Improvements of Bayesian Matting

Mikhail Sindeyev, Vadim Konushin, Vladimir Vezhnevets
Department of Computational Mathematics and Cybernetics, Graphics and Media Lab
Moscow State Lomonosov University, Moscow, Russia
{msindeev, vadim, vvp}@graphics.cs.msu.ru

Abstract

Digital image matting is a process of extracting a foreground object from an arbitrary natural image. Unlike the image segmentation task it is required to process fuzzy objects (like hair, feathers, etc.) and produce correct opacity channel for them. The result can then be composited onto a new background or edited by processing foreground and background layers separately.

Digital image matting has become a compulsory step in many photo-editing and video-compositing tasks. Currently professional digital artists have to accurately trace objects contours and paint the details to achieve maximum quality. Our aim is to create a convenient workflow for automating this process and make it possible to effectively handle high-resolution images.

In this paper we show how a smoothness constraint can be incorporated into Bayesian matting algorithm framework as additional regularization to improve the result quality without affecting the computation speed. We also demonstrate the hierarchical approach that significantly increases processing speed without noticeable loss of quality. This allows us to create convenient digital image matting system.

Keywords: *Bayesian matting, image editing, digital compositing, foreground extraction, alpha estimation.*

1. INTRODUCTION

In the matting problem it is assumed that the source image C is a composite of two images F and B (foreground and background) with opacity channel α. These values should satisfy the RGB-space compositing equation in each pixel:

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

where C, F and B are 3D vectors of RGB values, $0 \le \alpha \le 1$. The task is to reconstruct the α, F and sometimes B images from the source image C using some additional user input.

Typically matting algorithms takes a source image and a trimap image as input. Trimap image is a user-specified segmentation of the image into three regions: foreground, background and unknown. While the former two provide the knowledge about the object to be extracted, the latter denotes the area to which the algorithm should be applied. The result of the algorithm is a foreground image layer with color and opacity information available for each pixel, and a background layer. When composited together, these two layers should produce exactly the source image.

The problem is to reconstruct F, B and α values at each pixel from single observation C from a limited user input. The trimap specifies the areas with $\alpha = 0$ ($B = C$), $\alpha = 1$ ($F = C$) and unknown F, B, α. Note that if two of these three values are known, the fourth one can be easily calculated. The problem is heavily under-constrained, since for each color C there is an infinite number of combinations of foreground and background colors. In order to constrain the problem and make it formally solvable some regularization is required.

In the next section we make an overview of most notable algorithms, that propose different regularizations of the problem. In the third section we describe Bayesian matting algorithm in more details and propose our improvements. We show how a smoothness constraint can be incorporated into Bayesian matting algorithm. We also demonstrate the hierarchical approach and discuss its possible integration with smoothness constraint. After it we show the results and comparisons with other algorithms and outline the future work.

2. PREVIOUS WORK

In this section we briefly overview several state-of-the-art matting algorithms and outline their main ideas.

Knockout algorithm requires a precise trimap, ideally with unknown region containing only pixels with $0 < \alpha < 1$. When processing an unknown region pixel, F and B values are estimated by averaging color along the foreground/background region border in the neighborhood of the pixel being processed. α value is then calculated for each color component independently and weighted average is used as final α value. While being very fast, this algorithm produces poor results when F and/or B values in the pixel are inconsistent with the color along the corresponding region boundary. This happens in many images and the algorithm produces incorrect and noisy results.

Ruzon-Tomasi method [6] is a color statistics based algorithm. The distributions for foreground and background colors are modeled as the mixtures of unoriented Gaussians. Color statistics is calculated for rather big image fragments. Then α value is calculated under assumption that color C comes from in-between distribution which is an interpolation of foreground and background distributions. This algorithm maximizes probability density of this distribution in point C. The disadvantage of the algorithm is its relying on the color statistics over large image sub-regions which usually contains many overlaps and cannot be handled correctly.

Bayesian matting [2] also uses color statistics, but performs per-pixel color distribution estimation. Pixels are processed starting from foreground and background region borders contracting unknown region step by step. Pixels processed on earlier steps provide new foreground and background samples in addition to pixels from known regions. Used color model is a set of oriented Gaussians. Algorithm involves Bayesian framework to maximize the likelihood of F, B and α values. Conditional probability for F, B and α given observed color C can be written using Bayes's rule as:

$$\mathrm{P}(\mathrm{F},\mathrm{B},\alpha \mid \mathrm{C}) = \frac{\mathrm{P}(\mathrm{C} \mid \mathrm{F},\mathrm{B},\alpha)\mathrm{P}(\mathrm{F})\mathrm{P}(\mathrm{B})\mathrm{P}(\alpha)}{\mathrm{P}(\mathrm{C})}, \quad (2)$$

where P(C|F,B,α) is estimated using the distance between *C* and the mix of *F* and *B* (i.e. by the norm of the difference of the left hand side and right hand side of equation (1)),

P(F) and P(B) are estimated via probability density of foreground and background Gaussians,

P(α) is ignored (assuming all α values to be equiprobable),

P(C) is constant relatively to maximization parameters.

There is an extension of Bayesian matting algorithm proposed in [1]. It introduces P(α) term (which is ignored in the original algorithm) based on learnt priors (joint distribution of image and α gradients) and some additional priors, e.g. image edge magnitude. However, it uses global non-linear minimization of the energy function which is probably very slow (there is no time comparison in [1]). Their edge prior is more suitable for matting hard edges and probably oversharpens smooth objects, e.g. hair.

Poisson matting algorithm [7] assumes that *F* and *B* images are smooth in the unknown region. *F* and *B* values are estimated at each pixel by propagating color values from boundary and blurring the result. Poisson's partial differential equation constructed by taking the gradient of equation (1) is used for finding α image. Then the unknown region is reduced by fixing pixels that are close to being pure foreground or pure background, and the procedure is iteratively repeated until convergence. Poisson matting produces poor results when foreground/background image is not smooth (i.e. contains edges) or contains colors that are much different from those on the unknown region border.

Belief propagation algorithm [9] produces good result with very sparse trimaps i.e. containing small foreground and background regions represented with a few strokes with the rest of the image being the unknown region. Discrete set of alpha values is used. The problem is formulated as energy minimization with the expression for energy consisting of data term, which forces *F* and *B* values to conform to the local statistics, and smoothness term.

Markov Random Field (MRF) is constructed for the image pixels and discrete α values and solved using Belief Propagation method. Then the color statistics is refined and the algorithm is applied iteratively until convergence. However, this process is rather slow even for a single iteration and takes a while to converge.

Closed Form Solution to image matting algorithm [4] deals with a quadratic cost function. The main assumption is that colors in *F* and *B* images are locally linear i.e. for each of those images they are approximately linear combinations of two colors. In this case α is linearly dependent on color *C* in small image windows:

$$\alpha_p \approx aC_p + b,$$

where p is a pixel in a small image window (e.g. 3x3), *a* and *b* are coefficients fixed inside this window.

For grayscale images *a* and *b* are related with F and B by the following equations:

$$a = 1 / (\mathrm{F} - \mathrm{B}),$$

$$b = -\mathrm{B} / (\mathrm{F} - \mathrm{B}).$$

For color images they can also be expressed in terms of *F* and *B* (with *a* being a 3D vector):

$$\alpha_p \approx \sum_k a^k C_p^k + b, \quad (3)$$

where k is a color component index.

The cost function is constructed which penalizes the difference between the pixel α value and the one computed using (3). *a* and *b* coefficients are eliminated by expressing them in terms of known *C* and to-be-found α values. Least squares method is used to express *a* and *b* in (3) using α values over the neighborhoods of nearby pixels. It is shown in [4] that this cost function is quadratic with respect to α.

The cost function is then minimized by solving a sparse system of linear equations with matrix of size N by N where N is number of pixels in unknown region. This gives the α image directly from the source image. *F* and *B* are calculated later using another quadratic cost function.

Disadvantages of this algorithm include low computation speed and the lack of color statistics. The latter does not affect many images but usually produces "glows" in alpha channel in small holes and thin grooves (because large number of nearby opaque pixels impedes the propagation of background color information). Also the assumption of local foreground/background color linearity may not hold for noisy images.

3. OUR IMPROVEMENTS OF BAYESIAN MATTING

We have chosen Bayesian Matting algorithm [2] because at the moment it is the best color statistics based non-iterative algorithm and demonstrates optimal speed/quality balance. It doesn't rely on any strong assumptions about alpha and color channels like Poisson matting does. It works with complex distribution of foreground/background colors and its processing time is linear of number of pixels. Usage of statistics-based algorithm allows us to perform recalculation of the result in a small region without need to recalculate the whole image.

In the next several paragraphs we are going to describe Bayesian matting algorithm in more detail to form the base for our improvements.

Taking a logarithm of (2) and omitting terms not affecting the parameters to be calculated, we get

$$\mathrm{L}(\mathrm{F},\mathrm{B},\alpha \mid \mathrm{C}) = \mathrm{L}(\mathrm{C} \mid \mathrm{F},\mathrm{B},\alpha) + \mathrm{L}(\mathrm{F}) + \mathrm{L}(\mathrm{B}), \quad (4)$$

where L(·) = log P(·).

The authors of [2] use the following estimations of L(C|F,B,α), L(F), L(B):

$$\mathrm{L}(\mathrm{C} \mid \mathrm{F},\mathrm{B},\alpha) = -\left\|\mathrm{C} - \alpha\mathrm{F} - (1-\alpha)\mathrm{B}\right\|^2 / \sigma_\mathrm{C}^2$$

(with user-specified σ_C),

$$\mathrm{L}(\mathrm{F}) = -(\mathrm{F} - \overline{\mathrm{F}})^\mathrm{T} \Sigma_\mathrm{F}^{-1} (\mathrm{F} - \overline{\mathrm{F}}) / 2,$$

where $\overline{\mathrm{F}}$ and Σ_F are mean and covariance matrix of foreground Gaussian,

L(B) – similarly to L(F).

The authors of Bayesian Matting left deriving P(α) from ground-truth alpha mattes for future work, but as far as we know did not publish any papers or results on this.

If there are several pairs of foreground/background clusters, optimal *F*, *B* and α are calculated for each pair, then the pair with the greatest likelihood value $\mathrm{L(F,B,}\alpha\,|\,\mathrm{C)}$ is selected.

In order to maximize the non-quadratic function (4) the authors use an iterative procedure by alternately assuming α and F, B to be constant, what gives them two quadratic sub-problems. They use the mean α value over the neighborhood of the pixel being processed as the initial guess.

For constant α the following 6x6 system of linear equations for F and B is derived:

$$\begin{bmatrix} \Sigma_F^{-1} + I\alpha^2/\sigma_C^2 & I\alpha(1-\alpha)/\sigma_C^2 \\ I\alpha(1-\alpha)/\sigma_C^2 & \Sigma_B^{-1} + I(1-\alpha)^2/\sigma_C^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \\ = \begin{bmatrix} \Sigma_F^{-1}\overline{F} + C\alpha/\sigma_C^2 \\ \Sigma_B^{-1}\overline{B} + C(1-\alpha)/\sigma_C^2 \end{bmatrix} \tag{5}$$

For constant F and B the solution for α is simply the projection of C onto line segment FB:

$$\alpha = \frac{(C-B)\cdot(F-B)}{\|F-B\|^2} \tag{6}$$

Calculation of F, B and α by alternating formulas (5) and (6) is repeated until convergence.

3.1 Smoothness constraint

Bayesian matting is sensitive to overlapping of foreground and background Gaussians. In the original algorithm such makes α estimation unstable and usually produces impulse noise in generated opacity channel. Simple blur and median filters can improve alpha channel quality, but small details in the matte can be lost. Instead, we propose to add smoothness term into the Bayesian framework to regularize the estimation process in such cases. We model smoothness as 1D Gaussian with mean value α_0 being the average among already processed pixels, i.e. the same value that is used as initial guess for α when solving the system (5). Our smoothness term is introduced into (2) as P(α).

We use the following term for L(α):

$$\mathrm{L}(\alpha) = -\|\alpha - \alpha_0\|^2/\sigma_\alpha^2$$

Thus we are maximizing the following log-likelihood:

$$\mathrm{L(F,B,}\alpha\,|\,\mathrm{C)} = \mathrm{L(C}\,|\,\mathrm{F,B,}\alpha) + \mathrm{L(F)} + \mathrm{L(B)} + \mathrm{L}(\alpha), \tag{7}$$

Forcing partial derivative of (7) with respect to α to equal zero gives us the following solution for α:

$$\alpha = \frac{\alpha_0/\sigma_\alpha^2 + (C-B)\cdot(F-B)/\sigma_C^2}{1/\sigma_\alpha^2 + \|F-B\|^2/\sigma_C^2} \tag{8}$$

Formula (8) replaces formula (6) in the optimization procedure.

We can define σ_α in several ways. First, we can use fixed user-adjustable value. Second, we can base it on the distance between foreground and background Gaussian to prevent oversmoothing while keeping regions of high uncertainty (caused by overlapping of these Gaussians) consistent with nearby pixels:

$$\sigma_\alpha = \sigma_\alpha^0 + \lambda \cdot dist(P(F), P(B)),$$

where σ_α^0 and λ are user-specified values (in our experiments we set $\sigma_\alpha^0 = \lambda = 0.1$) and $dist(\cdot,\cdot)$ is the distance metrics between two distributions (we use the distance between Gaussians centers).

Third, we can use two-pass Bayesian matting using likelihood values calculated on the first pass for estimating σ_α on the second pass (it is similar to uncertainty map used in [9]).

To make smoothing less uniform and force it to conform to the color changes, we use weighted average for α_0 in pixel *q*:

$$\alpha_0 = \frac{1}{W}\sum_p \alpha_p \cdot w_p$$

where the sum is taken over the already processed (or known) pixels in the neighborhood of pixel *q* with the following weight (W is a sum of all weights w_p for the pixel *q*):

$$w_p = \exp\left(-\frac{\|C_p - C_q\|^2}{2\sigma_w^2}\right)$$

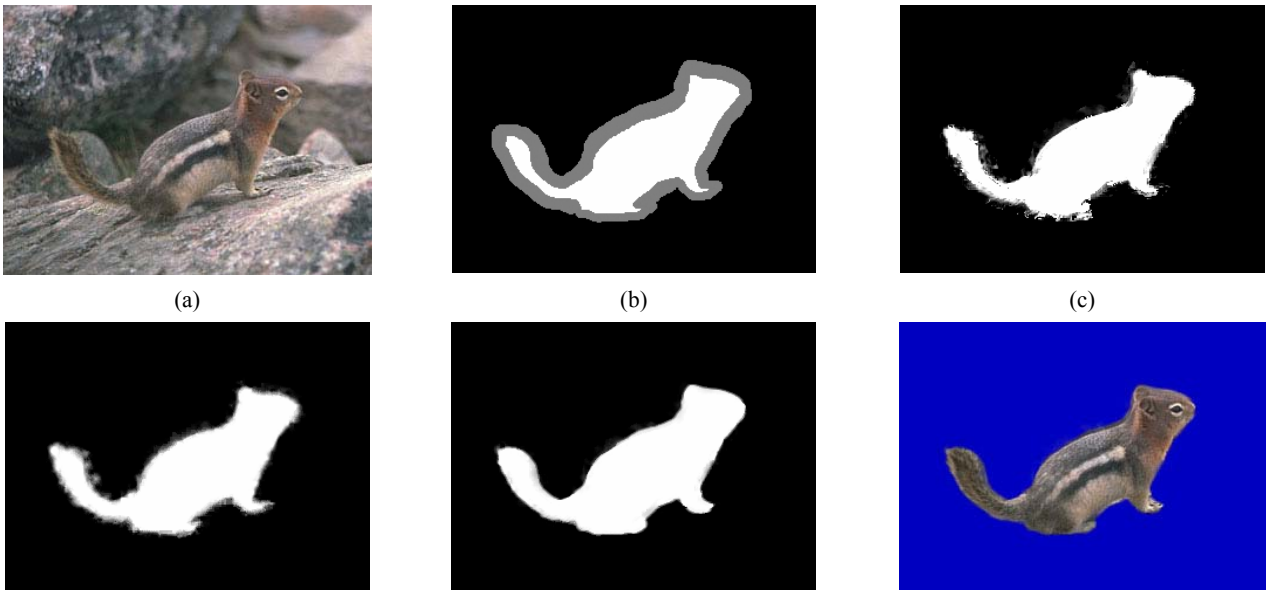
(we have empirically chosen value of 0.2 for σ_w). This α_0 value is also used as initial guess for α in fixed-alpha equation (5). Similar weighted-averaging method is used in many segmentation-related publications.

The usage of smoothness term practically does not affect computation time. The example results are shown in figures 1, 2 and 3. Computation times are compared in section 4.

3.2 Hierarchical approach

Another improvement is the hierarchical Bayesian matting. It aims to reduce processing time without losing the matte quality. A straightforward way to do this is to process small-scale image first, then revert to the source size and perform Bayesian matting again with much smaller sampling radius. But there is a more effective way to do this: by applying Closed Form Solution [4] hierarchical approach to Bayesian matting result we calculate *a* and *b* coefficient images using equation (3) from smaller image and use them to upsample α image back to original size. This allows us to completely eliminate second pass of Bayesian algorithm since we usually get accurate alpha channel. To restore *F* and *B* channels we can also assume that their RGB channels are linear combinations of channels of *C* (though we can also perform second pass of Bayesian matting for constant α).

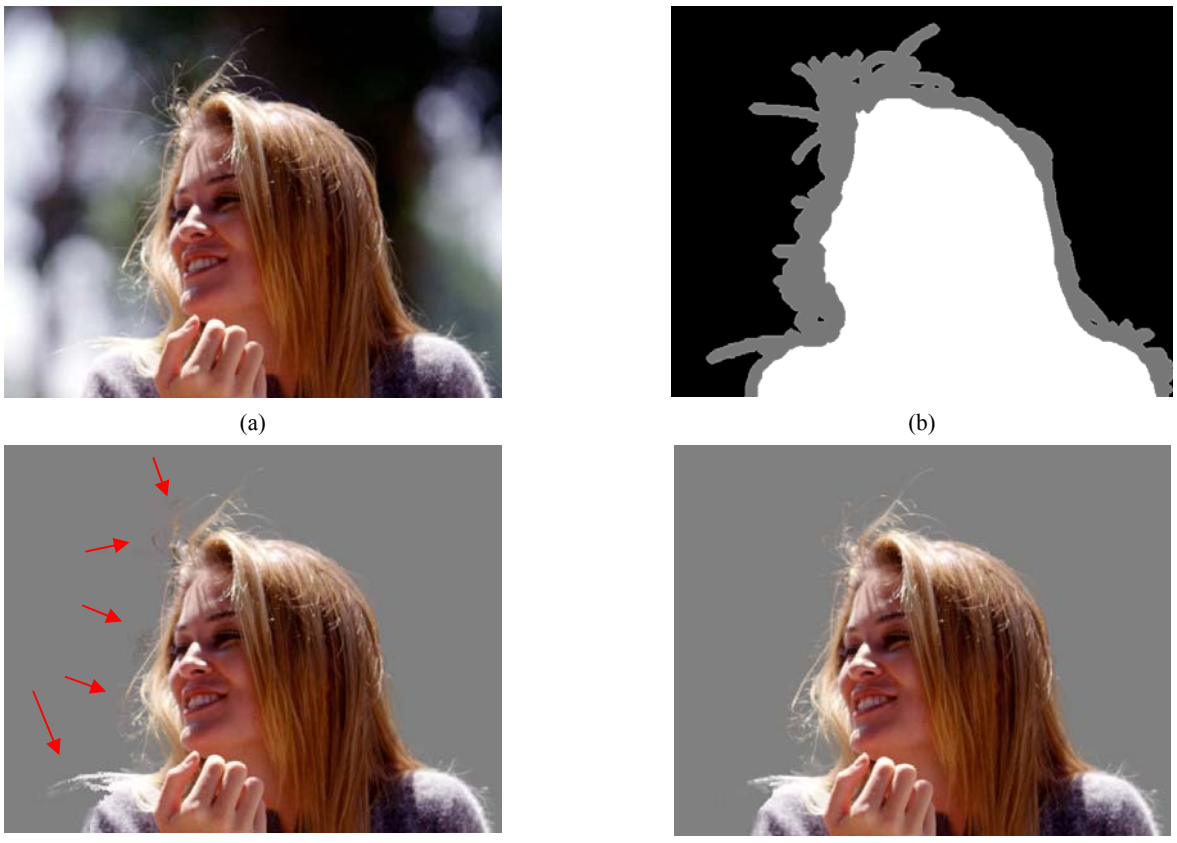
We generate a smaller image using bilinear downsampling. The trimap is downscaled using the resampling filter that marks the pixel of smaller image as foreground/background only if all corresponding pixels of the source trimap are foreground/background, otherwise it is marked as unknown. Bayesian matting parameters such as sigma value used for spatial weight of the sample are also downscaled. Bayesian matting is performed on a smaller image/trimap pair and produces α, *F* and *B* images. These images are required to be upsampled back to the original size. We apply the following procedure to α image and each channel of *F* and *B* images (but we will refer to the channel being processed as α):



(a) (b) (c)

(d) (e) (f)

Figure 1 Example of Bayesian matting with smoothness term. (a) Chipmunk image from Berkeley data set [5]. (b) Our trimap. (c) Alpha obtained by standard Bayesian matting (our implementation). (d) Alpha obtained by Closed Form Solution [4]. (e) Alpha obtained by Bayesian matting with our smoothness term. (f) Composite on a constant-color background using the result of (e).



(a) (b)

(c) (d)

Figure 2 Example of Bayesian matting with smoothness term. (a) Image from [2] website. (b) Trimap from [9]. (c) Composite obtained by standard Bayesian matting. Artifacts and problem areas denoted by red arrows. (d) Composite obtained by our improved Bayesian matting algorithm.

1. Calculate a and b coefficients at each pixel of the downscaled image that give the least squares approximation over a 3x3 window as in equation (3).
2. Resize the coefficient images using bilinear upsampling filter.
3. Produce upscaled α image (of original size) by applying equation (3) to the source image C using a and b coefficients from upscaled coefficient images obtained on step 2.

Apparently we can use scale factors which may be non-power-of-two and even non-integer (however, the upsampling procedure optimized for power-of-two factors works a little faster). We use 3x3 windows around the pixel.

It can be noticed that applying equation (3) to a downscaled image blurs the alpha channel. To prevent this on step 1 we constrain alpha value at the pixel being processed (i.e. at the central pixel of 3x3 window) to equal the right-hand side of (3) exactly.

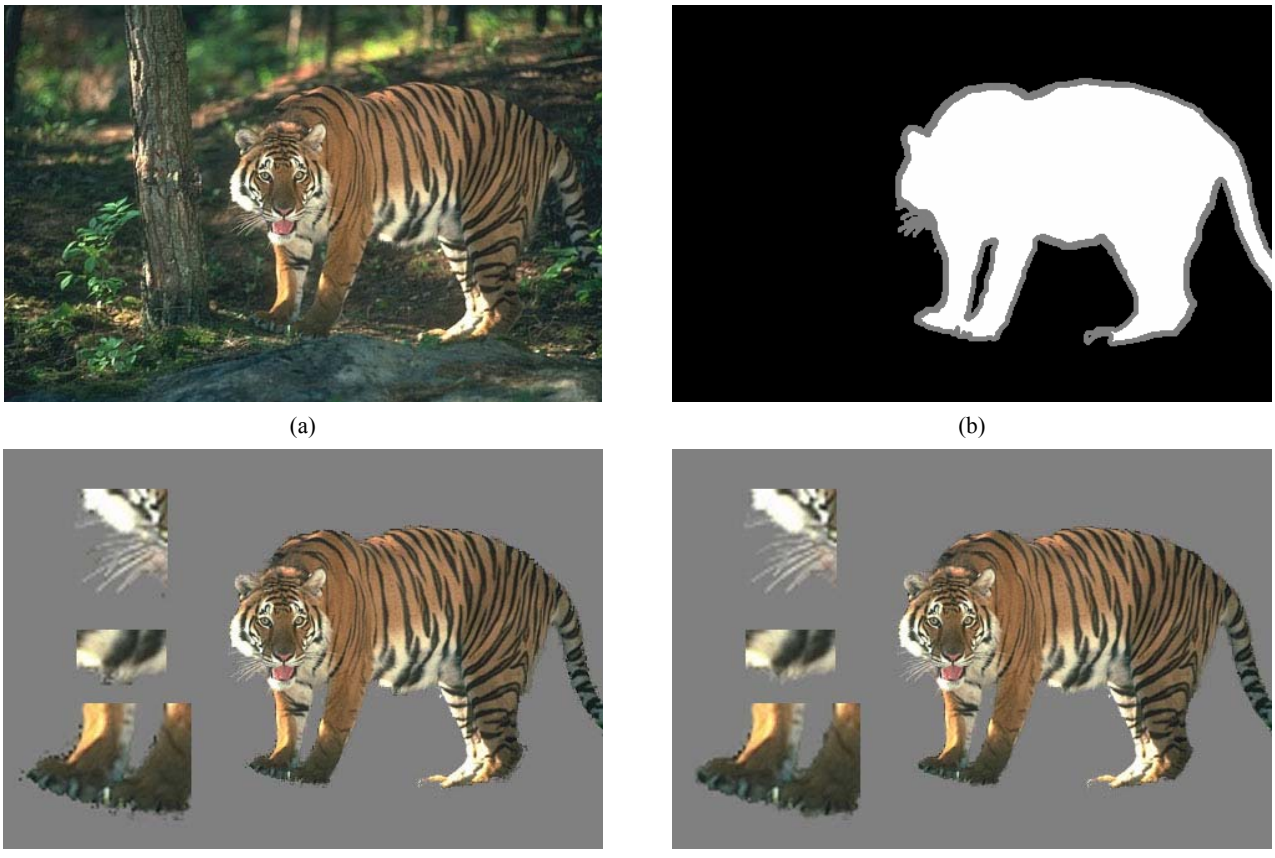
Using Bayesian matting with smoothing on a downsampled image instead of Closed Form Solution, as in [4], increases total speed of the algorithm, and gives better results on some images, like in Figure 1.

Performing Bayesian matting on the image of smaller size gives us a non-linear speed up. For reasonable downscale factors (≤4) the matte quality does not decrease in any noticeable way.

4. RESULTS AND COMPARISONS

Here we provide some results obtained with our improved algorithm compared to the original algorithm (our own implementation is used) both in quality and processing time. We also compare the result to Closed Form Solution matting algorithm [4] using the MATLAB code provided by the authors. Unfortunately, we could not do time comparisons in this case because the MATLAB code is very slow.

In Figure 1 we show how the introduced smoothness constraint helps to matte a rather complicated image with many color similarities between foreground and background. Processing time for both standard and smoothing Bayesian algorithms is 0.8 seconds on AMD Sempron 3100+ (1800 MHz) processor (image size is 299x219 pixels). Note that in case when foreground and background color statistics are hardly distinguishable, object boundary is attracted to the center line of the unknown region (a set of pixels that are equidistant from the foreground and background regions in L_1-distance). This happens when maximization of $L(F)+L(B)+L(C)$ terms fails to achieve high likelihood value and the smoothness term $L(\alpha)$ overpowers the



(a) (b)

(c) (d)

Figure 3 Example of Bayesian matting with smoothness term. (a) Image from Berkeley data set [5]. (b) Our trimap. (c) Composite obtained by standard Bayesian matting (with several close-ups). (d) Composite obtained by our improved Bayesian matting algorithm (with several close-ups).

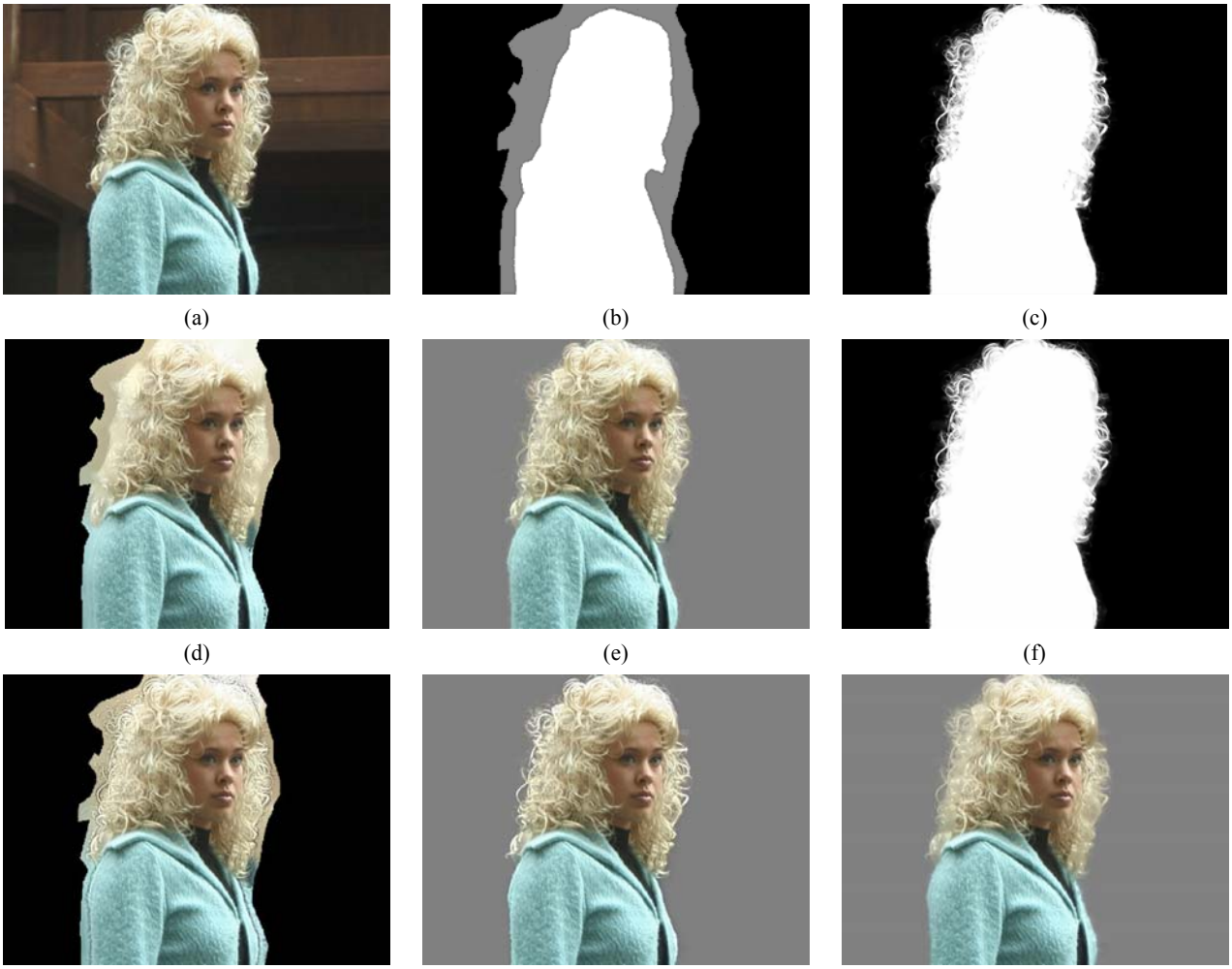


Figure 4 Example of hierarchical Bayesian matting. (a) An image from [3] (also used in [7]). (b) Corresponding trimap. (c), (d) Alpha and foreground images obtained by standard Bayesian matting. (e) The result of (c), (d) composited onto constant-color background. (f), (g) Alpha and foreground images obtained by hierarchical Bayesian matting with downscale factor of 8. (h) The result of (f), (g) composited onto a constant-color background. (i) A composite of (f) using the foreground obtained by simple bilinear upscaling. Shown to emphasize the sharpness of details reconstructed in (h).

other terms. This usually gives a good spatial-based rather than color-statistics based guess of object contour. However, many small details, absent in the unknown region shape but found using color statistics, are reconstructed.

In Figure 2 we show how smoothness term improves foreground color and small hair details. In this image small inaccuracies in the estimated alpha matte lead to incorrect estimation of foreground color. Smoothness term improves alpha matte insignificantly but this is enough to get good color estimation. Areas of similar foreground and background colors (especially on the lower left of the image) are also improved.

The same can be seen in Figure 3. Small black spots and small holes along the edge are effectively handled by our algorithm. The result image can be used for creating a composite without the need for manual clean-up.

In Figure 4 we demonstrate our hierarchical approach. Downscaling factor of 8 was used. Standard Bayesian algorithm took 5.2 seconds to compute on the processor mentioned above (image size is 640x480). The hierarchical approach reduced this time to 0.16 seconds while preserving the quality of the result. In Figure 4 (h) we show a composite obtained using the alpha matte from Figure 4 (e) with bilinearly upscaled foreground. Hair details, though preserved in alpha matte, become blurred with loose foreground color image. This demonstrates the importance of applying the upsampling algorithm to F and B images. We can also approximately compare the processing time with that of Poisson matting [7] assuming that their and our implementation of standard Bayesian matting runs the same time. This gives us a rough estimate of 0.23 seconds processing time for Poisson matting on our processor.

Table 1 shows the time comparison between the standard and the hierarchical Bayesian Matting algorithms on several images:

woman image from Figure 4, car, flower and lighthouse images from Figure 5. For each image we chose maximal downscaling factor which produced insignificant deviation of the result from the standard Bayesian Matting result.

5. FUTURE WORK

5.1 Selecting optimal parameters for smoothness constraint

One of the natural improvements is selecting optimal expression for σ_α used in (8) to produce the best results. In addition, we can evaluate several workflows involving smoothness and hierarchical passes to find the best speed/quality compromise.

Image	Resolution	TS (sec)	D	TH (sec)
Woman image	640x480	5.2	8	0.16
Car image	723x489	3.8	4	0.19
Flower image	600x450	3.9	2	0.43
Lighthouse image	320x480	5.3	2	0.46

Table 1. Processing time comparison of the standard and hierarchical Bayesian matting algorithms. TS – time for the standard algorithm, D – downscaling factor, TH – time for the hierarchical algorithm.

We also consider adding other user-controlled parameters besides trimap which could improve the result. For example, we can add smoothness brush which would allow the user to roughly specify areas where σ_α should be increased. We also want to make recalculation of the result as quick as possible (based on previous refinement step result) to save the user from waiting full image processing time after adding only several strokes to the trimap.

5.2 Trimap generation from user strokes

Algorithms producing good results from rough strokes could be more preferable than those which require precise trimap, if they were fast. At the moment there are no algorithms that can quickly perform full processing from several user strokes. So the following approach can be a good compromise: using a fast algorithm to produce more-or-less precise trimap from several user strokes can precede full processing algorithm. We consider using GrowCut algorithm [8] for this step. The produced trimap can also be made user-editable.

5.3 Video

One of the challenging fields of matting algorithms application is video compositing. In spite of recent achievements in natural image matting, video/film-editing studios continue to use chroma-keying and rotoscoping for foreground object extraction from video sequence.

Used chroma-keying algorithms require the object to be filmed on accurately lit constant color background and produce acceptable yet far from ideal results since they use simple heuristics to compute F and α from C.

For footage without chroma-key background rotoscoping is used, which means that object contours should be accurately traced in each frame by human operator. After the object is precisely traced boundary feathering can produce very good α channel, but F channel is again computed either by using simple heuristics or assumed to equal source image C. This produces halos around objects which are usually removed by contracting alpha matte. Usage of matting algorithms can significantly improve the result and increase working speed even if the trimap has to be hand-drawn for each frame.

6. CONCLUSION

Automatic user-guided matting is important for many image- and video-compositing tasks. In this paper we proposed two improvements to Bayesian matting algorithm. We compared the results of improved algorithm with the original one demonstrating that our algorithm produces smooth alpha image and handles color ambiguities by propagating alpha values from nearby pixels.

Our smoothness term significantly improves the result of Bayesian matting without increasing the computation time while the hierarchical approach effectively decreases this time still producing the acceptable result. This allows us to create user-friendly matting environment for use in image and (in future) video processing.

7. REFERENCES

[1] Apostoloff, N. and Fitzgibbon, A. *Bayesian video matting using learnt image priors*, Proc. of IEEE CVPR, pp. 407–414, 2004.

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

[3] Chuang, Y., Agarwala, A., Curless, B., Salesin, D. and Szeliski, R. *Video matting of complex scenes*, ACM Trans. Graph., 21(3):243–248, 2002.

[4] Levin, A., Lischinski, D., Weiss, Y. *A Closed Form Solution to Natural Image Matting*, Proc. of IEEE CVPR, pp. 61–68, 2006

[5] Martin, D., Fowlkes, C., Tal, D., Malik, J. *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, Proc. of ICCV, vol. 2, pp. 416–423, 2001.

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

[7] Sun, J., Jia, J., Tang, C.-K., and Shum, H.-Y. *Poisson matting*, ACM Trans. Graph., 23(3):315–321, 2004.

[8] Vezhnevets, V. and Konouchine, V. *GrowCut: Interactive multi-label N-D image segmentation by cellular automata*, Proc. of Graphicon, pp.150–156, 2005.

[9] Wang, J. and Cohen, M. F. *An iterative optimization approach for unified image segmentation and matting*, Proc. of ICCV, vol.2, pp. 936–943, 2005.

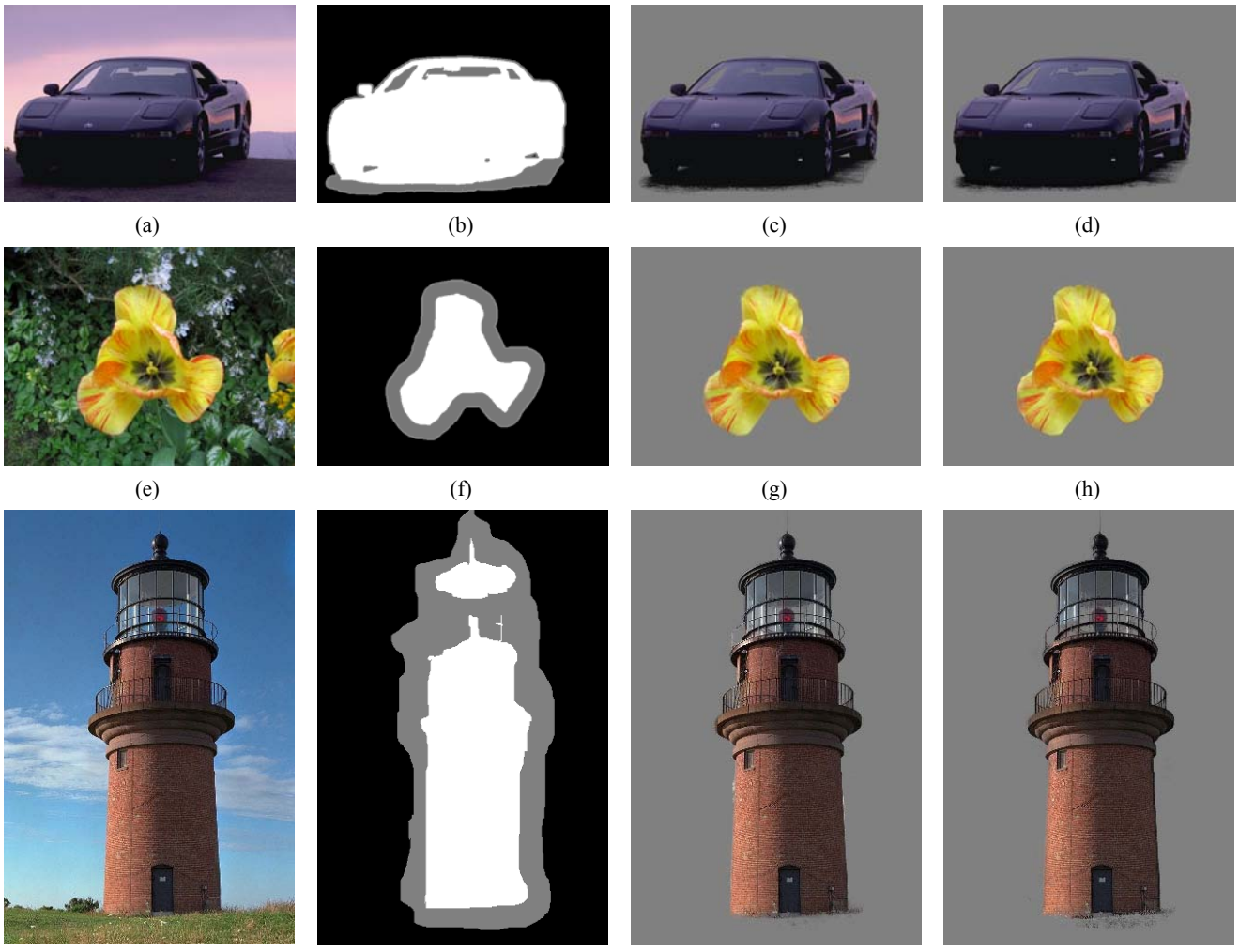


Figure 5 Examples of hierarchical Bayesian Matting. (a), (e), (i) Source Images. Car image is taken from the website of [2], flower image is taken from the dataset [5], and the lighthouse image is taken from [2]. (b), (f), (j) Trimaps. (c), (g), (k) Standard Bayesian Matting results. (d), (h), (l) Hierarchical Bayesian Matting results (composites). See Table 1 for processing time comparison.

About the authors

Mikhail Sindeyev is a 3rd year student at Graphics and Media Laboratory of Moscow State Lomonosov University, Department of Computational Mathematics and Cybernetics. His research interests include image and video processing, 3D reconstruction, computer vision and adjacent fields. His email address is msindeev@graphics.cs.msu.ru.

Vadim Konushin is a 5th year student at Graphics and Media Laboratory of Moscow State Lomonosov University, Department of Computational Mathematics and Cybernetics. His research interests include image and video processing, pattern recognition, computer vision and adjacent fields. His email address is vadim@graphics.cs.msu.ru.

Vladimir Vezhnevets is a vice head of Graphics and Media Laboratory of Moscow State Lomonosov University, Department of Computational Mathematics and Cybernetics. He graduated with honors from Faculty of Computational Mathematics and Cybernetics of Moscow State University in 1999. He received his PhD in 2002 in Moscow State University also. His research interests include image processing, pattern recognition, computer vision and adjacent fields. His email address is vvp@graphics.cs.msu.ru.