

Image Matting with Transductive Inference

Jue Wang

Adobe Systems, Seattle, WA 98103, USA
juewang@adobe.com

Abstract. Various matting methods have been proposed to isolate objects from images by extracting alpha mattes. Although they typically work well for images with smooth regions, their ability to deal with complex or textured patterns is limited due to their inductive inference nature. In this paper we present a *Transductive Matting* algorithm which explicitly treats the matting task as a statistical transductive inference. Unlike previous approaches, we assume the user marked pixels do not fully capture the statistical distributions of foreground and background colors in the unknown region of the given trimap, thus new foreground and background colors are allowed to be recognized in the transductive labeling process. Quantitative comparisons show that our method achieves better results than previous methods on textured images.

1 Introduction

Using image matting techniques for creating novel composites or facilitating other editing tasks has gained considerable interests from both professionals and consumers. In the matting problem, an observed image I is modeled as a convex combination of a foreground image F and a background image B as $I = \alpha F + (1 - \alpha)B$, and matting techniques try to estimate the alpha matte α (and sometimes with F) from I with the help of additional constraints provided by the user. Once estimated, the alpha matte can be used as a soft mask for applying a variety of object-based editing operations.

Recently proposed matting techniques are capable of generating fairly accurate mattes for images with smooth regions and homogeneous color distributions, as demonstrated in the quantitative studies conducted in [1], [2] and [3]. The test images used in these studies usually contain a single or few dominant foreground colors which remain stable towards the foreground boundary, along with significantly blurred backgrounds. In this case the smoothness assumption on image statistics made in these approaches typically holds, leading to satisfying results.

Unfortunately, as we will demonstrate later, for images containing textured foreground and/or background regions, the performance of these approaches degrades rapidly. The reason is twofold. First, most approaches assume foreground and background colors remain constant or vary smoothly in a local window. This assumption will not hold over strong edges inside the foreground or background region. Second, alpha values are often estimated in an aggressive way in previous approaches. In order to fully capture the fine details of fuzzy objects such as hair

and further, previous methods try to estimate fractional alpha values for all pixels under consideration, which often leads to erroneous mattes. We argue that both limitations come from the inductive inference nature of these approaches.

One way to solve this problem is to always require the user to provide an accurate trimap where most pixels are marked as either foreground or background, and only transparent pixels are marked as unknown. However this is often a labor-intensive process. To improve matting performance over complex images with less accurate trimaps, we treat the matting task as a transductive statistical inference, under the assumption that new foreground and background regions may exist in the unknown region of the given trimap (see Figure 5). These new regions are close, but not equal to user-marked foreground and background regions in some feature spaces. With transductive inference, our method is able to identify these regions and mark them correctly as either definite foreground or background, and only estimate fractional alpha values for real mixed pixels, which is not possible for an inductive inference setting. To the best of our knowledge our method is the first to explicitly solve the matting problem as a transductive inference.

A quantitative evaluation is conducted on different data sets. Experimental results suggest that our algorithm outperforms previous approaches on highly-textured images in terms of both accuracy and robustness.

2 Related Work

Recent image and video matting approaches have been well summarized in a comprehensive survey in [1]. They are classified into three categories, sampling-based, affinity-based, and combined approaches.

Given a user-specified trimap, sampling-based approaches collect a set of nearby known F and B colors, and use them as close approximations of the true F and B colors of unknown pixels, which leaves alpha estimation to be relatively straightforward. Earlier representative sampling-based techniques include Ruzon and Tomasi’s method [4] and Bayesian matting [5]. The recent Robust matting algorithm [2] proposes an improved color sampling procedure to selectively evaluate color samples, which is further improved in [6]. All these methods use color samples in an inductive way: user-specified known pixels are used as training data to build parametric or nonparametric models, and then the models are applied to unknown pixels for alpha estimation. For complex images, if the sampled colors do not represent the true B and F colors of unknown pixels, these methods tend to produce large errors.

Affinity-based approaches define constraints on the gradient of the alpha matte based on local image statistics. Poisson matting [7] estimates the matte by solving a set of Poisson equations. The random walk matting algorithm [8] uses the classic exponential affinity for matting. The geodesic matting technique [9] measures the weighted geodesic distances that a random walker will travel from an unknown pixel to reach the foreground and the background, and use the distance ratio as the alpha value. The closed-form matting [3] derives a matting

Laplacian by assuming that F and B colors are a linear mixture of two colors in a small window, which is used also in the automatic Spectral matting approach [10] and the multi-layer matting system [11]. For complex images with large local color variations, the smoothness assumptions often do not hold, leading to less accurate results.

Combined approaches integrate sampling methods and matting affinities together through an optimization process. Representative techniques include the iterative matting approach [12], Easy matting [13], Robust Matting [2], and the high-res matting system [14]. Although combined approaches often generate higher quality mattes [1], the inductive nature of these approaches limits their performance on complex images, as we will demonstrate later.

Our work is also inspired by recent success on applying transductive inference for image segmentation [15]. This method is based on the Laplacian graph regularizer, and segmentation is modeled as finding a labeling function (alpha matte) which is only allowed to vary in low density areas in the feature space. Although it estimates continuous alpha values in the intermediate step, this algorithm does not accurately model the shape of the matte in the foreground-to-background transition area, thus is not able to generate accurate mattes.

3 Transductive v.s. Inductive Matting

In machine learning tasks, transductive inference is often employed in cases where both labeled (training) and unlabeled (test) data is presented. Since all the data is available, transductive inference algorithms will take this advantage and produce a mapping function in such a way that the statistics of unlabeled data is also respected. In inductive inference, the test data is unknown beforehand, thus the mapping function is designed solely on the training data to map any possible input data to the output space. Obviously, inductive inference has a higher requirement on the “quality” of the training data, or in other words, how well the limited training data can represent the statistical characteristics of the test data. In areas of the feature space where test data may exist but no training data has been collected, inductive inference tends to make mistakes, as visualized in Figure 2 in [15].

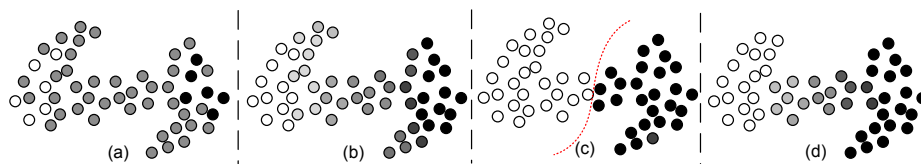


Fig. 1. (a). F (white), B (black) and U (gray) for matting. (b). Previous matting approaches will generate fractional α s for both mixed points and new F and B points. (c). Transductive segmentation generates a binary classification. (d). Our algorithm generates fractional α s for mixed points and also labels new F s and B s correctly.

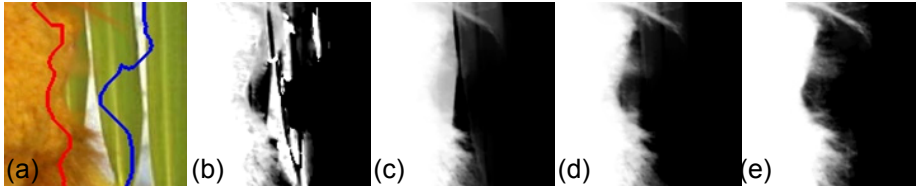


Fig. 2. (a). Input image with trimap boundaries. (b) Bayesian matting. (c). Closed-form matting. (d). our method. (e). ground-truth.

In the matting task, we assume the user has already provided a relatively loose trimap where the majority of pixels have been marked as either F or B as training data, and the rest are marked as U as test data. Since U is known, matting can be treated naturally as a transductive inference, which has a number of advantages over inductive inference when the color distribution in U is complicated and is not fully captured by both F and B . An analytic example is shown in Figure 1. Suppose F , B and U are distributed as in 1(a) in the feature space, note that mixed points (points between the two clusters) as well as new F s and B s are unmarked. Previous matting approaches tend to generate fractional α s aggressively, thus will label new F s and B s with fractional α s (1(b)). Transductive segmentation can label new F s and B s correctly, but is not able to generate correct α s for mixed points (1(c)). Our proposed method can deal with unlabeled data correctly as shown in 1(d).

A real example is shown in Figure 2, which is generated using one of the ground-truth foreground objects in the data set proposed in [2]. In this local region shown in 2(a), the background is highly textured and the white region between green leaves in U is not marked as B (in local sense). Consequently, previous approaches have difficulties to deal with the white region and its edges, as shown in 2(b)-(d). Our method is able to correctly identify the white pixels as new background colors, and generate a matte that is much closer to the ground-truth.

4 The Algorithm

4.1 Optimization Formulation

Our algorithm is designed to explicitly meet the following three objectives:

1. it should be able to identify new F or B colors presented in U ;
2. it should be able to correctly label new F and B colors;
3. it should be able to accurately estimate α s for real mixed pixels in U .

Previous approaches mostly ignore Objective 1 and 2 and only focus on Objective 3. We show here how additional transductive inference can be added to meet all three objectives.

Recall that in the transductive segmentation work [15], the labeling functions f is only allowed to vary in low-density regions in the feature space, and segmentation is modeled as the following optimization problem:

$$\min_f \sum_{i \in \{F, B\}} c_i [Y_i - f(X_i)] + \int_U \|\Delta f\|^2 p^s dV, \quad (1)$$

where the summation is over all known pixels, X_i are feature vectors of unknown pixels and Y_i are user-provided labels. c_i is a positive weight controlling how much we want to trust the known labels, which typically is set to be $+\infty$. The integral term is a *s-weighted Laplacian operator* which only allows f to vary where the density estimation p is low. Given the fact that a direct solution of this optimization cannot be obtained, graph Laplacian methods are used to solve for its discrete approximation:

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i \in \{F, B\}} c_i [Y_i - f(X_i)] + \alpha^t L \alpha, \quad (2)$$

where α is the vector of α values of all pixels, and L is the Laplacian matrix whose coefficients are determined by a kernel function $k(X_i, X_j)$ (for instance a Gaussian kernel), which measures the similarity between two feature vectors X_i and X_j .

However, this approach cannot be directly applied to the matting problem as for fuzzy objects, a large number of pixels may present fractional α s, thus the density estimation p does not necessarily correspond to where the alpha matte should vary. In other words, mixed pixels may form high density regions in the feature space, and directly optimizing Equation 1 will force the alpha matte to stay constant in these regions, resulting in matting errors.

In our algorithm we force f to vary not in low density regions, but in *high density regions of real mixed pixels*. This can be achieved if we have a mixed pixel detector, which for each X_i calculates a probability γ_i , indicating how likely this pixel has a fractional alpha value. With this detector X_i can thus be decomposed into two components: $X_i = \gamma_i X_i^m + (1 - \gamma_i) X_i^b$. Let $X_i^m = \gamma_i X_i$ and $X_i^b = (1 - \gamma_i) X_i$ be two subsets, applying density estimation on subset X_i^b will allow f to vary in the correct regions. Denoting p_b as density of X_i^b , we then replace p in Equation 1 with p_b . Furthermore, if we relax the kernel function and allow each X_i to be associated with a weight w_i , and define the weighted kernel function as $k(X_i, X_j, w_i, w_j) = w_i w_j k(X_i, X_j)$, the discrete approximation of the modified optimization problem becomes:

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i \in \{F, B\}} c_i [Y_i - f(X_i)] + \alpha^t L^b \alpha, \quad (3)$$

where in the Laplacian matrix L^b the similarity between two pixels is computed as:

$$k^b(X_i, X_j) = \tilde{k}^b(X_i, X_j, 1 - \gamma_i, 1 - \gamma_j). \quad (4)$$

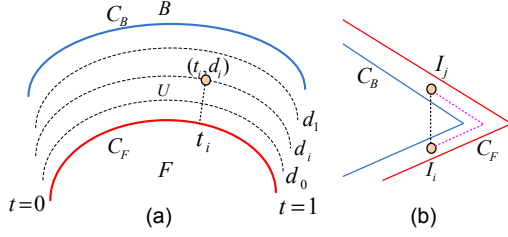


Fig. 3. (a). Parametrization of the unknown region. (b). Respecting sharp corners when computing distance between I_i and I_j .

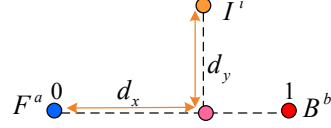


Fig. 4. Linearity analysis among F^a , B^b and I^i .

The solution of the updated optimization problem in Equation 3 will satisfy the Objective 1 and 2, however not 3, since the exactly shape of f is not characterized for mixed pixels. To achieve this we add another term to the optimization problem as

$$\sum_{i \in U} \lambda_i \gamma_i^2 \|\alpha_i - \hat{\alpha}_i\|^2 + \alpha^t L^m \alpha, \quad (5)$$

where $\hat{\alpha}_i$ is the prior alpha value for X_i (data term), L^m is a matting Laplacian defined by a matting kernel function $k^m(X_i, X_j) = \tilde{k}^m(X_i, X_j, \gamma_i, \gamma_j)$. λ_i is a weight to balance the two terms.

Combining 3 and 5, the final optimization problem is defined as

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i \in \{F, B\}} c_i [Y_i - f(X_i)] + \sum_{i \in U} \lambda_i \gamma_i^2 \|\alpha_i - \hat{\alpha}_i\|^2 + \alpha^t (L^b + L^m) \alpha. \quad (6)$$

In our system we usually set $c_i = +\infty$ as we treat the input trimap as a hard constraint, and fix $\lambda_i = 0.1$ as it generates good results in our tests. In next sections we will describe how we compute γ_i , $\hat{\alpha}_i$, and the two kernel functions \tilde{k}^b and \tilde{k}^m .

4.2 Kernel Functions

In our system for a pixel i the feature vector X_i contains three components: the *RGB* color vector I_i , the color level of a local 3×3 patch I_i^p as the texture feature, and a geometric position G_i which is not defined by the absolute (x, y) coordinates of the pixel, but by its relative location in the unknown region parameterized using level set curves, as shown in Figure 3a. We first parameterize the boundary curve of the F region as $C_F(t)$, then apply a distance transform in U to parameterize the region using level set curves. for pixel i , G_i is parameterized as (t_i, d_i) .

This parametrization allows us to compare relative locations of two points in U instead of on the image lattice, thus shape corners of the foreground boundary

can be respected. An illustration is shown in 3b, where two pixels I_i and I_j are on the two sides of a sharp corner. If we use absolute coordinates, the two pixels have a short distance and a strong affinity value, which will encourage α_i and α_j to be the same, thus statistical changes along C_F will not be respected. Using our parametrization these two points will have a much smaller affinity value.

The primary goal of the Laplacian L^b is to classify new F s and B s in a binary sense. We use a weighted Gaussian kernel for it as

$$\tilde{k}^b(X_i, X_j, a_i, a_j) = a_i a_j \exp(-(\|I_i - I_j\|^2/2h_c^2 + \|I_i^p - I_j^p\|^2/2h_p^2 + \|G_i - G_j\|^2/2h_g^2)). \quad (7)$$

To reduce the complexity, similar to [15], we use a truncated version of the Gaussian kernel for the geometric distance, by setting $h_g = 1.25$ and applying a threshold at 0.05. In this way L^b becomes a sparse matrix. h_c and h_p are color and texture variances which can be either fixed as user-specified constants, or computed dynamically using local image statistics as proposed in [16]. We found the latter usually works better when the input image contains both smooth regions and textured regions.

Recall that in [15], the kernel is further normalized as

$$k(X_i, X_j) = \frac{\tilde{k}(X_i, X_j)}{[\tilde{d}(X_i)\tilde{d}(X_j)]^\tau}, \quad (8)$$

where $\tilde{d}(X_i) = \sum_{j=0}^n \tilde{k}(X_i, X_j)$, and $\tau = 1 - s/2$ (s is the free parameter in Equation 1). However, in our system we do not want this normalization to happen since each X_i is associated with a weight, and normalizing the kernel will undesirably cancel out the effects of the weights. We thus set $s = 2$ and $\tau = 0$ for both L^b and L^m .

Denoting W^b as the $n \times n$ matrix where $W_{ij}^b = k^b(X_i, X_j)$ (see Equation 4 and 7), D^b as the diagonal $n \times n$ matrix where $D_{ii}^b = \sum_{j=0}^n k^b(X_i, X_j)$, then L^b is defined as $L^b = D^b - W^b$.

The goal of L^m in the optimization problem 6 is to accurately estimate alpha values for real mixed pixels in U , which have been extensively studied in previous matting approaches. Although the same weighted Gaussian kernel can be defined for L^m as described in [8], the recently proposed matting Laplacian [3] has been shown to be able to generate the most accurate mattes among affinity-based matting approaches [1]. In our system we use this affinity and define $\tilde{k}^m(X_i, X_j, a_i, a_j) = a_i a_j \mu(i, j)$, where $\mu(i, j)$ is the matting Laplacian coefficient defined in Equation 12 in [3]. Similarly, we define W^m as the $n \times n$ matrix where $W_{ij}^m = k^m(X_i, X_j) = \tilde{k}^m(X_i, X_j, \gamma_i, \gamma_j)$, D^m as the diagonal $n \times n$ matrix where $D_{ii}^m = \sum_{j=0}^n k^m(X_i, X_j)$, and L^m as $L^m = D^m - W^m$.

4.3 Estimation of γ_i and $\hat{\alpha}_i$

Recall the convex combination assumption of the matting problem: $I = \alpha F + (1 - \alpha)B$. Under this assumption, and given a relatively tight input trimap (compared

with a few scribbles), we assume that if a pixel I_i can be well approximated as a linear combination of a known foreground color \hat{F} and background color \hat{B} , then it has a higher probability to be a mixed pixel. Similar to the sampling scheme proposed in [2], for an unknown pixel I_i , we sample a relatively large number of nearby foreground and background colors $F^k, B^k, i = 1, \dots, M$, and try to find a good linear approximation of I_i among them.

Specifically, for a sample pair $(F^a, B^b)(a, b \in [1, M])$, we first normalize the distance $|F^a - B^b|$ to 1 and align B^b to $(0, 0)$ and F^a to $(1, 0)$ in the 2D plane defined by the three points F^i, B^j and I_i in the 3D color space, as shown in Figure 4. We then compute the coordinates of I_i in this plane as (d_x, d_y) , and compute an estimated α and a mixture probability γ as

$$\hat{\alpha}(F^a, B^b, I_i) = \Gamma(d_x), \quad (9)$$

$$\gamma_{a,b,i} = P(\hat{\alpha}) \cdot \exp\left(-\frac{\delta(d_y - \varepsilon_y)(d_y - \varepsilon_y)}{\sigma_y}\right), \quad (10)$$

where $\Gamma(x)$ is a truncation function whose output is 1 if $x > 1$, 0 if $x < 0$, and x otherwise. $\delta(x)$ is a standard step function where $\delta(x) = 1$ for $x \geq 0$ and $\delta(x) = 0$ otherwise. ε_y and σ_y are two constants which are empirically chosen as $\varepsilon_y = 0.1$ and $\sigma_y = 0.025$, which generate good results in our tests. Intuitively, if I_i is closer to the line, which means d_y value is smaller, then γ is higher, indicating the three points can be better approximated using a line in the color space. $P(\hat{\alpha})$ is a weighting function defined as

$$P(\hat{\alpha}) = 4\hat{\alpha}(1 - \hat{\alpha}). \quad (11)$$

$P(\hat{\alpha})$ has its maximal value of 1 at $\hat{\alpha} = 0.5$ and gradually goes to 0 as $\hat{\alpha}$ approaches either 0 or 1. The intuition for applying such a weighting function is that if $\hat{\alpha}$ is closer to 0 or 1, I_i is closer to known F and B and actually has a higher probability to be a new foreground or background color.

We do this analysis for every pair of (F^a, B^b) , and the top three pairs are chosen which generate the highest $\gamma_{a,b,i}$, and their average γ_i and $\hat{\alpha}_i$ are computed as the final results for I_i at the color sampling step.

Finally, individually estimated γ_i is still somewhat noisy, since there is no spatial smoothness constraint in γ_i estimation. However, for two neighboring pixels I_i and I_j , if their colors are similar, then their mixture probabilities γ_i and γ_j should also be close. To generate a smoother mixture map which respects the local image statistics, we apply the matting affinity proposed in [3] as a spatial smoothness constraint for the mixture map, and use the matting Laplacian coefficients defined in that approach as smoothing weights between neighboring pixels. Mathematically, the smoothing operation is applied as

$$\gamma_i^{t+1} = (1 - \lambda_s)\gamma_i^t + \lambda_s \sum_{j \in N(i)} (\mu(i, j) \cdot \gamma_j^t) / \sum_{j \in N(i)} \mu(i, j), \quad (12)$$

where $N(i)$ is a 3×3 window centered at i . $\mu(i, j)$ are coefficients in the matting Laplacian matrix. t stands for smoothing iteration, which is fixed to be 20 in our system. λ_s is the step width parameter which is set to be 0.5.



Fig. 5. Example of estimated mixture maps. Each example from left to right: original image with trimap boundaries, γ_i before adaptive smoothing, γ_i after adaptive smoothing.

Figure 5 shows some examples of estimated mixture probability maps. Note how the mixture maps capture the real foreground edge for near-solid boundaries (first row) as well as large fuzzy regions (bottom row).

4.4 Iterative Refinement

One may have noticed that the mixture map estimation largely depends on the available F and B training data. After the optimization problem in Equation 6 is solved as a large linear system, some pixels in the unknown region may have been classified as F and B , giving us new F and B samples which could be used to refine the mixture map estimation. In this way the whole process can be iterated until convergence. The convergence is guaranteed since the upper limit of the number of possible new F and B samples is all the pixels in U , and in practice we found the matte usually becomes stable after 2 to 3 iterations.

5 Link with Other Approaches

Many previous matting and segmentation approaches can be treated as special cases of the proposed algorithm. If we simply set $\gamma_i = 1$ everywhere in U , then L^b becomes an empty matrix and our algorithm degrades to a regular matting algorithm, which shares similar components with the state-of-the-art matting algorithms. For example, L^m incorporates the matting Laplacian [3], and the matte prior $\hat{\alpha}$ is computed in a similar way as in [2]. On the contrary, if we set $\gamma_i = 0$ for all pixels, then L^m becomes an empty matrix and the algorithm degrades to a transductive segmentation algorithm which is similar to the one proposed in [15]. By automatically varying γ_i at different regions, our algorithm combines the advantages of transductive labeling and matting together, thus is able to generate accurate alpha mattes in a more robust way.

Some tri-level segmentation algorithms have been recently proposed which are able to generate relatively accurate trimaps based on user-specified scribbles [17, 14]. These approaches usually build color models not only for F and



Fig. 6. Six test images containing textured backgrounds, with trimap boundaries overlaid and ground-truth mattes.

	Bayesian	Clo.form	Robust	Our
T1	793.2	346.7	392.0	94.6
T2	2395.2	451.2	420.7	280.9
T3	263.0	123.0	152.7	79.6
T4	1786.5	401.9	331.4	117.3
T5	3233.0	339.6	320.1	216.6
T6	971.9	106.5	91.8	55.7

Fig. 7. MSE of mattes generated by different algorithms on the data set in Figure 6.

B , but also for U by linearly blending F and B models, thus are similar to the linear mixture analysis proposed in our approach on the concept level. However, our approach differs from these approaches from two major aspects. First, in trimap generation systems the trimap generation and alpha matting are treated as separate steps, thus any errors in trimap generation will be magnified in the matting step. In our system the matting and transductive labeling are integrated together and they help each other. Second, trimap generation methods mostly use inductive inference, relying on the user to provide enough color samples to construct the proper statistical models (for instance Gaussian Mixtures). In our system the new F and B labeling is done under a more robust transductive inference framework.

Nevertheless, one can imagine integrating these techniques together to build a more efficient system. Given an input image, a trimap can be interactively generated using trimap segmentation algorithms. Since the resulting trimap will not be perfect where U region may still contain some F and B colors, our algorithm can be applied to improve the matting quality, especially for complex images.

6 Evaluations and Comparisons

To quantitatively evaluate the algorithm, a test data set is constructed which, unlike data sets used in previous approaches, contains highly-textured backgrounds, as shown in Figure 6. For image $E5$ and $E6$ we shoot the foreground dolls against multiple known backgrounds, and use triangular matting methods [18] to extract the ground-truth mattes. The foregrounds and ground-truth mattes in $E1$ to $E4$ are borrowed from the data sets in [2] and [3], but we compose them onto more complicated backgrounds to create test images. Note that the data set contains both hairy foreground objects and near-solid ones. For each example a relatively loose trimap is specified as the user input.

Four algorithms are applied on the test data set, including Bayesian matting [5], closed-form matting [3], Robust matting [2], and the proposed transductive matting algorithm. Specifically, Bayesian matting is chosen as a representative sampling-based approach, closed-form matting as the most accurate

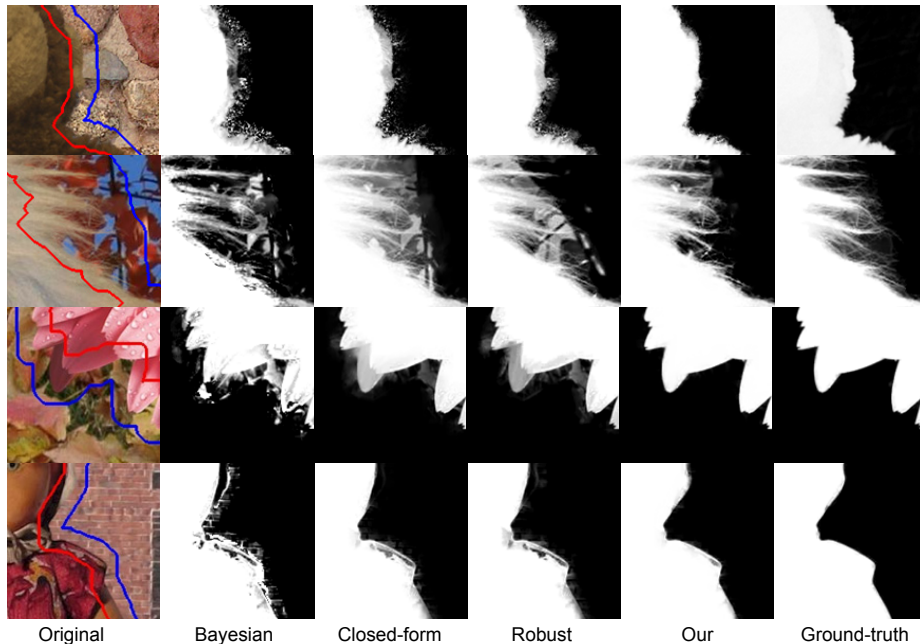


Fig. 8. Partial mattes extracted by different algorithms.

Table 1. MSE of mattes generated by our algorithm on the data set in [1], and their ranks among total of 8 test systems. Format: $Min^{rank} : Max^{rank}$.

T1	T2	T3	T4	T5	T6
$58.9^2 : 93.5^2$	$51.6^2 : 142.3^2$	$41.8^2 : 70.5^2$	$74.7^1 : 248.5^3$	$154.2^2 : 355.3^1$	$36.5^3 : 47.6^1$

affinity-based method, and robust matting as a well-balanced optimization-based algorithm which combines sampling and affinities.

Figure 7 shows the Mean Squared Errors (MSE) of extracted mattes against the ground-truth. Alpha values are stretched to 0 – 255 for MSE calculation. Figure 8 and 2 shows partial mattes generated by different algorithms. There results clear suggest the proposed algorithm outperforms previous approaches on these complex images.

To evaluate the performance of the proposed algorithm on simpler images with smooth F and B regions, we apply it on the test data set proposed in [1], which contains 6 test images, each with a ground-truth matte and a series of trimaps. Table 1 shows the MSE values of our extracted mattes, and their ranks comparing with the other 7 matting algorithms. The results suggest that the proposed algorithm performs comparably well with other matting methods when the input image does not contain complex textures.

7 Conclusion

Previous inductive-inference-based matting algorithms tend to produce erroneous mattes when dealing with textured objects. In this paper we propose a transductive matting algorithm which explicitly models the trimap-based matting task under a transductive inference framework, thus not only is able to produce higher quality results on textured or non-homogeneous images, but also can produce accurate mattes for regular images with smooth foreground and background regions.

References

1. Wang, J., Cohen, M.: Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision* **3** (2007) 97–175
2. Wang, J., Cohen, M.: Optimized color sampling for robust matting. In: *Proc. of IEEE CVPR*. (2007)
3. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. *IEEE Trans. Pattern Analysis and Machine Intelligence* **30** (2008) 228–242
4. Ruzon, M., Tomasi, C.: Alpha estimation in natural images. In: *Proceedings of IEEE CVPR*. (2000) 18–25
5. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: *Proceedings of IEEE CVPR*. (2001) 264–271
6. Rhemann, C., Rother, C., Gelautz, M.: Improving color modeling for alpha matting. In: *Proc. of BMVC*. (2008)
7. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. In: *Proceedings of ACM SIGGRAPH*. (2004) 315–321
8. Grady, L., Schiwietz, T., Aharon, S., Westermann, R.: Random walks for interactive alpha-matting. In: *Proceedings of VIIP 2005*. (2005) 423–429
9. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: *Proc. of IEEE ICCV*. (2007)
10. Levin, A., Rav-Acha, A., Lischinski, D.: Spectral matting. In: *Proc. of IEEE CVPR*. (2007)
11. Singaraju, D., Vidal, R.: Interactive image matting for multiple layers. In: *Proc. of IEEE CVPR*. (2008)
12. Wang, J., Cohen, M.: An iterative optimization approach for unified image segmentation and matting. In: *Proceedings of ICCV 2005*. (2005) 936–943
13. Guan, Y., Chen, W., Liang, X., Ding, Z., Peng, Q.: Easy matting. In: *Proc. of Eurographics*. (2006)
14. Rhemann, C., Rother, C., Rav-Acha, A., Sharp, T.: High resolution matting via interactive trimap segmentation. In: *Proc. of IEEE CVPR*. (2008)
15. Duchenne, O., Audibert, J.Y.: Segmentation by transduction. In: *Proc. of IEEE CVPR*. (2008)
16. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: *Proc. of IEEE CVPR*. (2001)
17. Juan, O., Keriven, R.: Trimap segmentation for fast and user-friendly alpha matting. In: *Proc. of IEEE Workshop on VLISM*. (2005)
18. Smith, A., Blinn, J.: Blue screen matting. In: *Proceedings of ACM SIGGRAPH*. (1996) 259–268