An Iterative Optimization Approach for Unified Image Segmentation and Matting

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

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

Abstract

Separating a foreground object from the background in a static image involves determining both full and partial pixel coverages, also known as extracting a matte. Previous approaches require the input image to be pre-segmented into three regions: foreground, background and unknown, which is called a trimap. Partial opacity values are then computed only for pixels inside the unknown region. This pre-segmentation based approach fails for images with large portions of semi-transparent foreground where the trimap is difficult to create even manually. In this paper we combine the segmentation and matting problem together and propose a unified optimization approach based on Belief Propagation. We iteratively estimate the opacity value for every pixel in the image, based on a small sample of foreground and background pixels marked by the user. Experimental results show that compared with previous approaches, our method is more efficient to extract high quality mattes for foregrounds with significant semi-transparent regions.

1. Introduction

Image matting refers to the problem of estimating an opacity (alpha value) and foreground and background colors for each pixel in the image. Specifically, the observed image $I(z)$ ($z = (x, y)$) is modeled as a linear combination of foreground image $F(z)$ and background image $B(z)$ by an alpha map:

$$I(z) = \alpha_z F(z) + (1 - \alpha_z)B(z) \quad (1)$$

where α_z can be any value in $[0, 1]$. If we constrain the alpha value to be either 0 or 1, then the matting problem degrades to be the segmentation problem, where each pixel is assigned to be either fully foreground or fully background.

For natural images, all values α, F and B need to be estimated for every pixel thus it is inherently an under-constrained problem. Previous approaches and the one reported here depend on user interaction combined with prior assumptions on image statistics to obtain good mattes.

As in earlier work we will require minimal user input. In contrast to previous methods we will not first segment the image into foreground, background, and unknown regions. Such methods fail in images with significant partial pixel coverage and for foreground objects containing many holes. Rather, we call on a *belief propagation* optimization method to directly determine α values.

1.1. Previous Work

Early matting approaches try to simplify the problem by photographing foreground objects against a constant-colored background, which is called *blue screen matting* [1]. However, even knowing the background color is still insufficient to fully constrain the problem thus some simple constraints are made, which require expert level parameter tuning and can fail on fairly simple foregrounds. Mishima [8] improves this technique by introducing statistical methods based on representative foreground and background samples.

Recent approaches attempt to extract foreground mattes directly from natural images. The most successful systems include Knockout 2 [3], the approach proposed by Ruzon and Tomasi [7], Bayesian matting [12] and Poisson matting [5]. All these systems start by having the user segment the image into three regions: definitely foreground, definitely background and unknown, which is often referred to as a *trimap*. The problem is thus reduced to estimating F, B and α in the unknown region. The Knockout system extrapolates known foreground and background colors into the unknown region and estimates αs according to them. Ruzon and Tomasi are the first to take a probabilistic view of the problem. They analyze foreground and background

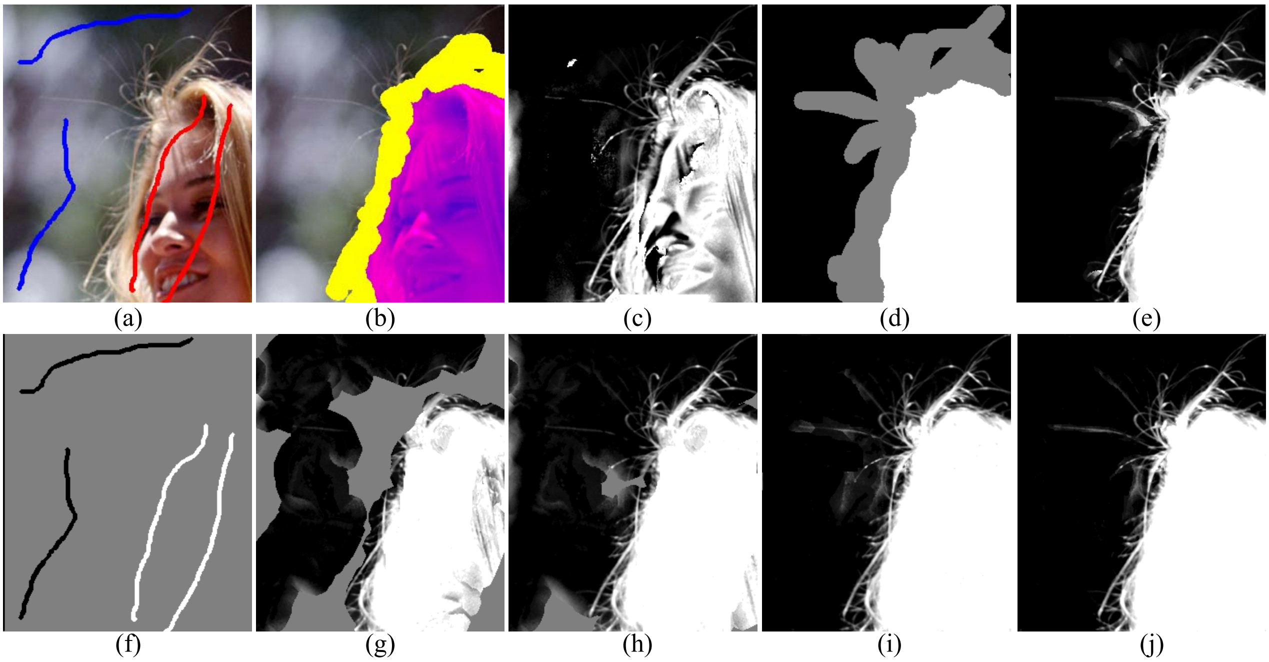

Figure 1. (a). Original image with user specified foreground and background strokes. (b). Applying graph-cut based segmentation followed by erosion-dilation operations to automatically create a trimap. The unknown region(shown in yellow) is insufficient to cover all hair strands. (c). Directly applying Bayesian matting by treating all unmarked pixels as unknown ones results in an erroneous matte. (d). Manually specified trimap. (e). Bayesian matting result on trimap (d). (f)-(j). Our algorithm iteratively estimates a matte based on the user specified strokes in (a). From (f) to (j): initial matte, matte after 3 iterations, matte after 6 iterations, matte after 9 iterations, final matte after 15 iterations.

color distributions and use them for alpha estimation. This approach has been further improved by the Bayesian matting system, which formulates the problem in a well-defined Bayesian framework and solves it using the *maximum a posteriori* (MAP) technique. This technique has been further applied on video with the help of optical flow for trimap propagation [13]. However, the trimap propagation becomes the bottleneck of the system since the optical flow calculation is not fully reliable. The Poisson matting approach estimates the matte from an image gradient by solving Poisson equations using boundary information from the trimap. Given the fact that solving the global Poisson equation is erroneous for complex foreground and background patterns, some interactive local Poisson matting techniques have been proposed to improve the global matting results, which demand a large amount of user interaction.

Recently, interactive image segmentation systems have been proposed, which can be used to efficiently generate the trimap needed for matting. The Lazy Snapping [9] and GrabCut [2] systems use a graph cut based optimization approach to extract foregrounds from images according to a small amount of user input, such as a few strokes or a bounding box. The GrabCut system then erodes and dilates the extracted foreground region uniformly to automatically create an unknown region for matting. They also propose a border matting method which assumes a strong parametric function to the matte, thus can only apply for smooth object boundaries.

1.2. Limitations of a Trimap

As we mentioned before, previous natural image matting approaches heavily rely on the user specified trimap. Ideally, the unknown region in the trimap should only cover pixels whose actual alpha values are not 0 or 1. In other words, the unknown region in the trimap should be as thin as possible to achieve the best matting result. The reason is straightforward: for Bayesian matting, the underline assumption is the local spatial coherence of the image, thus we can use nearby foreground and background colors to estimate F and B for an unknown pixel. If the unknown region is too thick and the unknown pixel is far away from the known regions, this sampling-based estimation will be less accurate, even erroneous. For Poisson matting, the assumption is the smooth intensity change in the foreground and background. One can imagine that an unnecessarily thick unknown region may cover some foreground and background textures, where the intensity can vary unevenly.

To generate good mattes, all these approaches require the user to "carefully" specify the trimap. However, it requires a considerable degree of user interaction to construct a "good" trimap for an experienced user, and it is almost impossible to manually create an optimal trimap. When images contain large portions of semi-transparent foregrounds or partial pixel coverage, such as the spider web image in Figure 3, manually creating a trimap is a very tedious process.

Automatically generated trimaps based on the binary segmentation result is non-optimal, since it always has uniform thickness regardless of local image characteristics. Furthermore, without considering the opacities of pixels in the segmentation process, the segmentation itself is erroneous. An example is shown in Figure 1. We used the graph-cut based segmentation approach described in the Lazy Snapping system to segment the foreground with a few foreground and background strokes drawn by the user (Figure 1a). Without considering the α map, some hair strands are missing in the segmentation result thus the automatically generated trimap will not be able to cover them (Figure 1b). On the other hand, directly applying Bayesian matting approach by treating all unmarked pixels as unknown results in an erroneous matte as shown in Figure 1c. The user thus has no choice but to manually create a trimap as shown in Figure 1d. Applying Bayesian matting, even on the manually generated trimap, results in the matte shown in Figure 1e, which still contains some noise because the user specified trimap is not optimal.

In this paper we propose an iterative optimization approach which unifies the segmentation and matting problems to generate a matte directly from a few user specified foreground and background strokes. For example, starting from the foreground and background strokes in Figure 1a, our approach can iteratively generate a good matte as shown in Figure 1f-j.

2. Iterative Matte Optimization

2.1. Overview

The goal of the iterative matte optimization is to determine for each pixel, p, a foreground color, F, a background color, B, an alpha value between 0 and 1, α, and to reduce the uncertainty (also between 0 and 1), u, of these values. As input, we have the mixed color of each pixel, C, and few foreground and background pixels marked by the user. The marked pixels are given an uncertainty of 0, an α of 0 (background) or 1 (foreground), and their foreground or background colors set. All other pixels are initialized to have $\alpha = 0.5$, and $u = 1$.

We divide the pixels into two groups: group U_c includes all pixels whose alpha values have been estimated in previous iterations; and group U_n includes all other pixels left unconsidered. Initially pixels the user marked are in U_c and all others are in U_n.

The approach proceeds iteratively. In each iteration, the pixels in U_n which are nearby (within 15 pixels) to ones in U_c are added to U_c, and (F, B, α, u) are estimated or re-estimated for each pixel in the expanded set. The algorithm stops when U_n is null and the uncertainty for the whole image (sum of uncertainties of pixels) cannot be reduced any further. Thus, the algorithm works in a front-propagation fashion: the estimated alpha map is propagated out from user marked pixels to rest of the image.

The overall optimization can be considered as solving a Conditional Random Field (CRF)[4] problem. Each iteration acting on the U_c region is modelled as a Markov Random Field (MRF), and a belief propagation algorithm estimates a new alpha map for it as described in the next section. The intrinsic idea for this front-propagation algorithm is to estimate alpha values for pixels with high confidence first (i.e., pixels near the user marked ones) and use them later for help to estimate alpha values for pixels which are initially far away from user marked ones.

2.2. Belief Propagation for Matte Estimation

2.2.1 MRF Construction

Pixels in U_c can be further classified into two groups based on their uncertainties: those pixels whose uncertainties are 0, which means their estimated alpha values are fixed and will not be changed in later iterations. Obviously at the beginning they are only the user marked ones. $\tilde{U}_c$ includes all pixels in U_c but whose uncertainty is less than 1. Newly joined pixels from U_n to U_c are in $\tilde{U}_c$. The goal is to estimate/update the matte for $\tilde{U}_c$ based on all pixels in U_c.

Like many other computer vision tasks such as stereo and optical flow estimation, our a priori expectation for the matte is twofold. First, we want the estimated alpha values, the foreground and the background colors to fit the actual image colors. Secondly, we expect the estimated alpha matte to be smooth. In previous approaches, the Bayesian matting system employs a statistical sampling method to meet the accuracy requirement, and the Poisson matting approach employs a Poisson equation formulation to meet the smoothness constraint. However, none of these approaches combine these two constraints together for optimization. In our approach, these two constraints are explicitly treated as optimization objectives. Motivated by recent advances in optimization methods for computer vision problems, we model each iteration of the matte estimation as an MRF optimization problem. We employ a belief propagation (BP) based approach to solve each step of the iteration. BP has been proven to be an efficient method for solving a number of low level vision tasks [6, 11].

In detail, each pixel in $\tilde{U}_c$ is treated as a node in the MRF. In addition, adjacent pixels in U_c which are not in $\tilde{U}_c$ are also included in the MRF. All the nodes (denoted as ψ) are connected to their spatial neighbors with 4-connectivity. Since $\tilde{U}_c$ changes at each iteration, the graph structure is dynamically updated. The optimization goal in each iteration is to minimize the total energy defined as:

$$V = \sum_{p \in \psi} V_d(\alpha_p) + \sum_{p,q \in \psi} V_s(\alpha_p, \alpha_q) \quad (2)$$

where V_d is the data energy describing how well the estimated alpha value α_p, and foreground and background colors for p fit with the actual color C_p, and V_s is the smoothness energy which penalizes inconsistent alpha value changes between two neighbors p and q. V_d and V_s will be described in detail in the next section.

2.2.2 Data and Neighborhood Terms

For the belief propagation optimization, we discretize the possible alpha value to 25 levels between 0 and 1, denoted as $\alpha^k, k = 1, ..., 25$, and each level corresponds to a possible state for a node in the MRF. Similar to the Bayesian matting approach, to estimate α_p we seek for each pixel a local estimate of the foreground or background color. We sample a group of known or previously estimated foreground and background colors from the neighborhood of the node p. The local neighborhood area is defined to have a radius of $r = 20$ around p.

In addition to being nearby, for a sample to be considered valid to estimate the local foreground it must have an α larger than the α_p, and likewise must be less than α_p to be a valid sample of the local background.

The set of valid foreground samples, p_i, are then weighted by their uncertainty and distance, by

$$w_i^F = (1 - u(p_i))) \cdot exp(\frac{-s(p, p_i))^2}{\sigma_w^2}) \quad (3)$$

where $s(p, p_i))$ is the spatial distance between the two points, and $\sigma_w = r/2$. We pick out N (set in our system to 12) samples with the largest weights as foreground samples. A similar method is applied to get background samples.

It is possible (likely in early iterations) that there will not be N valid foreground or background samples for a pixel. We then resort to a global sampling method. Similar to the GrabCut system [2], we first train a Gaussian Mixture Model (GMM) on the user specified foreground (background) pixels, then assign each marked pixel to a single Gaussian in the GMM. We then randomly select pixels from each Gaussian to assemble the sample set. Note that this initial global sampling is not accurate since it samples a mixture of all possible foreground or background colors, but as the algorithm progresses, foreground and background regions will be propagated close to p thus the local model will take over.

Given the foreground and background samples and corresponding weights, we can build a probability histogram of alpha levels. Specifically, we compute the likelihood of each alpha level α^k as

$$L_k(p) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_i^F w_j^B \cdot \quad (4)$$

$$exp\left(-d_c\left(C_p, \alpha^k F_i^p + (1-\alpha^k) B_j^p\right)^2 / 2\sigma_d^{k2}\right)$$

where F_i^p and B_j^p are foreground and background samples and C_p is the actual color of p. We calculate Euclidian distance in RGB space as distance between colors which is denoted as $d_c(\cdot, \cdot)$. To compute the covariance σ_d^k, we first compute the distance covariance among foreground and background samples, denoted as σ_F and σ_B, then calculate σ_d^k as $\alpha_k \sigma_F + (1 - \alpha_k)\sigma_B$.

The data term in Equation 2 can be computed for each of the possible states, α^k, for α_p. The data cost is defined as

$$V_d(\alpha_p^k) = 1 - \frac{L_k(p)}{\sum_{k=1}^{K} L_k(p)} \quad (5)$$

Note that unlike the Bayesian matting approach which fits Gaussians on the sampled foreground and background colors, we do not fit any parametric models on them thus our algorithm is more robust to sampling outliers and avoids degenerate cases in computation, such as non-invertible matrices in the covariance analysis for Bayesian matting.

The neighborhood term in Equation 2 encourages the labels to vary smoothly. It is defined based on the degree of difference between two neighboring α values, say α_1 and α_2 . The smoothness cost is defined as

$$V_s(\alpha_1, \alpha_2) = 1 - exp\left(-(\alpha_1 - \alpha_2)^2 / \sigma_s^2\right) \quad (6)$$

We set σ_s empirically to be 0.2 in our system.

2.2.3 Belief Propagation Optimization

With the MRF defined above, finding a labeling (α level for each pixel) with minimum energy corresponds to the MAP estimation problem. We use the loopy belief propagation (BP) algorithm [10] to solve it. Belief propagation has been recently exploited for early vision problems such as stereo and image restoration as an efficient approximation method

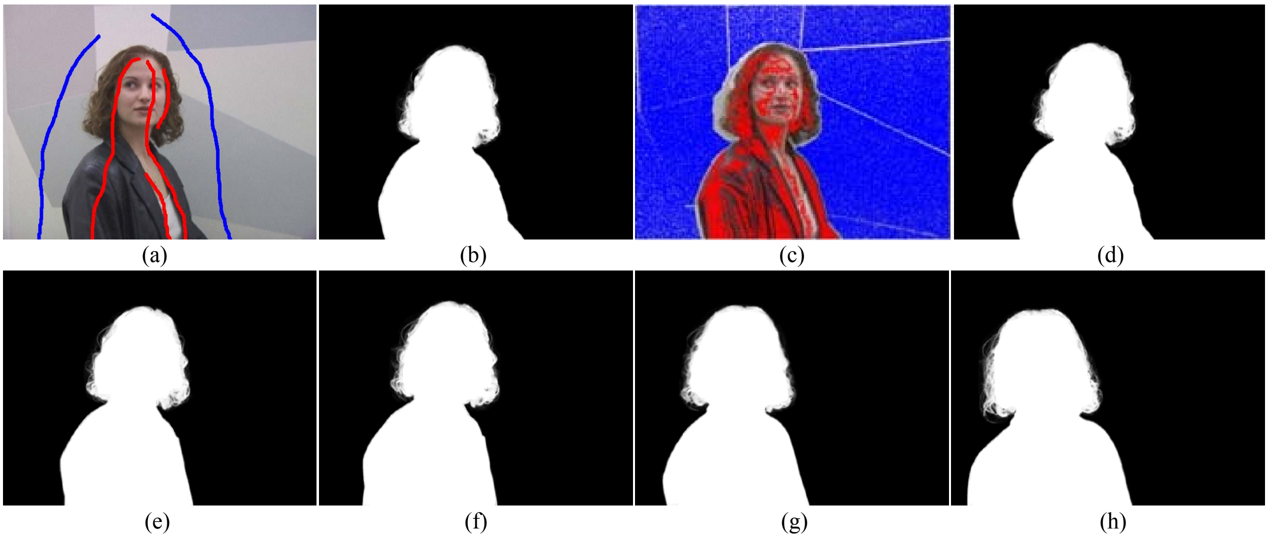

Figure 2. (a). One frame of a video sequence with user specified foreground and background strokes. (b). Extracted matte by our approach. (c). Automatically propagating foreground and background marks onto the next frame. (d). Extracted matte on the next frame. (e)-(h). Automatically generated mattes for the 5th, 10th, 15th and 20th frames.

to solve the MAP formulation of these problems. Although a full description of this algorithm is beyond the scope of this paper, it essentially works as follows.

The BP algorithm itself is an iterative process. It works by passing messages along links in the constructed graph. Each message is a vector of dimension K, the number of possible states/labels. Following the notation in previous papers, we let m_{pq}^t to be the message that node p sends to a neighboring node q at iteration t. In each iteration, new messages are computed for each possible state in the following way in our system:

$$m_{pq}^t(k_q) = c \max_{k_p} \left(V_s(\alpha_p^{k_p}, \alpha_q^{k_q}) \cdot V_d(\alpha_p^{k_p}) \cdot \Pi_{\tau \in H(p) \setminus q} m_{\tau p}^{t-1}(k_p) \right) \quad (7)$$

where $H(p) \setminus q$ denotes the neighbors of p other than q. c is a normalization factor and V_d and V_s are the data and neighborhood terms defined in the previous section. After T iterations a belief vector is computed for each node as

$$b\left(\alpha_p^{k_p}\right) = cV_d\left(\alpha_p^{k_p}\right) \cdot \sum_{q \in H(p)} m_{qp}^T(k_p) \quad (8)$$

Finally, the k_p state that maximizes $b(\alpha_p^{k_p})$ at each node is selected as the estimated alpha level, α_p^*.

After running the belief propagation algorithm, we get an estimated alpha value α_p^* for each pixel p in $\tilde{U}_c$. If $\alpha_p^* = 1$, we set the actual color C_p as a new foreground sample and remove it from $\tilde{U}_c$ assigning $u(p) = 0$. Similar operations are applied when $\alpha_p^* = 0$ for background pixels. Otherwise we choose the pair of foreground and background colors from the group of foreground and background samples which minimizes the fitting error:

$$F^*(p), B^*(p) = arg \min_{F_i^p, B_j^p} \left(C_p - \alpha_p^* F_i^p - (1 - \alpha_p^*) B_j^p \right)^2 \quad (9)$$

Note that we use real foreground and background colors as the estimated ones, in this way we can avoid the ”color bleeding” artifact that the Bayesian matting approach may have as addressed in [2], in which case the estimated foreground color is a mixture of real foreground and background colors. The uncertainty value $u(p)$ is updated as

$$u^*(p) = 1 - (w_i^{F*} \cdot w_i^{B*})^{\frac{1}{2}} \quad (10)$$

where w_i^{F*} and w_i^{B*} are weights for the selected pair of foreground and background samples. Since the sample weights will generally increase as the U_c region grows, the uncertainty values will decrease during the iterative process. The algorithm halts when the total uncertainty of the whole matte cannot be reduced any further.

2.3. The Algorithm

Iterative Belief Propagation for Image Matting

1. Initially, user marked pixels are in U_c and all other ones are in U_n.
2. Compute GMMs for foreground pixels and background pixels the user marked, then assign each marked pixel to one Gaussian for further global sampling.
3. Repeat until U_n is null and the total uncertainty of the estimated matte is minimized:
   (a) If U_n is not null, apply a spatial distance transform to U_c region and transfer pixels within 15 pixels of U_c from U_n to $\tilde{U}_c$.
   (b) Build an MRF based on pixels in U_c as described in Section 2.2.1.
   (c) For each node in the MRF, sample a group of foreground and background colors by either neighborhood sampling or global sampling, and compute its data cost as described in Section 2.2.2.
   (d) Run belief propagation algorithm to estimate a matte for pixels in the graph.
   (e) Update the uncertainty, the estimated foreground and background colors for each pixel in the graph. Also update $\tilde{U}_c$ based on the updated matte.
4. End.

3 Extension to Video

Our algorithm can also be applied for the purpose of video matting by propagating the user's input from frame to frame. As shown in Figure 2a, the user draws a few foreground and background strokes on one frame of the video sequence, and our algorithm generates a good matte on this frame as shown in Figure 2b. To automatically generate an initial set of foreground and background pixels on the next frame, we first examine every location z where the estimated alpha value $\alpha_z^t = 1$, and compare the color distance between p_z^t and p_z^{t+1}. If the distance is small enough, we mark p_z^{t+1} as a definite foreground pixel. We apply the same operation on background pixels, and can get an initial foreground and background map on the next frame, as shown in 2c. Applying our algorithm on the next frame based on this map results in a good matte shown in 2d. In this way we can automatically estimate good mattes for more than 20 frames based on a few strokes on one single frame, as shown in 2e-h. Note that applying Bayesian matting on this sequence is very time-consuming since automatically propagating a good trimap from one frame to others is difficult and unreliable, as demonstrated in the video matting system [13]. By loosening the requirement for an accurate trimap on each single frame, our approach thus is more efficient for video matting.

4. Results and Comparisons

The proposed approach has been tested on a variety of different input images. One example is shown in Figure 1 with intermediate results. In Figure 3a, we show four example images along with user marked foreground and background strokes. Note that these foreground objects either have large transparent regions or have a lot of holes. Our algorithm can successfully extract good mattes based on a few user specified strokes. In contrast, Bayesian matting requires the user to manually create complex trimaps for them, as shown in Figure 3c, to obtain the mattes shown in Figure 3d. Note that some of the mattes computed by the Bayesian matting approach have noticeably lower quality compared with results generated by our approach, even the user has taken much more efforts to create the trimaps.

Figure 4 shows composite images with extracted foregrounds shown in Figure 3 and novel backgrounds.

Our approach works generally well in our tests. However, for difficult images where foreground and background have very similar colors and complex patterns, like many other approaches, our approach does not always work well. An example is shown in Figure 5a. Both the foreground and background have a quite few dominant colors and some of them, such as the white strips on the sheet and white body of the toy rabbit, cannot be easily distinguished, especially under the effects of shadows. Due to the color ambiguity, our algorithm thus produces an erroneous result as shown in Figure 5b. However, another benefit of our system is that it can also start with an initial trimap instead of foreground and background strokes. Figure 5c shows a rough trimap the user created. Bayesian matting result based on this trimap is shown in Figure 5d, and matting result of our approach based on the same trimap is shown in Figure 5e. One can imagine that given an initial rough trimap, our approach can iteratively refine it to be more accurate so our system produces a better matting result than the Bayesian approach from the same starting point.

Brute force belief propagation is a computational expensive process. By iteratively running belief propagation, our initial implementation of the system runs about 15-20 minutes for an input image (640*480). Since the initial submission paper, dramatic speed ups to the system improved

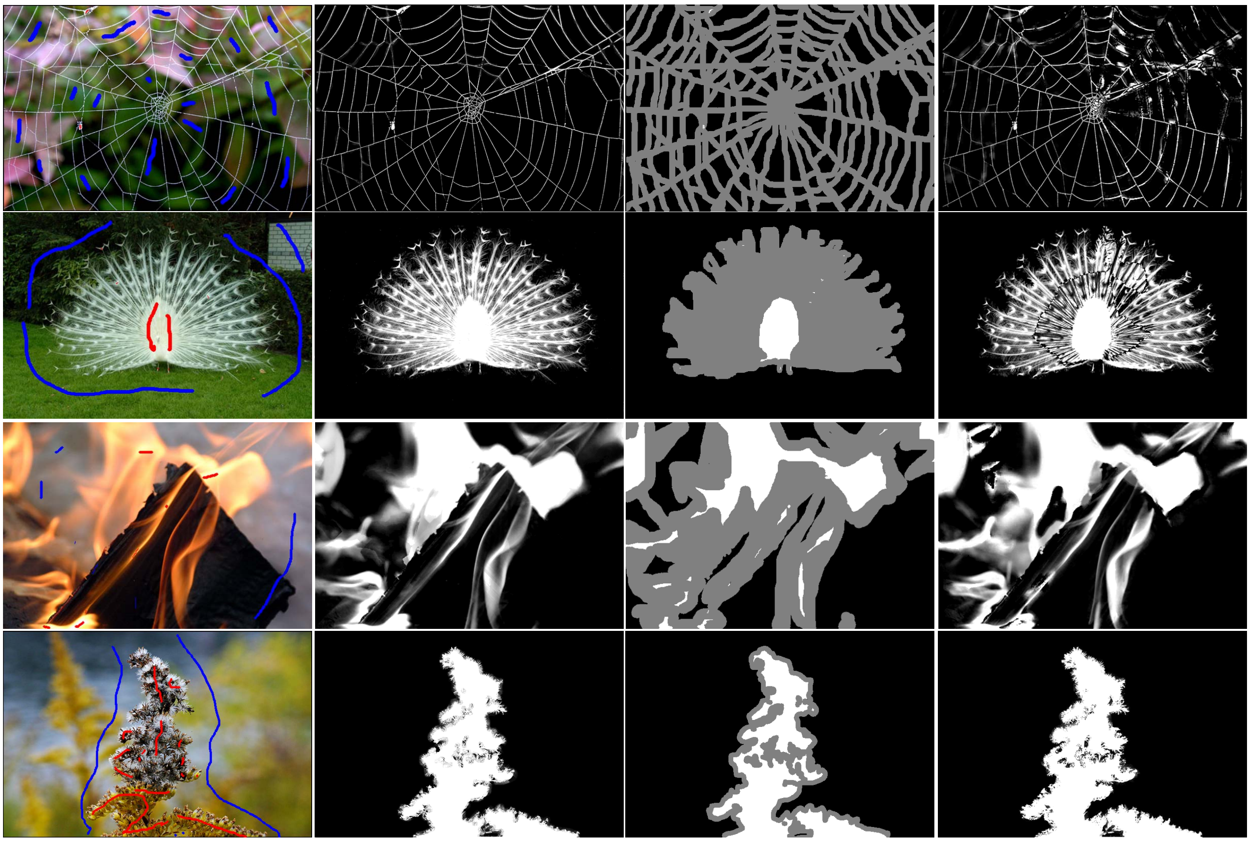

Figure 3. (a). Original images with user specified foreground and background strokes. (b). Extracted alpha mattes by our approach. (c). For Bayesian matting, complex trimaps must be manually created by the user. (d). Bayesian matting results based on trimaps in (c).

running times by a factor of 50 without loss of quality. In particular the fast belief propagation algorithm described in [11] was used. We also employed a hierarchical strategy for efficient matting. These techniques will be detailed in an technical extension of this paper.

5. Summary and Conclusions

We have proposed and demonstrated an iterative optimization approach to solve the image matting problem. Our approach combines the problems of segmentation and matting into a unified formula that iteratively estimates an alpha value for every pixel in the image based on a small amount of foreground and background pixels marked by the user. Unlike previous approaches, our method (usually) does not require a well specified trimap, thus it is more efficient to extract mattes for semi-transparent or foreground objects with many holes.

Our method is based on statistical sampling of known foreground and background colors, thus it has similar weakness with Bayesian matting method for images where foreground and background colors are too ambiguous. How to build more accurate color models is the crucial challenge for both this approach and many other color-based segmentation and matting approaches.

Acknowledgement

The authors would like to thank Colin Ke Zheng and Yi Li for sharing their initial experimental code, and Qi Miao for help collecting test images. One of the authors of this work is supported by a grant from Microsoft Research.

References

[1] A.R. Smith and J.F. Blinn. Blue screen matting. *Proceedings of ACM SIGGRAPH*, pp. 259-268, 1996.

[2] C. Rother, V. Kolmogorov and A. Blake. GrabCut - Interactive foreground extraction using iterated graph cut. *Proc. of ACM SIGGRAPH*, pp. 309-314, 2004.

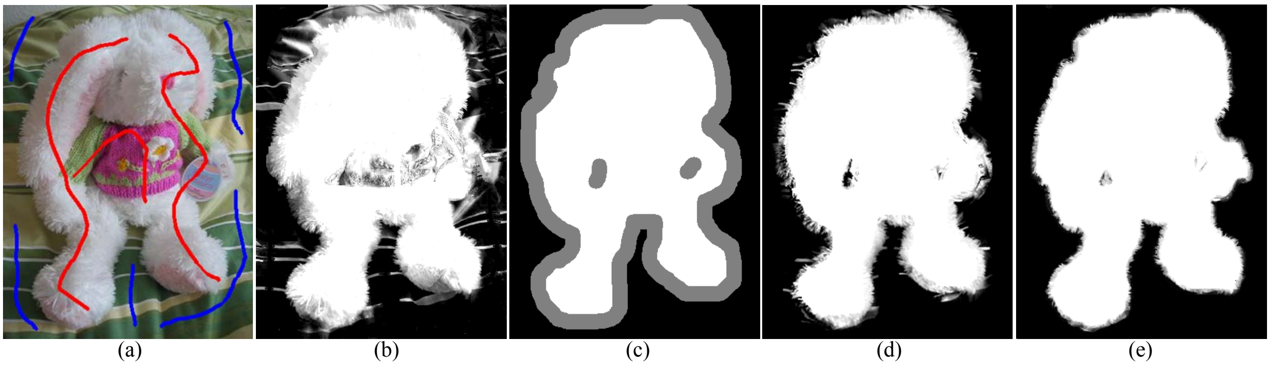

Figure 5. (a). Original image with foreground and background strokes. (b). Extracted matte by our approach is erroneous due to color ambiguity. (c). The user specified rough trimap. (d). Bayesian matting result based on trimap (c). (e) Our matting result based on trimap (c).

[3] COREL CORPORATION. Knockout user guide, 2002.

[4] J. Weinman, A. Hanson and A. McCallum. Sign detection in natural images with conditional random fields. *Proc. IEEE Workshop on Machine Learning for Signal Processing*, Sep. 2004.

[5] J. Sun, J.Y. Jia, C.K. Tang and H.Y. Shum. Poisson matting. *Proc. of ACM SIGGRAPH*, pp. 315-321, 2004.

[6] J. Sun, H.Y. Shum and N.N. Zheng. Stereo matching using belief propagation. *IEEE Trans. on PAMI*, Vol. 25, No. 7, pp. 787-800, 2003.

[7] M.A. Ruzon and C. Tomasi. Alpha estimation in natural images. *Proc. of IEEE CVPR*, pp. 18-25, 2000.

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

[9] Y. Li, J. Sun, C.K. Tang and H.Y. Shum. Lazy snapping. *Proc. of ACM SIGGRAPH*, pp. 303-308, 2004.

[10] Y. Weiss and W.T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arebitrary graphs. *IEEE Trans. on Information Theory*, Vol. 47, No. 2, pp. 303-308, 2001.

[11] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. *Proc. of IEEE CVPR*, pp. 261-268, 2004.

[12] Y.Y. Chuang, B. Curless, D. Salesin and R. Szeliski. A Bayesian Approach to Digital Matting. *Proc. of IEEE CVPR*, pp. 264-271, 2001.

[13] Y.Y. Chuang, A. Agarwala, B. Curless, D. Salesin and R. Szeliski. Video matting. *Proc. of ACM SIGGRAPH*, pp. 243-248, 2002.

Figure 4. Composite images with new backgrounds.