FuzzyMatte: A Computationally Efficient Scheme for Interactive Matting

Yuanjie Zheng[1] Chandra Kambhamettu[1] Jingyi Yu[1] Thomas Bauer[2] Karl Steiner[3]
[1]Department of Computer Science, University of Delaware, Newark, DE, USA
[2]Helen F. Graham Cancer Center, Christiana Care Health Services, Newark, DE, USA
[3]Delaware Biotechnology Institute, University of Delaware, Newark, DE, USA

Abstract

In this paper, we propose an online interactive matting algorithm, which we call FuzzyMatte. Our framework is based on computing the fuzzy connectedness (FC) [20] from each unknown pixel to the known foreground and background. FC effectively captures the adjacency and similarity between image elements and can be efficiently computed using the strongest connected path searching algorithm. The final alpha value at each pixel can then be calculated from its FC. While many previous methods need to completely recompute the matte when new inputs are provided, FuzzyMatte effectively integrates these new inputs with the previously estimated matte by efficiently recomputing the FC value for a small subset of pixels. Thus, the computational overhead between each iteration of the refinement is significantly reduced. We demonstrate FuzzyMatte on a wide range of images. We show that FuzzyMatte updates the matte in an online interactive setting and generates high quality matte for complex images.

1. Introduction

Image matting refers to the process of decomposing an observed image I into a foreground object image F, a background image B, and an alpha matte α as

$$I = \alpha F + (1 - \alpha)B. \quad (1)$$

The resulting foreground and the alpha matte can then be used to composite a new image with a different background.

It is well understood that estimating the matte from a single image is inherently under-constrained: there are more unknowns $(F, B \text{ and } \alpha)$ than the constraints (Eqn 1). Additional constraints have been proposed to resolve the ambiguity problem, including the ones based on user inputs like scribbles[22] or trimaps [3] and the ones using multiple images or video [2, 15, 10, 18].

Recent image matting methods [22, 9, 8, 1] have focused on improving the user interactivity and reducing the algorithm running time. For example, these methods prefer using scribbles rather than specifying a complete trimap. The user can also incrementally add more scribbles to improve the quality of the matte. However, the time between each iteration of the interactive loop can be very long, as shown in Figure 5. This is because many previous methods do not make use of the matting result from prior iterations. Instead, they recompute the matte from the scratch even if the user refinement is minimal (e.g., adding a single stroke or change the bounding box) [22, 8, 9, 1, 13]. A notable exception is the soft scissor [21], which estimates the matte and foreground color in a local region covered by the stroke. However, the soft scissor algorithm relies on accurately tracing the foreground edge to produce high-quality mattes.

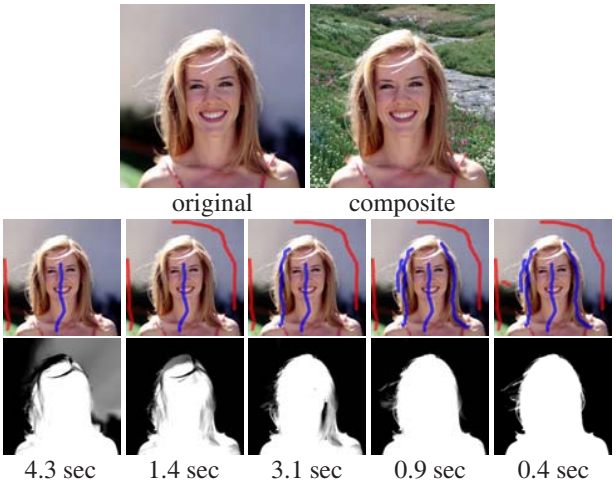

Figure 1. Interactive matting process and run time with FuzzyMatte method. The images in the last row represent alpha values.

In this paper, we propose a scribble-based online matting algorithm, which we call FuzzyMatte. FuzzyMatte significantly reduces the time between each iteration of the interactive loop by effectively integrating the new input with the estimated matte from the previous iteration. FuzzyMatte is inspired by modern techniques in fuzzy medical segmentation [20] that are used to determine the partial volume from multiple tissues in a single image. Given an image and the initial user input (scribbles or a trimap), we first compute, for each pixel p with an unknown alpha, its fuzzy con-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

nectedness (FC) to the known foreground and background. FC is a concept that effectively captures fuzzy "connectedness" (adjacency and similarity) between image elements [20]. We show that the FC at each pixel can be computed by searching for its strongest connected path to the known foreground and background. The alpha value of the pixel can then be calculated from FCs.

When additional inputs become available, FuzzyMatte does not require recomputing all pixels' FC by searching again for the strongest connected path. Instead, it partitions the pixels into three subsets based on their old FC values and only a small number of pixels in one of the three subsets require searching for the strongest connectedness path. All other pixels simply reuse the previously estimated FC. This significantly reduces the computational overhead between each iteration of the refinement. In addition, for pixels that need recomputing of the FC by searching the strongest path, we develop an efficient algorithm by confining the search space for the strongest path. This further reduces the computational cost for re-estimating the matte. We have demonstrated FuzzyMatte on a wide range of images and experiments have shown that FuzzyMatte updates the matte in an online interactive setting and generates high-quality mattes for complex images.

2. Previous Work

Single Image vs. Multiple Images The problem of alpha matting has been studied for over 40 years. Single image based matting methods rely on image statistics [3, 22] or the image gradient [16] to resolve the ambiguity problem. They are often assisted with user inputs in form of trimaps or scribbles. Multi-image based matting methods can also remove the ambiguity. Smith and Blinn [15] used two different backgrounds to make the matting problem over-determined. Defocus video matting [10] used a special imaging system to capture multiple images of the scene with different focus. It then automatically segments the image into foreground, background, and an unknown region by analyzing the defocus. Flash matting [17] uses a flash/no-flash pair to extract the foreground layers by assuming the foreground is significantly bright. However, many multi-image based methods are sensitive to calibration errors, camera shake, and foreground motions.

Trimap vs. Scribbles We can also categorize the existing matting methods by the form of user inputs. Trimaps and scribbles are the two most commonly used inputs for identifying foreground, background, and the unknown region.

Bayesian matting and Poisson matting [16] are two important matting techniques using trimaps as the input. Bayesian matting [3] builds a probabilistic model of foreground and background from the trimap and then uses a maximum-likelihood criterion to estimate the matte. Bayesian matting can also be extended to videos by propagating the trimap using optical flow [2]. Poisson matting [16] assumes the matte gradient is proportional to the image gradient and solves the Poisson equation in the region specified by the trimap. To generate a high-quality matte, the trimap often needs to be manually and carefully specified. Thus, it is usually impractical to use the trimap-based methods for online interactive processing.

Recently, many interactive image matting methods proposed have been using scribbles or a bounding box [13] as the input. The random walk [6] algorithm assigns a label to each pixel with maximal probability that it will reach the scribbles through a random walk. Wang&Cohen [22] proposed to use a Markov Random Field (MRF) to simultaneously segment foreground and background and compute the matte. Bai and Sapiro [1] proposed to use the geodesic distance from the scribbles to an arbitrary pixel to assign the likelihood that a pixel belongs to foreground or background and then uses this likelihood to compute its alpha value. Spectral matting [8] computes a set of soft mattes by solving the eigenvectors of the matting Laplacian matrix [9]. The user can then use scribbles to refine the soft matte.

The major advantage of using scribbles is that the user can easily refine the inputs to improve the quality of the matte. However, the wait time between each iteration of refining the scribbles can be very long for most of these methods since they recompute the matte from the scratch without using the previously estimated result even if the refinement is minimal. The recently proposed soft scissor method [21] allows the user to roughly track the boundary with a self-adjustable brush. By estimating the matte only in local regions, soft scissor achieves real-time performance. However, its result heavily depends on the accuracy of the specified boundary.

We propose a scribble-based online matting algorithm called FuzzyMatte based on the notion of fuzzy connectedness (FC) introduced by Rosenfeld [12]. FC captures fuzzy "togetherness" (adjacency and similarity) of image elements [20]. FC has been used in medical image segmentation and tissue density quantification [20, 11], as well as in color image or video segmentation [7]. Dijkstra's algorithm [4] can be used to efficiently compute the FC. Nyul *et al*. [11] further accelerated this computation using a Fibonacci heap. In this paper, we use the method in [11] for computing the FC and we show that with this acceleration, FuzzyMatte provides an interactive tool for online estimation of alpha mattes.

3. Modeling Alpha Matting Using Fuzzy Connectedness

In this section, we show how to use the fuzzy connectedness (FC) to model the alpha matting problem. The FC

between two pixels models the "hanging togetherness" in terms of their intensity similarity, spatial distance, and color consistency [20]. In this paper, we model the image as a graph with four-connectivity relationships between neighboring pixels. We first define the similarity metric (which we refer to as affinity $\mathcal{A}$ [20] in this paper) between any two neighboring pixels (Section 3.1). We then compute the fuzzy connectedness $\mathcal{FC}(p,q)$ between two pixels p and q using the strongest connected path between p and q (Section 3.2). For image matting, we also define the FC between a pixel and the known scribbles and we show that the alpha value at each pixel can be estimated from its FCs to the foreground scribbles and to the background scribbles (Section 3.3).

3.1. Affinity

We start by defining the affinity $\mathcal{A}$ between two adjacent pixels p_1 and p_2. We adopt a notion of $\mathcal{A}$ similar to [14] as:

$$\mathcal{A}^o(p_1,p_2) = \lambda\mu^o_\psi(p_1,p_2) + (1-\lambda)\mu^o_\phi(p_1,p_2), \quad o \in \{f,b\} \tag{2}$$

where superscript o distinguishes the foreground ($o=f$) and the background ($o=b$) affinity. μ_ψ measures the color similarity between the two pixels, which we call pixel-pixel similarity. μ^o_ϕ measures the color similarity between p_1 and p_2 and the color of scribbles in o and takes high value when p_1 and p_2 are both close to the scribbles' color, which we call pixels-scribble similarity. Finally, λ balances the two similarity measures and is between $[0\ 1]$.

We first fit a Gaussian Mixture Model (GMM) to both the foreground scribble colors and the background scribble colors. In this paper, we use the CIE LUV color representation. We can then compute the pixel-pixel similarity μ^o_ψ as

$$\mu^o_\psi(p_1,p_2) = \exp\left(-\tfrac{1}{2}[I(p_1)-I(p_2)]^T(\Sigma^o_{\max})^{-1}[I(p_1)-I(p_2)]\right). \quad o \in \{f,b\} \tag{3}$$

For each Gaussian in the GMM, we average the variance of all three channels and $\Sigma_{\max}$ corresponds to the covariance matrix of the Gaussian that has the largest average variance. We choose to use the largest variance to improve the robustness of FC computation in highly textured regions [14].

We define the similarity metric $S^f(p)$ between a pixel p's color to the foreground scribble color as

$$S^f(p) = \max_i \exp\left(-\frac{1}{2}[I(p)-m^f_i]^T(\Sigma^f_i)^{-1}[I(p)-m^f_i]\right)$$

where i is the index of the Gaussian in the GMM of the foreground color, and m^f_i, Σ^f_i are the mean vector and covariance matrix of the corresponding Gaussian. $S^b(p)$ can be similarly computed with the background scribbles' color. We compute the pixels-stroke similarity μ^f_ϕ (foreground strokes) as

$$\mu^f_\phi(p_1,p_2) = \begin{cases} 1 & \text{if } p_1 = p_2 \\ \frac{W^f_{min}(p_1,p_2)}{W^f_{min}(p_1,p_2)+W^f_{max}(p_1,p_2)} & \text{if } W^f_{min}(p_1,p_2) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where

$$W^f_{min}(p_1,p_2) = \min\left[S^f(p_1),\ S^f(p_2)\right]$$

and

$$W^f_{max}(p_1,p_2) = \max\left[S^b(p_1),\ S^b(p_2)\right].$$

We can compute the other pixels-stroke similarity μ^b_ϕ (background strokes) in a similar way to μ^f_ϕ.

Finally, we can combine μ_ϕ and μ_ψ for computing affinity $\mathcal{A}$ with equation (2).

Notice, the affinity $\mathcal{A}$ is reflective (i.e. $\mathcal{A}(p,p)=1$) and symmetric (i.e. $\mathcal{A}(p_1,p_2)=\mathcal{A}(p_2,p_1)$). In Section 4, we will use these two properties to develop highly efficient algorithms for computing the fuzzy connectedness between pixels.

Our affinity measure $\mathcal{A}(p_1,p_2)$ is similar to the ones described in [14]. However, we extend the definitions to color images, and extend the known pixels to scribbles. In addition, we preprocess the image with Bilateral Filtering [19] with a small spatial support (in our examples, $\sigma_d=2$ and $\sigma_r=5$) to reduce the noise when computing affinity.

3.2. Fuzzy Connectedness

To define the fuzzy connectedness (FC) between two pixels p_1 and p_2, we represent a path γ between p_1 and p_2 in image Ω as a sequence of n_γ pixels $\{q_1,q_2,\cdots,q_{n_\gamma}\}$ where $q_1=p_1$ and $q_{n_\gamma}=p_2$ and p_k and p_{k+1} are the two neighboring pixels in Ω, as shown in Figure 3. The strength of path γ is defined as:

$$str^o\left(\gamma(p_1\rightarrow p_2)\right) = \min_{j=2\cdots,n_\gamma} \mathcal{A}^o(q_{j-1},q_j), \text{ for } o\in\{f,b\}. \tag{5}$$

Then, the FC between p_1 and p_2 is defined as

$$\mathcal{FC}^o(p_1,p_2) = \max_{\text{all }\gamma} str^o(\gamma), \text{ for } o\in\{f,b\} \tag{6}$$

where $\mathcal{FC}^f$ and $\mathcal{FC}^b$ represent the FC with respect to the foreground and background scribbles.

Notice that the min/max metric in Eqn (5) and Eqn (6) guarantees that the FC between two pixels inside the same region of the same object is large. More importantly, even if the pixel intensities are different due to the graded composition of the heterogeneous material of the object, their

Original image FC value

Figure 2. Fuzzy connected values to the circle region highlighted with light green color.

fuzzy connectedness can still be large as there exists a path along which the color changes smoothly.

Udupa *et al*. [20] have proved that the FC between two pixels $\mathcal{FC}(p_1, p_2)$ (Eqn 6) is a similitude relation. It is reflective (i.e. $\mathcal{FC}(p, p) = 1$), symmetric (i.e. $\mathcal{FC}(p_1, p_2) = \mathcal{FC}(p_2, p_1)$), and transitive (i.e. $\mathcal{FC}(p_1, p_2) = max_{p\in\Omega, p\neq p_1, p\neq p_2} min(\mathcal{FC}(p_1, p), \mathcal{FC}(p, p_2))$).

We can further define the FC between a pixel p and the foreground scribbles $\mathcal{FC}^f(p)$ and the background scribbles $\mathcal{FC}^b(p)$ as

$$\mathcal{FC}^o(p) = \mathcal{FC}(\Omega^o, p) = \max_{p'\in\Omega^o} \mathcal{FC}^o(p', p), \quad o \in \{f, b\} \tag{7}$$

where Ω^f and Ω^b represent the foreground and the background scribbles, respectively. In Fig. 2, we represent the FC for all pixels with respect to a circular scribble.

Finally, we can also define the FC between two sets of pixels P and Q using pixel-pixel FCs (Eqn 6) as:

$$\mathcal{FC}(P, Q) = \max_{p\in P} \max_{q\in Q} \mathcal{FC}(p, q). \tag{8}$$

We can prove that the FC relations defined in Eqn 7 and Eqn 8 are also a similitude relation. We provide the proof in the supplementary material on our webpage.

3.3. Estimating Matting Using Fuzzy Connectedness

The fuzzy connectedness (FC) between pixels are closely related to the matting problem. A pixel with larger alpha values should be more closely connected to the foreground scribbles, and hence, have a larger FC value to the foreground than to the background. Similarly, a pixel with smaller alpha values should have a larger FC value to the background. Therefore, we can derive every pixel p's α value by using its foreground and background FC as

$$\alpha(p) = \begin{cases} 1 & (\mathcal{FC}^f(p) > \nu_1) \& (\mathcal{FC}^b(p) < \nu_2) \\ 0 & (\mathcal{FC}^b(p) > \nu_1) \& (\mathcal{FC}^f(p) < \nu_2) \\ \frac{\mathcal{FC}^f(p)}{\mathcal{FC}^f(p) + \mathcal{FC}^b(p)} & \text{otherwise} \end{cases} \tag{9}$$

where thresholds $\nu_1 = 0.95$ and $\nu_2 = 0.05$ are used to determine if p is foreground, background, or transparent.

Once we estimate the alpha matte using FC, we can compute the foreground and background colors using a similar method to [22]. Specifically, we fit a Gaussian Mixture Model (GMM) to represent both the foreground and background colors of the pixels whose FC exceeds the thresholds. We then find the optimal pair of foreground and background colors that minimize the fitting error in Eqn 9.

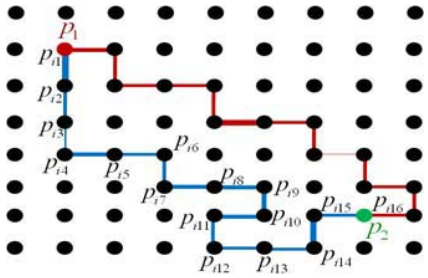

Figure 3. Demonstration of the definition of fuzzy connectedness. To compute the fuzzy connectedness value between two pixels p_1 (red filled circle) and p_2 (green filled circle), we need to find the strongest path among all the possible paths between them. The blue-color lines and yellow-color lines show two variant paths, in which larger line width represents larger affinity value. For the path in blue color, the strength is defined as the weakest affinity value between the two pixels q_{i3} and q_{i4}.

4. Estimating Fuzzy Connectedness

In this section, we develop efficient algorithms to compute, for pixels with unknown alpha, their fuzzy connectedness (FC) to the foreground and the background scribbles. The computed FC can then directly be used to evaluate the matte with Eqn (9).

Computing FC between two pixels is, in essence, the single-source shortest path problem. The only difference is that the cost of each path is the sum of the edge weights in traditional problems, whereas it is the minimal affinity between all neighboring pixels in FC computation. Nevertheless, the classical Dijkstra's algorithm [4] can still be used to compute FC [11] and its running time can be further improved using a Fibonacci heap [5, 11]. In addition, since the Dijkstra's algorithm computes the shortest path from a single node to all nodes, we can easily compute the FC from each pixel to the scribbles. In Appendix, we show the pseudo code for the Dijkstra FC estimation algorithm, as a Dijkstra-FC-T algorithm. Note that we add a template T in the algorithm to indicate, which pixel can be inserted in the heap, hence confining the search space for the strongest path.

Recall that our goal is to reduce the computational overhead when new scribbles are added. Notice that if the new scribbles introduce new colors in GMM, we will change i) the affinity between the neighboring pixels, and ii) the FC value that is based on the strongest path from the pixel to the foreground and background scribbles. For affinity, the key computational cost is to calculate the pixel-pixel similarity (Eqn 3) and pixels-scribble similarity (Eqn 4). To

minimize this cost, one can draw the scribbles to cover most of the main colors of foreground or background in the first iteration so that the cost for updating affinity will be kept minimal when new scribbles are added. To recompute the FC, we prove that we can partition the pixels into three subsets and only one subset of pixels needs re-estimating the FC by searching for the strongest path within a smaller domain and the other two subsets can directly update their FC values based on the previously computed FC.

4.1. First Iteration

At the first iteration of user interaction, we set the FC of known foreground or background pixels to be 1. We then directly use the Dijkstra-FC-T algorithm to compute the FC values to foreground or background by setting the template T to be 1 for all pixels. Counter-intuitively, larger strokes will usually result in lower computation cost. This is because the manually specified pixels in the strokes will never fit the condition in the 5th step of the Dijkstra-FC-T algorithm, therefore, will be put in the queue only once at the first step. In contrast, other pixels will be put in the queue possibly more than once.

Our initial FC computation also works for the trimap. In fact, the computational cost for FC using a trimap can be much lower than using scribbles as there are fewer pixels with unknown FC. Furthermore, we can constrain the strongest path to only pass the foreground regions and the unknown regions to compute the FC to foreground and impose a similar constraint to compute the FC to the background to further reduce the search space. In practice, we set the T values as 0 for pixels in Ω_b and other pixels as 1 when computing FC to foreground, and set the T values as 0 for pixels in Ω_f and other pixels as 1 when computing FC to background (in the Dijkstra-FC-T algorithm).

4.2. Efficient Fuzzy Connectedness Updates

At each iteration, when new scribbles are added, FuzzyMatte attempts to avoid recomputing all pixels' FC using the strongest path search algorithm. Instead, it tries to reuse the computed FC from the last iteration. To do so, we present two propositions and then develop an efficient algorithm based on the propositions that only requires recomputing the strongest path for a much smaller set of pixels. We provide a complete proof to all two propositions in the supplementary material on our webpage. Similar theoretical results first appeared in [23].

Proposition 1. Assume that the FC values to a pixel p have been computed to all the pixels in Ω, given a different pixel p', Ω can be partitioned into three subsets $\Omega^1_{pp'}$, $\Omega^2_{pp'}$, and $\Omega^3_{pp'}$, such that, $\forall q \in \Omega^1_{pp'}$, $\mathcal{FC}(p,q) > \mathcal{FC}(p,p')$; $\forall q \in \Omega^2_{pp'}$, $\mathcal{FC}(p,q) = \mathcal{FC}(p,p')$; and $\forall q \in \Omega^3_{pp'}$, $\mathcal{FC}(p,q) < \mathcal{FC}(p,p')$. To compute the FC values to p', $\forall q \in \Omega^1_{pp'}$, we always have $\mathcal{FC}(p',q) = \mathcal{FC}(p,p')$; $\forall q \in \Omega^3_{pp'}$, we always have $\mathcal{FC}(p',q) = \mathcal{FC}(p,q)$; and $\forall q \in \Omega^2_{pp'}$, we always have $\mathcal{FC}(p',q) \geq \mathcal{FC}(p,p')$, and when $\mathcal{FC}(p',q) > \mathcal{FC}(p,p')$, the strongest path between q and p' are included in $\Omega^2_{pp'}$.

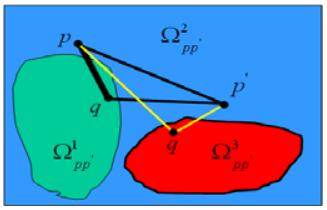

Figure 4. Illustration for the proposition 1, in which the black dots denote pixels, and thicker lines connecting two pixels mean larger values of fuzzy connectedness between them.

Proposition 1 implies that once we obtain the FC value from each pixel to one pixel p, to compute its FC value of all pixels to a new pixel p', only pixels in $\Omega^2_{pp'}$ (blue region in Figure 4) need to recompute their FC using the strongest path search algorithm. The remaining pixels in $\Omega^1_{pp'}$ and $\Omega^3_{pp'}$ (the green and red regions in Figure 4) can simply reuse their previously computed FC. Furthermore, the FC for a pixel in $q \in \Omega^2_{pp'}$ is either equal to $\mathcal{FC}(p,p')$ or the strongest path for computing FC is in the domain $\Omega^2_{pp'}$. Thus, we can simply initialize the FC values of pixels in domain $\Omega^2_{pp'}$ as $\mathcal{FC}(p,p')$, and then use the Dijkstra-FC-T algorithm to search for the strongest path within the constrained domain of $\Omega^2_{pp'}$ instead of the whole image. This can significantly reduce the computation cost.

Proposition 1 aims to simplify the computation of pixel-pixel FC. We now extend it to pixel-stroke FC.

Proposition 2. Assume the FC values to some strokes have been computed for all the pixels in Ω. Denote the set of pixels covered by the scribbles as P. Given some new strokes that cover a new set of pixels P', we can partition Ω into three subsets: $\Omega^1_{PP'}$, $\Omega^2_{PP'}$ and $\Omega^3_{PP'}$, such that, $\forall q \in \Omega^1_{PP'}$, $\mathcal{FC}(P,q) > \mathcal{FC}(P,P')$; $\forall q \in \Omega^2_{PP'}$, $\mathcal{FC}(P,q) = \mathcal{FC}(P,P')$; and $\forall q \in \Omega^3_{PP'}$, $\mathcal{FC}(P,q) < \mathcal{FC}(P,P')$. To compute the FC values to P', $\forall q \in \Omega^1_{PP'}$, we always have $\mathcal{FC}(P',q) = \mathcal{FC}(P,P')$; $\forall q \in \Omega^3_{PP'}$, we always have $\mathcal{FC}(P',q) = \mathcal{FC}(P,q)$; and $\forall q \in \Omega^2_{PP'}$, we always have $\mathcal{FC}(P',q) \geq \mathcal{FC}(P,P')$, and when $\mathcal{FC}(P',q) > \mathcal{FC}(P,P')$, all of the pixels on the strongest path between q and P' are included in $\Omega^2_{PP'}$.

Similar to proposition 1, Proposition 2 reveals that given the FC values of all pixels to some stroke pixels P, to compute the FC values of all pixels to the newly added stroke pixels P', we do not need to recompute the strongest path for pixels in $\Omega^1_{PP'}$ and $\Omega^3_{PP'}$. Instead, we can directly compute their FC from their previously computed FC to P. For the remaining pixels in $\Omega^2_{PP'}$ that require recomputing the strongest path, the search domain is within $\Omega^2_{PP'}$. To compute the FC values for pixels in $\Omega^2_{PP'}$ to the newly added

scribbles, we initialize the FC values of pixels in the new scribbles to be 1, and set the template T values in $\Omega^2_{PP'}$ as 1 and others as 0. We then use the Dijkstra-FC-T algorithm to do the computation.

Finally, after we update the FC values of all pixels to the newly added strokes P', the FC value for each pixel q to all strokes $P \cup P'$ is simply the maximum of the FC values of q to P and q to P'.

5. Results and Discussions

FuzzyMatte updates the matte in an online interactive setting and generates high-quality mattes for complex images. The images in Figure 1 show one example of using FuzzyMatte. As new scribbles are gradually added, the resulting matte gets further improved. The run time between each iteration also significantly decreases. The only exception is at the third iteration where the computation time increases. This is because the newly added strokes at the third iteration have introduced new foreground colors and the affinity values need to be recomputed and the fuzzy connectedness (FC) values need to be recomputed from scratch, as is discussed in Section 4.

In Figure 5, we compare FuzzyMatte with other state-of-art approaches: interactive BP [22] and Spectral Matting [8]. Spectral Matting [8] was implemented in Matlab while interactive BP [22] in C++. For Spectral Matting, we treat the computation of the spectral components as a preprocessing and do not count it towards the final processing time at each iteration. FuzzyMatte is implemented in C++ and for all examples in this paper, except for the peacock image (Figure 6), we set $\lambda = 0.7$ in Eqn. (2). All algorithms are run on a 2.39 GHz PC.

In Figure 5, we compare the run time for each iteration. Although it is difficult to fairly compare these algorithms (some implemented in MatLab and some in C++), it is still easy to see that the run time of FuzzyMatte is significantly reduced at each iteration of scribble refinement, whereas it remains the same when using other methods. FuzzyMatte best compares with interactive BP [22] (both implemented in C++). The processing time of the first iteration is approximately the same. However, at subsequent iterations, when more and more user scribbles are added, FuzzyMatte re-estimates the matte much faster than interactive BP [22]. We only need to recompute the FC values for a very small subset of the pixels. For example, the last iteration of FuzzyMatte in Figure 5 only requires updating 9.6% of all pixels. This makes FuzzyMatte a highly useful tool for interactive matting, particularly at the later stage of interactive refinements, when a lot of small strokes need to be added to recover the fine details.

We also find that all three methods produce high-quality mattes with sufficient scribbles, however, they improve the matte in different ways at each iteration of refinement. For

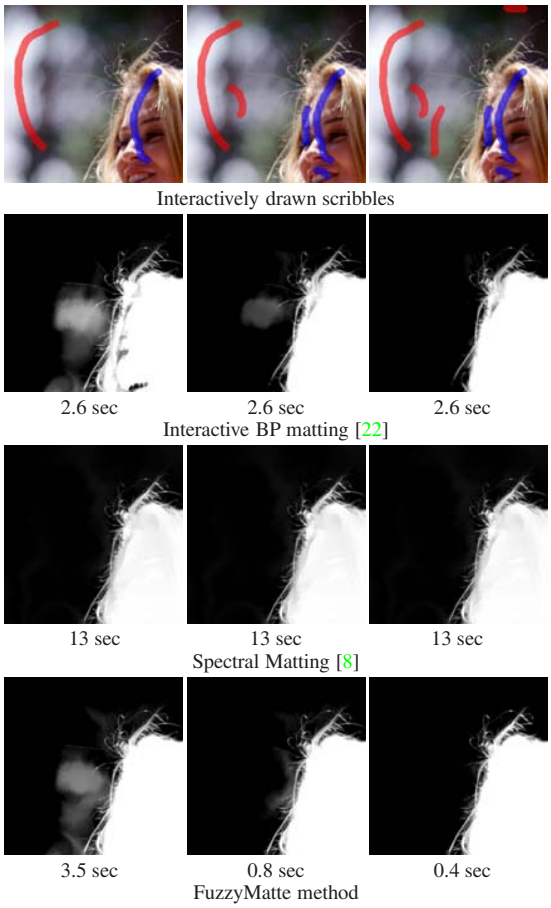

Figure 5. Comparison of different matting schemes and their speeds with interactive scribbles.

example, spectral matting generates high-quality mattes for pixels near the foreground with only two simple strokes (top row of Figure 5). However, the errors in background do not reduce much with additional scribbles, whereas FuzzyMatte, and interactive BP significantly improve the quality of the matte near the background with more scribbles. Furthermore, interactive BP and Spectral Matting both generate large errors for pixels in the foreground that have similar colors to the background (e.g., the black regions near the "forehead" of the matte images in Figure 5). FuzzyMatte, on the other hand, does not. In FuzzyMatte, if a pixel's surrounding neighbors all have a high FC to the scribbles, the pixel itself has a good chance to have a high FC because of the usual high pixel-pixel similarity (Eqn 3). The only exception is when the color difference between the pixel and the strokes is large, i.e., the pixels-scribble similarity is low (Eqn 4). FuzzyMatte balances the two terms by setting an appropriate λ when computing the affinity in (2).

In Figure 6, we show FuzzyMatte results for highly complex images. For the peacock image, we use about 3 strokes at the first iteration, one on the foreground and two on the background. We set $\lambda = 0.5$ in the equation (2) for com-

(a) (b) (c)

Figure 6. Results with the FuzzyMatte. (a). Original images with foreground and background strokes. (b). Extracted mattes with FuzzyMatte. (c). Composite images.

puting affinity and it takes about 4 seconds for the initial matte. Notice that the peacock has a lot of fine details at its tail. Thus, we gradually add a total of around 7 additional strokes near the tail and the feet over about 3 iterations. The computational cost at each iteration is about 0.5 second, much faster than most previous matting methods. The flame image usually requires a lot of gradual refinement using small strokes [22, 9] to generate a satisfactory matte. Since FuzzyMatte is particularly efficient with gradual refinement, it is highly suitable to process these kinds of images. In Figure 6, we composite the extracted foreground into a new background using the estimated matte created with FuzzyMatte.

We have also conducted a quantitative evaluation of our method using the ground truth dataset provided in [8]. In Figure 7, we compare our method with the interactive BP matting [22] and Spectral Matting [8]. We plot the average SSD errors proposed by [8]. Our evaluation shows that FuzzyMatte generates accurate matting results comparable to Spectral Matting. It is also possible to further refine our matting results by first thresholding the computed FC values to create a trimap and then applying more accurate matting estimation algorithms [22, 16] using the trimap.

6. Conclusion and Future Work

An ideal scribble-based image matting method should interactively update the estimated matte when additional scribbles are added. In this paper, we have presented an almost real-time interactive matting tool called FuzzyMatte. FuzzyMatte significantly reduces the wait time between each iteration of the interactive loop by effectively integrating the new inputs with the previously estimated matte. FuzzyMatte is based on the notion of fuzzy connectedness originally proposed in fuzzy medical segmentation [20]. Given an image and the initial user inputs (scribbles or a trimap), FuzzyMatte first computes, for each pixel p with an

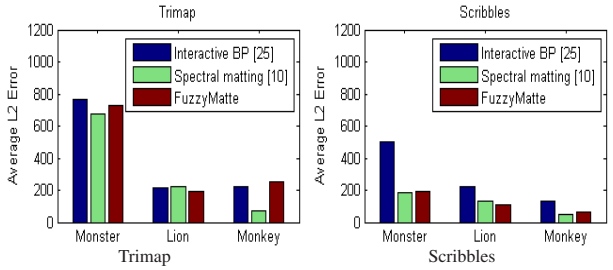

Figure 7. Comparisons of the errors in mattes computed by trimaps and scribbles, with the public ground truth matting data [8].

unknown alpha, its fuzzy connectedness (FC) to the known foreground and background by searching for the strongest connected path. The alpha value of p can be computed from FC.

When additional inputs become available, FuzzyMatte does not require recomputing of all pixels' FC by searching again for the strongest connected path. Instead, it partitions the pixels into three subsets based on their old FC values and only a small number of pixels in one of the three subsets requires searching for the strongest connectedness path. All other pixels simply reuse the previously estimated FC. This significantly reduces the computational overhead between each iteration of the interactive refinement. In addition, for pixels that need recomputing the FC using the strongest path, we present an efficient algorithm by confining the search space for the strongest path. This further reduces the computational cost for re-estimating the matte. We have demonstrated FuzzyMatte on a wide range of images and experiments have shown that FuzzyMatte updates the matte in an online interactive setting and generates high-quality mattes for complex images.

The major limitation of FuzzyMatte is when a user wants to arbitrarily remove a scribble, the FC of all pixels need to be recomputed using the strongest connected path algorithm. Thus it will eliminate the key advantage of our

approach on efficiently reusing the previously estimated matte. Although most existing matting methods also require completely recomputing the matte from scratch in these situations, we believe efficiently updating the matte instead of recomputing it is a desirable feature in an interactive matting system. In the future, we intend to explore novel algorithms to handle these cases using a similar technique for adding the scribbles as described in Section 4.2. Furthermore, we plan to extend FuzzyMatte to processing 3D stacks of images and videos.

APPENDIX

Algorithm Name: Dijkstra-FC-T
Symbols:
Ω: the set of pixels in the image
Ω_o: the set of pixels in the strokes for foreground when $o = f$ or background when $o = b$
$\mathcal{FC}^o(p)$: fuzzy connectedness of pixel p to Ω_o
$\mathcal{A}^o(p_1, p_2)$: affinity between pixels p_1 and p_2
p, p': pixels
$\emptyset$: empty set
Q: priority queue managed by a Fibonacci heap, in which the fuzzy connectedness value is treated as the priority
$\max(Q)$: element with largest priority in Q
T: template for determining if a pixel can be inserted in the queue, where 1 means "yes" and 0 mean "no".
EXTRACT, INSERT: operations for Q, to extract or insert one element, respectively.
begin
Set fuzzy connectedness of the pixels in $\{\Omega - \Omega_o\}$ as 0, and in Ω_o as 1
1: INSERT(Q, Ω_o)
2: **while** $(Q \neq \emptyset)$
3: $p :=$ EXTRACT-max(Q)
4: **for** each p' such that $\mathcal{A}^o(p, p') > 0$ & $T(p) = 1$
5: **if** $(\min(\mathcal{FC}^o(p), \mathcal{A}^o(p, p')) > \mathcal{FC}^o(p'))$
6: $\mathcal{FC}^o(p') := \min(\mathcal{FC}^o(p), \mathcal{A}^o(p, p'))$
7: **if** $p' \in Q$
8: update p' in Q
9: **else**
10: INSERT(Q, p')
11: **end if**
12: **end if**
13: **end for**
14:**end while**
end

References

[1] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV07*, 2007. 1, 2

[2] Y. Chuang, A. Agarwala, B. Curless, D. Salesin, and R. Szeliski. Video matting of complex scenes. *SIGGRAPH'02*, 2002. 1, 2

[3] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *CVPR01*, 2001. 1, 2

[4] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 2, 4

[5] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34:596–615, 1987. 4

[6] L. Grady, T. Schiwietz, S. Aharon, and R. Westerman. Random walks for interactive alpha-matting. In *VIIP05*, 2005. 2

[7] G. T. Herman and B. M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23:460–474, 2001. 2

[8] A. Levin, A. R. Acha, and D. Lischinski. Spectral matting. In *CVPR07*, 2007. 1, 2, 6, 7

[9] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *CVPR06*, 2006. 1, 2, 7

[10] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, , and F. Durand. Defocus video matting. *SIGGRAPH'05*, 2005. 1, 2

[11] L. G. Nyul, A. X. Falcao, and J. K. Udupa. Fuzzy-connected 3D image segmentation at interactive speeds. *Graphical Models*, 64:259–281, 2002. 2, 4

[12] A. Rosenfeld. Fuzzy digital topology. *Information and Control*, 40:76–87, 1979. 2

[13] C. Rothers, V. Kokmogorov, and A. Blake. GrabCut-Interactive foreground extraction using iterated graph cut. *SIGGRAPH'04*, 2004. 1, 2

[14] P. K. Saha, J. K. Udupa, and D. Odhner. Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation. *Computer Vision and Image Understanding*, 77:145–174, 2000. 3

[15] A. R. Smith and J. Blinn. Blue screen matting. *SIGGRAPH'96*, 1996. 1, 2

[16] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *ACM Trans. Graph*, volume 23, pages 315–321, 2004. 2, 7

[17] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Flash matting. *SIGGRAPH'06*, 2006. 2

[18] J. Sun, J. Sun, S. B. Kang, Z. Xu, X. Tang, and H.-Y. Shun. Flash cut: Foreground extraction with flash and no-flash image pairs. In *CVPR07*, 2007. 1

[19] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV98*, 1998. 3

[20] J. K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996. 1, 2, 3, 4, 7

[21] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: An interactive tool for realtime high quality matting. *SIGGRAPH'07*, 2007. 1, 2

[22] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV05*, 2005. 1, 2, 4, 6, 7

[23] Y. Zheng. *Study On Precise and Automatic Segmentaton of Digital Image*. PhD thesis, Shanghai Jiaotong University, December 2005. In Chinese. 5