Optimized Color Sampling for Robust Matting

Jue Wang
Electrical Engineering
University of Washington
Seattle, WA 98195, USA
juew@u.washington.edu

Michael F. Cohen
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
Michael.Cohen@microsoft.com

Abstract

Image matting is the problem of determining for each pixel in an image whether it is foreground, background, or the mixing parameter, "alpha", for those pixels that are a mixture of foreground and background. Matting is inherently an ill-posed problem. Previous matting approaches either use naive color sampling methods to estimate foreground and background colors for unknown pixels, or use propagation-based methods to avoid color sampling under weak assumptions about image statistics. We argue that neither method itself is enough to generate good results for complex natural images.

We analyze the weaknesses of previous matting approaches, and propose a new robust matting algorithm. In our approach we also sample foreground and background colors for unknown pixels, but more importantly, analyze the confidence of these samples. Only high confidence samples are chosen to contribute to the matting energy function which is minimized by a Random Walk. The energy function we define also contains a neighborhood term to enforce the smoothness of the matte. To validate the approach, we present an extensive and quantitative comparison between our algorithm and a number of previous approaches in hopes of providing a benchmark for future matting research.

1. Introduction

Matting refers to the problem of soft and accurate foreground extraction from an image. The results play an important role in image and video editing. Specifically, an input image C is modelled as a convex combination of a foreground image F and a background image B as

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

where $z = (x, y)$ refers to the image lattice, and α_zs are the pixels' foreground opacities. The problem is underconstrained since F, B and α are all unknown.

Although the problem is severely ill-posed, the strong correlations between nearby image pixels can be leveraged to alleviate the difficulties. Roughly speaking, previous matting approaches can be classified into two categories based on how they make use of natural image statistics: *sampling-based* approaches and *propagation-based* ones. Sampling-based methods assume that the foreground and background colors of an unknown pixel can be explicitly estimated by examining nearby pixels that have been specified by the user as foreground or background. These color samples are then used to directly estimate the alpha value. Propagation-based methods do not explicitly estimate foreground and background colors, instead they assume foreground and background colors are locally smooth, for example, that they can be modelled as constant or linearly varying. In this way foreground and background colors can be systematically eliminated from the optimization process and the matte can be solved in a closed form.

Although various successful examples have been shown for these approaches, their performance rapidly degrades when foreground and background patterns become complex. The intrinsic reason is that in many images the foreground and background regions contain significant textures and/or discontinuities; thus direct color sampling may be erroneous, as well as only fitting low-order models to them.

In this paper we propose a robust matting algorithm to explicitly avoid these limitations as much as possible. The kernel of our algorithm contains a robust color sampling method, which not only estimates foreground and background colors for unknown pixels, but also self-evaluates the confidences of the estimates. Combining the optimized color sampling method with propagation-based approaches, we propose an iterative optimization process to select truly mixed pixels from all the unmarked ones, and estimate alpha values for them in closed form at each iteration.

Another limitation of previous matting research is the lack of systematic comparisons of various methods. In this paper we conduct a quantitative, objective and comprehensive comparison between this algorithm and a number of

previous approaches. Results show our algorithm performs the best in terms of both robustness and accuracy.

2. Related Work

2.1. Sampling-based Approaches

Sampling representative foreground and background colors and analyzing their statistics for image matting was first proposed in [8]. Following this idea, the KnockOut2 system [2] extrapolates known foreground and background colors into the unknown region to estimate alphas. Ruzon and Tomasi [10] analyze the statistical distributions of foreground and background samples for alpha estimation. Their approach is improved by the Bayesian matting system [1], which formulates the problem in a well-defined Bayesian framework and solves it using the MAP technique. The recently proposed Belief Propagation matting system [14] and easy matting system [6] solve a matte directly from a few user specified scribbles instead of a carefully specified trimap. Since the user input is very sparse in this case, global sampling methods are proposed to assist the local sampling procedure to generate enough color samples. We will demonstrate later that all these approaches often suffer from inaccurate foreground and background estimation.

2.2. Propagation-based Approaches

Propagation-based approaches solve the matte without explicit foreground and background color estimation. The Poisson matting algorithm [12] assumes the foreground and background colors are smooth in a narrow band of unknown pixels. Thus, the gradient of the matte matches with the gradient of the image, which can be calculated by solving Poisson equations. A similar method based on Random Walks is proposed in [5]. The Closed-form matting [7] approach assumes foreground and background colors can be fit with linear models in local windows, which leads to a quadratic cost function in alpha that can be minimized globally.

2.3. Extensions

Instead of a single input image, additional information can be used for reducing unknowns in matte estimation if available, such as using multiple backgrounds [11], or flash and non-flash image pairs [13]. In this paper we only consider the general problem of matte estimation from a single image with limited user input.

3. Failure Modes for Previous Approaches

Before introducing the proposed robust matting algorithm, we first analyze why previous matting approaches fail in the context of complex foreground and background patterns or sparse user input. These insights will motivate the new algorithm in the following sections.

For an unknown pixel, sampling-based approaches collect a group of nearby foreground and background colors for alpha estimation. As illustrated in Figure 1a, these samples form clusters in color space, and the alpha value is estimated by projecting the pixel under consideration onto the line between foreground and background cluster centers to fit a linear model as in Equation 1. In the figure, we can see that pixel P_A fits the linear model very well, thus it has a high probability of being a true mixed pixel between the foreground and background clusters. On the contrary, pixel P_B is far away from the interpolation line thus it is very unlikely to be generated by a linear combination of the two clusters. Such a pixel is more likely to be an unmarked foreground or background pixel. Unfortunately, previous approaches ignore this fact and simply estimate an alpha value for P_B based on its projection to the line P'_B.

For complex foreground and background patterns, samples collected from local regions do not have a uniform color distribution, as shown in Figure 1b. In this case, propagation-based approaches will fail (at least partially) since the smoothness assumption is violated. For sampling-based methods which treat each sample equally, such as the Belief Propagation matting system [14], these samples will produce erroneous alpha values. Although Bayesian matting [1] will try to fit multiple Gaussians to color samples, it is still insufficient to determine the specific color samples (F' and B' in Figure 1b) that best explain the observed pixel, and thus should be used for alpha estimation.

Figure 2 demonstrates these limitations on a real image. Figure 2a shows the original image with the user specified foreground (red) and unknown region (yellow). Note that although the foreground color is relatively uniform, the background contains complex patterns, in which dark regions match well with the foreground color. Figure 2 b, c, and d show that Bayesian matting, Belief Propagation matting and closed-form matting will produce noticeable artifacts due to color sampling errors or assumption violations. In contrast, the robust matting approach proposed in this paper is able to generate a good matte.

4. Robust Matting

We propose a new matting algorithm to counter the sampling problems faced by previous approaches. Our algorithm is based on an optimized color sampling scheme, as we believe (and shown by experimental results) that this non-parametric technique is more robust for natural images. A Random Walk optimizer is employed to solve for the matte.

4.1. Optimized Color Sampling

Given the input image and a roughly specified trimap, for a pixel z with unknown α, our algorithm first assem-

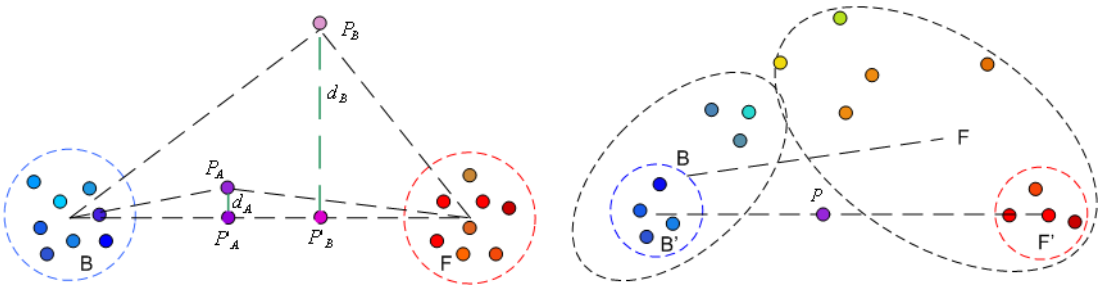

Figure 1. Two failure modes of previous matting approaches. Left: a linear blend of samples does not explain the color observed at P_B. Right: only a particular subset of samples are valid for alpha value estimation.

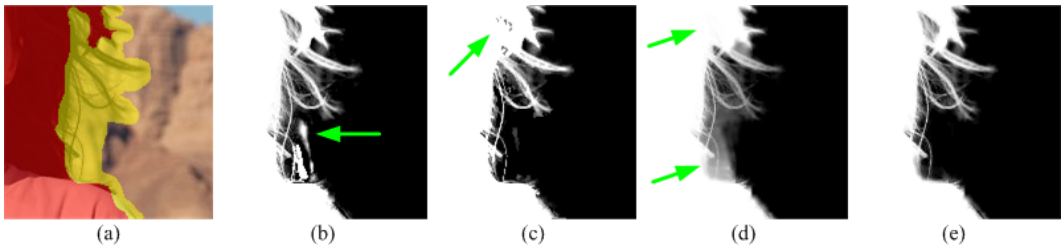

Figure 2. (a). Original image with user input. Following images show matte extracted by (b). Bayesian matting[1]. (c). Belief Propagation matting[14]. (d). Closed-form matting[7]. (e). Our algorithm. Green arrows highlight artifacts.

bles a large number of foreground and background samples as candidates for estimating the true foreground and background colors at this location. Well discuss how these samples are drawn later. We make the assumption that for any mixed pixels, the true foreground and background colors F_z and B_z are close (in color space) to some samples in this large sample set.

The challenging task is to pick out "good" samples from this large candidate set. Good sample pairs should explain any mixed foreground/background pixels as linear combinations of the samples. For example, as shown in Figure 1a, two samples define a line in color space. If this line passes through (or near) the color of the pixel under consideration, it successfully explains the pixel's color as a convex combination. Specifically, for a pair of foreground and background colors F^i and B^j, the estimated alpha value is

$$\hat{\alpha} = \frac{(C - B^j)(F^i - B^j)}{\| F^i - B^j \|^2} \tag{2}$$

We define a *distance ratio* $R_d(F^i, B^j)$, which evaluates this sample pair by examining the ratio of the distances between (1) the pixel color, C, and the color it would have, $\hat{C}$, predicted by the linear model in Equation 1, and (2) the distance between the foreground/background pair:

$$R_d(F^i, B^j) = \frac{\| C - (\hat{\alpha}F^i + (1 - \hat{\alpha})B^j) \|}{\| F^i - B^j \|} \tag{3}$$

In the example shown in Figure 1a, the distance ratio will be much higher for P_B than P_A, indicating the samples are not as good for estimating alpha for P_B.

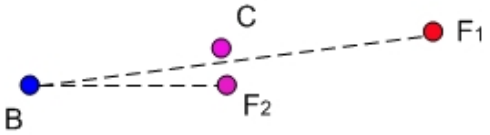

Figure 3. The distance ratio defined in Equation 3 may not define the best sampling. The proximity of samples should also be taken into account. For example, for C, because of proximity in color space, F_2 is a better foreground sample even though F_1 creates a better fit for the linear blend model.

The distance ratio alone will favor sample pairs that are widely spread in color space since the denominator $\| F^i - B^j \|$ will be large. Since we expect most pixels to be fully foreground or background, pixels with colors that lie nearby in color space to foreground and background samples are more likely to be fully foreground or background themselves. Thus, for each individual sample we define two more weights $w(F^i)$ and $w(B^j)$ as

$$w(F^i) = exp\left\{- \| F^i - C \|^2 / D_F^2\right\} \tag{4}$$

and

$$w(B^j) = exp\left\{- \| B^j - C \|^2 / D_B^2\right\} \tag{5}$$

where D_F and D_B are the minimum distances between foreground/background sample and the current pixel, i.e., $\min_i(\| F^i - C \|)$ and $\min_j(\| B^j - C \|)$.

Combining these factors, we calculate a final *confidence*

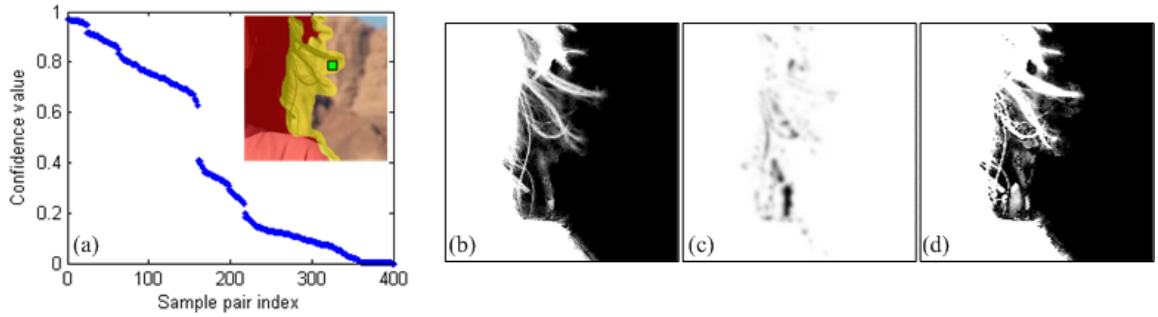

Figure 4. (a). Sorted confidence values for 400 foreground-background sample pairs for green pixel. (b). Estimated alpha values from pairs with highest confidences. (c). Confidence values. White = higher confidence. (d). Estimated alpha values from lowest confidence pairs.

value $f(F^i, B^j)$ for a sample pair as

$$f(F^i, B^j) = exp\left\{-\frac{R_d(F^i, B^j)^2 \cdot w(F^i) \cdot w(B^j)}{\sigma^2}\right\} \quad (6)$$

where σ is fixed to be 0.1 in our system.

We examine the confidence of every pair of foreground and background samples from the large number of candidates. Finally, we select a small number of pairs (3 in our system) with the highest confidences. The average estimated alpha value and confidence of these three sample pairs are taken as the final values in the color sampling step. These values are used in the optimization process described later to generate the final matte for the image.

In Figure 4a, we first selected 20 foreground and 20 background samples corresponding to the highlighted pixel. This results in 400 foreground/background pairs. Each pair provides an estimated alpha and confidence. The confidence values are sorted and plotted in the figure.

By selecting those pairs with highest confidence values, we generate the initial alpha matte shown in Figure 4b, which will be further improved by the optimization process. Figure 4c shows the accompanying confidence map (i.e., the average of the three highest confidence values). Note the correlation between pixels whose alpha values are mis-estimated and the dark (low confidence) regions in the confidence map. For comparison, Figure 4d shows a much poorer matte generated by using sample pairs with lowest confidences.

4.2. Collecting the Sample Set

Finally, one remaining question is how to construct the initial sample set. Previous approaches such as Bayesian matting and Belief Propagation matting collect pixels known to be fully foreground or background that have the shortest spatial distances to the target pixel as samples. As shown in Figure 5, in regions with complex structure, the spatially nearest pixels may not fully span the variation in foreground and background colors.

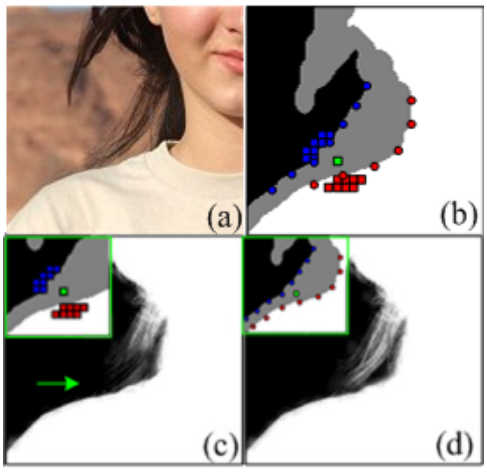

Figure 5. (a). Original image. (b). Sampling methods: nearest spatial (square) vs. sparse samples (round). (c). Matte generated with square samples. (d). Matte generated with round samples.

We thus spread the sampling of foreground and background samples along the boundaries of known foreground and background regions. In this way the sample set can better capture the variation of foreground and background colors. The higher variation benefits the robust sampling method outlined above but may be problematic for previous approaches that simply average over the full sample set. Figure 5 shows that a sample set we collect performs better at least for this example.

4.3. Matte Optimization

As we have seen, the sampling process leads to a good initial alpha estimate and a confidence value for each pixel. This initial estimate can be further improved by leveraging a priori expectations about more global aspects of the alpha matte. In particular, we expect the matte to exhibit local smoothness. We also expect alpha values of one or zero (fully foreground or background) to be much more common than mixed pixels. As we can see in Figure 4b and c, the pixels with low confidence values are those unmarked

foreground and background pixels.

Our expectation for the matte is thus two fold: firstly, it should respect the alphas chosen for each individual pixel (*data constraint*) especially when the confidence value is high; secondly, the matte should be locally smooth and robust to image noise (*neighborhood constraint*). As shown in previous graph-based image labelling approaches [14, 9], this expectation can be satisfied by solving a graph labelling problem shown in Figure 6, where Ω_F and Ω_B are virtual nodes representing pure foreground and pure background, white nodes represent unknown pixels on the image lattice, and light red and light blue nodes are known 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 neighboring pixels to enforce the neighborhood constraint.

The data weights correspond to relative probabilities of a node being foreground or background. For nodes for which we have high confidence values, $\hat{f}_i$, we rely on the alpha that fits the linear model from the selected samples. When the confidence is low, we have a higher expectation that the node is fully foreground or background (i.e., alpha is either 1 or 0). Which one to bias alpha towards is determined by the initial alpha estimate.

Specifically, as shown in Figure 6, for an unknown pixel i, two data weights $W(i,F)$ and $W(i,B)$ are assigned to the links between pixel i and each virtual node. They are defined as

$$W(i,F) = \gamma \cdot [\hat{f}_i \hat{\alpha}_i + (1-\hat{f}_i)\delta(\hat{\alpha}_i > 0.5)]$$

and

$$W(i,B) = \gamma \cdot [\hat{f}_i (1-\hat{\alpha}_i) + (1-\hat{f}_i)\delta(\hat{\alpha}_i < 0.5)] \quad (7)$$

where $\hat{\alpha}_i$ and $\hat{f}_i$ are estimated alpha and confidence values, and δ is boolean function returning 0 or 1.

γ is a free parameter in our system which balances the data weight and the edge weight. In general, setting the weight to be too low will result in an over-smoothed matte, while setting it to be too high will generate a noisy one. We set $\gamma = 0.1$ in our system for all examples.

We also need to specify the weights, $W_{i,j}$, between nodes to encourage alpha to have local smoothness. The closed-form matting system [7] sets the weights between neighboring pixels based on their color difference computed from local color distributions. In this work, we also choose to use this formulation for edge costs W_{ij} due to its simplicity and efficiency.

Formally, the neighborhood term, W_{ij}, is defined by a sum over all 3×3 windows that contain pixels i and j.

$$W_{ij} = \sum_{k}^{(i,j)\in w_k} \frac{1}{9}(1 + (C_i - \mu_k)(\Sigma_k + \frac{\epsilon}{9}I)^{-1}(C_j - \mu_k)) \quad (8)$$

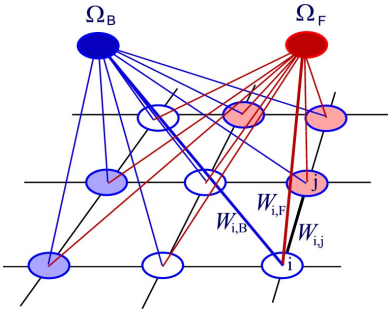

Figure 6. Our system estimates the matte by using a Random Walk to solve a graph labelling problem.

where w_k represents the set of 3×3 [1] windows containing pixels i and j, and k iterates over those windows. μ_k and Σ_k are the color mean and variance in each window. ϵ is a regularization coefficient which is set to be 10^{-5} in our system. More details and justifications of this neighborhood term can be found in [7].

4.3.1 Solving for Optimal αs

Given the fact that alpha values are continuous, we avoid discrete labelling optimizations such as graph cut. Instead, we solve the graph labelling problem as a Random Walk [3, 4], which has been shown to minimize the total graph energy over real values. Although a detailed description of Random Walk theory is beyond the scope of this paper, it essentially works as follows.

First, we construct a Laplacian matrix for the graph as

$$L_{ij} = \begin{cases} W_{ii} & : \text{ if } \quad i = j, \\ -W_{ij} & : \text{ if i and j are neighbors,} \\ 0 & : \text{ otherwise,} \end{cases} \quad (9)$$

where $W_{ii} = \sum_j W_{ij}$. L is thus a sparse, symmetric, positive-definite matrix with dimension $N \times N$, where N is the number of all nodes in the graph, including all pixels in the image plus two virtual nodes Ω_B and Ω_F.

We then decompose L into blocks corresponding to unknown nodes P_u, and known nodes P_k including user labelled pixels and virtual nodes, as

$$L = \begin{bmatrix} L_k & R \\ R^T & L_u \end{bmatrix} \quad (10)$$

It has been shown [4] that the probabilities of unknown pixels belonging to a certain label (for example, foreground) is the solution to

$$L_u A_u = -R^T A_k \quad (11)$$

where A_u is the vector of unknown alphas we wish to solve for, and A_k is the vector encoding the boundary conditions,

[1] The window size can be potentially varied but 3×3 window generated the best results in our tests.

Figure 7. Top: test images. Bottom: ground-truth mattes.

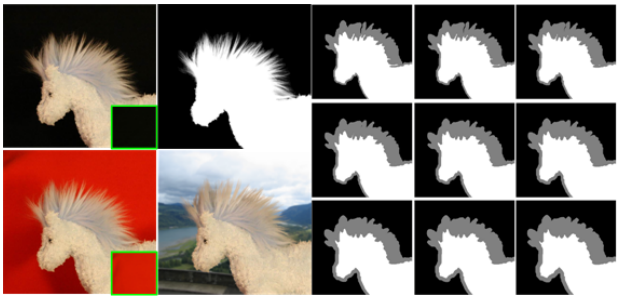

Figure 8. Left: same foreground against two known backgrounds. Middle: extracted matte and new composite. Right: fine-to-coarse trimaps.

(i.e., 1's and 0's for the known alpha values of the virtual and user specified nodes). L_u is guaranteed to be nonsingular for a connected graph, thus the solution A_u is guaranteed to exist and be unique with values guaranteed to lie between 0 and 1. We use Conjugate Gradient (CG) to solve the linear system.

5. A Quantitative Evaluation

Previous matting papers only show a few successful mattes to verify their methods. Although limited quantitative comparisons have been shown in [7], there is no objective and quantitative evaluation on how robust these methods to various user inputs. In this paper we present such an evaluation and comparison.

5.1. Materials and Methods

Our test set includes 8 test images along with *ground-truth*[2] mattes, as shown in Figure 7. Images T1 to T3 are generated using the blue screen matting technique described in [11]. As shown in Figure 8, a fuzzy foreground is shot against two known backgrounds, which enables us to use triangulation matting techniques to pull out an accurate matte. The matte is then used composited over a new image which serves as the test image. Images T4 to T6 use real people and animals as foregrounds. Only one of each of these images were available, however over a simple solid-color background, thus an easy matting problem. In this case, we used the Bayesian matting technique to extract the initial matte and create a synthetic composite for the test.

To evaluate how the algorithms perform on real natural images in addition to synthetic ones, we have included two natural images T7 and T8. Since the ground-truth mattes are harder to derive in this case, we generate the best possible approximations using a variety of methods along with extensive user assistance. We first manually create a very accurate trimap for each image. We then apply previous matting algorithms to get the best matte from each individual algorithm. We then manually combine these mattes by using the best matte regions of all candidates. Finally, a manual correction process which is similar to the one proposed in Poisson Matting system [12] is employed for further improvement.

To evaluate the accuracy and robustness of different algorithms, we first create a fairly accurate trimap T_0 for each test image, then dilate the unknown region gradually to create a series of less accurate trimaps $T_1, T_2, ..T_8$. We apply a number of previous algorithms on each test image and each trimap, and calculate the Mean Squared Error (MSE) of the estimated mattes against the ground-truth. In this way we can quantitatively compare different methods.

We compare the performances of 7 algorithms: Bayesian matting [1], Belief Propagation matting [14], Global Poisson matting [12], Random Walk matting [5], KnockOut 2 [2], closed-form matting [7], and the robust algorithm proposed in this paper. Note that some approaches such as Belief Propagation matting and closed-form matting have the added ability to work with user defined scribbles. However, as shown in their examples, the scribbles must be carefully placed in order to get good mattes. A roughly specified trimap, which can be drawn with a single fat stroke, creates a similar burden for the user as creating scribbles and thus stands in for user scribbles. This allows us to test all algorithms in a uniform way.

[2]We use the term ground-truth to indicate the best matte we could extract using both analytic and extensive user input as described.

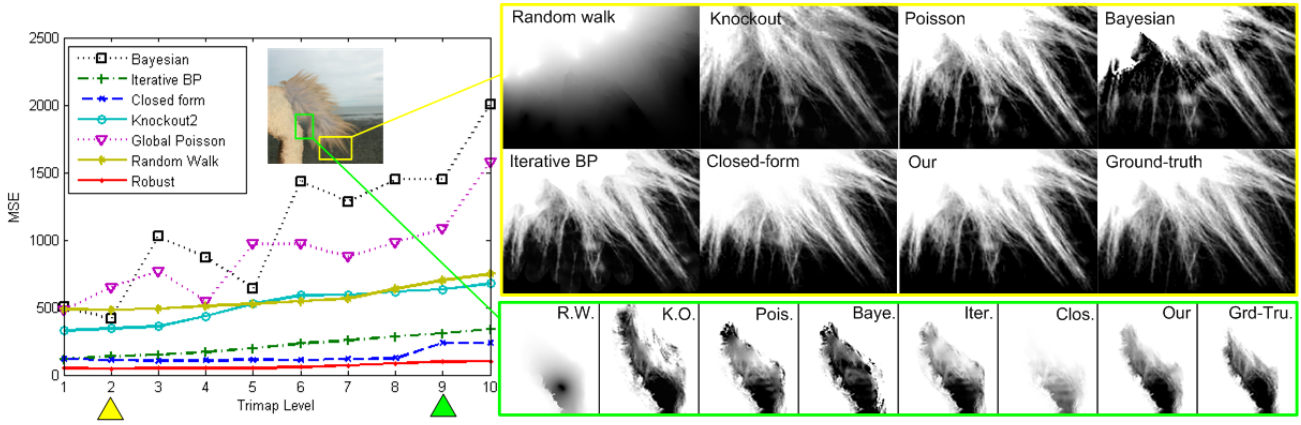

Figure 9. Left: MSE curves for test image T2. Right: Partial mattes computed at two trimap levels (indicated by colored triangular marks).

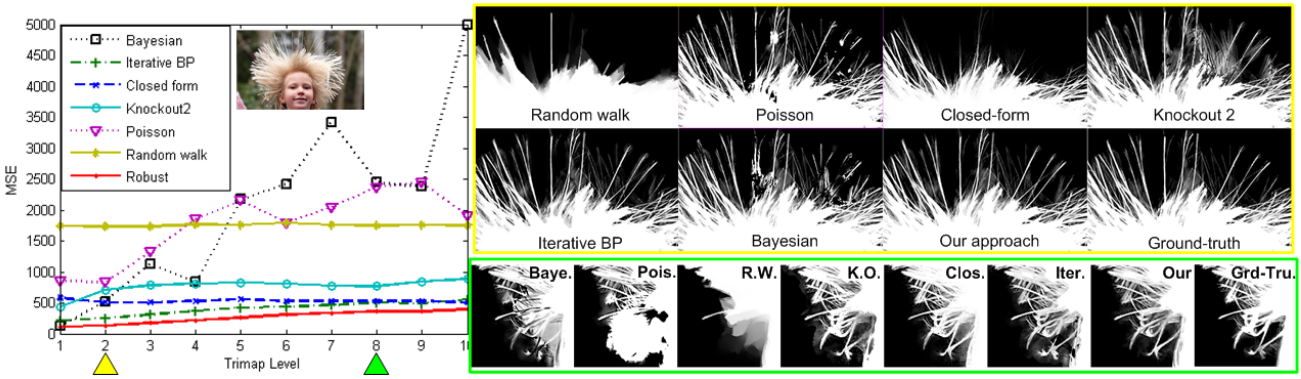

Figure 10. Left: MSE curves for test image T8. Right: Mattes computed at two trimap levels

5.2. Results and Discussion

Figure 9 shows the MSE curves of different algorithms for one test image. As expected, most algorithms achieve their best performance with the finest trimap. As the trimaps becomes coarser, their performances generally degrade, but at different rates. We also show partial mattes at two trimap levels (2 and 8) since MSE values do not always correlate exactly with visual quality. Our algorithm gives the best result at all trimap levels, both quantitatively and visually. Figure 10 shows the result on another test image.

We define two numerical indicators I_a and I_r to measure the accuracy and robustness of different algorithms, respectively. For an algorithm and a test image, we compute a series of MSE values using different trimaps, and define I_a as the minimal MSE value, and I_r as the difference between the maximal and minimal values. Table 1 shows the I_a and I_r values for all algorithms on all the test images along with their ranks. It demonstrates that our algorithm consistently ranks the top in terms of both accuracy and robustness.

This study reveals that pure sampling-based approaches, such as Bayesian matting, can generate fairly good mattes with tightly defined trimaps. However, when the trimap becomes coarser, the performance rapidly degrades since their assumptions on color samples are violated. Although our approach cannot fully avoid the performance degradation, the robust color sampling method we propose helps our system generate more accurate result with fine trimaps, and be more robust to the variance of user input.

On the other hand, propagation-based methods, such as the closed-form matting, can generate stable results across all trimap levels. It is not a surprise that for some examples, at coarser trimap levels, the closed-form approach estimates mattes with lower MSEs than our algorithm does, which means the color samples are no longer reliable when known pixels are far away from unknown ones. However, the major limitation of this approach is that it cannot generate very accurate matte with fine trimaps. As shown in Figure 10, some fine details of the foreground is always missing in its estimation. This fact demonstrates the contribution of the color sampling method in our approach to the accuracy of the extracted mattes. In practice we can potentially lower the data weight γ in Equation 7 for coarse trimaps where samples are not reliable.

Figure 11 shows the best mattes we extracted for a few other test images. Note that some backgrounds are very complex.

Figure 11. Best mattes extracted by our algorithm for other examples.

	Poisson	Rand. Walk	Knockout2	Bayesian	Iterative BP	Closed-form	Our
T1	717^7:1959^7	286^5:343^5	155^4:321^4	453^6:1198^6	61^2:182^3	79^3:97^2	46^1:85^1
T2	477^6:1577^6	482^7:748^5	326^4:678^4	415^5:2006^7	118^3:337^3	105^2:234^2	44^1:101^1
T3	843^7:1736^7	218^6 :278^5	113^3:180^3	137^5:1005^6	130^4:205^4	61^2:80^2	38^1:67^1
T4	340^7:1330^7	198^6:307^3	154^5:596^5	82^4:724^6	69^3:356^4	59^2:137^2	41^1:95^1
T5	451^7:2891^7	151^6:393^5	33^5:336^4	28^4:687^6	27^3:227^2	23^2:356^3	10^1:155^1
T6	879^7:3174^7	279^5:638^3	338^6:1387^6	194^3:938^5	207^4:903^4	157^2:237^1	69^1:381^2
T7	359^7:1830^7	274^6:401^4	150^5:516^6	69^2:406^5	78^4:362^3	77^3:143^1	31^1:165^2
T8	832^6:2442^6	1732^7 :1795^5	435^4:888^4	120^2:4994^7	214^3:553^2	503^5:582^3	114^1:394^1
Rank	6.9:6.8	6.0:4.4	4.5:4.5	3.9:6.0	3.3:3.1	2.6:2.0	1.0:1.3

Table 1. I_a and I_r values for different algorithms on different test images and their ranks(Format: $I_a^{rank} : I_r^{rank}$). Bottom line shows the average ranks.

6. Conclusion

We have proposed a robust matting approach that combines the advantages of sampling-based approaches and propagation-based approaches. Specifically, we propose an optimized color sampling method, which explicitly avoids the weak assumptions of previous approaches, thus enable our algorithm to generate accurate mattes in a robust way for complex images. A quantitative and objective evaluation is also presented to demonstrate the efficiency of the proposed algorithm.

References

[1] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR*, pages 264–271, 2001.

[2] C. CORPORATION. Knockout user guide. 2002.

[3] L. Grady. Multilabel randomwalker image segmentation using prior models. In *Proceedings of IEEE CVPR*, pages 763–770, 2005.

[4] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2006.

[5] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP 2005*, pages 423–429, 2005.

[6] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting. In *Proceedings of Eurographics*, 2006.

[7] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *Proceedings of IEEE CVPR*, 2006.

[8] Y. Mishima. Soft edge chroma-key generation based upon hexoctahedral color space. In *U.S. Patent 5,355,174*, 1993.

[9] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cut. In *Proceedings of ACM SIGGRAPH*, pages 309–314, 2004.

[10] M. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Proceedings of IEEE CVPR*, pages 18–25, 2000.

[11] A. Smith and J. Blinn. Blue screen matting. In *Proceedings of ACM SIGGRAPH*, pages 259–268, 1996.

[12] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *Proceedings of ACM SIGGRAPH*, pages 315–321, 2004.

[13] J. Sun, Y. Li, S.-B. Kang, and H.-Y. Shum. Flash matting. In *Proceedings of ACM SIGGRAPH*, 2006.

[14] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proceedings of ICCV 2005*, pages 936–943, 2005.