A Spatially Varying PSF-based Prior for Alpha Matting

Technical report

Supplementary material to CVPR submission #1291

Note that this technical report is not necessary to understand the submitted paper

Abstract

In this paper we considerably improve on a state-of-the-art alpha matting approach by incorporating a new prior which is based on the image formation process. In particular, we model the prior probability of an alpha matte as the convolution of a high-resolution binary segmentation with the spatially varying point spread function (PSF) of the camera. Our main contribution is a new and efficient deconvolution approach that recovers the prior model, given an approximate alpha matte. By assuming that the PSF is a kernel with a single peak, we are able to recover the binary segmentation with an MRF-based approach, which exploits flux and a new way of enforcing connectivity. The spatially varying PSF is obtained via a partitioning of the image into regions of similar defocus. Incorporating our new prior model into a state-of-the-art matting technique produces results that outperform all competitors, which we confirm using a publicly available benchmark.

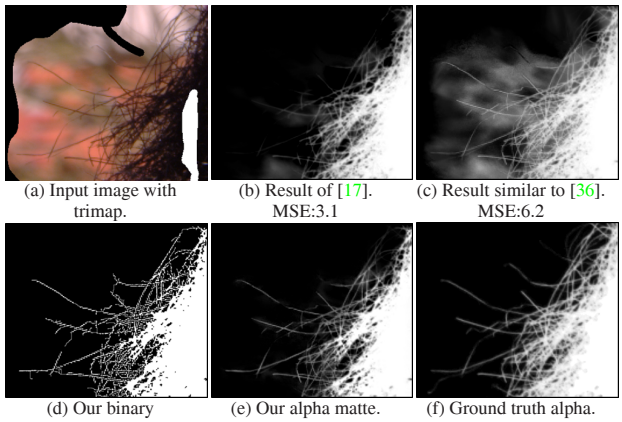
1. Introduction

Alpha matting is the process of extracting a foreground object that is composed with its background. Formally, the observed color C is modeled as a convex combination of the foreground color F and background color B as

$$C = \alpha F + (1 - \alpha)B, \quad (1)$$

where the mixing factor α is referred to as the alpha matte. Recovering alpha given only a single input image C is a severely ill-posed problem. Hence, strong prior models for the alpha matte are necessary to restrict the solution space.

In this paper we use a new prior that is based on the image formation process, studied with respect to the super-resolution (e.g. [3]) and deblurring tasks (e.g. [11, 10, 31]). The image formation process gives useful insights into the reasons that cause the appearance of mixed pixels, i.e. pixels having non-binary α ($0<\alpha<1$): Mixed pixels can be caused by a number of factors such as defocus blur, motion blur, discretization artifacts or light-transmitting scene objects. Thus, apart from light-transmitting objects (e.g. window glass), it is reasonable to assume that mixed pixels are mainly caused by the camera's point spread function (PSF), which accounts for the transparency effects. Hence, we model the prior distribution of the alpha matte as the convolution of a high-resolution binary segmentation with the spatially varying point spread function of the camera.



(a) Input image with trimap. (b) Result of [17]. MSE:3.1 (c) Result similar to [36]. MSE:6.2

(d) Our binary segmentation (e) Our alpha matte. MSE:2.4 (f) Ground truth alpha.

Figure 1. **Why is our prior useful?** Ambiguities in alpha matting are often not resolved by state-of-the-art algorithms (b,c). Our strong prior, based on a PSF and segmentation (d), can better resolve matting ambiguities as reflected in our final alpha matte (e).

This is in contrast to all previous matting approaches (except [24], which we discus below), which infer alpha directly from eq. (1) without committing to an explicit image formation model. However, this has a major drawback as we discuss now. Levin et al. [17] were the first to show that if one assumes the colors of the fore- and background to vary linearly inside a small patch, the alpha matte can be derived in closed form. The resulting matte of [17] is shown in fig. 1(b), given the image and trimap in fig. 1(a). The result is imperfect (some hairs cut-off). It has been observed (e.g. [19]) that a major problem is that for insufficient user input (i.e. large trimap) the cost function used in [17] has a large space of (nearly) equally likely solutions[1]. There have been several approaches to overcome this deficiency. Wang et al. [36] introduced data terms in the framework of [17], based on color models of the fore- and background regions. However, the result is even worse, see fig. 1(c). The prob-

[1] Another problem is that the color line model does not hold for highly textured patches, which is however in our experience less important.

lem is that some dark-green areas in the image background are explained as semi-transparent layers, i.e. dark-green is a mix of dark foreground with green background. This is a plausible solution given the color observations, however it is a solution which is physically very unlikely. Hence, previous work (e.g. [19, 36, 23]) used a "generic sparsity" prior, which forces as many pixels as possible to an alpha value of 0 or 1. Our prior, based on the image formation process naturally encodes sparsity of the matte. This is because under our model it is very likely that transparencies occur only at the object boundary and most parts of the alpha matte are either 0 or 1. In contrast to a generic sparsity prior which is employed to each pixel *independently*, our prior depends on the underlying binary segmentation. We will show that our prior achieves better results than "generic sparsity" priors.

The work closest to our approach is [24], where the idea of a prior motivated by an image formation model has been introduced. They showed that their prior can effectively resolve ambiguities in the alpha matte (we confirm this observation in our experiments). However, [24] models the prior probability of alpha as the convolution of a binary segmentation with a *spatially constant* PSF. This model is an oversimplification of the reality, where the PSF can vary over the image with respect to the scene depth. An example is shown in fig. 2, where two PSFs are necessary to describe the alpha matte of the foreground object.

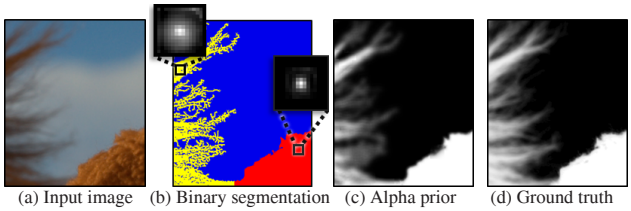
In contrast to [24], we model the prior distribution of the alpha matte as a convolution of an underlying, potentially higher resolution, binary segmentation α^b with a *spatially varying* point spread function K, whose result is potentially downsampled:

$$\alpha = D(K \otimes \alpha^b), \tag{2}$$

where $\otimes$ denotes convolution and D is the downsampling function. Note that there are other major differences to the approach of [24], detailed in sec. 2. We will show that our approach generates superior results.

To construct our prior, the key challenge is to solve the blind deconvolution problem, which is the reconstruction of the binary segmentation α^b and spatially varying PSF K in eq. (2) from an input alpha matte. Thus the main contribution of this paper is a new and efficient approach for the deconvolution of alpha mattes. Our method assumes that the spatially varying PSF is a single peaked kernel, which is in general true for optical or very slight motion blur (a limitation is complex motion blur). If our assumption is met, it has been shown by Joshi et al. [11] that the binary segmentation can be recovered from the edges in the blurred alpha matte. Hence, we infer the binary mask with a new MRF-based segmentation technique. Also, our approach exploits flux and a new efficient way to enforce connectivity of the foreground object.

To recover a spatially varying PSF, our algorithm partitions the foreground object into regions of similar defocus blur and recovers a PSF in each of these regions. Here, our main contribution is a new, efficient approach to infer the amount of defocus at each pixel of the foreground object. Our defocus estimation method generates results that compare well to specialized approaches proposed for this task.



(a) Input image (crop of a soft toy) (b) Binary segmentation and spatially var. PSF (c) Alpha prior (d) Ground truth alpha matte

Figure 2. **Our PSF prior.** For image (a) our approach computes the binary segmentation and defocus of the foreground (b). The color of the foreground (red/yellow) indicates small/large defocus. PSFs computed for the red/yellow regions are shown in (b). Convolving the segmentation (b) with the corresponding PSFs gives an alpha prior (c) that is close to the ground truth (d).

Convolving the recovered binary segmentation with the PSF gives an alpha matte which typically is of high quality (see e.g. fig. 2(c)). However, to account for potential artifacts in the alpha matte (due to e.g. discretization or inaccurate PSF), we use the convolved segmentation as prior in the matting method of [23]. The result is a matte whose quality exceeds the current state-of-the-art.

It is interesting to note that our matting approach can be seen as generalization of the segmentation-based "border matting" method of GrabCut [26]. In fact [26] fits an alpha profile to the binary segmentation, which could be generated from a (spatially constant) Gaussian PSF. However, the authors of [26] conclude that PSF-based border matting is not applicable to "difficult mattes", resulting from e.g. hair (a similar conclusion was recently made in [21]). This work shows that even for complex mattes such an approach is feasible and moreover outperforms state-of-the-art methods.

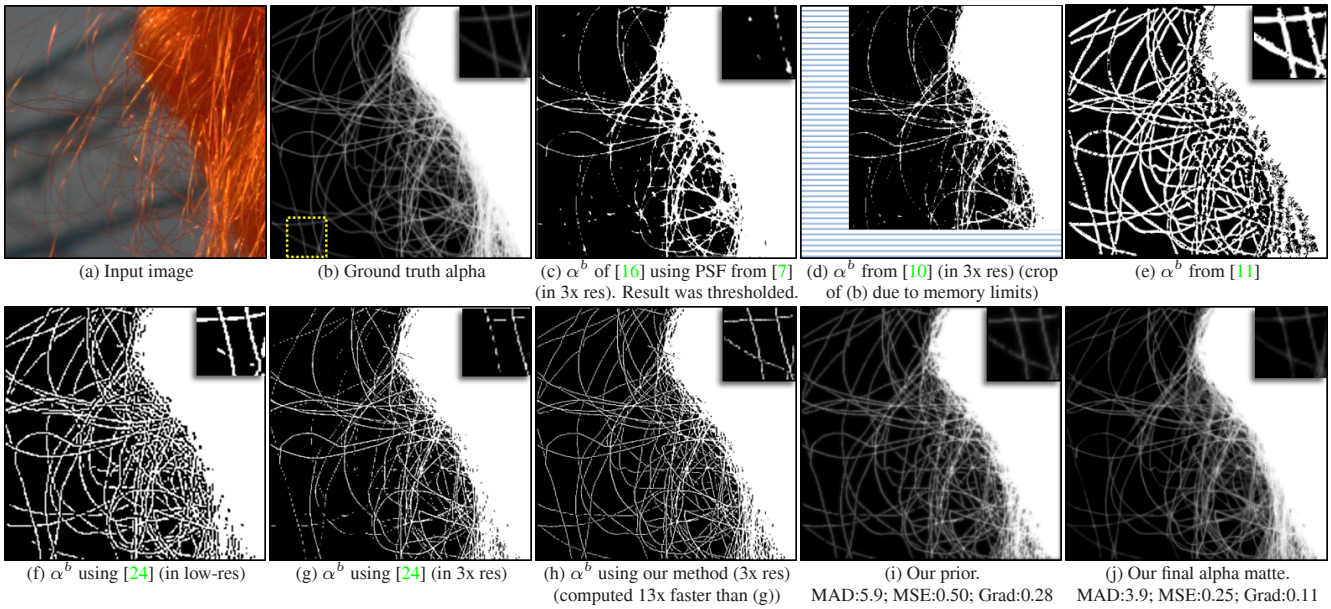
Finally, note that in the near future our segmentation-based matting approach might become even more applicable, since the depth information provided by emerging consumer 3D cameras (e.g. Fuji 3D W1) could be used to greatly simplify the PSF estimation procedure.

In the following, we first review and compare related work in section 2 and section 3.4. In section 3 we detail our approach to estimate the prior model. Section 4 gives an experimental comparison.

2. Related work

There are two main areas of related work: alpha matting and blind deconvolution. We discussed related matting approaches in sec. 1 and the reader is referred to the survey of [34] for more details. Recovering the binary segmentation and PSF from an alpha matte is the task of blind deconvolution and we discuss the related work in the following.

In this section we use the *ground truth* alpha matte α^* from [25] for comparing deconvolution methods. However, for matting (sec. 3) we use an alpha matte, *computed* from



(a) Input image (b) Ground truth alpha (c) α^b of [16] using PSF from [7] (in 3x res). Result was thresholded. (d) α^b from [10] (in 3x res) (crop of (b) due to memory limits) (e) α^b from [11]

(f) α^b using [24] (in low-res) (g) α^b using [24] (in 3x res) (h) α^b using our method (3x res) (computed 13x faster than (g)) (i) Our prior. MAD:5.9; MSE:0.50; Grad:0.28 (j) Our final alpha matte. MAD:3.9; MSE:0.25; Grad:0.11

Figure 3. **Comparison of blind (and non-blind) deconvolution methods from a ground truth alpha matte (b).** Our deconvolution approach (h) estimates the underlying binary segmentation better than previous approaches for this task (c-g). Note that all results were computed in 3x higher resolution and downscaled afterwards. Thus segmentation results may not be completely binary. See text for details.

the *input image* with a standard matting algorithm. To ensure that the underlying segmentation is more likely to be binary, we upscaled α^* by a factor of 3 before applying the methods discussed below (we discuss upscaling in sec. 3.2).

In theory one should be able to perfectly reconstruct α^b with deconvolution algorithms, given the true α^* and the true K, respectively. (We also confirmed this in a synthetic experiment). However, in practice we found the results obtained with state-of-the-art blind deconvolution (i.e. simultaneously estimating α^b and K) approaches, e.g. [29], to be inappropriate for our purposes. More specifically, we observed that the deconvolved alpha mattes were usually far away from being binary. This empirical observation was recently confirmed in the work of Levin et al. [20] which shows that the simultaneous MAP estimation of both K and α^b mostly favors the no-blur explanation (i.e. K is the delta kernel). To overcome this problem, Levin et al. [20] suggested to first estimate the PSF using the approach of [7] and then perform (non-blind) deconvolution using [16]. We tested this approach, using the authors' implementations, but unfortunately the results were still non-binary. Hence, to obtain α^b we had to threshold the deconvolution results, which resulted in the loss of many details like hair strands. Figure 3(c) is an example. Since [7] was mainly designed for large motion blur, we also used [11] to initialize the PSF for [16] but found it to give non-binary results as well.

A possible explanation for this failure is that state-of-the-art deblurring approaches are based on natural image statistic priors that are not applicable to alpha mattes. In particular, the desired deblurred alpha matte is a two-tone image, thus has a much simpler structure than a natural image. Experiments in Levin et al. [20] suggest that a prior which favors two-tone images could potentially overcome the undesired no-blur solution. Therefore, one could follow the approach of Jia [10] and incorporate in the deconvolution process the assumption that the unblurred alpha matte is binary. The authors of [10] kindly applied their method on a crop of a ground truth matte (fig. 3(b)). The result is shown in fig. 3(d), where unfortunately many fine details were lost.

One could also employ the sparsity prior of [19] directly on α^b, as proposed in Dai et al. [5]. However, Rhemann et al. [24] found such an approach to be inferior to their own method. Also, [5] additionally applies an edge smoothness prior to α^b, which is, however, invalid at hairy boundaries according to [5]. Finally, α^b in [5] is not necessarily binary.

Another class of deconvolution approaches explicitly detect edges in the image to infer a binary segmentation. For instance, the recent approach by Joshi et al. [11] detects the location of the step edge in the (unknown) sharp image by applying a sub-pixel accurate edge detector to the blurred image. If the deblurred image is two-toned (which is true for alpha mattes), the location and orientation of the sharp image edges is sufficient to infer α^b around the detected edges. We found this method to perform reasonably well on solid boundaries, but it severely over-estimated α^b in the presence of thin structures like hair strands, which can be attributed to an incorrect edge localization, see e.g. fig. 3(e).

The work most closely related to our approach is Rhemann et al. [24], where α^b is iteratively obtained from the deconvolved alpha matte using an MRF that preserves the edges in the deblurred alpha. This method can effectively preserve thin structures like hair strands. The result of [24] is shown in fig. 3(f). Although most details could be pre-

served, α^b was overestimated and originally connected hair strands are fragmented (see upper right corner of fig. 3(f)).

In this work we improve on the approach of [24] in several respects. Firstly, we propose to work on the higher-resolution (upscaled) alpha matte, where the underlying segmentation of thin structures is more likely to be binary. We also found this to greatly improve the result of [24], an example of which is shown in fig. 3(g). Secondly, our approach works directly on the alpha matte as opposed to [24], where computationally expensive deconvolution methods were applied to alpha before binarization. (We observed a speed up factor of about 13 compared to [24].) Thirdly, we apply a different procedure to estimate α^b based on flux and connectivity (sec. 3.3). Finally, we estimate the spatially-varying amount of blur over the foreground object, which relaxes the assumption of a spatially constant PSF in [24].

Fig. 3(h) shows α^b obtained with our method using the ground truth α^*. We see that most of the fine details were nicely recovered and the foreground is connected. Convolving our computed α^b with our estimated PSF yields the result in fig. 3(i), which is very close to the ground truth, both visually and in terms of error rates. To further refine this result, we use it as prior in the approach of [23], see fig. 3(j). This example shows that our prior has the potential to approximate even very detailed mattes with high accuracy.

3. Our matting approach

We now detail our matting approach, which comprises five steps: (i) Given an image and trimap, compute an initial (usually imperfect) alpha matte α with the matting method of [23]; (ii) upscale α to a resolution where the underlying segmentation is more likely to be binary (apart from discretization); (iii) estimate the binary segmentation α^b with an MRF; (iv) downsample α^b and compute the spatially varying PSF; (v) convolve α^b with the PSF and use the result as prior in the framework of [23] to compute the final alpha matte. Each step is now described in detail.

3.1. Estimating the initial alpha matte

We have seen in sec. 2 that the binary segmentation and PSF may be derived using deconvolution approaches from the ground truth alpha. To apply our approach to natural images where the ground truth is unknown, we infer the segmentation and PSF from an alpha matte computed from the natural image with a conventional matting algorithm. (Note, the same task was addressed in [24, 10].) In this work we use the matting method of Rhemann et al. [23]. In short, they first compute a pixel-wise estimate of alpha denoted as $\hat{\alpha}$, which defines the data term. The data term is combined with the smoothness term of [17], giving the objective function:

$$J(\alpha) = \alpha^T L \alpha + (\alpha - \hat{\alpha})^T \hat{\Gamma} (\alpha - \hat{\alpha}), \tag{3}$$

where α and $\hat{\alpha}$ are treated as column vectors and L is the matting Laplacian of [17]. The diagonal matrix $\hat{\Gamma}$ weights the data against the smoothness term. The objective function is minimized by solving a set of sparse linear equations, subject to the user constraints. To obtain high-resolution mattes we solve (3) in overlapping windows as in [23].

3.2. Upsampling alpha

It is possible that small structures like hair strands project to a camera sensor area which is smaller than a pixel. To ensure that the underlying binary structure is at least of the size of one pixel, we compute α on a higher-resolution pixel grid. Thus we bicubically upscale the image to a resolution where the underlying segmentation is likely to be binary.

To determine a good scaling factor f, let us imagine a high-resolution 3x3 pixel alpha matte where the center pixel is completely opaque and all other pixels are completely transparent. Bicubicly downsampling this alpha matte by a factor of $f = 3$, gives a single pixel with an opacity value of $\alpha = 1/f^2$. Hence, using a scaling factor of 3 we can recover all structures with $\alpha \geq 1/9$.

In practice we can recover even more details with the same scaling factor because of additional defocus blur (which was neglected in the above analysis). Thus we found that a scaling factor of 3 is sufficient to preserve most details in our test images. However, further work could be conducted to learn the optimal scaling factor in a user study.

3.3. Estimating the binary segmentation

Assuming that the PSF is a single-peaked kernel, our approach recovers the binary mask α^b from the upscaled α by solving the following submodular energy with graph cut:

$$E(\alpha^b) = \sum_{i \in \mathcal{I}} D_i(\alpha_i^b) + \theta_1 F_i(\alpha_i^b) + \theta_2 \sum_{\{i,j\} \in \mathcal{N}} V_{ij}(\alpha_i^b, \alpha_j^b), \tag{4}$$

where α^b is the binary labeling and $\mathcal{N}$ denotes an 8-conn. neighborhood on the set of image pixels $\mathcal{I}$. The parameters θ_1,θ_2 balance the terms in eq. (4) and were set as in sec. 4.

The data term D_i encourages α^b to be close to α:

$$D_i(\alpha_i^b) = \delta(\alpha_i^b = 1) \cdot L_i, \tag{5}$$

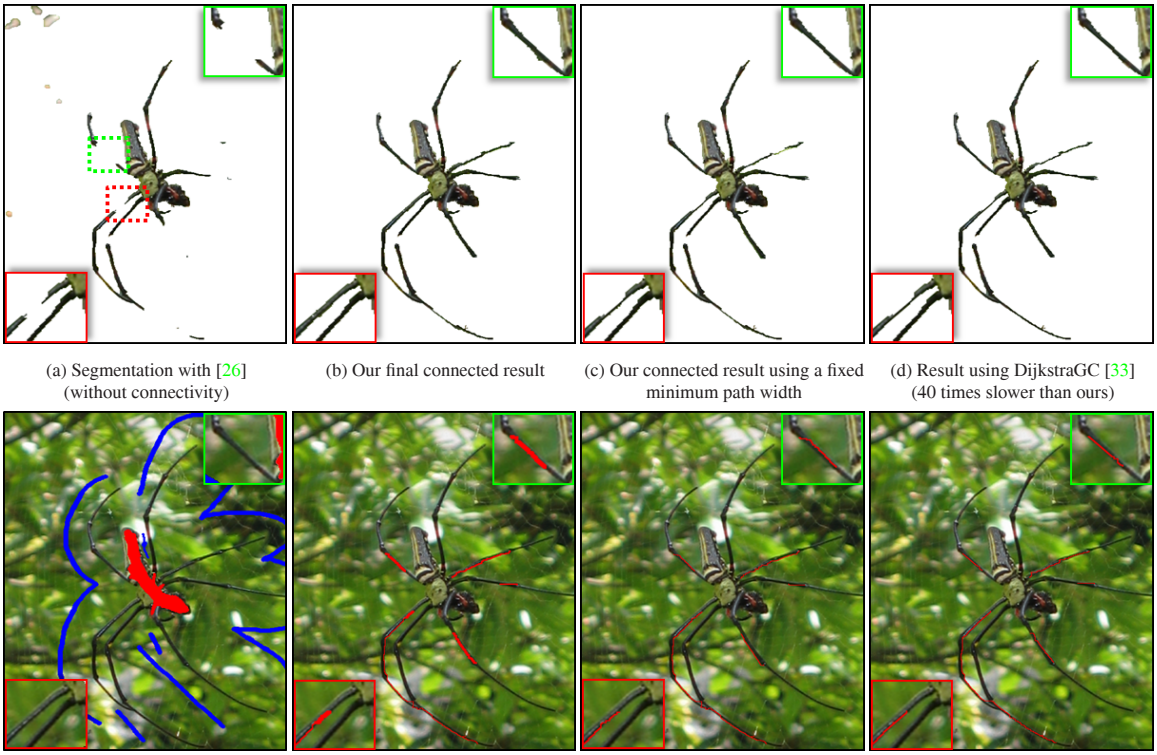
where δ is the Kronecker delta and $L_i = -\log(2\alpha_i) + \log(2(1-\alpha_i))$ is the difference of the negative log likelihood that a pixel i with alpha value α_i belongs to the fore- or the background, respectively.[2]

To detect edges and to preserve thin structures like hair strands in the segmentation, we use flux which has been shown to be effective for segmenting thin objects in medical grayscale images [32] and has been demonstrated to be amenable for graph cut minimization [14]. The unary term F_i represents the flux of the gradient in L_i:

$$F_i(\alpha_i^b) = \delta(\alpha_i^b = 0) \cdot div\left(\nabla L_i \cdot \exp\left(-|L_i|/\sigma\right)\right), \tag{6}$$

where ∇ and div denote the gradient and divergence and σ was fixed to 2. In F_i, the exponential function is used

[2] The diff. of the log likelihoods is a re-parameterization of the energy.



(a) Segmentation with [26] (without connectivity)
(b) Our final connected result
(c) Our connected result using a fixed minimum path width
(d) Result using DijkstraGC [33] (40 times slower than ours)
(e) Image with scribbles (blue=bkg; red=fgd)
(f) Input image. Computed connected paths from (b) are marked red.
(g) Input image. Computed connected paths from (c) are marked red.
(h) Input image. Computed connected paths from (d) are marked red.

Figure 4. **Enforcing connectivity.** Given an input image and user constraints (e), GrabCut [26] gives a disconnected segmentation (a). Our approach automatically connects or excludes disconnected islands in (a) to the foreground. Our final segmentation (b) includes most of the spider legs and shows no background artifacts. The result of our approach, where we disable the automatic estimation of the minimum width of the "connection path" (hence, we use a fixed minimum width of 1 pixel) is shown in (c). As expected it is worse than (b). Our results (b,c) are comparable to the result of DijkstraGC (d), which is, however, 40 times slower than our approach. We show the "connection paths" for the results in (b-d) in (f-h). Note that for this example we replaced eq. (4) with the energy in [26].

to truncate the gradient in places where the foreground and background likelihoods in L_i are approximately equal. To avoid that the flux term is affected by noise, we smooth the gradient of L_i with a Gaussian filter of variance 1.5 before computing the divergence in eq. (6). We observed that the upsampling process leads to a "fattening" of F_i. To compensate for this, we lower the magnitude of F_i in places where F_i is not a local maximum.

Note that to preserve thin structures, [24] used a pairwise MRF term. However, the flux term used in our approach has a better theoretical justification and is easier to optimize.[3]

Finally, our pairwise term V_{ij} encodes the ising prior:

$$V_{ij}(\alpha_i^b, \alpha_j^b) = \delta(\alpha_i^b \neq \alpha_j^b). \quad (7)$$

An example of α^b is shown in fig. 1(d).

Enforcing connectivity

To additionally regularize the binary segmentation, we enforce the foreground object to be a single 4-connected component. In general, this assumption is true for all non-occluded objects as well as for all images used for evaluation in sec. 4. Recently, a solution to this task has been presented in [22]. Unfortunately, their solution to this $\mathcal{NP}$-hard problem requires the image to be segmented into large superpixels for computational reasons. Thus it is impractical for segmenting fine structures like hair strands. An interactive solution to this problem was proposed in Vicente et al. [33]. They start by computing a segmentation without connectivity constraints (e.g. fig. 4(a)). Then the user manually marks a pixel, which has to be connected to the main part of the foreground object, and also manually selects a minimum width for the "connection path". The method finds a connected component which fulfills these constraints.

In this work we propose a new approach to compute an entirely connected segmentation, which in contrast to previous work is very efficient and fully automatic. In essence, we automate the user interactions of [33] while maintaining a low energy, and also make the core algorithm of [33] much more efficient while keeping high quality results.

In detail, we first compute a segmentation $\hat{\alpha}^b$ by minimizing (4) without connectivity constraints (fig. 4(a)). Then those regions in $\hat{\alpha}^b$ which are disconnected from a source region s are identified. We define s to be all pixels in $\hat{\alpha}^b$ that are 4-connected to the user marked foreground pixels (e.g. spider body in fig. 4(a)). Then for each disconnected region t a segmentation $\hat{\alpha}^{b'}$ is computed by minimizing (4) under

[3] We found that the pairwise term in [24] gives a non-submodular energy, although differently stated in [24].

the constraint that s and t must be connected. (This step is discussed in detail below.) We also determine an alternative solution $\hat{\alpha}^{b''}$, by simply removing region t from $\hat{\alpha}^{b}$. Now we keep the solution with lower energy, i.e. we keep e.g. $\hat{\alpha}^{b'}$ if $E(\hat{\alpha}^{b'}) \leq E(\hat{\alpha}^{b''})$. In this manner all disconnected regions are processed, which gives the final result (fig. 4(b)).

The difficult step in the above procedure is to find a segmentation subject to the condition that regions s and t are connected. Vicente et al. [33] suggested a heuristic method called *DijkstraGC*. It works by computing the "shortest path" in a graph where the "distance" between two nodes measures the value of the energy (4) under the constraint that all pixels on the path from s to t belong to the foreground. Unfortunately, DijkstraGC is computationally very expensive, since it requires many calls to the maxflow algorithm to minimize function (4).[4] Hence, we found it impractical to compute a solution for many disconnected islands.

The key idea of our approach is to compute the shortest path on a graph where the weight of each node is its min-marginal energy under (4), which is given by

$$M(i) = \min_{\alpha^b, \alpha^b_i = 1} E(\alpha^b) - \min_{\alpha^b} E(\alpha^b), \qquad (8)$$

and can be computed very efficiently using graph recycling [12]. (The path to all disconnected islands can be computed in a single run of Dijkstra.) A segmentation is then computed by minimizing (4) under the constraint that all pixels on the shortest path in the min-marginals belong to the foreground. Hence, our approach approximates DijkstraGC but gives comparable results (for instance, compare our result in fig. 4(b) with the result of DijkstraGC in fig. 4(d)).

Finally, we address the problem of finding the minimum width of the "connection path". It has been observed in [33] that DijkstraGC might result in undesired one-pixel-wide segmentations (see e.g. fig. 4(c,d)). In [33] this problem was fixed by manually specifying a minimum width for each connecting path (see [33] for details). We automate this process by computing multiple shortest paths with different widths $\varphi \in \{1, .., 4\}$ for each disconnected island and choose the path which gives the segmentation with the lowest costs under (4). Note that we encourage thicker paths by dividing the costs of paths where $\varphi > 1$ by a factor of 1.005.

3.4. Estimating a spatially varying PSF

Most previous work that can be used to estimate a PSF from alpha, assumes a constant blur kernel over the whole image (e.g. [24, 10]). However, in real world scenes the PSF may vary over the image due to lens imperfections, motion blur or defocus blur that varies with the scene depth.

To account for spatially varying motion blur, [28] proposed an interactive deblurring method which is, however, limited to rotational motions. Another approach is to estimate the PSF in local sub-windows, assuming constant blur in each window (see e.g. [11]). Clearly, such an approach fails if the PSF changes rapidly due to depth discontinuities. In the limit, a window-based approach could be used to compute a PSF for every pixel. However, there might not be enough constraints to reliably estimate a PSF at each pixel locally. Hence, smoothness priors on neighboring kernels could be used to regularize the result, as in the Filter Flow framework [27]. A drawback of such an approach are the immense runtime and memory requirements ([27] reported several hours of runtime for low-res. images). Moreover, the smoothness prior in [27] is limited to linear metrics, which might oversmooth depth discontinuities.

The basic idea of our approach is to segment the image into regions exhibiting similar defocus blur and then estimate a PSF in each of these regions separately.[5] Thus the key challenge is to estimate the amount of defocus, which can be characterized by the radius R (i.e. the spatial extent) of the PSF K. Recently, a solution to this task has been proposed in [16]. However, it requires the image to be captured using a camera with a modified aperture. Also their method is potentially slow, since computationally expensive deconvolution algorithms are applied to the image several times (the authors report a runtime in the magnitude of hours). The method closest to our approach is Bae et al. [1], where the level of blurriness is automatically computed at image edges (similar to [6]) and then propagated to the rest of the image by adapting the approach of [18]. We will qualitatively compare [16] and [1] to our approach in fig. 6.

Our approach differs from [1] in several ways. Firstly, we compute local defocus measures along the boundary of α^b, which usually coincides with the object outline. This is potentially more reliable than using interior edges for blur estimation, which might originate from shading or attached shadows. Secondly, we use a different method for local blur estimation. Thirdly, by working on the alpha matte, as opposed to the image, we can formulate an effective confidence measure for the amount of blur. Finally, we propagate the local defocus information using discrete optimization, enabling the use of edge preserving affinities.

In more detail, we formulate the defocus estimation of the blur kernel radius inside the foreground object as the following MRF and optimize it using alpha expansion:

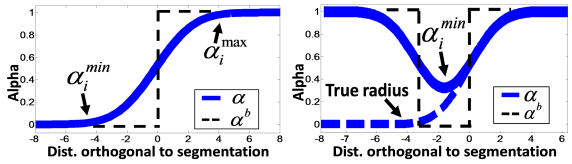
$$E(R) = \sum_{i \in \Omega} B_i(R_i) \cdot \rho_i + \sum_{\{i,j\} \in \mathcal{N}} W_{ij}(R_i, R_j), \quad (9)$$

where Ω denotes the set of pixels at the boundary of α^b and $\mathcal{N}$ is an 8-connected neighborhood defined over all foreground pixels of α^b (i.e. where $\alpha^b = 1$). Here, ρ_i is the confidence of the data term at pixel i, and R_i is the discretized radius of the PSF at pixel i (we use 12 radii $R \in \{1, .., 12\}$).

To construct the data term B_i consider fig. 5(a). It shows the 1D profile of α orthogonal to the boundary of α^b. The

[4] In [33] the computational burden was reduced by recycling flow and search trees[13], but the authors of [33] found that its effectiveness was significantly reduced, since nodes had to be (un)fixed in an unordered fashion.

[5] This is similar to e.g. [15], where the image was segmented into motion layers before deconvolution.



(a) PSF radius (b) Confidence

Figure 5. **Data term estimation.** (a) The radius of the PSF is determined by the max/min values in the alpha profile (see text for details). (b) The data term might be unreliable two step edges of the segmentation boundary are close to each other. Our confidence measure handles such cases (see text for details).

distance of the local minimum alpha value α_i^{min} along the edge profile to the segmentation boundary gives an estimate of the blur radius.[6] The data term B_i is then defined as $B_i(R_i) = |\alpha_i^{R_i} - \alpha_i^{min}|$, where $\alpha_i^{R_i}$ is the alpha value of the pixel which is at distance R_i away from pixel i in the direction orthogonal to the segmentation boundary.

The data term at pixel i might be unreliable due to artifacts in alpha. Thus we define a pixel-wise confidence for the data term as $\rho_i{=}exp(-\alpha_i^{min}/\theta_3)$, where $\theta_3 = 1.2$. Intuitively, the confidence at pixel i is high if α_i^{min} is zero and lower otherwise (α_i^{min} is zero in a perfect matte).

Another case where our confidence measure is useful is illustrated in fig. 5(b). It shows the alpha profile (solid blue line) generated by a binary segmentation (thin dashed line) whose edges are close to each other. In this case α_i^{min} does not longer coincide with the true PSF radius. The true PSF radius is located at the minimum of the alpha profile that would result from convolving only the right step edge in fig. 5(b) with the PSF (thick dashed line). Our confidence measure accounts for such a failure, since in such cases α_i^{min} is larger than zero (hence, the confidence is lower).

We also construct a data term using the local max. alpha value along the edge profile in the same way. Finally, at each pixel the data term with the higher confidence is chosen.

The pairwise term W_{ij} encodes our assumption that neighboring pixels should have similar kernel radii if they have similar colors in the input image. We implement this assumption using a contrast sensitive truncated linear term:

$$W_{ij}(R_i, R_j) = \delta(R_i \neq R_j) \cdot g(R_i, R_j), \qquad (10)$$

where δ is the Kronecker delta and $g(R_i, R_j)$ is a function based on the difference of colors C_i and C_j in neighboring pixels of the input image C:

$$g(R_i, R_j) = \theta_4 + min(|R_i - R_j|, \theta_5) + \theta_6 \exp(-\beta |C_i - C_j|^2),$$

where θ_5 was fixed to 2 and $\beta = \left(2\left\langle (C_i - C_j)^2 \right\rangle\right)^{-1}$, where $\langle\cdot\rangle$ denotes expectation over the image. The weights θ_4 and θ_6 were chosen such that the smoothness is higher along the object boundary Ω:

$$\{\theta_4, \theta_6\} = \begin{cases} \{0.4, 2\} & \text{if } i \vee j \in \Omega \\ \{0, 0.0001\} & \text{otherwise.} \end{cases}$$

[6] Our approach was inspired by the sharp-edge prediction method in [11].

Optimizing eq. (9) gives an estimate of the PSF radius R for each pixel of the foreground object. We then split the foreground into regions of uniform kernel radii and estimate a PSF in each of these regions separately. In each region, we model the PSF as a kernel K with estimated radius R that comprises non-negative elements that sum up to one. We apply a smoothness prior to K that is given by $\gamma||\nabla K||^2$, where $\gamma = (2R+1)^2$ normalizes the kernel area. Given α^b and α, we obtain K by minimizing the quadratic energy function for all pixels in each region of constant defocus:

$$||\alpha^b \otimes K - \alpha||^2/\sigma^2 + \theta_7 \gamma ||\nabla K||^2, \qquad (11)$$

where $\sigma = 0.005$ denotes the noise level and $\theta_7 = 2$ weights the smoothness prior.[7] For computational reasons we compute K in the original image resolution, thus we bicubicly downsample α^b before PSF estimation.[8]

To give a rough impression about the quality of our approach, we compare the result of our interactive defocus estimation method with the automatic approaches of [1] and [16] in fig. 6 (see discussion in figure caption).[9] In the future, one could try to use our defocus map for further image manipulations such as re-focusing.

3.5. Re-estimating alpha with our PSF Prior

Once the binary segmentation α^b and the spatially-varying PSF K are computed, we construct the prior for alpha as $\alpha^{prior} = (\alpha^b \otimes K)$. We then re-estimate α by using α^{prior} as a data term in the framework of [23]. This was done by replacing $\hat{\alpha}$ in eq. (3) with the term:

$$\hat{\alpha} = \hat{\alpha} + \theta_8 \alpha^{prior}, \qquad (12)$$

where $\theta_8 = 0.08$ is the relative weight of the prior. An example of the final alpha matte is shown in fig. 1(e).

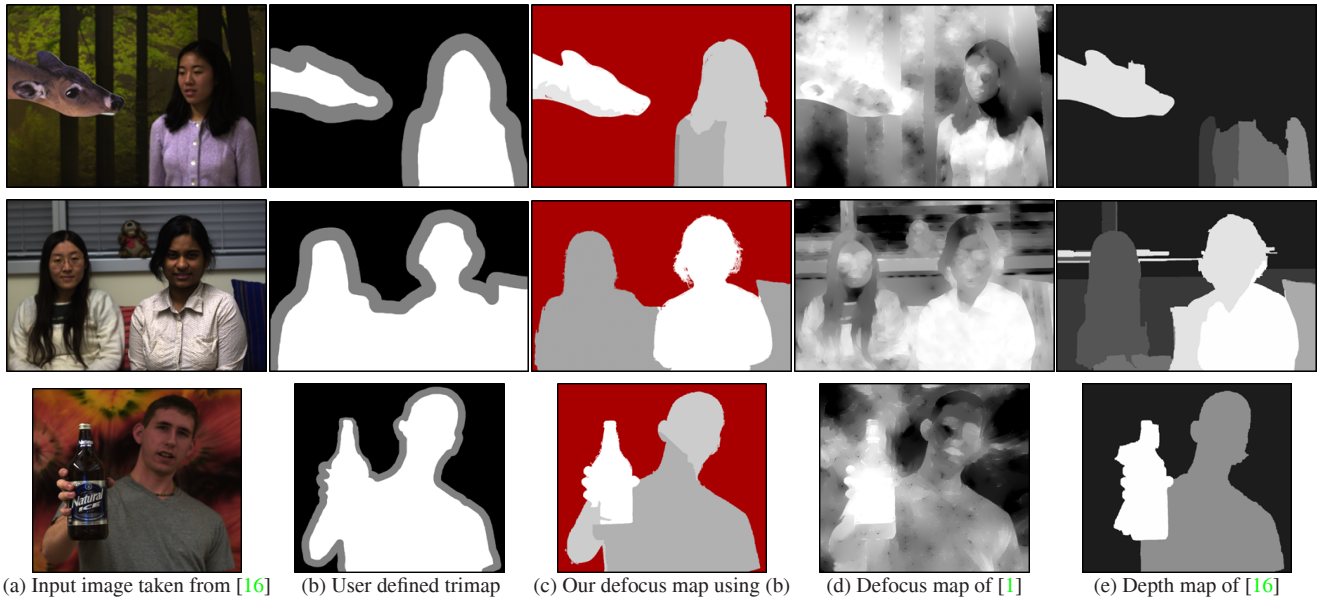
4. Matting results on natural images

We quantitatively evaluated our approach on the recently proposed ground truth benchmark of [25]. At the time of submission the benchmark compares 10 state-of-the-art matting algorithms on 8 (low-resolution) natural images with respect to 4 error metrics. As user input 3 different trimaps per input image are provided. Results of different methods are shown in fig. 1 as well as figures 7-10. Our results for all low-resolution images were computed by setting the parameters (θ_1,θ_2) in eq. (4) to $(200, 0.005)$. We show the overall ranks of selected algorithms obtained from the benchmark of [25] in columns "low-res" of table 1. We see that our method is the top performer on three out of four error metrics. Our approach performs less well on the connectivity metric despite enforcing connectivity of the binary segmentation. This is because the final alpha matte might

[7] In [24], K was derived in a similar way. However, they constrained K to be symmetrical, which cannot account for potential slight motion blur.

[8] We found this to give similar results compared to computing the PSF from the upscaled matte and then downsample the convolved result.

[9] Although the images in fig. 6(a) were recorded with an aperture that generates a multi-peaked PSF, we found our method to work well.



(a) Input image taken from [16] (b) User defined trimap (c) Our defocus map using (b) (d) Defocus map of [1] (e) Depth map of [16]

Figure 6. **A loose comparison of different defocus estimation methods.** Our defocus map (c) was generated with the user-defined trimap (b). The methods of [1, 16] (d,e) are automatic. Here, white encodes small defocus/depth, black means large defocus/depth, and red means background region which is not estimated by our approach. Note, that our result is much cleaner than that of [1] (d) and is of comparable quality to [16] (e). It is important to note that [16] requires the image to be captured with a specialized aperture as well as an exact calibration of the PSF at several depths. Also our solution was computed in a few seconds, thus is orders of magnitudes faster than [16].

still be disconnected. In the future one could investigate approaches that enforce connectivity directly on alpha.

As an additional competitor we replaced the prior in our method (i.e. convolved segmentation) with the one computed by Rhemann et al. [24]. As expected, this competitor performs better then the original method of [24], due to the better initial alpha matte. However, the results are still inferior to our approach which shows the quality of our prior.

Note that the test set used in [25] includes one image that shows a light-transmitting object (translucent plasticbag), which largely violates our assumptions. We excluded this image from the test set and show the overall rankings for the remaining 7 images in column "low-res*" of table 1. As expected, the ranking of our method improves.

It should be noted that the benchmark of [25] is performed on low-resolution ($\approx$ 1Mpix) images where our assumption that the underlying segmentation is binary, might not always be met (even after upscaling). Fortunately, [25] provides additionally 27 high-resolution ($\approx$ 6Mpix) images with public ground truth alpha, which were originally intended for parameter training. We use these images as an additional test set for our matting approach. For the high-resolution data we set the parameters (θ_1,θ_2) in eq. (4) to $(200, 0.05)$. We show the average ranks in column "high-res" of table 1. Our approach is best on all error metrics.

Note that on the high-resolution dataset we only compare against the 5 methods that performed best on the low-resolution data. High-resolution results for [36, 17] were obtained in a multi-resolution framework, as in [24].

We qualitatively compare our method on the crop of a high-resolution image showing fuzzy hair (fig. 7(b)). The results of our competitors (fig. 7(c-g)) show large background artifacts or underestimate alpha inside the foreground object. Even replacing the prior in our method with that of [24] gives inferior results (see the background artifacts in fig. 7(g)). The result of our method (fig. 7(h)) is closest to the ground truth (fig. 7(i)).

Another example is shown in fig. 8(b) which depicts a crop of a high-resolution image showing fine hair strands. The approach of [24] (fig. 8(c)) could partially recover the hair strands but introduced large artifacts in the foreground region. All other competitors (fig. 8(d-g)) underestimated the fine hair strands. In contrast, our approach could better preserve the hair (fig. 8(h)) and is visually close to the ground truth (fig. 8(i)).

Fig. 9 shows results on the crop of a high-resolution image depicting the out-of-focus boundary of a soft toy (fig. 9(b)). The results of all competing approaches (fig. 9(c-g)) show artifacts in the background. The result of [17] (fig. 9(g)) is closest to our result, but still shows slight artifacts around the object boundary (arrows in fig. 9(g) point to the artifacts). Interestingly, the prior of [24] had almost no effect on the result (compare fig. 9(e) and (f)). This is presumably because [24] is limited to a single PSF, hence cannot handle out-of-focus regions (in out-of-focus-regions the method of [24] degrades to a standard matting method). Our approach has no such limitations, which is reflected in our result (fig. 9(h)) that is close to the ground truth (fig. 9(i)).

Finally, we show results on the crop of a low-resolution image (fig. 10(a)). We see that the approach of [17] is clos-

Method	Ranking for SAD			Ranking for MSE			Ranking for Grad.			Ranking for Conn.		
	low-res	low-res*	hi-res	low-res	low-res*	hi-res	low-res	low-res*	hi-res	low-res	low-res*	hi-res
Our result	**2.4**	**2.1**	**2.1**	**2.5**	**2.1**	**2.0**	**2.0**	**1.8**	**1.9**	5.1	4.6	**2.1**
Imp. Col. Mat. [23] with prior of [24]	2.6	2.5	2.8	3.5	3.4	3.2	2.8	2.3	3.8	4.2	3.5	3.8
Improved Color Matting [23]	3.0	3.1	2.8	2.6	2.8	2.4	2.6	2.8	3.1	4.2	3.8	2.8
Closed-Form Matting [17]	3.5	3.5	2.8	3.8	3.9	3.4	4.6	5.0	3.0	3.1	3.2	2.2
Robust Matting [36]	5.0	5.4	4.7	4.6	5.0	4.3	4.8	4.9	4.0	7.0	6.9	4.4
High-res Matting [24]	6.0	5.8	5.8	5.5	5.1	5.7	5.2	5.0	5.3	5.4	5.7	5.7
Random Walk Matting [8]	7.7	7.5	-	7.8	7.7	-	7.8	8.1	-	**2.0**	**2.1**	-

Table 1. **Comparison on alphamatting.com.** We show the overall ranks (as defined in [25]) of the top performing matting approaches on the benchmark of [25] wrt. four error metrics. Our approach performs best wrt. three out of four error metrics. See the text for a discussion.

est to our method, but oversmoothed the hole in the foreground (fig. 10(l)). The other approaches (fig. 10(b-k)) either introduced large background artifacts or completely cut off the hair. Our approach (fig. 10(m)) shows the cleanest result. The ground truth for this test image, obtained from the benchmark of [25] is hidden from the public.

5. Conclusions

In this work we have shown that state-of-the-art alpha matting approaches can be improved by incorporating a prior that models the alpha matte as convolution of a binary segmentation with the spatially varying PSF. We proposed a new and efficient deconvolution approach, based on flux and connectivity that recovers this binary segmentation. We further introduced a new and efficient method to infer the amount of defocus at each pixel of the foreground object. This enabled us to recover a PSF which varies due to scene depth. We demonstrated that our method improves over the state-of-the-art on a ground truth matting benchmark.

References

[1] S. Bae and F. Durand. Defocus magnification. In *Eurographics*, 2007. 6, 7, 8

[2] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, 2007. 12

[3] S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *CVPR*, 2000. 1

[4] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *CVPR'01*. 12

[5] S. Dai and Y. Wu. Removing partial blur in a single image. In *CVPR*, 2009. 3

[6] J. Elder and S. Zucker. Local scale control for edge detection and blur estimation. *PAMI*, 1998. 6

[7] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. *SIGGRAPH*, 2006. 3

[8] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. In *VIIP'05*. 9, 12

[9] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting: A stroke based approach for continuous image matting. In *Eurographics*, 2006. 12

[10] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007. 1, 3, 4, 6

[11] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *CVPR*, 2008. 1, 2, 3, 6, 7

[12] P. Kohli and P. Torr. Measuring uncertainty in graph cut solutions. In *ECCV*, 2006. 6

[13] P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in markov random fields. *PAMI*, 2007. 6

[14] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *ICCV*, 2005. 4

[15] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2006. 6

[16] A. Levin, R. Fergus, F. Durand, and W. Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 2007. 3, 6, 7, 8

[17] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *CVPR'06*. 1, 4, 8, 9, 11, 12

[18] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *SIGGRAPH*, 2004. 6

[19] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *CVPR*, 2007. 1, 2, 3

[20] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009. 3

[21] J. Liu, J. Sun, and H. Shum. Paint selection. *SIGGRAPH*, 2009. 2

[22] S. Nowozin and C. Lampert. Global connectivity potentials for random field models. In *CVPR*, 2009. 5

[23] C. Rhemann, C. Rother, and M. Gelautz. Improving color modeling for alpha matting. In *BMVC*, 2008. 2, 4, 7, 9, 11, 12

[24] C. Rhemann, C. Rother, A. Rav-Acha, and T. Sharp. High resolution matting via interactive trimap segmentation. In *CVPR*, 2008. 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12

[25] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *CVPR09*, to appear. 2, 7, 8, 9, 11, 12

[26] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004. 2, 5

[27] S. Seitz and S. Baker. Filter flow. In *ICCV*, 2009. 6

[28] Q. Shan and W. X. anb J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007. 6

[29] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *SIGGRAPH*, 2008. 3

[30] J. Sun, J. Jia, C. Tang, and H. Shum. Poisson matting. *SIGGRAPH*, 2004. 12

[31] Y. Tai, H. Du, M. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *SIGGRAPH ASIA*, 2008. 1

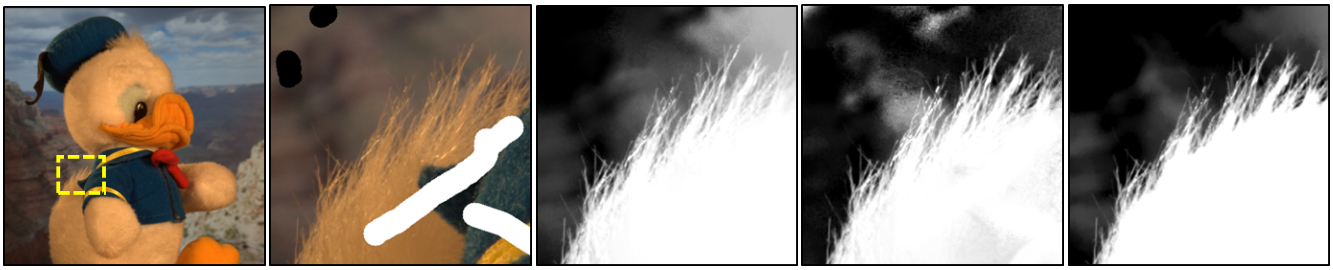
[32] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. *PAMI*, 2002. 4

[33] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. *CVPR08*. 5, 6

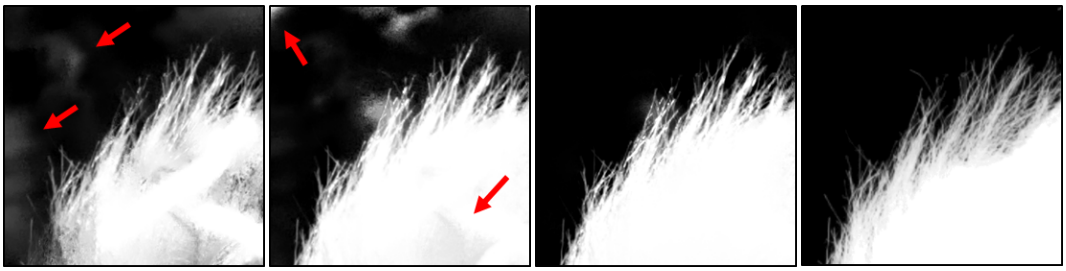
[34] J. Wang and M. Cohen. Image and video matting: A survey. *Foundations/ Trends Comp. Graphics and Vision*, 2007. 2

[35] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, 2005. 12

[36] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007. 1, 2, 8, 9, 11, 12

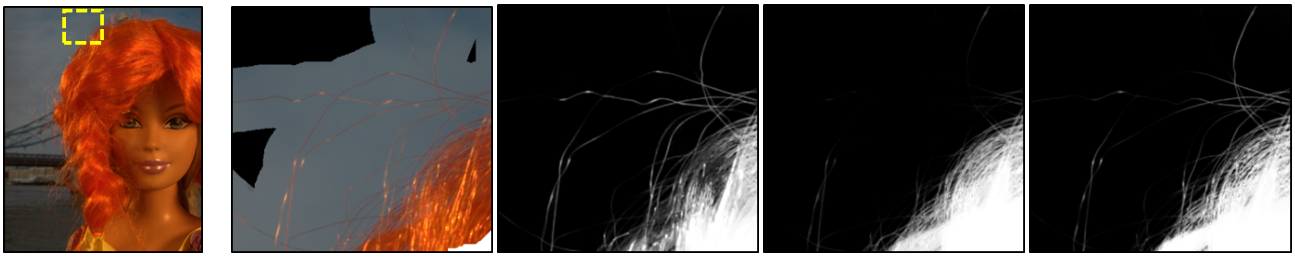


(a) Full image (b) Image crop + trimap (c) Closed-Form Matting [17]. SAD:19.6 (d) Improved Color Matting [23]. SAD:13.0 (e) Robust Matting [36]. SAD:10.7

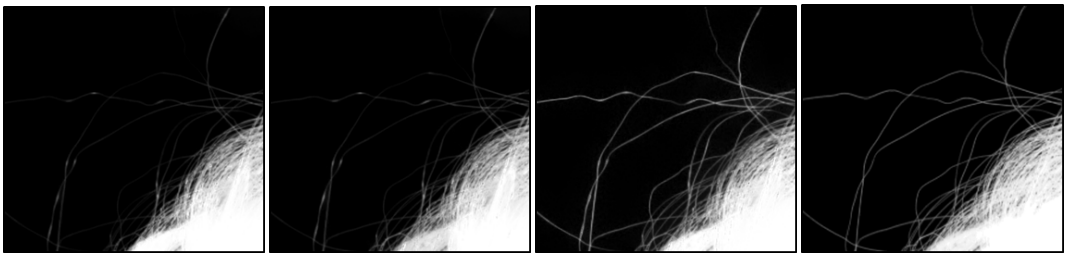


(f) High-resolution Matting [24]. SAD:10.6 (g) Improved Color Matting [23] with prior of High-res. Matting [24]. SAD:10.1 (h) Our result. SAD:5.0 (i) Ground truth alpha

Figure 7. **High-resolution matting comparison (1).** (c-h) Results for a crop of an image (obtained from the benchmark of [25]) (b). Arrows point to minor artifacts. See the text for a discussion.

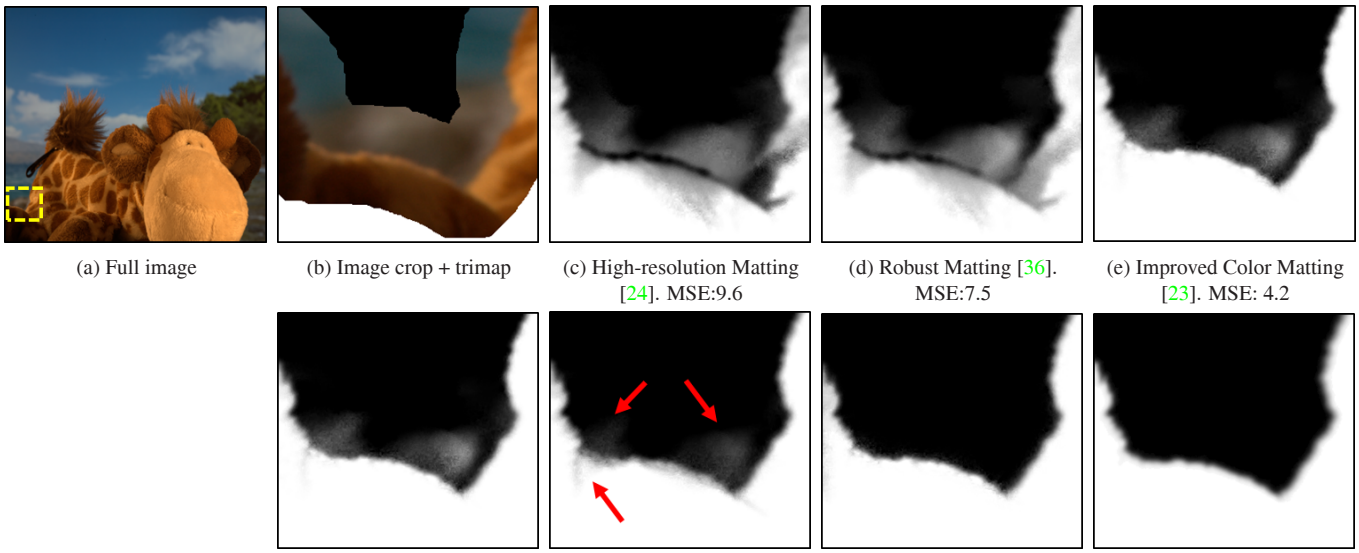


(a) Full image (b) Image crop + trimap (c) High-resolution Matting [24]. MSE:1.78 (d) Closed-Form Matting [17]. MSE:0.35 (e) Improved Color Matting [23] with prior of High-res. Matting [24]. MSE:0.34



(f) Improved Color Matting [23]. MSE:0.26 (g) Robust Matting [36]. MSE:0.24 (h) Our result. MSE:0.23 (i) Ground truth alpha

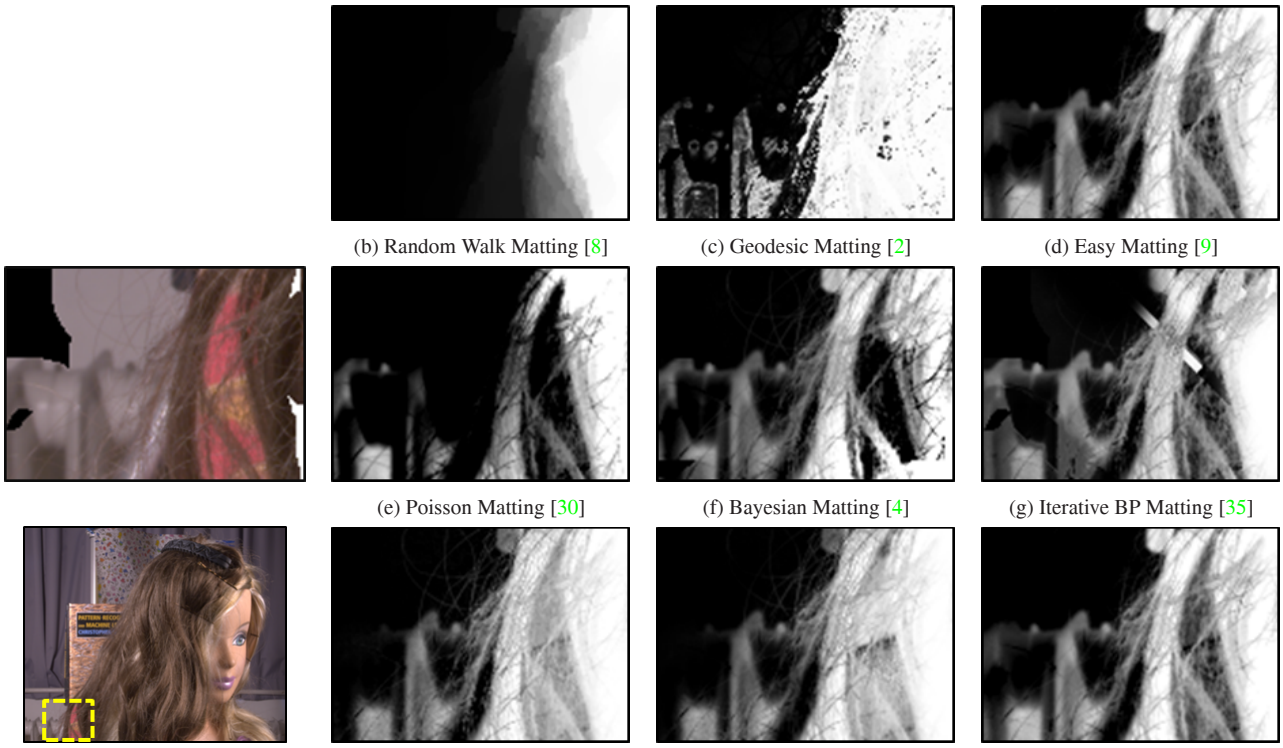
Figure 8. **High-resolution matting comparison (2).** (c-h) Results for a crop of an image (obtained from the benchmark of [25]) (b) showing fine hair strands of a doll. See the text for a discussion.



(a) Full image (b) Image crop + trimap (c) High-resolution Matting [24]. MSE:9.6 (d) Robust Matting [36]. MSE:7.5 (e) Improved Color Matting [23]. MSE: 4.2

(f) Improved Color Matting [23] with prior High-res. Matting [24]. MSE:4.2 (g) Closed-Form Matting [17]. MSE:2.1 (h) Our result. MSE:1.8 (i) Ground truth alpha

Figure 9. **High-resolution matting comparison (3).** (c-h) Results for a crop of an image (obtained from the benchmark of [25]) (b) showing the out-of-focus boundary of a soft toy. Arrows point to small artifacts in (g). See the text for a discussion.



(b) Random Walk Matting [8] (c) Geodesic Matting [2] (d) Easy Matting [9]

(e) Poisson Matting [30] (f) Bayesian Matting [4] (g) Iterative BP Matting [35]

(a) Full image (bottom) and image crop + trimap (top) (h) Improved Color Matting [23] (i) Improved Color Matting [23] with prior of High-res. Matting [24] (j) Robust Matting [36]

(k) High-resolution Matting [24] (l) Closed-Form Matting [17] (m) Our result

Figure 10. **Low-resolution matting comparison.** On the crop of this challenging test image (a) our approach (m) could better resolve the matting ambiguities than its competitors (b-l). The ground truth for this test image of [25] is hidden from the public. See the text for details.