Trimap Segmentation for Fast and User-Friendly Alpha Matting

Olivier Juan and Renaud Keriven

CERTIS, ENPC,
77455 Marne-la-Vallée, France
{juan,keriven}@certis.enpc.fr

Abstract. Given an image, digital matting consists in extracting a foreground element from the background. Standard methods are initialized with a *trimap*, a partition of the image into three regions: a definite foreground, a definite background, and a *blended region* where pixels are considered as a mixture of foreground and background colors. Recovering these colors and the proportion of mixture between both is an under-constrained inverse problem, sensitive to its initialization: one has to specify an accurate trimap, leaving undetermined as few pixels as possible.
First, we propose a new segmentation scheme to extract an accurate trimap from just a coarse indication of some background and/or foreground pixels. Standard statistical models are used for the foreground and the background, while a specific one is designed for the blended region. The segmentation of the three regions is conducted simultaneously by an iterative Graph Cut based optimization scheme. This user-friendly trimap is similar to carefully hand specified ones.
As a second step, we take advantage of our blended region model to design an improved matting method coherent. Based on global statistics rather than on local ones, our method is much faster than standard Bayesian matting, without quality loss, and also usable with manual trimaps.

1 Introduction

The commonly used model of digital or alpha matting is the following. An image I is considered as a mixture between a foreground I_F and a background I_B, mixture quantified by an *alpha mask* $\alpha \in [0, 1]$. For each pixel x, this writes

$$I(x) = \alpha(x)I_F(x) + (1 - \alpha(x))I_B(x) \tag{1}$$

Such a blending has multiple reasons: transparent objects, aliasing, blur or motion blur. The problem is to recover I_F, I_B and α from I.

This inverse problem is under-constrained and can not be solved without priors. Historically, a solution was proposed in the case of a known constant background, e.g. a blue screen [1]. Recently, inspired by computer vision techniques, methods based on a model of the foreground and of the background were

proposed that greatly improve the matte quality, even without a blue screen. Since the pioneering work of Ruzon and Tomasi [2], several methods have been proposed [3–6].

As a prerequisite of any method, the user has to specify a so-called *trimap*, partitioning the image into three regions: a set Ω_F of definitely foreground pixels (where α will always be 1), a set Ω_B of definitely background pixels ($\alpha = 0$), and a blended region Ω_M where α, I_F and I_B are unknown. Ω_M has to be an intermediate region, separating Ω_F from Ω_B. Matting methods suffer from sensitivity to this initial condition and one has to specify it accurately, leaving undetermined as few pixels as possible. Moreover, when too small Ω_F and Ω_B are given, the matting process generally does not work at all.

Digital matting was primitively developed for movie production. For a specialist, carefully specifying a trimap is a long but feasible process (actually faster and easier than alpha masking). Today, extracting a subject from a picture for editing purpose becomes a standard in a non professional context. Speed becomes also an issue, particularly with the ever increasing resolution of digital cameras.

This paper addresses both user-friendly trimap design and speed. First, we propose a trimap segmentation scheme from just a small subset of the background and/or foreground, that can be for instance specified by the user with a brush-like tool. Standard Gaussian Mixture Models (GMMs) are used for foreground and background modeling, while a specific statistical model is proposed for the blended region. To save the user from specifying some obscure number of components, the GMMs parameters are determined with a coupled Expectation Maximization (EM) / Minimum Description Length (MDL) scheme. For the sake of speed, the segmentation of the three regions is conducted simultaneously by an iterative Graph Cut based optimization. The resulting trimap proves to be similar to carefully hand specified ones.

As a second step, we take advantage of our blended region model to design an improved matting method. Based on global statistics rather than on local ones, our method is much faster than the original Bayesian matting, although without quality loss. It can also be used with manually designed trimaps.

2 Related work

The original work of Ruzon and Tomasi [2] laid the foundations of most of the actual methods, for which the key point consist in modeling the background and the foreground with some statistical model. In their famous *Bayesian Matting*, Chuang et al. [3] improved both the statistical model and the way to use it to recover the alpha mask and the original background and foreground colors. Since then, Rother et al. proposed *GrabCut* [7, 8], a method inspired by Boykov and Jolly work [9], where the image is actually segmented into two regions using an iterated Graph Cuts [10] scheme. A smooth alpha mask is then modeled as a ramp of variable width to be estimated. As a result, it is unadapted to non smooth objects like hairs or trees. The GrabCut method does not need

a trimap. It can be seen more as a two regions segmentation with a smooth transition between the two regions, than as a strictly speaking digital matting method. However, as another member of iterated Graph Cuts methods [9], our trimap segmentation has similarities with the segmentation step of GrabCut.

In their *Poisson Matting*, Sun et al. propose another prior on α, based on its gradient and Poisson equations, already used in image editing [5]. Their method supply different modes, refinements and filters, manually invoked by the user. Again, priors on α or its gradient can be questionable as the blending might have different origins and the blended objects different scales with respect to pixels size. Moreover, manual decisions might limit the usability of this technique for non specialists. In conclusion, Bayesian matting can be considered as the less *ad-hoc* method so far. Its weak points are the need of an accurate trimap (a problem common with other matting techniques) and its slowness due to many local statistics estimations. Figure 1 demonstrates how a coarse trimap affects digital matting.

To our knowledge, the only works addressing trimap design are video oriented. Following the original work by Mitsunaga et al. [11], one can specify trimaps for some *key frames* and interpolate them in the intermediate frames. In their recent work, Xiao and Shah [12] proposed an occlusion based trimap extraction couple with motion layer segmentation. However, it is unusable not only for still pictures but also in real film production where motion is often fast and/or heavily blurred.

This paper is organized as follows. First, section 3 exposes the background-foreground model and our parameters estimation method. Then, section 4 introduces the trimap segmentation, details its implementation, and compares it with manual segmentations. Finally, section 5 proposes an improved fast, global and accurate matting method, and shows results.

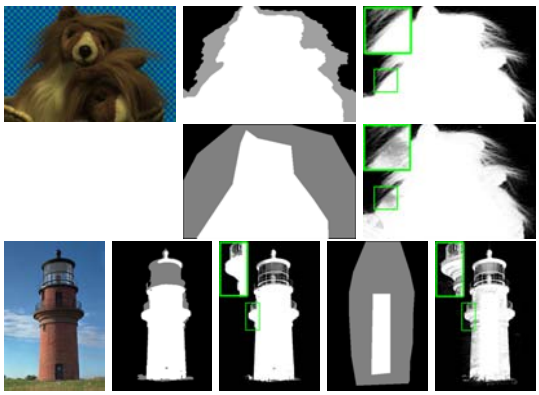


Fig. 1. Sensitivity of Bayesian matting [3] to the trimap. First row: the original image, an accurate trimap and its corresponding alpha. Second row: a coarse trimap and its alpha. Third row: same as rows 1 and 2 for another image.

3 Unsupervised Two-Regions Segmentation

As a first step toward our trimap segmentation, we first focus on segmenting an image into two regions, each of them having its own characteristics, a-priori unknown. This often called *Unsupervised Segmentation* has recently received a lot of attention from the Computer Vision community. Many approaches have been proposed, among which some Level Set [13, 14] based methods (e.g. [15, 16]). More recently, using the Graph Cuts framework, Boykov and Jolly initiated an iterated method [9], further developed by Rother et al. in [7] and in their GrabCut scheme [8].

In this section, we briefly describe the segmentation part of the GrabCut scheme. Already known to the GrabCut aware reader, the content of this section introduces definitions and notations. The slight difference with the original work is that we plead for a more sophisticated parameter estimation method, EM + MDL based, mathematically more justified, more user-friendly, and yielding somehow better results.

Let I be a color image defined over a domain Ω. For all $x \in \Omega$, $I(x)$ is a pixel defined in a color space (e.g. RGB or CieLab). Let Ω_U be a part of Ω specified by the user. Our goal here is to segment Ω into two *"coherent"* regions that we will abusively still call the background and the foreground, respectively still denoted by Ω_B Ω_F, such that $\Omega_U \subset \Omega_B$.

3.1 Region Modeling

Following previous work and using a statistical approach, each region Ω_X ($X = F$ or B) is modeled by a Probability Density Function (PDF) approximated by a Gaussian Mixture Model (GMM):

$$p_X(I) = \sum_{i=1}^{N_X} \pi_i^X G_{\mu_i^X, \Sigma_i^X}(I) \text{ with } \sum_{i=1}^{N_X} \pi_i^X = 1 \text{ and } \pi_i^X \in [0,1]$$

Each component is represented by a Gaussian of mean μ_i^X and covariance Σ_i^X: $G_{\mu,\Sigma}(I) = \frac{|\Sigma|^{1/2}}{(2\pi)^{3/2}} e^{-(I-\mu)^T \Sigma^{-1}(I-\mu)/2}$ and π_i^X is the prior of the i^{th} component with respect to all components, i.e. its proportion in the mixture.

Estimating the parameters $\Theta_X = \{N_X, (\pi_i^X, \mu_i^X, \Sigma_i^X)_{i=1..N_X}\}$ is a widely studied problem. For a given N_X, one can use the K-Means algorithm (see [17]), a fast but approximate method. This method is widely sensitive to its initialization. Moreover it does not provide a likelihood maximum, which is not appropriate for a segmentation based on likelihood maximization. Indeed the K-Means just solves:

$$(\mu_i^X, \Sigma_i^X) = \arg\min_{(\mu_i, \Sigma_i)} \sum_{x \in \Omega_X} \left\| I(x) - \mu_{k(I(x))} \right\|^2_{\Sigma_{k(I(x))}}$$

with $k(I) = \arg\min_k \|I - \mu_k\|^2_{\Sigma_k}$, $\|I - \mu\|_\Sigma$ being the Mahalanobis distance between I and μ with respect to Σ. Note that [7] suggests [18] as a variant and that [3] uses the method in [19]. We prefer the EM algorithm [20, 21]. It is much

more robust with respect to the initial parameters and provides a likelihood maximum, solving:

$$(\pi_i^X, \mu_i^X, \Sigma_i^X) = \underset{(\pi_i, \mu_i, \Sigma_i)}{\arg\min} \sum_{x \in \Omega_X} p(I(x))$$

Finally, we combine the EM algorithm with a MDL [22] estimation of N^X, saving the user from manually adjusting the number of Gaussian components. Note that we have also tested more recent algorithms like Split and Merge EM [23], without any significant improvement.

3.2 Energy Design

Let γ be the partition function of Ω into Ω_F and Ω_B: $\gamma(x) = F$ if $x \in \Omega_F$, $\gamma(x) = B$ otherwise. Under the hypothesis that regions are independent with respect to their color distribution, it is natural to use the posterior probability of the pixels as a segmentation criterion, thus stating the problem as minimizing an energy:

$$E_{data}(\gamma) = \int_{\Omega} -log\ p_{\gamma(x)}(I(x))\ dx \tag{2}$$

An extra control term should be added to constrain the smoothness of the solution which is often addressed as a local smoothness constrain: neighbor pixels should belong to the same region. This yields an additional smoothness energy:

$$E_{smooth}(\gamma) = \int_{\Omega} \left(\int_{y \in \mathcal{N}(x)} \mathcal{V}(x, y)\ dy \right) dx \tag{3}$$

where $\mathcal{N}(x)$ is a local neighborhood of x and $\mathcal{V}(x, y) = \mathcal{V}^0(x, y)$ if $\gamma(x) \neq \gamma(y)$ with $\mathcal{V}(x, y) = 0$ otherwise.

Under the assumption that the frontier between the two regions corresponds to high image gradients, a frequent choice is $\mathcal{V}^0(x, y) = \kappa \exp(-\frac{\|I(x) - I(y)\|^2}{2\sigma^2})$ where κ is some positive constant controlling the degree of smoothness and σ is set as in [9]. The global energy to minimize ends to:

$$E(\gamma) = E_{data}(\gamma) + E_{smooth}(\gamma) \tag{4}$$

3.3 Implementation and Comparison

When minimizing E either with a Level Sets Method approach [24, 25] or with a Graph Cuts one [9], one should be aware of the dependency of the PDFs upon γ. This leads to an iterated process that is usual in the Level Sets gradient descent, but is not in the case of Graph Cuts. As we do not need sub-pixel accuracy, we opt for a Graph Cuts approach, mainly for speed reasons. Using EM instead of K-means is theoretically important: the algorithm consists in alternately updating the PDFs according to the segmentation and in segmenting according to the PDFs. At each step, the energy decreases:

– Updating the PDF using EM ensures that E_{data} decreases, E_{smooth} being fixed.
– The Graph Cuts step ensures that E decreases.

Let us just recall useful notations [26]. We consider a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ that is a set of nodes $\mathcal{V}$ and directed edges $\mathcal{E}$ connecting them. Two special terminal nodes are present: the source s and the sink t. Each edge (p, q) connecting a node p to a node q is assigned a weight $t_{p,q}$. Edges are broken in two groups: n-links and t-links. A n-link is an edge connecting two non-terminal nodes. A t-link connects a non-terminal node to a terminal node. A cut C is a partitioning of the nodes of the graph into two disjoint subsets S et T such that the source $s \in S$ and the sink $t \in T$. Its cost is the sum of the weights of all edges (p, q) such that $p \in S$ and $q \in T$. A minimum cut is a cut with minimal cost and one minimum cut can be determined in polynomial time with a max-flow extraction algorithm.

Here, each pixel of the image is associated to a node and to edges for each of its neighbors. Each node is also connected to the sink and the source. The weights on the t-links deal with data constrain and those on the n-links account for smoothness. For a pixel x associated to node p, let D_X be the negative logarithm of the probability density function associated to region Ω_X: $D_X(p) = -\log\ p_X\left(I\left(x\right)\right)$. The Graph is built according to table 3.3. After the cut, the nodes that are still connected to the source, are assigned to Ω_F, the others to Ω_B. Figure 2 shows the result of this segmentation process on some test image using both the method in [7] and a method using an EM/MDL estimation. Note that some details misclassified by the original method are correctly handled by the EM/MDL approach. Yet, these improvements are not decisive. More important is the fact that the MDL based estimation of N_X proves to be reliable and masks one annoying parameter from the user.

Table 1. Weights associated to node p

link	weight	for
$t_{s,p}$	0	$p \in \Omega_U$
$t_{p,t}$	∞	$p \in \Omega_U$
$t_{s,p}$	$D_B(p)$	$p \notin \Omega_U$
$t_{p,t}$	$D_F(p)$	$p \notin \Omega_U$
$t_{p,q}$	$\mathcal{V}(p,q)$	$q \in \mathcal{N}(p)$

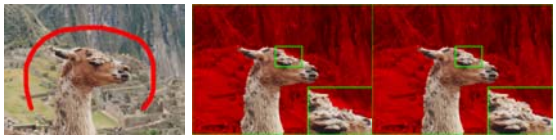


Fig. 2. Importance of then EM estimation and reliability of the MDL criterion. Original image with background specification in red (left) and the corresponding segmentations using the method in [7] with fixed $N_F = N_B = 5$ (middle) and our EM/MDL approach (right)

4 Trimap segmentation

With these notations in hand, let us go back to our main goal of segmenting a trimap. Assuming that the blended region will also be modeled by a PDF $p_M(I)$,

still to be modeled, the data driven part of the energy is unchanged and given by equation (2) with a new partition function that reflects the 3 regions.

However, keeping the same smoothing term is a nonsense. A high image gradient does not indicate a frontier between two regions anymore. Instead, we use the length of the frontiers separating the regions as a smoothing energy. Classical in the Level Set framework and inducing mean curvature motion, this can also be handled rigorously in an Markov Random Field framework (see [27]). Here, we will restrict ourselves to an approximation, just replacing the previous $\mathcal{V}^0(x,y)$ to a decreasing function of the distance between x and y (e.g. $\mathcal{V}^0(x,y) = \kappa/(1+d(x,y))$).

Keeping the same GMM models for p_B and p_F, we still have to design a model for p_M in order to define the energy to minimize.

4.1 A PDF for Ω_M

A straightforward solution would be to take a third GMM for p_M and to estimate its parameters $\Theta_M = \{N_M, (\pi_i^M, \mu_i^M, \Sigma_i^M)_{i=1..N_M}\}$ via the same EM/MDL scheme as for p_B and p_F. It would be a mistake. Indeed, p_M is not independent from p_B and p_F: in Ω_M, I, I_F and I_B are related by equation (1). Despite this, one could willingly ignore this dependency and try to segment $(\Omega_B, \Omega_M, \Omega_F)$ as three regions with each one its own independent GMM. Unfortunately, it is not obvious that the resulting iterated minimizing process will converge to the desired regions without a very accurate initialization, specifying pixels of the three regions. On the contrary, making p_M depend on p_B and p_F will turn out to be sufficient to keep a coarse initialization Ω_U.

Let us examine equation (1). We will assume for simplicity that both I_F and I_B come from one single Gaussian of the respective GMMs p_F and p_B. In their Bayesian estimation of layers from multiple images, Wexler et al. [4] assume that α follows a Beta law. Yet, they choose the parameters of the Beta distribution by estimating them on some reference image. Thus, although Kitamoto gives in [28, 29] a Gaussian approximation of a mixture of two Gaussian distributions when the mixture coefficient follows a Beta Law, we prefer to simply consider that α follows a uniform law. In that case, if I_F comes from $G_{\mu_i^F, \Sigma_i^F}$ and I_B from $G_{\mu_j^B, \Sigma_j^B}$, the distribution of I can be approximated by another Gaussian $G_{\mu_{ij}^M, \Sigma_{ij}^M}$, given also by Kitamoto in [28, 29] as:

$$\mu_{ij}^M = \frac{\mu_i^F + \mu_j^B}{2} \quad \text{and} \quad \Sigma_{ij}^M = \frac{1}{3}\left(\Sigma_i^F + \Sigma_j^B\right) + \frac{1}{12}\left(\mu_j^B - \mu_i^F\right)\left(\mu_j^B - \mu_i^F\right)^T \quad (5)$$

Note that this is, again, an approximation and that more sophisticated models could be investigated. Actually, our simple assumption of a uniform α, and of a Gaussian approximation for I, will turn out to give good results. With this choice, it is natural to model p_M with another GMM, whose $N_M = N_F N_B$ components are now fixed and dependent on p_F and p_B:

$$p_M(I) = \sum_{i=1}^{N_F} \sum_{j=1}^{N_B} \pi_{ij}^M G_{\mu_{ij}^M, \Sigma_{ij}^M}(I) \quad (6)$$

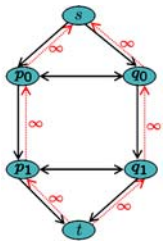
where $\sum_{ij} \pi_{ij}^M = 1$ and where the $(\mu_{ij}^M, \Sigma_{ij}^M)$ are given by equation (5). The only free parameters are the (π_{ij}^M), and we estimate them with an EM algorithm on Ω_M.

4.2 Graph Cuts Implementation

As we assume that the blended region Ω_M separates Ω_F from Ω_B, we can use the Graph Cuts implementation described in [30] which is simpler than the usual α-expansion based algorithm and provides a global minimum. Each pixel x is represented by two nodes p_0 and p_1. The graph is built according to figure 3. After the cut, each node is labeled according to the following rule:

- If the link between $\{s, p_0\}$ is cut, the node is assigned to the foreground.
- If the link between $\{p_0, p_1\}$ is cut, the node is assigned to the blended region.
- If the link between $\{p_1, t\}$ is cut, the node is assigned to the background.

Here we use the method described by Kolmogorov and Zabih in [10] to force the algorithm to cut one and only one of the three links $\{s, p_0\}$, $\{p_0, p_1\}$ and $\{p_1, t\}$. It consists in adding infinite reverse edges on the graph (see red links on figure 3). Like in the two regions case, we use an iterative scheme. However we found that using the two regions segmentation as a first step gives good initial estimates for p_B and p_F and speeds up the convergence.



link	weight	for
t_{s,p_0}	∞	$p \in \Omega_U$
t_{p_0,p_1}	∞	$p \in \Omega_U$
$t_{p_1,t}$	0	$p \in \Omega_U$
t_{s,p_0}	$D_F(p)$	$p \notin \Omega_U$
t_{p_0,p_1}	$D_M(p)$	$p \notin \Omega_U$
$t_{p_1,t}$	$D_B(p)$	$p \notin \Omega_U$
t_{p_i,q_i}	$\mathcal{V}(p,q)$	$q \in \mathcal{N}(p)$

Fig. 3. Trimap segmentation. Graph representation for two nodes p and q and associated weights.

4.3 Results

Figure 4 shows the trimap obtained for the reference image in [3] from just a coarse indication of the background. It is similar to the hand designed one used in the original work. For comparison purposes, we show also the trimap obtained when naively modeling the blended region with an independent GMM, even when starting from a more accurate initialization. Figure 6**a** and 6**b** in next section show many other automatic trimaps. Table 4.3 gives the running times of the trimap extraction (and of the first step of two regions segmentation) for some

of our test images, on a standard 2.4GHz PC without any specific optimization. These are the times for a complete convergence and the process might be stopped before. A multi-scale approach would also improve speed significantly. Anyway, these are to be compared with the times needed for a cautious manual segmentation, depending on the user's ability and/or equipment. Note that the more complex a manual segmentation would be, the more the automatic segmentation seems to require time to converge (see images on figures 6**a** and 6**b**).

Table 2. Running times for trimap segmentation on some test images.

Image	First step	Total time
Teddy Bear	36s	94s
Butterfly	14s	28s
Light	48s	133s

Fig. 4. Automatic trimaps. First column: the original image and the hand designed trimap used in [3]. Second column: background/foreground initialization (in red/white) and the obtained trimap, naively considering p_M as an independent GMM. Third column: background only initialization (in red) and the trimap obtained with our method

5 An improved matting method

In this section, we propose a new matting algorithm taking advantage of our blended region model. Based on global statistics rather than on local ones, it is faster than the original Bayesian matting, although without quality loss.

Chuang et al.'s Bayesian matting algorithm is based on estimating local statistics of the foreground and of the background. For each pixel in the blended region, a neighborhood is considered, where the foreground and the background are respectively modeled by two Gaussian distributions $G_{\mu^F_{loc},\Sigma^F_{loc}}$ and $G_{\mu^B_{loc},\Sigma^B_{loc}}$.

Estimating a local mean and covariance for each pixel is inefficient from a computational point of view. Moreover, limiting the distribution of the neighborhood of a pixel to one single Gaussian may sometimes be a too coarse approximation. We propose to take advantage of our global GMM analysis of the foreground and the background carried out during the segmentation process. Keeping the assumption that I_F and I_B come from one Gaussian each, we choose these two Gaussian distributions respectively among the components of p_F and p_B. We use $\pi^M_{ij} G_{\mu^M_{ij},\Sigma^M_{ij}}(I)$ to measure which Gaussian distributions most probably explain I. Thus, we simply:

1. choose the pair (i_0, j_0) that maximizes $\pi^M_{ij} G_{\mu^M_{ij},\Sigma^M_{ij}}(I)$
2. use Chuang et al.'s solving scheme with $G_{\mu^F_{i_0},\Sigma^F_{i_0}}$ and $G_{\mu^B_{j_0},\Sigma^B_{j_0}}$ as priors for I_F and I_B instead of the local estimations $G_{\mu^F_{loc},\Sigma^F_{loc}}$ and $G_{\mu^B_{loc},\Sigma^B_{loc}}$.

The resulting process turns out to be faster than the original method and the results are similar. Note that it is essential that the GMMs have enough components to explain all the colors/textures locally present in the image. Our EM/MDL estimation ensures this.

6 Results

In their original work on Bayesian matting, Chuang et al. proposed a real bench image, supplying a ground truth for the alpha mask (see [3]). Figure 5 shows this *true* mask, compared those obtained with their and our matting algorithm, using our automatic trimap in both cases. The result are similar and the relative errors, in L^2 norm in region Ω_M, respectively gives 1.5% and 1.4% errors.

As expected, the main advantage of our method is its computational efficiency. Table 6 gives the running times of both methods for some of our test images, under the same conditions as previously (standard 2.4GHz PC, no specific optimization). We observed a speedup of about 100. Please note that this would also stand when starting from a manual trimap. The only overhead for our matting would be to estimate the global statistics from this trimap before running, which is actually negligible with respect to the matting process.

Finally, figures 6**a** and 6**b** show the complete process of our method on several test images: the original images (figure 6**a** only), the user's initialization, the segmented trimap, the mask obtained with Bayesian matting (fig. 6**a** only), the one obtained with our method, and a recompositing from our (α, I_F, I_B) estimation. It demonstrates how a simple initialization without any additional parameter (e.g. number of Gaussian distributions) is enough to get accurate trimaps, and how our fast matting method gives results similar to the ones obtained with the original but slower Bayesian matting.

Table 3. Running times for the standard Bayesian matting and for our method on some test images.

Image	Bayesian matting	Our matting
Teddy Bear	47s	0.36s
Butterfly	2.7s	0.027s
Light	37s	0.27s

Fig. 5. From top to bottom, left to right: three alpha masks (ground truth, Bayesian matting using our trimap, our method using our trimap), a recompositing using our mask and foreground estimations

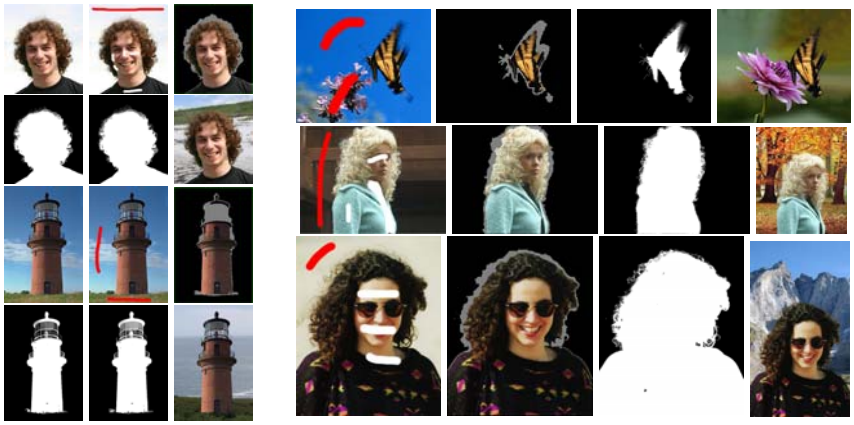


Fig. 6. a. For each image, in reading order: original image, user's initialization, automatic trimap, Bayesian matting, our matting, recompositing. **b.** On each line, from left to right: user's initialization, automatic trimap, our matting, recompositing.

7 Conclusion

In this paper, we propose a segmentation method aimed at extracting an accurate trimap for the digital matting problem. A statistical model is specifically designed for the blended region and an iterative Graph Cut based optimization scheme allows this trimap segmentation from just a coarse specification of some background and/or foreground pixels. This trimap is similar with those obtained by a meticulous hand drawing. Finally, taking advantage of this blended region model, we describe a improved digital matting method, based on global statistics, much faster than the original Bayesian matting, although without quality loss. This method is also usable starting from a manual trimap.

References

1. Smith, A.R., Blinn, J.F.: Blue screen matting. In: 23rd annual conference on Computer graphics and interactive techniques, ACM Press (1996) 259–268
2. Ruzon, M., Tomasi, C.: Alpha estimation in natural images. In: CVPR. (2000)
3. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: CVPR. Volume 2. (2001) 264–271
4. Wexler, Y., Fitzgibbon, A., Zisserman, A.: Bayesian estimation of layers from multiple images. In: ECCV. (2002)
5. Perez, P., Gangnet, M., Blake, A.: Poisson image editing. In: SIGGRAPH. (2003)
6. Sun, J., Jia, J., Tang, C., shum, H.: Poisson matting. In: SIGGRAPH. (2004)
7. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive gmmrf model. In: ECCV. (2004)
8. Rother, C., Kolmogorov, V., A., B.: Grabcut - interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004)
9. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV. (2001) 105–112

10. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. on Pattern Analysis and Machine Intelligence (2004) 65–81
11. Mitsunaga, T., Yokoyama, T., Totsuka, T.: Autokey: Human assisted key extraction. In: SIGGRAPH. (1995)
12. Xiao, J., Shah, M.: Accurate motion layer segmentation and matting. In: CVPR. (2005)
13. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on the Hamilton–Jacobi formulation. Journal of Computational Physics **79** (1988) 12–49
14. Osher, S., Paragios, N., eds.: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer Verlag (2003)
15. Brox, T., Rousson, M., Deriche, R., Weickert, J.: Unsupervised segmentation incorporating colour, texture, and motion. In: 10th International Computer Analysis of Images and Patterns. LNCS 2756, Springer Verlag (2003) 353–360
16. Kadir, T., Brady, M.: Unsupervised non-parametric region segmentation using level sets. In: Proceedings of ICCV 2003. (2003)
17. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In Cam, L.M.L., Neyman, J., eds.: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1. (1967) 281–297
18. X. Descombes, M. Sigelle, F.P.: Estimating gaussian markov random field parameters in a nonstationary framework: Application to remote sensing imaging. IEEE Trans. on Image Processing **8** (1999) 490–503
19. Orchard, M.T., Bouman, C.A.: Color Quantization of Images. IEEE Trans. on Signal Processing **39** (1991) 2677–2690
20. McLachlan, G., Krishnan, T.: The EM algorithm and extensions. Wiley, New York (1997)
21. McLachlan, G., David, P.: Finite Mixture Models. Wiley, New York (2000)
22. Figueiredo, M., Leitao, J.M.N., Jain, A.K.: On fitting mixture models. In: Energy Minimization Methods in Computer Vision and Pattern Recognition. (1999) 54–69
23. Ueda, N., Nakano, R., Ghahramani, Z., Hinton, G.E.: Smem algorithm for mixture models. Neural Computation **12** (2000) 2109–2128
24. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based space. In: International Conference on Computer Vision and Pattern Recognition. Volume 2., Madison, Wisconsin, USA (2003) 699–704
25. Juan, O., Keriven, R., Postelnicu, G.: Stochastic Motion and the Level Set Method in Computer Vision: Stochastics Active Contours. International Journal of Computer Vision (in press)
26. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence **23** (2001) 1222–1239
27. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: ICCV. (2003)
28. Kitamoto, A.: The moments of the mixel distribution and its application to statistical image classification. In Amin, A., Ferri, F., Inesta, J., Pudil, P., eds.: Advances in Pattern Recognition (SPR'00), LNCS 1876. (2000) 521–531
29. Kitamoto, A., Takagi, M.: Area proportion distribution – relationship with the internal structure of mixels and its application to image classification. Systems and Computers in Japan **31** (2000) 57–76
30. Ishikawa, H.: Exact optimization for markov random fields with convex priors. IEEE Trans. on Pattern Analysis and Machine Intelligence **25** (2003) 1333–1336