DISTANCECUT: INTERACTIVE SEGMENTATION AND MATTING OF IMAGES AND VIDEOS

Xue Bai *and* **Guillermo Sapiro**

Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455
baixx015@umn.edu , guille@umn.edu

ABSTRACT

An interactive algorithm for soft segmentation and matting of natural images and videos is presented in this paper. The technique follows and extends [10], where the user first roughly scribbles/labels different regions of interest, and from them the whole data is automatically segmented. The segmentation and alpha matte are obtained from the fast, linear complexity, computation of weighted distances to the user-provided scribbles. These weighted distances assign probabilities to each labeled class for every pixel. The weights are derived from models of the image regions obtained from the user provided scribbles via kernel density estimation. The matting results follow from combining this density and the computed weighted distances. We present the underlying framework and examples showing the capability of the algorithm to segment and compute alpha mattes, in interactive real time, for difficult natural data.

Index Terms**—** Distance functions, Segmentation, Matting, Natural, Interactive

1. INTRODUCTION

Interactive image and video segmentation, and matting, where the user starts the automatic algorithm by providing rough scribbles labeling the regions of interests, has received a lot of attention in recent years, see for example [1, 2, 3, 4, 5, 6, 8, 9, 11, 13, 14] and references therein, and [10] for a discussion on these works and the key attributes of distance-based techniques as the one pursued in this paper.

In order to address the challenges of real-time interactive image segmentation, the authors of [10] proposed to exploit the colorization work in [17], where the goal is to add color (or other special effects) to a given mono-chromatic image following color hints provided by the user via scribbles (see also [7]). The added color depends on the geodesic distance between the scribble and the pixel being processed. Being more specific, let s and t be two pixels of the image Ω and $C_{s,t}$ a path over the image connecting them. The geodesic distance between s and t is defined by:

$$d(s,t) := \min_{C_{s,t}} \int_0^1 W dp, \tag{1}$$

where p stands for the Euclidean arc-length and W is a weight that depends on the application (see below). This distance (1) can be efficiently computed in linear time [16], making the algorithm appropriate for interactive applications. Let now Ω_c be the set of pixels labeled by the user provided scribbles l_i, $i \in [1, N_l]$, with color indications in [17] or segment labels in [10]. Then, the distance from a pixel t to a single label l_i, $i \in [1, N_l]$, is $d_i(t) = \min_{s \in \Omega_c \,:\, label(s)=l_i} d(s,t)$, and the probability $P(t \in l_i)$ for t to be assigned to the label l_i representing the class (color or segment) i is given by $\Pr(t \in l_i) = \frac{d_i(t)^{-1}}{\sum_{j \in [1,N_l]} d_j(t)^{-1}}$.

In [17], $W = |\nabla Y \cdot \dot{C}_{s,t}(p)|$, where Y is the given luminance (and the gradient is 2D for images and 3D for video), and this probability $\Pr(t \in \ l_i)$ weights the amount of color the pixel t receives from the color in the scribble (label) l_i.

For segmentation, in order to compute W, [10] starts by modeling, via a Gaussian PDF, each region of interest from the collection of pixels labeled by the user via the scribbles l_i (one Gaussian PDF per label l_i). From this PDF, the likelihood of a pixel to belong to the same class as label l_i is derived considering competing PDF's (competing labels). When multiple colors and channels are used, these likelihoods are further weighted according to the capability of each channel to discriminate between the provided labels (these weights are automatically computed). These likelihoods form the basis for the computation of W in (1), see below and [10] for complete details.

In this paper we extend the work in [10] at a number of important levels. First, with enhanced models for the scribbled/labeled pixels provided by the user, we significantly reduced the user effort and further improve the computational time. Improvements are also obtained following a two stage application of the above described segmentation approach (with the enhanced models). Second, we compute explicit alpha matting (foreground opacity), based on the geodesic distance combined with a function of the actual pixel value. This is critical for composition applications. Third, we extend the work to video. The rest of this paper presents these enhancements and numerous examples.

2. SEGMENTATION AND MATTING FRAMEWORK

We now present the basic extensions mentioned above.

2.1. Improved Foreground and Background Models

Following the distance based work [10], we first propose a more general model for the labeled pixels provided by the user. In [10], the user specifies scribbles on each "uniform" region, in which the pixel features (intensities, colors, or filtered responses) are assumed to be samples from a single Gaussian. Then, as briefly mentioned above, the algorithm computes the likelihoods and weighted distances for every pair of competing foreground/background scribbles. This puts on the user the burden to scribble many regions, virtually one per

Work partially supported by NSF, ONR, NGA, DARPA, and the McKnight Foundation. We thank Kedar Patwardhan for help with kernel methods and other suggestions.

uniform region in the image/video, process which becomes very tedious for complicated images. Ideally, we would like the user to just provide a single scribble for the foreground and a single one for the background, or in general, a single scribble per region the user wants to label together. Aiming at this goal, we enhance the Gaussian model via the standard non-parametric kernel density estimation [12]. The user places single scribbles roughly across the foreground (F) and background (B) and let them automatically propagate throughout the image via the fast weighted distance computations. In contrast to [10], where the weights W in (1) are linear combination of likelihoods from a set of channels, we use the gradient of these likelihoods (in agreement with [17]), which shows better responses to strong edges.

Specifically, let Ω_F be the set of pixels with label F and Ω_B those corresponding to the background. We first estimate the PDF $Pr(x|F)$ of Ω_F, in Luv color space, via kernel density estimation,[1] where x is a color vector. The likelihood $P_F(x)$ of a given pixel x to belong to F according to this PDF computation is then expressed as

$$P_F(x) = \frac{Pr(x|F)}{Pr(x|F) + Pr(x|B)}, \tag{2}$$

and $P_B(x) = 1 - P_F(x)$. We employ the well-developed Fast Gaussian Transform algorithm to efficiently calculate this probability, e.g., [15]. The weighted distance (geodesic) from each of the two labels for every pixel x is then computed as

$$d_l(x) = \min_{s \in \Omega_l} d(s, x), \quad l \in \{F, B\}, \tag{3}$$

where $d(s, x)$ is the distance defined in (1) with weights W computed as in [10, 17], from (the gradient of) the modified likelihood described above. From this weighted distance, the probability of assignment can be obtained as explained in the Introduction.

2.2. Alpha Matting Computation

As detailed before, this distance can be used for color blending, [17], or soft segmentation [10]. We now extend this work to obtain an explicit estimate for the alpha value so that our framework can intrinsically handle image matting problems. The alpha channel is explicitly computed as

$$\omega_l(x) = d_l(x)^{-r} \cdot P_l(x), \quad l \in \{F, B\}, \tag{4}$$

$$\alpha(x) = \frac{\omega_F(x)}{\omega_F(x) + \omega_B(x)}, \tag{5}$$

where r is a constant trading between the distances and the probabilities. In our experiments, r is typically between 0 and 2. Intuitively, pixels that are close to a scribble (in the weighted distance sense), and have similar colors (in the likelihood sense following the kernel density estimation), are assigned higher probabilities for the region represented by that scribble. This alpha matting combines both the weighted distance and the probability previously estimated, and it is much more efficiently computed (interactive real time) and with (at least) competitive results when compared with those works mentioned in the introduction.

As a consequence of the improved image modeling via kernel density estimation and the explicit alpha matting computation, we significantly reduce the user input and are capable of dealing with difficult images such as the one in Figure 1. A comparison with [10] is presented in Figure 2. For analysis on the robustness of the scribble positions, see [17].

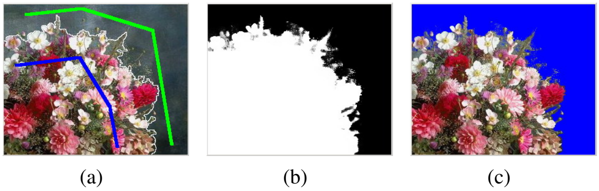


Fig. 1. *(a) The user provides foreground (blue) and background (green) scribbles. The binary segmentation boundary is shown in a white line. The segmentation is obtained by selecting for every pixel the corresponding label with minimal distance. (b) The resulting alpha mask. (c) Foreground segments composite on blue screen*

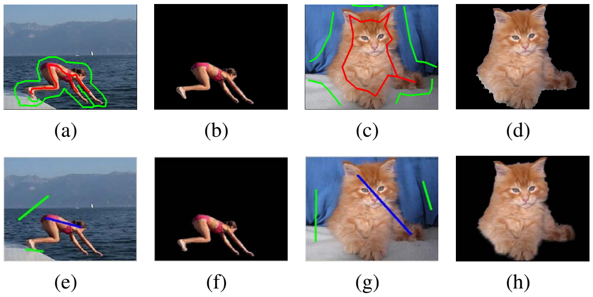


Fig. 2. *Figures (a)-(d) show the user inputs and results from [10]. Figures (e)-(h) correspond to the new inputs and results for the same images, leading to similar results with less user-marked scribbles.*

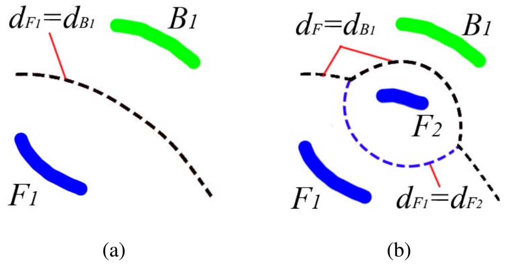
2.3. Interactive Refinement

The proposed algorithm (as the ones in [10, 17]), allows the users to interactively add new scribbles to achieve the desired segmentation in a progressive fashion. By learning from the samples on the new scribbles being added, the weighted distances get updated and the new segmentation result is shown to the user. This process is repeated until the desired segmentation is obtained. Figures 3 and 4 (image from the authors of [11]) illustrate the process of adding one new foreground scribble F_2 to the image. The distance of every pixel to the foreground labels, as defined previously, is updated to the smaller value, i.e., $d_F = \min\{d_{F_1}, d_{F_2}\}$. The propagation of F_2 stops once d_{F_2} exceeds d_{F_1} or d_B . When computing the distance to the new scribble F_2, we use the weights, or probabilities/likelihoods, between only F_2 (not F_1) and the previous background scribbles, giving more accurate local color estimation.

2.4. Additional Speedup Strategies

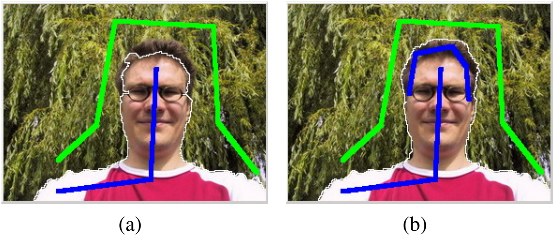
Additional computational improvements can be obtained motivated by the assumption that an object can have semi-opacities only around its border and alpha should be solid elsewhere. This holds true for a large variety of natural images and videos. The main idea then is to quickly find an approximate boundary and generate a narrow stripe around it (trimap). Then the refined computation is limited within

[1] Kernel density has superior performance and computational times than models such as mixtures of Gaussians.



(a) (b)

Fig. 3. *The effect of adding a new F scribble. Dotted line shows the equal-distance line. The F_2 scribble only propagates in a limited area.*



(a) (b)

Fig. 4. *An example showing how the user adds a new scribble to fix the misclassified hair region.*

this stripe.[2]

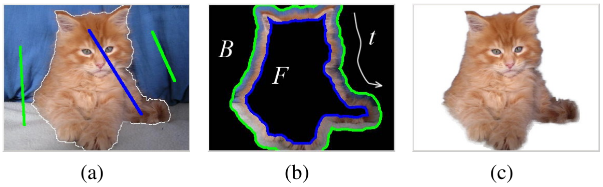
In the first stage, we decompose the Luv color space into three channels, each of which is quantized into 256 levels. A pixel's likelihood is quickly obtained by multiplying via a look-up table the three independently estimated probabilities (from the user provided scribbles as detailed above). A binary segmentation follows, Figure 5(a), which would be less accurate than the full model described before, but often good enough to get a rough initial segmentation. In the second stage, a narrow band is spanned by a distance transform and its borders serve as new foreground and background scribbles (Figure 5(b)), parameterized by $t \in [0, 1]$ (periodic with period 1 if the contour is closed). The band-width depends on the data and can be interactively adjusted by the user. The likelihoods for pixels inside the band are then locally recomputed using the feature vector (L, u, v, t), giving more accurate local estimation. Then we proceed as before to segment with the weighted distance approach. This two-step-framework further reduces the user intervention and computational time yet makes the algorithm more robust and accurate, see Figure 6 for examples.

Our algorithm preserves the important linear complexity. Without any code optimization, it runs for $0.44sec$ and $3.36sec$ (excluding the user operating time) for images with sizes 480×452 and 1500×1500 respectively, on a $1.7GHz$ CPU with 512 MB RAM.

3. EXTENSION TO VIDEOS

Our framework can be easily extended to videos, which can be as 3D images (no explicit optical flow computation, see also [17]). Instead

[2] If the band large enough such that it ends with zero width around the user provided scribbles (basically covering the whole image), we are back into the previously described framework, thereby no obtaining any computational speedup while remaining fully general.



(a) (b) (c)

Fig. 5. *(a) A hard segmentation is quickly found by a few scribbles. The white line indicates the binary segmentation boundary. (b) Automatically generated trimap and new scribbles parameterized by t. (c) Segmented result.*

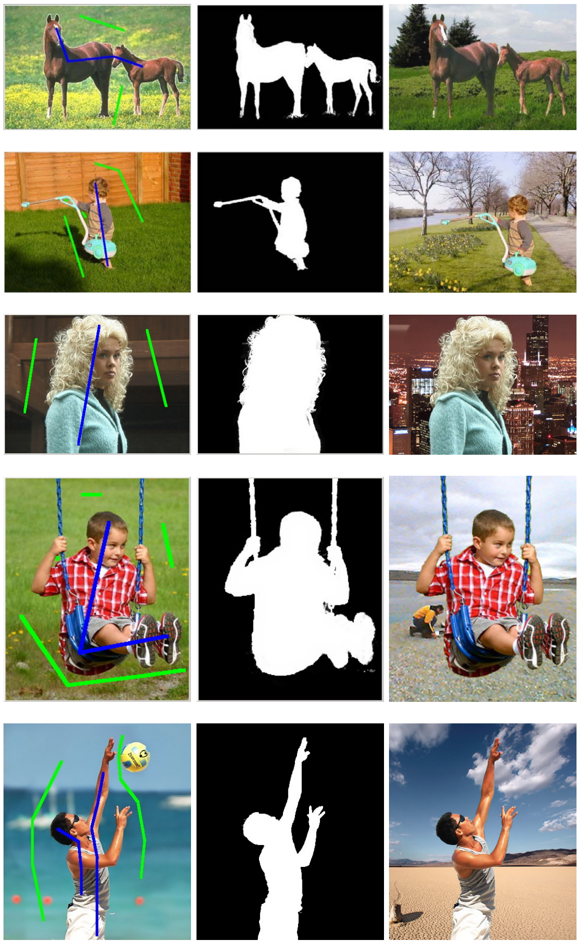


Fig. 6. *Five additional examples. For each set, the user places a few scribbles to obtain different objects of interest (left), computed segmentation and matting (middle), and composition into a new background (right).*

of cutting out a region, the algorithm segments a spatial-temporal tube. The user scribbles on one or more frames and then they propagate throughout the whole video (weighted distances in 3D). See results in figures 7 and 8.

4. CONCLUSIONS AND FUTURE WORK

In this paper we presented a distance-based algorithm for fast interactive natural image and video segmentation and matting. Following the work of [10], we introduced a number of improvements which greatly simplify user input, reduce computational complexity, and produce pleasant matting results. Various difficult examples were presented supporting this. We are currently working on further improving the video results to handle more difficult scenarios with occlusions and dynamic background.

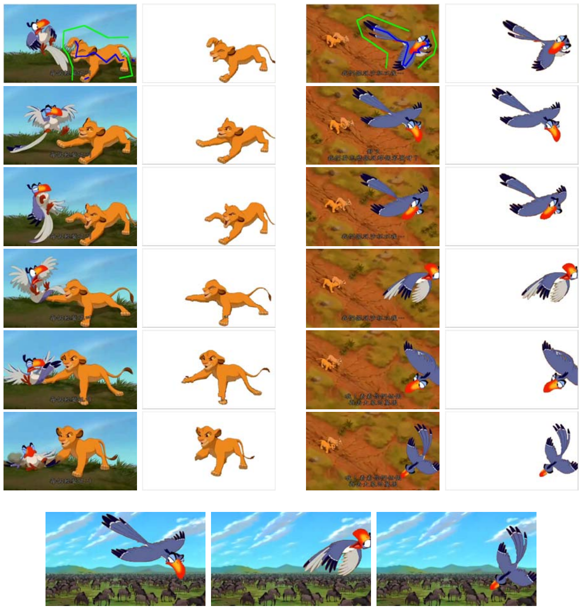


Fig. 7. *Two examples of video segmentation (pair of left and pair of right columns; first six rows). The user draws scribbles on one frame of the video and the algorithm automatically segments the whole video (a total of* 50 *and* 72 *frames). The last row shows an example of video composite.*

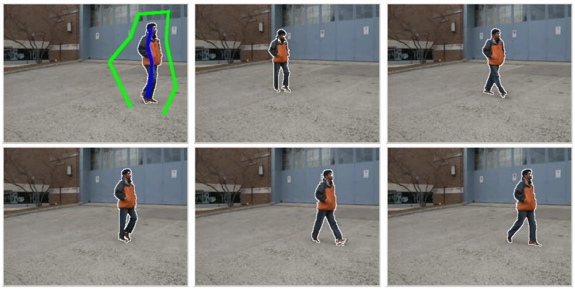


Fig. 8. *Third video example (a total of* 66 *frames).*

5. REFERENCES

[1] A. Agarwala, A. Hertzmann, D. Salesin, and S. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH'04)*, 23:584–591, 2004.

[2] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(1):106–118, 2006.

[3] A. Bartesaghi, G. Sapiro, and S. Subramaniam. An energy-based three dimensional segmentation approach for the quantitative interpretation of electron tomograms. *IEEE Trans. Image Processing*, 14:1314–1323, 2005. Special Issue on Molecular and Cellular Bioimaging.

[4] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *IEEE ICCV*, I:105–112, 2001.

[5] L. Cohen and R. Kimmel. Global minimum for active contours models: A minimal path approach. *International Journal of Computer Vision*, 24:57–78, 1997.

[6] L. Grady and G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, ECCV 2004 Workshops*, pages 230–245, 2004.

[7] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transctions on Graphics (SIGGRAPH'04)*, 23(3):689–694, 2004.

[8] Y. Li, J. Sun, C. K. Tang, and H. Y. Shum. Lazy snapping. *ACM Transactions on Graphics (SIGGRAPH'04)*, pages 303–308, 2004.

[9] E. N. Mortensen and W. Barrett. Intelligent scissors for image composition. *ACM Transactions on Graphics (SIGGRAPH'95)*, pages 191–198, 1995.

[10] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Trans. Image Processing*, 2007. to appear.

[11] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH'04)*, pages 309–314, 2004.

[12] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, 1995.

[13] F. Wang, J. Wang, C. Zhang, and H. C. Shen. Semi-supervised classification using linear neighborhood propagation. *Proceedings IEEE CVPR*, pages 160–167, New York, 2006.

[14] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. *Proceedings of IEEE ICCV*, pages 936–943, Beijing, China, 2005.

[15] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *International Conference on Computer Vision, Nice, France, 2003.*, pages 464–471, 2003.

[16] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(n) implementation of the fast marching algorithm. *Journal of Computational Physics*, 212:393–399, 2006.

[17] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Trans. Image Processing*, 15:5:1120–1129, 2006.