on identifying unmarked foreground and background pixels. Details are given in the following section.

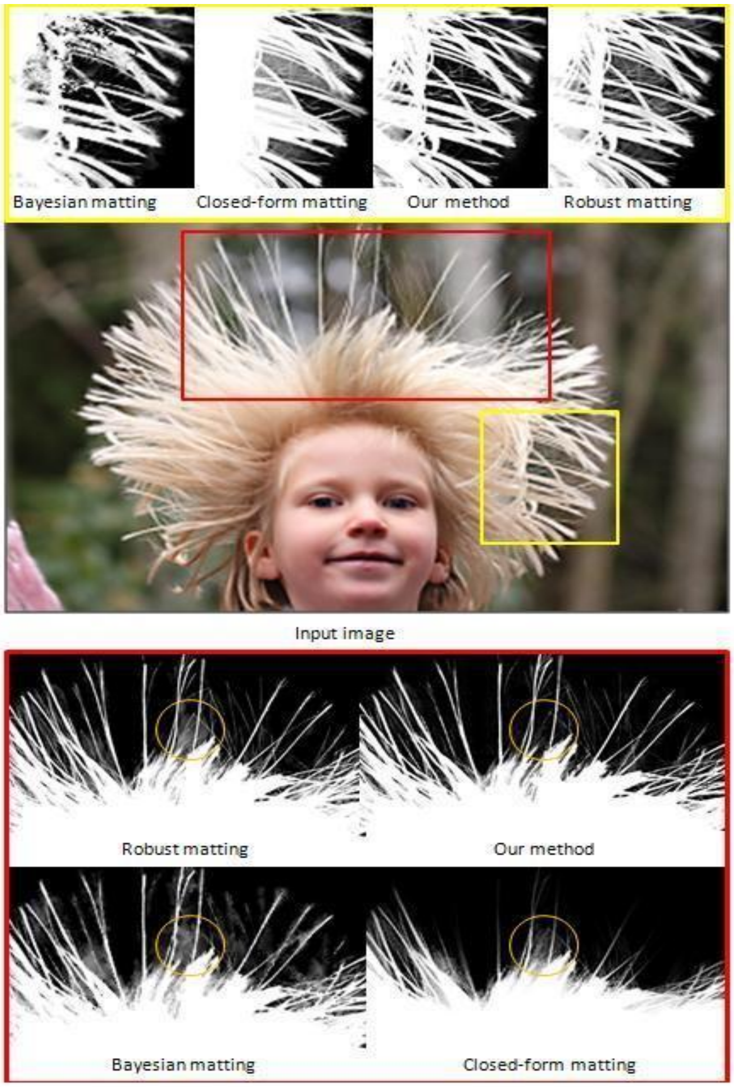
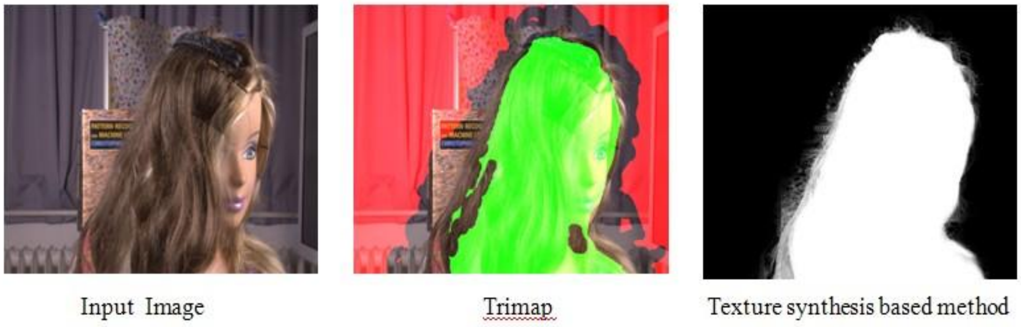


Figure 3-10: Complex image background with relatively regular textures.



Input Image

Trimap

Texture synthesis based method

Figure 3-11: Example image with complex background.

3.3 Complex Image Matting by Unmarked Foreground and Background Pixels Identification

In this section we propose a new matting method based on identifying unmarked foreground/background pixels which can effectively handle the challenges of large color variation introduced by complex images. The main idea of our method is to identify pure foreground and background pixels in the unknown region as shown in Figure 3-12(d). The turquoise pixels are identified background pixels and the pink pixels are identified foreground pixels. These identified foreground and background pixels can serve as “alpha seeds” to overcome the alpha blockage issue discussed in section 3 and facilitate the alpha propagation through unmarked background and foreground regions. This unmarked foreground and background pixels identification based matting method is called UFBPI matting. We present in the following sections several heuristics designed to select the pure foreground and background pixels with high confidence.

3.3.1 Interpolation error thresholding

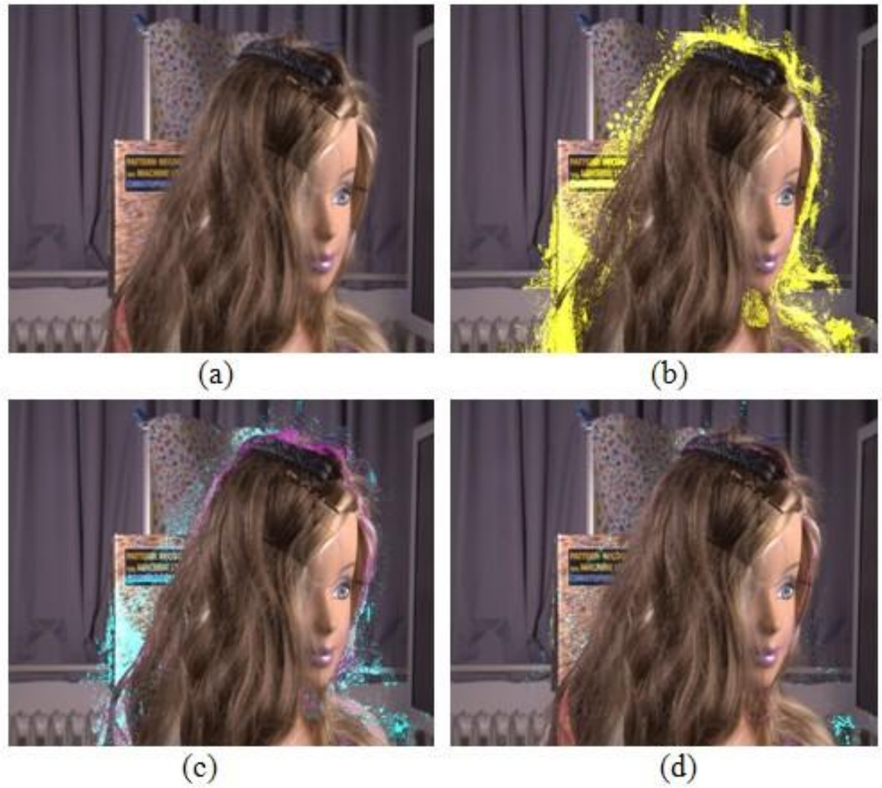


Figure 3-12: Candidates of pure foreground/background pixels after different selecting steps of our proposed method.

In robust matting, for each unknown pixel, a set of foreground and background samples are collected and all the possible foreground/background sample pixel pairs are tried to find the best interpolation for the unknown pixel. We follow the same sampling scheme as in robust matting, however, with the purpose of screening out those unknown pixels with large interpolation errors as candidates for pure foreground and background pixels. In particular, the foreground and background samples are collected along the boundaries of known foreground and background regions. For each unknown pixel C, all the sample pairs $\{F^i, B^j\}$ are tried to estimate $\hat{\alpha}$:

$$\tilde{\alpha} = \frac{\left(C - B^j\right)\left(F^i - B^j\right)}{\|F^i - B^j\|^2} \tag{3.13}$$

and the interpolation error for the pair $\{F^i, B^j\}$ is defined as:

$$E\left(F^i, B^j\right) = \left\|C - \left(\tilde{\alpha}F^i + (1 - \tilde{\alpha})B^j\right)\right\| \tag{3.14}$$

We denote the minimal interpolation error for the unknown pixel C as E_C. A smaller interpolation error E_C indicates that there is a higher chance that the unknown pixel C is a composite pixel. Since our purpose is to identify pure foreground/background pixels, we can use the interpolation error to first filter out true composite pixels. So only the unknown pixels with interpolation error E_C larger than a threshold value T_E are selected for further analysis. The setting of the threshold value T_E is discussed in section 3.3.3. One example of the result of interpolation error thresholding is shown in Figure 3-12(b) with yellow pixels denoting selected pixels.

3.3.2 Adaptive foreground/background distance thresholding

The main criterion we use to decide whether an unknown pixel is a pure foreground/background pixel or a composite pixel is its colour distance to the definite foreground and background regions. In robust matting, each unknown pixel C is compared with its foreground and background samples and the colour distance is used as an indication of whether or not C is a pure foreground or background pixel. As discussed in section 3, when the image is complex, the foreground and background samples collected from the boundaries of the definite foreground and background regions may not be sufficient to capture the large variation of the foreground and background colour. So we propose that, in order to find good foreground and background samples for the unknown pixel C, a direct search for similar pixels in the definite foreground and background regions is used. Denote a 5 by 5 window centered at the unknown pixel C as W_c. Then the most similar window W_{c-bg} in the definite background region is found based on the accumulated colour distance over all the pixels in the window. The same search is conducted for the definite foreground region and the best matching window W_{c-fg} is found. For each unknown pixel P in the window W_c, all the pixels in the background matching window W_{c-bg} are added to the set of P's background samples B_P and all the pixels in the foreground matching window W_{c-fg} are added to the set of P's foreground samples F_P. Now for an unknown pixel C, its distance to the background and foreground is defined as the minimum colour distance to its background and foreground samples B_c and F_c, respectively:

$$D_B(C) = min_i\left(\left\|B_C^i - C\right\|\right), \qquad B_C^i \in B_c \tag{3.15}$$

$$D_F(C) = min_i\left(\left\|F_C^i - C\right\|\right), \qquad F_C^i \in F_c \tag{3.16}$$

A smaller background distance $D_B(C)$ indicates a higher chance that the unknown pixel C is a pure background pixel and the same inference applies to the foreground distance $D_F(C)$. Our goal is to set the threshold values $T_F(C)$ and $T_B(C)$ for each unknown pixel C adaptively so that we consider C as a candidate of pure background pixel if

$$D_B(C) \leq T_B(C) \tag{3.17}$$

and a candidate of pure foreground pixel if

$$D_F(C) \leq T_F(C) \tag{3.18}$$

The threshold value $T_B(C)$ for the unknown pixel C is computed as follows. The definite background region specified in the trimap is first dilated using a 3 by 3 square structuring element. We denote the set of pixels which are in the dilated background region but not in the original background region as B_{dilate} . Since pixels in B_{dilate} are near the boundary of the definite background region, they are most likely pure background pixels and their distance to the background are quite informative for defining the threshold value for the background distance of unknown pixels. Again, for a 5 by 5 window W_c centered at an unknown pixel C, we search for the most similar window W_{c-bg} whose center pixel is in B_{dilate} based on the accumulated colour distance over all the pixels in the window. The threshold value $T_B(C)$ is then defined as:

$$T_B(C) = \frac{1}{\#W_{c-bg}} \sum_i D_B(W_{c-bg}^i) \tag{3.19}$$

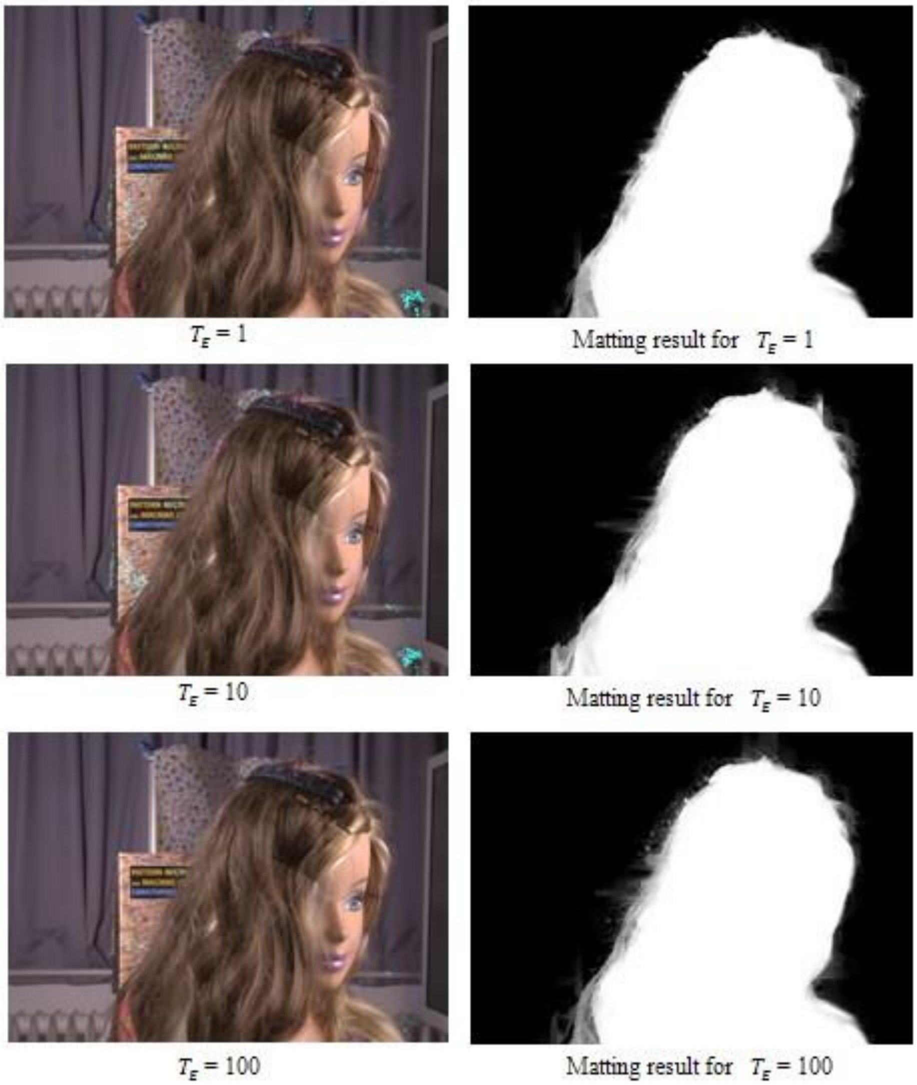
where pixel W_{c-bg}^i is an unknown pixel in window W_{c-bg} and $\#W_{c-bg}$ is the total number of such pixels (which is at least 1 since the center pixel of W_{c-bg} is always an unknown pixel by definition). The rationale behind the design of $T_B(C)$ is that if C is a composite pixel or pure foreground pixel, then its background distance should be larger than the background distance of those pixels in the matching window W_{c-bg} , most of which are assumed to be pure background pixels. And if C is a pure background pixel, its background distance will be similar to those pixels in the matching window W_{c-bg}, which makes $T_B(C)$, the average background distance of the pixels in the matching window, a good candidate of the threshold value. The foreground distance threshold value $T_F(C)$ is defined using the foreground matching window W_{c-fg} with a similar definition as $T_B(C)$.

After applying the foreground/background distance thresholding, candidates of pure foreground/background pixels for the doll example are shown in Figure 3-12(c), with the turquoise pixels denoting candidates of pure background pixels and pink pixels denoting candidates of pure foreground pixels. Unknown pixels with both foreground and background distance smaller than its corresponding foreground and background threshold values are not selected due to their ambiguities. From the thresholded results, we can see that many unknown pixels are identified correctly as pure foreground and background candidates, but some unknown pixels are obviously misclassified. We refine the results by imposing a smoothness constraint that an identified background candidate is declared as a pure background pixel only when the number of identified background candidates in its neighbourhood is larger than a threshold value T_{nbr} . In our implementation, we set the neighbourhood to be a 3 by 3 square and T_{nbr} to be 7. The same constraint is applied to the foreground candidates. Unknown pixels declared as pure foreground and background pixels after applying the smoothness constraint are shown in Figure 3-12(d).

After the unmarked background and foreground pixels are identified, they can serve as the "alpha seeds" to facilitate the alpha propagation through unmarked background and foreground regions. We follow [4] to do alpha optimization using the user specified trimap together with the identified background and foreground pixels. The final matting result is shown in Figure 3-14. We can see that the matting result has significant improvement over the three existing popular matting methods.

3.3.3 Setting of interpolation error threshold value

The threshold value T_E for interpolation error introduced in section 3.3.1 controls how many unknown pixels can enter the next step of foreground and background distance thresholding. We show in Figure 3-13 how the choice of different T_E values can affect the final matting result of the doll example. We can see that a larger value of T_E will filter out more unknown pixels and fewer foreground and background pxiels are identified. Hence the complex background may not be handled effectively as shown in the case of $T_E = 100$. On the other hand, a smaller value of T_E allows more unknown pixels to proceed to the next thresholding



$T_E = 1$

Matting result for $T_E = 1$

$T_E = 10$

Matting result for $T_E = 10$

$T_E = 100$

Matting result for $T_E = 100$

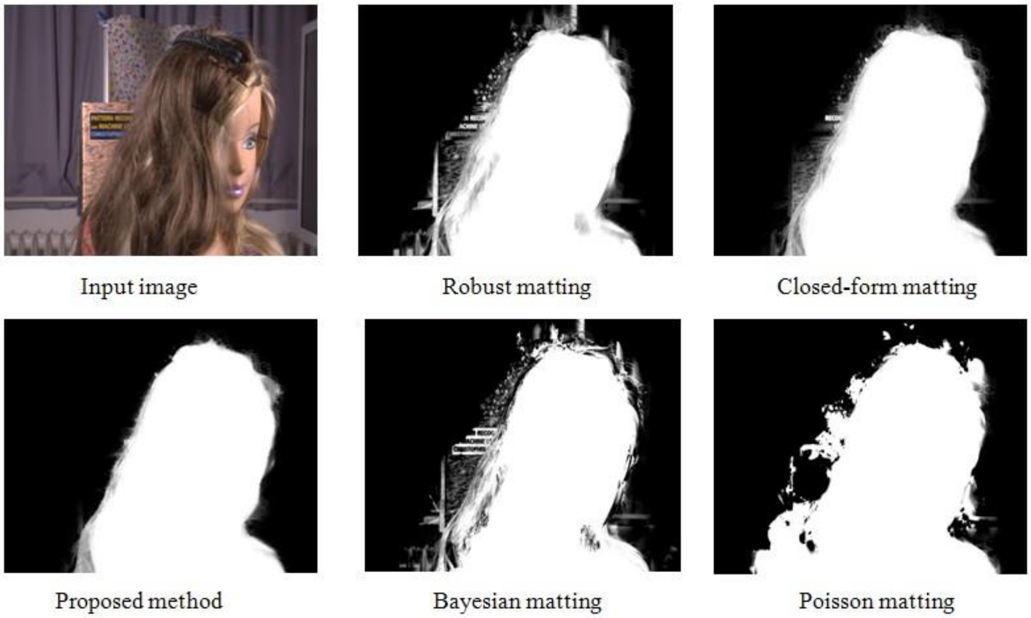
Figure 3-13: The matting results of choosing different threshold values for interpolation error.

step which also increases the chance of introducing false positves of identified foreground and background pixels, as shown in the case of $T_E = 1$ that on the right side of the doll there are some semi-transparent hair pixels identified as pure foreground pixels. The general rule based on our experience is that a large value of T_E is suitable for images with relatively smooth background and foreground and a small value of T_E for images with relatively complex background and foreground.

3.3.4 Experimental results and discussion

The final matting result of the proposed method (with T_E = 1) for the doll example is shown in Figure 3-14. We can see that with the help of the identified foreground and background pixels, the background with complex texture patterns is effectively handled and the matting result of our proposed method is much cleaner than the four popular matting methods. Other than the doll example, we also conduct several qualitative and quantitative experiments to verify the effectiveness of our methods in handling complex images.

The test case in Figure 3-15 is taken from [30]. The challenge of this image is that in the upper right part of the image, the bridge in the background is cluttered by the hair of the toy. From the trimap we can see that there is no suitable background samples in the nearby definite background boundary for the cluttered region. Hence, the color sampling scheme used in robust matting fails in this case and misclassifies the bridge as foreground. In closed-form matting, the bridge in the upper right part of the image is cut off from the bridge in the lower right part of the image by the hair of the toy, so the known alpha values from the lower right part of the bridge cannot propagate to the upper right part of the bridge. That is why closed-from matting also fails in this case. In the bottom left figure of Figure 3-15, we zoom in to the upper right part of the image and show the pure background pixels in turquoise and pure foreground pixels in pink identified by our proposed method. It can be seen from the matting result of our proposed method (with T_E= 100) that these identified background pixels are quite helpful in correctly recognizing the background bridge.


Input image

Robust matting

Closed-form matting

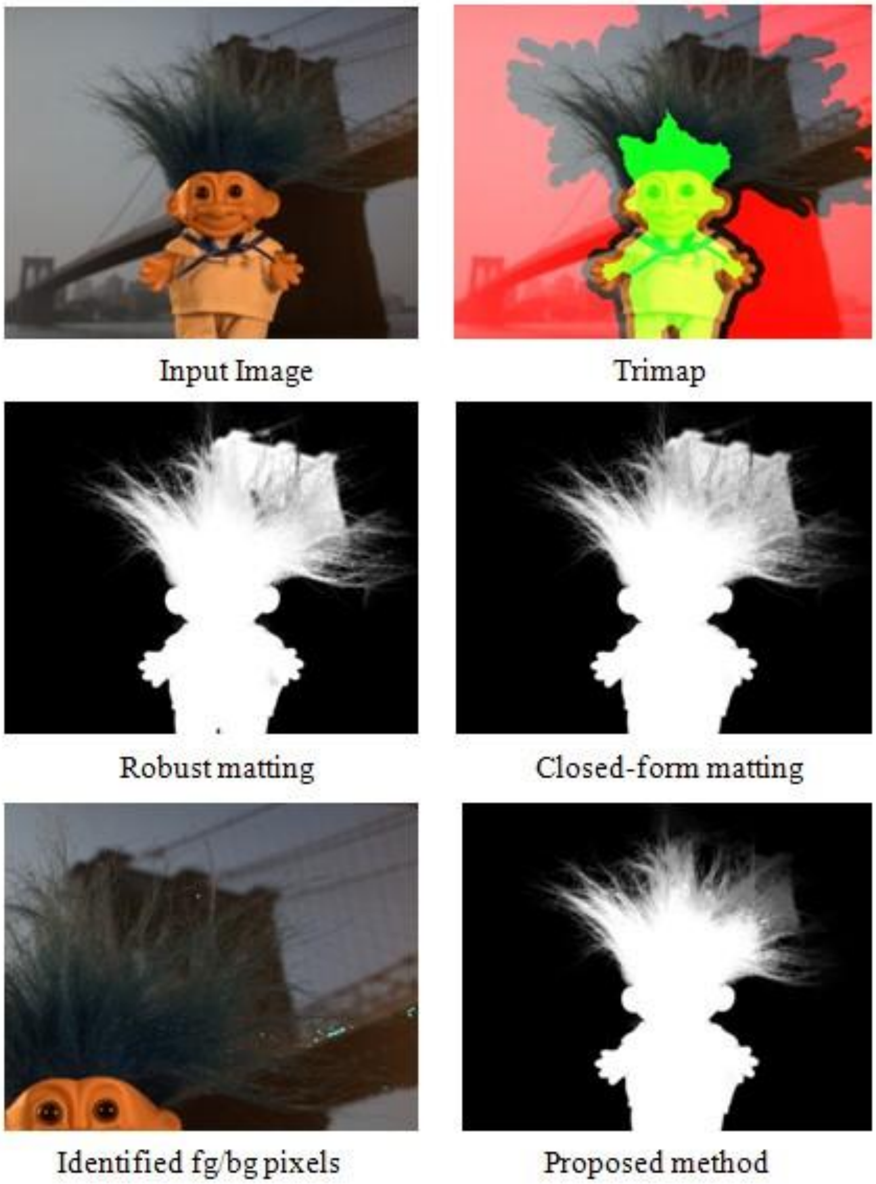
Proposed method

Bayesian matting

Poisson matting

Figure 3-14: Matting result of the proposed method for the doll example.

The lion example in Figure 3-16 is taken from [37]. We can see that the texture patterns in the background caused the existing matting methods to generate obvious artefacts while our method (with T_E = 1) handles the textured background very well and generates very clean matte results.



Input Image

Trimap

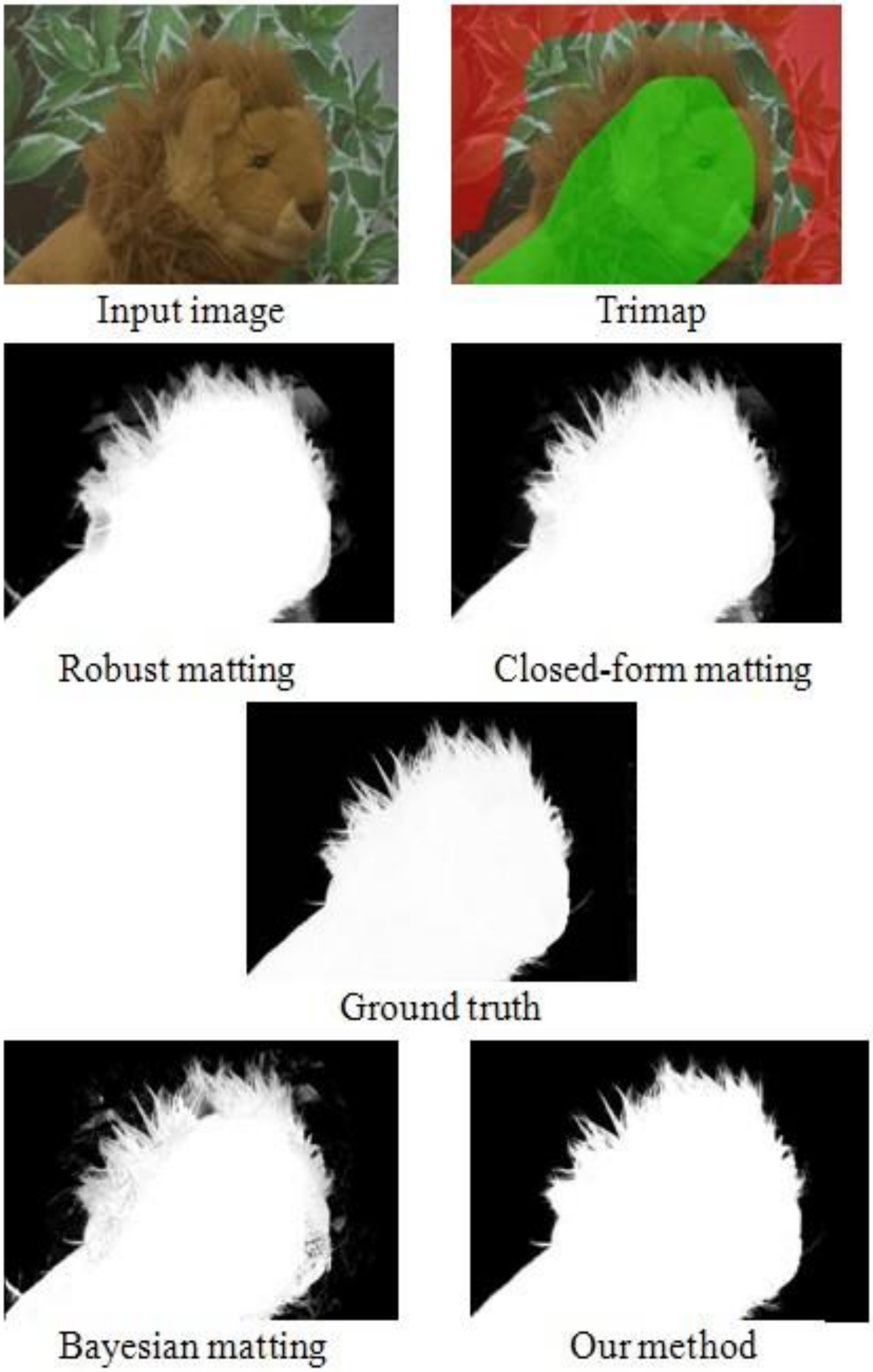
Robust matting

Closed-form matting

Identified fg/bg pixels

Proposed method

Figure 3-15: Comparison of different matting methods on the troll example.



Input image

Trimap

Robust matting

Closed-form matting

Ground truth

Bayesian matting

Our method

Figure 3-16: Lion example with complex background.

We also quantitatively compare the UFBPI matting with existing matting methods. Figure 3.17 – Figure 3.20 show the comparison of UFBPI matting with three other matting methods on 4 testing images taken from [30] using the average of the Sum of Absolute Difference (SAD) between the generated alpha matte and the ground truth as the error measure. The SAD value for each method is shown in the bracket associated with the name of each method in the figures. T_E is set to 1 in all the 4 test cases. We can see that for these quite challenging test images with complex background, UFBPI matting performs consistently better than the other 3 matting methods.

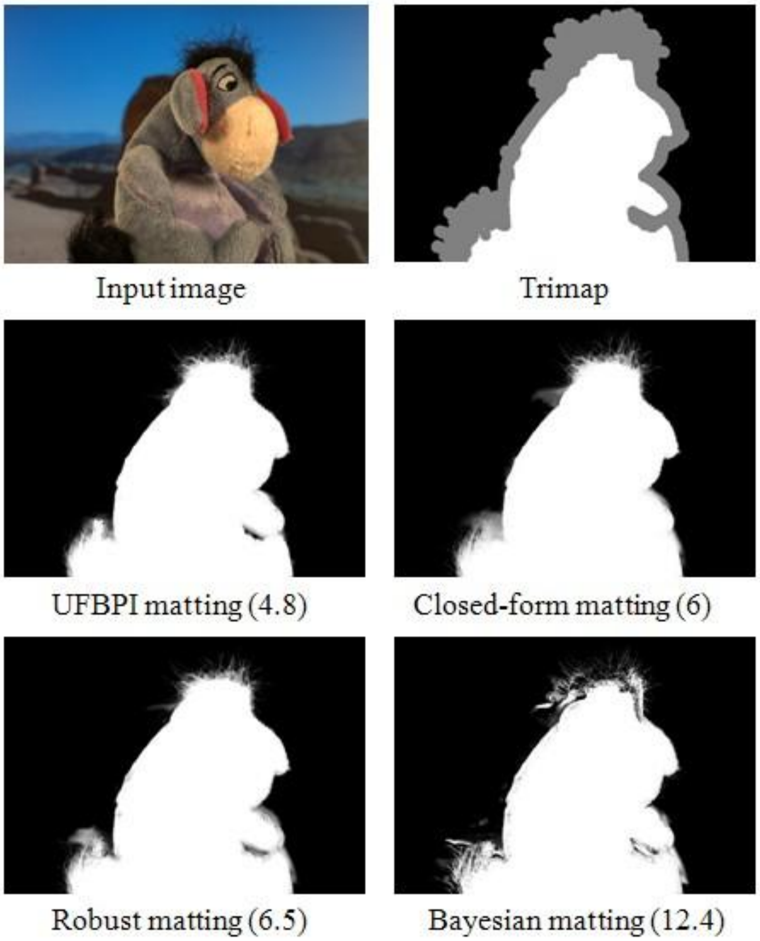
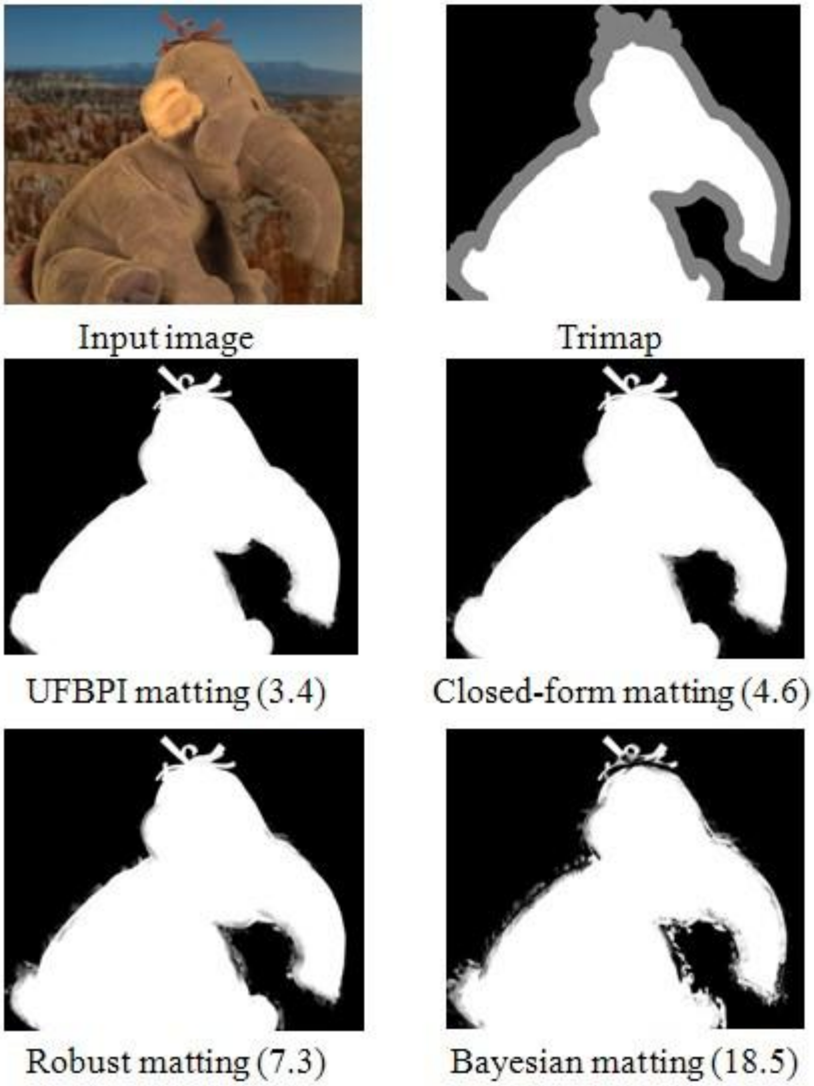


Figure 3-17: Donkey example



Input image

Trimap

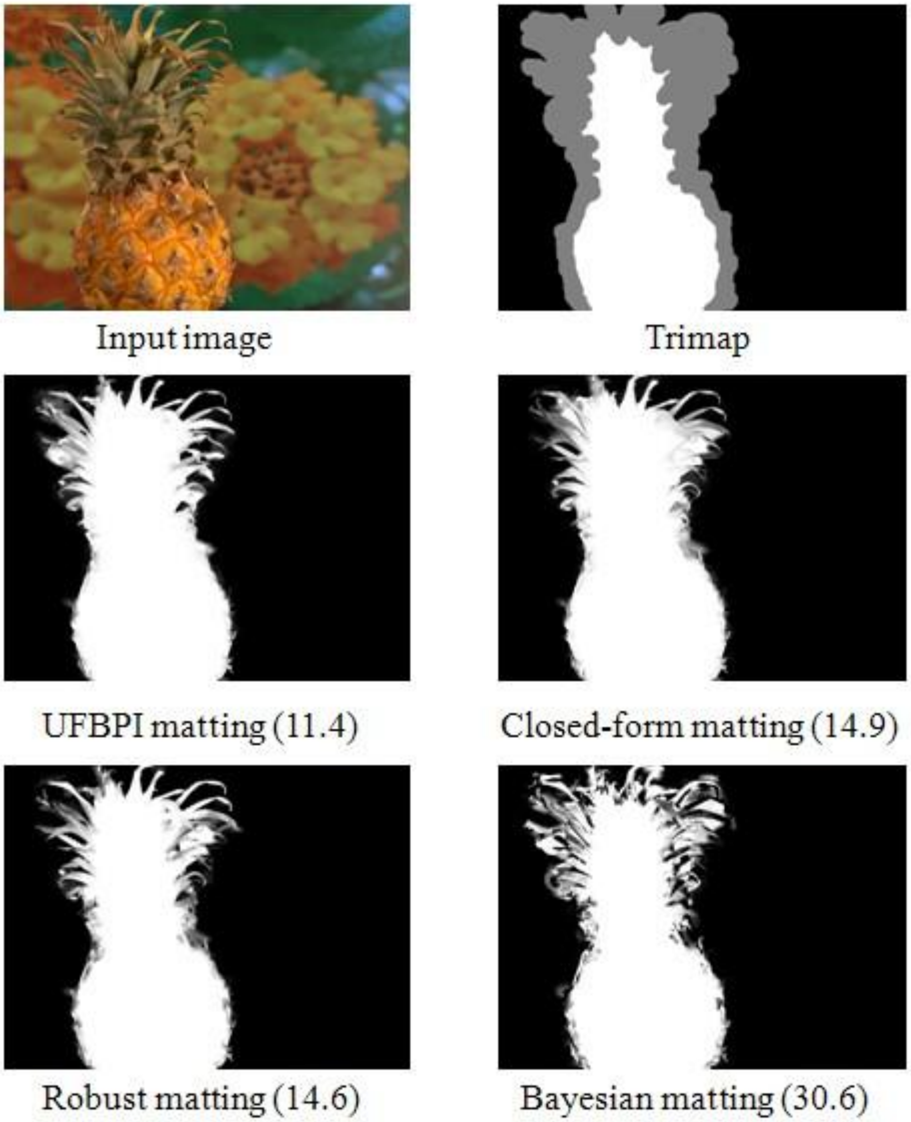
UFBPI matting (3.4)

Closed-form matting (4.6)

Robust matting (7.3)

Bayesian matting (18.5)

Figure 3-18: Elephant example



Input image

Trimap

UFBPI matting (11.4)

Closed-form matting (14.9)

Robust matting (14.6)

Bayesian matting (30.6)

Figure 3-19: Pineapple example

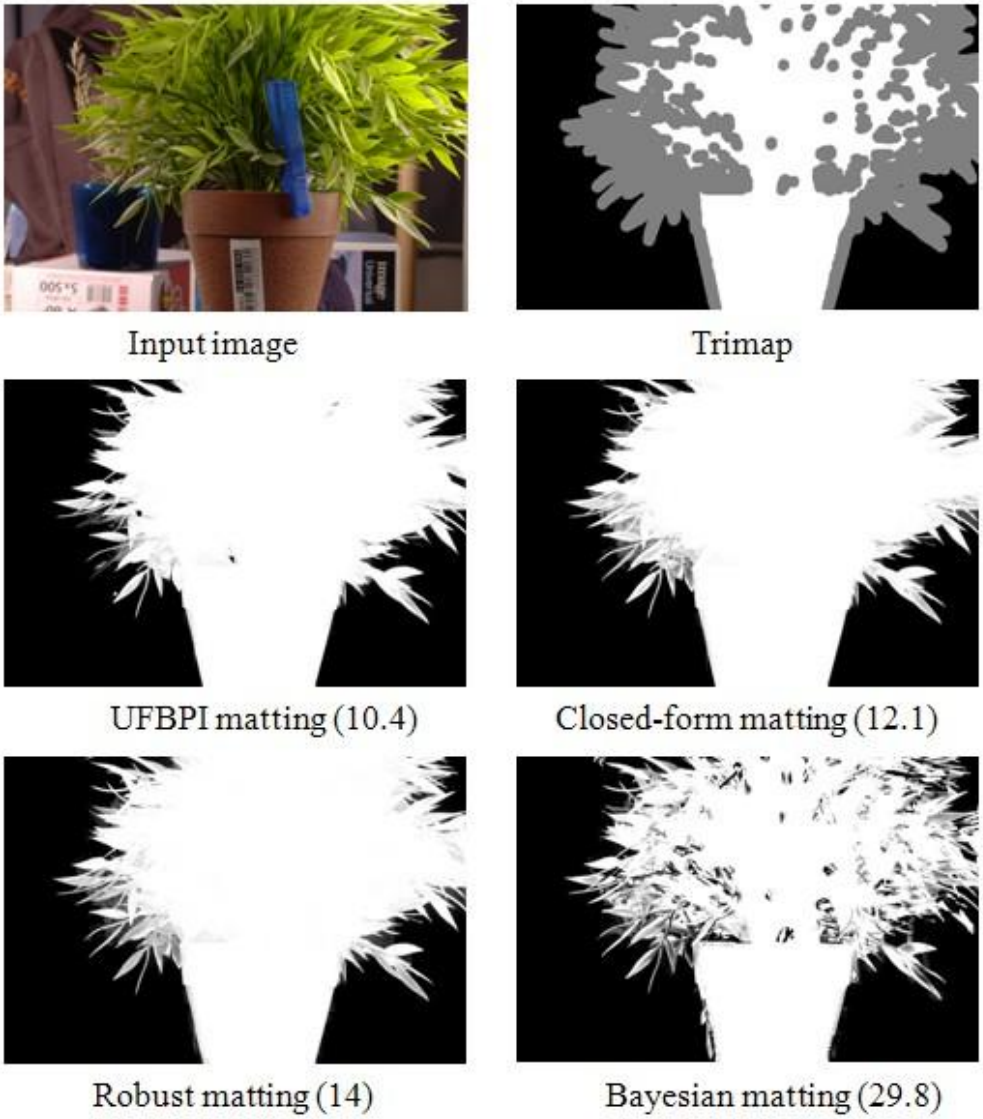


Figure 3-20: Plant example

Chapter 4: Conclusion

In this thesis, two novel matting algorithms are presented that can better handle complex images than the existing state-of-the-art matting methods. While the natural image can be complex in arbitrary ways, we mainly focus on complex images with highly textured background. The main challenge introduced by textured background is the large color variance and strong color discontinuity. Large color variance makes the popular color sampling scheme less reliable since nearby definite foreground and definite background pixels may no longer serve as good samples for the true foreground and background pixels of the unknown pixels. Strong color discontinuity within pure foreground and background regions may even be stronger than the real foreground/background edges, which confuses most alpha propagation based matting methods. Our understanding is that almost all the existing matting methods rely on making assumptions on local image statistics to develop local color models for color sampling, or build local alpha relationship for alpha propagation. In complex images, those local assumptions are frequently violated, that is the local color models built are no longer accurate due to high sample variance, and neighbouring pixels with negatively related alpha values are no longer a reliable indication of foreground/background edge.

Generally speaking, the algorithms we developed to tackle the challenges of complex images rely on seeking help from global image statistics. In the proposed texture-synthesis based matting method, we rely on the global texture model to provide good foreground and background samples for the unknown pixels, and in the proposed UFBPI matting method, we use a direct global search scheme to locate unmarked foreground and background pixels with high confidence. Actually global models and local models can be considered as different ways of answering a fundamental question when solving the matting problem: what is foreground and/or background? Local color models used in color sampling based matting algorithms consider an unknown pixel as foreground or background according to its similarity to nearby definite foreground and background pixels. Local alpha models used in alpha propagation methods models the boundary between foreground and background instead. Global models used in our proposed methods consider an unknown pixel as foreground or background according to whether or not there is a similar pixel --- in terms of texture similarity or patch similarity --- in the whole definite foreground or background regions that can be found. The advantage of using global models over local models is that in complex image global information is more informative and reliable than local information in finding the identity of unknown pixels.

Still, the global models we employed are far from perfect. For the texture synthesis based matting method to perform well, several requirements need to be satisfied. For example, the background texture pattern is better to be homogenous and regular and the definite background regions should provide enough texture samples for synthesizing the textures in the unknown regions. As for the UFBPI matting, if the foreground and background colors are not well separable in the color space, then the thresholding scheme may fail and a lot of noise may be introduced in the final alpha matte.

The ultimate goal of matting research is to develop user-friendly matting systems that have the ability comparable to human perception in differentiating between foreground and background pixels and extracting an accurate alpha matte for mixed pixels. To achieve this goal, there is much more to be done. Some possible direction for the future work would be the refinement of existing local color models and local alpha models, more robust global color model and more intelligent way of combining local and global image models.

References

1. Porter, T. and T. Duff. *Compositing digital images*. in *Proc. of ACM SIGGRAPH*. 1984.
2. Wang, J. and M. Cohen. *Optimized color sampling for robust matting*. in *Proc. of IEEE CVPR*. 2007.
3. Wang, J. and M. Cohen, *An iterative optimization approach for unified image segmentation and matting*, in *Proceedings of ICCV*. 2005. p. 936-943.
4. Levin, A., D. Lischinski, and Y. Weiss. *A closed form solution to natural image matting*. in *CVPR*. 2006.
5. *Knockout user guide*. 2002, Corel Corporation.
6. Chuang, Y.-Y., et al. *A bayesian approach to digital matting*. in *Proceedings of IEEE CVPR*. 2001.
7. Mishima, Y., *Soft edge chroma-key generation based upon hexoctahedral color space*. 1993: U.S. Patent 5,355,174.
8. Ruzon, M. and C. Tomasi. *Alpha estimation in natural images*. in *Proceedings of IEEE CVPR*. 2000.
9. Bai, X. and G. Sapiro. *A geodesic framework for fast interactive image and video segmentation and matting*. in *Proc. of IEEE ICCV*. 2007.
10. Grady, L., et al. *Random walks for interactive alpha-matting*. in *Proceedings of VIIP* 2005.
11. Levin, A., A. Rav-Acha, and D. Lischinski. *Spectral matting*. in *CVPR*. 2007.
12. Sun, J., et al. *Poisson matting*. in *Proceedings of ACM SIGGRAPH*. 2004.
13. Guan, Y., et al. *Easy matting*. in *Proc. of Eurographics*. 2006.
14. Smith, A. and J. Blinn. *Blue screen matting*. in *Proceedings of ACM SIGGRAPH*. 1996.
15. Sun, J., et al. *Flash matting*. in *Proceedings of ACM SIGGRAPH*. 2006.
16. Joshi, N., W. Matusik, and S. Avidan. *Natural video matting using camera arrays*. in *Proc. of ACM SIGGRAPH*. 2006.
17. Chuang, Y.-Y., et al. *Video Matting of Complex Scenes*. in *Proceedings of ACM SIGGRAPH 2002*. 2002.
18. McGuire, M., et al. *Defocus video matting*. in *Proceedings of ACM SIGGRAPH*. 2005.

19. Wyszecki, G. and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*. 1982: John Wiley and Sons, New York, NY.

20. Yang, C., et al. *Improved fast gauss transform and efficient kernel density estimation*. in *Proc. of IEEE ICCV*. 2003.

21. Rother, C., V. Kolmogorov, and A. Blake. *Grabcut - interactive foreground extraction using iterated graph cut*. in *Proceedings of ACM SIGGRAPH*. 2004.

22. He, X. and P. Niyogi. *Locality preserving projections*. in *Proc. of Advances in Neural Information Processing Systems (NIPS)*. 2003.

23. Omer, I. and M. Werman. *Color lines: Image specific color representation*. in *Proc. CVPR*. 2004.

24. Geman, S. and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.* IEEE Trans Pattern Anal Mach Intell, 1984(6): p. 721-741.

25. Li, S., *Markov Random Field Modeling in Computer Vision*. 1995: Springer-Verlag.

26. Agarwala, A., et al. *Interactive digital photomontage*. in *Proceedings of ACM SIGGRAPH*. 2004.

27. Li, Y., et al. *Lazy snapping*. in *Proceedings of ACM SIGGRAPH*. 2004.

28. Szeliski, R., et al. *A Comparative Study of Energy Minimization Methods for Markov Random Fields*. in *European Conference on Computer Vision*. 2006.

29. Weiss, Y. and W. Freeman, *On the optimality of solutions of the max-product belief propagation algorithm in arebitrary graphs.* EEE Trans. on Information Theory, 2001. **47**(2): p. 303-308.

30. *http://www.alphamatting.com/*.

31. Ni, K., S.R. Thiruvenkadam, and T.F. Chan. *Matting Through Variational Inpainting*. in *Proceeding of Signal and Image Processing*. 2007.

32. Kwatra, V., et al., *Texture optimization for example-based synthesis.* ACM Transactions on Graphics (TOG), 2005. **24**(3).

33. Rhemann, C., C. Rother, and M. Gelautz. *Improving Color Modeling for Alpha Matting*. in *BMVC*. 2008.

34. J.Han, et al., *Fast example-based surface texture synthesis via discrete optimization.* The Visual Computer: International Journal of Computer Graphics, 2006. **22**(9): p. 918-925.

35. H.Huang, X.Tong, and W.Wang, *Accelerated Parallel Texture Optimization.* Journal of Computer Science and Technology, 2007. **22**(5).

36. http://www.digitalfilmtools.com/ezmask/.

37. Wang, J. and M. Cohen, *Image and Video Matting: A Survey.* Foundations and Trends in Computer Graphics and Vision, 2007. **3**(2).

38. Rhemann, C., et al. *A Perceptually Motivated Online Benchmark for Image Matting*. in *CVPR*. 2009.