A GEODESIC FRAMEWORK FOR FAST INTERACTIVE IMAGE AND VIDEO SEGMENTATION AND MATTING

By

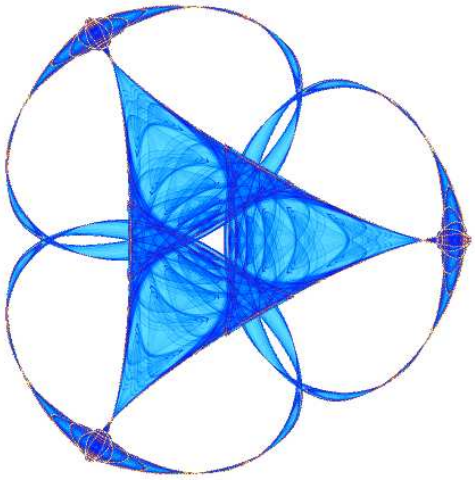
Xue Bai

and

Guillermo Sapiro

IMA Preprint Series # 2171

(August 2007)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
AUG 2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting (PREPRINT)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Minnesota,Institute for Mathematics and its Applications,207 Church Street SE,Minneapolis,MN,55455-0436**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An interactive framework for soft segmentation and matting of natural images and videos is presented in this paper. The proposed technique is based on the optimal, linear time, computation of weighted geodesic distances to the user-provided scribbles, from which the whole data is automatically segmented. The weights are based on spatial and/or temporal gradients, without explicit optical flow or any advanced and often computationally expensive feature detectors. These could be naturally added to the proposed framework as well if desired, in the form of weights in the geodesic distances. A localized refinement step follows this fast segmentation in order to accurately compute the corresponding matte function. Additional constraints into the distance definition permit to efficiently handle occlusions such as people or objects crossing each other in a video sequence. The presentation of the framework is complemented with numerous and diverse examples, including extraction of moving foreground from dynamic background, and comparisons with the recent literature.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**9**	

A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting*

Xue Bai and Guillermo Sapiro
University of Minnesota, Minneapolis, MN 55455, {baixx015,guille}@umn.edu

Abstract

An interactive framework for soft segmentation and matting of natural images and videos is presented in this paper. The proposed technique is based on the optimal, linear time, computation of weighted geodesic distances to the user-provided scribbles, from which the whole data is automatically segmented. The weights are based on spatial and/or temporal gradients, without explicit optical flow or any advanced and often computationally expensive feature detectors. These could be naturally added to the proposed framework as well if desired, in the form of weights in the geodesic distances. A localized refinement step follows this fast segmentation in order to accurately compute the corresponding matte function. Additional constraints into the distance definition permit to efficiently handle occlusions such as people or objects crossing each other in a video sequence. The presentation of the framework is complemented with numerous and diverse examples, including extraction of moving foreground from dynamic background, and comparisons with the recent literature.

1. Introduction

The segmentation of natural images and videos is one of the most fundamental and challenging problems in image processing. One of its applications is to extract the foreground object (or object of interest) out of the cluttered background, and, for example composite it onto a new background without visual artifacts (see also [4] for additional applications in video). For complex images, as well as subjective applications, there can be more than one interpretation of the foreground or objects of interest (in absence of higher level knowledge), thus making the task ill-posed and ambiguous. It is often imperative then to incorporate some user intervention, which encodes prior information, into the process. Specifically, the user can draw rough scribbles labeling the regions of interest and then the image/video is automatically segmented. The user is allowed to add more scribbles to achieve the ideal result, although of course, the goal is to minimize as much as possible the user effort.

Closely connected to the segmentation of objects of interest, image and video matting refers to the process of reconstructing the foreground/background components and the alpha value (transparency) of each pixel. This is important for applications such as extracting hair strands or blurry edges, as well as for compositing. Being inherently under-constrained (solving for three components, F (foreground), B (background), and α transparency, with only the observed color), the matting problem also requires priors, such as user interactions, which could be in the form of scribbles as in the segmentation task, or a complete trimap.

In this paper, we propose a fast weighted-distance-based technique for image and video segmentation and matting from very few and roughly placed user scribbles (often just one scribble for the foreground and one for the background). The distance (geodesic) computation is linear in time, and thereby optimal (with minimal memory requirements as well). The weights are based on simple properties such as spatial and temporal gradients, while more sophisticated features can be naturally included as well. The proposed framework can handle diverse data, including dynamic background, moving cameras, and objects crossing each other in the video.

Following a brief literature review, Section 2, we describe the framework for segmenting and matting still images, Section 3. Examples and comparison with the literature are presented in this section as well. Then, we extend it to video applications, where a long video can be processed with little user interaction, Section 4. We explain how to add constraints to the distance computation to handle moving objects occluding each other, e.g., people/objects crossing each other. We illustrate our method with additional video examples in Section 5, and conclude and discuss future research in Section 6. Before proceeding, let us explicitly present the key attributes of the proposed framework:

1. It is based on weighted distance functions (geodesics), thereby solving a first order geometric Hamilton-Jacobi equation in computationally optimal linear time. This makes the proposed framework natural for user-interactive

*Work supported by ONR, NGA, NSF, DARPA, and ARO.

processing of images and videos.
2. It produces very good, state-of-the-art results, with very few user provide scribbles and very simple attributes defining the weights in the distance computation. We often use just a couple of rough scribbles for still images (one for the foreground and one for the background) and scribble one frame every 70 or so for videos.
3. It applies to a large class of natural data, and since it avoids off-line learning, it is not limited to pre-observed and classified classes and to the availability of ground-truth and hand segmented data.
4. It can handle dynamic background in video as well as crossing objects of interest.
5. The framework is general so that additional attributes can be naturally included in the weights for the geodesic distances if so required for a particular type of data.

2. Related work

One important class of related works is based on energy formulations which are minimized via discrete optimization techniques. The pioneering graph cuts technique, [5], addresses the foreground/background interactive segmentation in still images via max-flow/min-cut energy minimization. The energy balances between the probability of pixels belonging to the foreground (likelihood) and the edge contrast, imposing regularization. The user-provided scribbles collect statistical information on pixels and also serve as hard constraints. The Grabcut algorithm, [14], further simplifies the user interaction. Scribbles can be interactively added to improve the initial segmentation. Full color statistics are used, modeled as mixtures of Gaussians (here, in contrast, we use fast kernel density estimation), and these are updated as the segmentation progresses. This can help but also hurt by propagating segmentation errors. Very good and fast results were demonstrated with this technique. A number of methods have been proposed extending this framework, aiming at devising more sophisticated energy formulations and at extending it to higher dimensions (video). The Bilayer approach, [8], segments videos with basically static background. It incorporates an additional second order temporal transition prior term and a motion likelihood term. Each frame is segmented via graph cuts, conditionally dependent on the previous two frames. Although excellent results are reported for a particular type of videos, this method makes assumptions about the different behaviors of foreground and background pixels and deals with videos with mostly static backgrounds (they do permit a moving object in the background as long as it is different enough from the foreground). Moreover, it needs to learn the motion statistics, which is very useful as they have cleverly incorporated in their system, but requires the availability of pre-segmented ground-truth training data and of video classes (to train and apply with videos having the same type of motion).

Interactive video cutout, [18], presents a system where the user draws scribbles in 3D space. A hierarchical mean-shift preprocess is employed to cluster pixels into super-nodes, which greatly reduces the computation of the min-cut problem. In [9], the author uses random walks for soft image segmentation. Each pixel is assigned the label with maximal probability that a random walker reaches it when starting from the corresponding scribbles. The authors of [20] propose an MRF framework to solve segmentation and matting simultaneously. The basic idea is to minimize the fitting error of the matte while maintaining its smoothness. The uncertainties (0 for the scribbles and 1 for all unknown pixels) are propagated to the rest of the image using belief propagation. Once the alpha values are found, the F and B components are estimated. In [12], a local linear relation between the alpha values and image intensities is assumed, that is, the pixel's alpha value can be immediately determined in a local region if its intensity is known. The matting problem is solved by minimizing a cost function combining the prediction error, the regularization of alpha values, and the user-supplied scribbles which indicate constraints to the optimization problem.

Poisson matting, [15], and Bayesian matting, [7], are two important matting techniques that use trimaps as inputs. Poisson matting computes the alpha matte by solving the second order Poisson equation with Dirichlet boundary conditions.[1] An assumption is made by neglecting the gradients of F and B, considering the matte gradient proportional to the image gradient. Additional operations are performed to adjust to local regions. Bayesian matting simultaneously estimates F, B, and α by maximizing a posterior probability. For each pixel in the trimaps region, it models the known F and B colors around as mixture of oriented Gaussians in color space (again, we use fast kernel densities instead). An (F, B, α) triplet is computed as the one that most probably generates the observed color of that pixel. This technique is applied to videos in [6], where the trimap is temporally propagated using optical flow and the matte is pulled out individually in each frame by the Bayesian matting algorithm. Explicit optical flow is not used in our method, although it could be incorporated as part of the weights in the geodesic computation.

After this paper was submitted for publication, a few additional matting techniques have been published. The spectral matting technique, [10], automatically computes a set of soft matting components via a linear transformation of the smallest eigenvectors of the matting Laplacian matrix [12]. These components are then selected and grouped into semantically reasonable mattes either in an unsupervised or supervised fashion. The main drawback of this algorithm

[1] Note that in contrast with this, we solve a first order Hamilton-Jacobi equation, which is computationally more efficient.

is its high computational cost – it takes several minutes to compute the matting components for small sized images. In addition, it is not intuitive where to place the constraints. The authors of [19] proposed an improved color sampling method for natural image matting, and demonstrated very good performance. The authors in [17] implemented an interface for interactive realtime matting. The user roughly tracks the boundary with a self-adjustable brush. Like in [19], the matte is pulled out in local regions, solving a soft graph-labeling problem. Flash cut, [16], extracts the foreground layers of flash/no-flash image pairs, using the prior information that only the foreground is significantly brightened. This information is incorporated in an graph cut energy framework. The segmentation algorithm is shown to tolerate some amount of foreground motion and camera shake.

Our work is inspired by [23], where the authors, following [11], show how to use distance functions for image colorization. As here, these distances are optimally computed in linear time [22]. This was then extended in [13] for segmentation. In contrast with this work, we use significantly less scribbles per image (thanks in part to a more efficient modeling of the corresponding probability distribution functions), see Figure 1, extend the work to video, and also produce explicit mattes (F, B, and α).

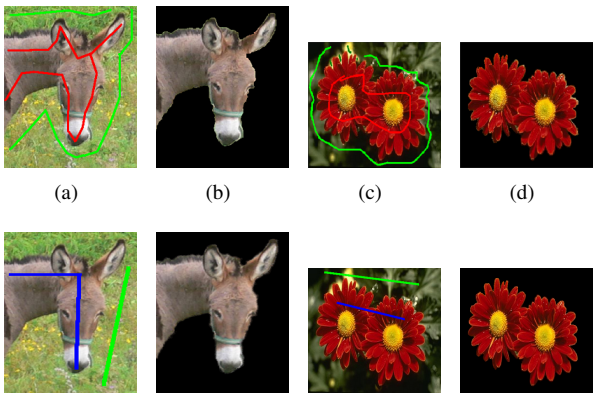


Figure 1. *Figures (a)-(d) show the user inputs and results from [13]. Figures (e)-(h) correspond to the new inputs and results for the same images, leading to better results with less scribbles.*

3. General framework: Still images

As discussed in the introduction, our algorithm starts from two types of user-provided scribbles, $\mathcal{F}$ for foreground and $\mathcal{B}$ for background, roughly placed across the main regions of interest. Now the problem is how to learn from them and propagate this prior information/labeling to the entire image.

We use the geodesic distance from these scribbles to classify the pixels, labeling them $\mathcal{F}$ or $\mathcal{B}$. The geodesic distance $d(x)$ is simply the smallest integral of a weight function over all paths from the scribbles to x. Specifically, let $\Omega_{\mathcal{F}}$ be the set of pixels with label/scribble $\mathcal{F}$ and $\Omega_{\mathcal{B}}$ those corresponding to the background scribble $\mathcal{B}$. The weighted distance (geodesic) from each of the two scribbles for every pixel x is then computed as

$$D_l(x) = \min_{s \in \Omega_l} d(s, x), \quad l \in \{\mathcal{F}, \mathcal{B}\}, \tag{1}$$

where

$$d(s_1, s_2) := \min_{C_{s_1,s_2}} \int_0^1 |W \cdot \dot{C}_{s_1,s_2}(p)| dp, \tag{2}$$

where $C_{s_1,s_2}(p)$ is a path connecting the pixels s_1, s_2 (for $p = 0$ and $p = 1$ respectively). The weights W are set to the gradient of the likelihood that a pixel belongs to the foreground (resp. background), i.e., $W = \nabla P_{\mathcal{F}}(x)$. This likelihood is obtained from the samples on the provided scribbles in Luv color space, i.e., $P_{\mathcal{F}}(x) = \frac{Pr(x|\mathcal{F})}{Pr(x|\mathcal{F})+Pr(x|\mathcal{B})}$, where $Pr(x|\mathcal{F})$ is the color PDF of $\Omega_{\mathcal{F}}$, obtained via the fast kernel density estimation ([21]) (same process for the background PDF). A pixel is close in this metric to a scribble in the sense that there exists a path along which the likelihood function does not change much, Figure 2(d). Following [22], we can efficiently compute the distances, in optimal linear time, and assign each pixel to the label with the shorter distance. The user can progressively add new scribbles to achieve the desired result, although often a single scribble for the foreground and one for the background (regardless of how cluttered it is), is sufficient. If a refinement step is needed, a narrow band is spanned across the

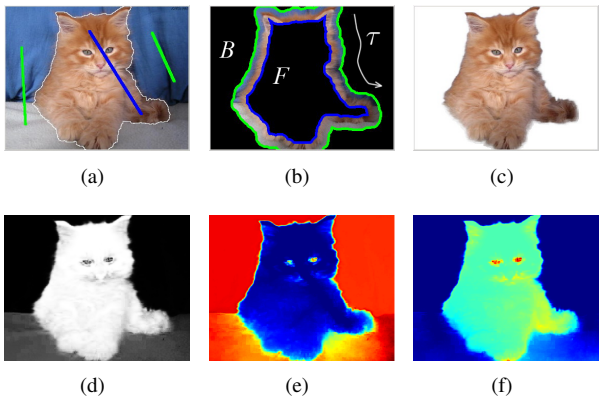


Figure 2. *(a) A hard segmentation (white curve) is quickly found by a few scribbles. (b) Automatically generated trimap, a narrow band around the white curve, and new automatically generated local scribbles (borders of the band). (c) Obtained segmentation and alpha matting. (d) $P_{\mathcal{F}}(x)$. Dark indicates low probabilities and white high probabilities. (Note that this is not the final alpha matte.) (e) $D_{\mathcal{F}}(x)$. (f) $D_{\mathcal{B}}(x)$. Blue indicates low distances and red high distances.*

current boundaries (see Figure 2(b)), and its borders serve as new $\mathcal{F}$ and $\mathcal{B}$ scribbles, thereby reducing the computational cost just to a few pixels in the band, while at the same time refining the likelihood functions and locally adapting them to the region of interest.

Once this distance has been obtained, the alpha channel inside the band is explicitly computed as

$$\omega_l(x) = D_l(x)^{-r} \cdot P_l(x), \quad l \in \{\mathcal{F}, \mathcal{B}\}, \tag{3}$$

$$\alpha(x) = \frac{\omega_{\mathcal{F}}(x)}{\omega_{\mathcal{F}}(x) + \omega_{\mathcal{B}}(x)}, \tag{4}$$

where $P_l(x)$ is locally recomputed using the feature vector (L, u, v, τ), $\tau \in [0, 1]$ parameterizes the band along the boundary (leading to local PDF estimations), and is periodic with period 1 if the curve is closed (see Figure 2(b)). r controls the smoothness of the edges. When $r = 0$, $\alpha(x) = P_{\mathcal{F}}(x)$; when $r \to \infty$, $\alpha(x)$ becomes hard segmentation (typically $0 \leq r \leq 2$ in our examples). This alpha matte combines the weighted distance (measuring how "close" the pixel is to the scribble) and the probability based on the fast kernel density estimation (measuring how probable is its color). Note that regularization, e,g, anisotropic diffusion of α, can be applied inside the band as well if needed. Since this is done locally, virtually no computational cost is added.

After the matte α is computed, we follow the method in [20] to estimate the F_x and B_x components (in Luv space) for each pixel x inside the band. We randomly sample the foreground and background colors in the neighborhood of x and use the pair that gives the minimal fitting error:

$$(F_x, B_x) = \arg \min_{F_i, B_j} \|F_i \alpha_x + B_j(1 - \alpha_x) - I_x\|, \tag{5}$$

where $i \in N(x) \cap \Omega_{\mathcal{F}}, j \in N(x) \cap \Omega_{\mathcal{B}}$, F_i, B_j are foreground and background colors sampled on the (band boundary) scribbles within the window $N(x)$ centered at x, and I_x is the observed color.

With these components, we can now paste the object onto a new background if desired, with no noticeable visual artifacts by the simple matting equation $C_x^* = F_x \alpha_x + B_x^*(1 - \alpha_x)$, where the composite color C_x^* is a linear combination of foreground color F_x and the new background color B_x^* for every pixel x in the image.

Figure 3 shows our results for still images. Note how simple scribbles can handle cluttered and diverse images. Figure 4 presents comparisons with the work in [20] , Photoshop Extract Filter [1], Photoshop CS3 Quick Selection & Refine Edge tools [3] , Corel Knockout2 [2], and Spectral Matting [10] (note how our proposed approach needs significantly less scribbles).

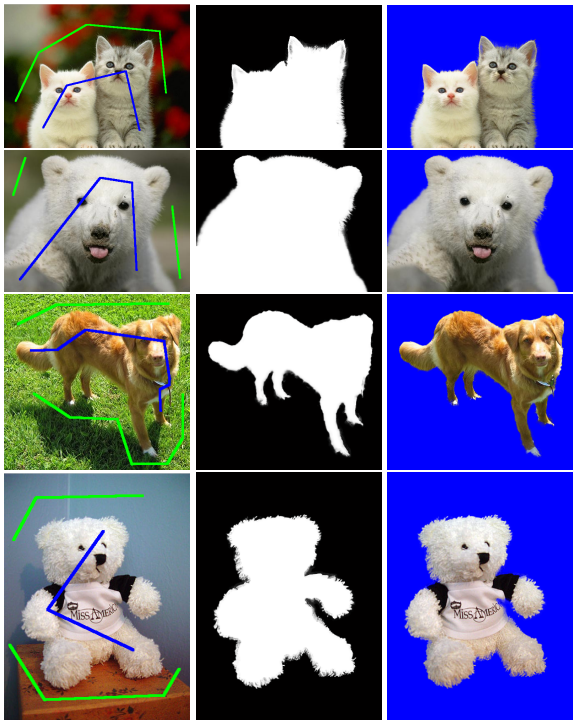


Figure 3. *Left column: original images with user-provided scribbles. Blue for foreground and green for background. Middle column: Computed alpha matte. Right column: Foregrounds pasted on blue backgrounds (blue (constant) backgrounds are selected since they often permit much more careful inspection of the results than pasting on cluttered backgrounds).*

4. Interactive video segmentation and matting

The above described framework is now extended to videos, modeled as 3D images, in which every pixel has six neighbors, four spatial and two temporal (except the ones on the frame borders). The scribbles, drawn on one or several frames, propagate throughout the whole video by weighted distances in spatio-temporal space. In particular, spatial and temporal gradients of the likelihood function are used to define the weight W in the geodesic computation in Equation (2). Note that there is no explicit use of optical flow in the framework (or motion models as in the works described in Section 2), thereby not only simplifying the computations but also permitting to deal with dynamic background and not limiting the work to pre-specified motion classes. As we will see in the experimental section, this simple model is already very useful for numerous scenarios. We now introduce some additional extensions to make it more general.

4.1. Constrained spatio-temporal distance

In still images, a single $\mathcal{F}$ scribble and a single $\mathcal{B}$ scribble always return two connected components. This can be easily proved by the triangle inequality property of the distance function (this also helps to prove the robustness of the

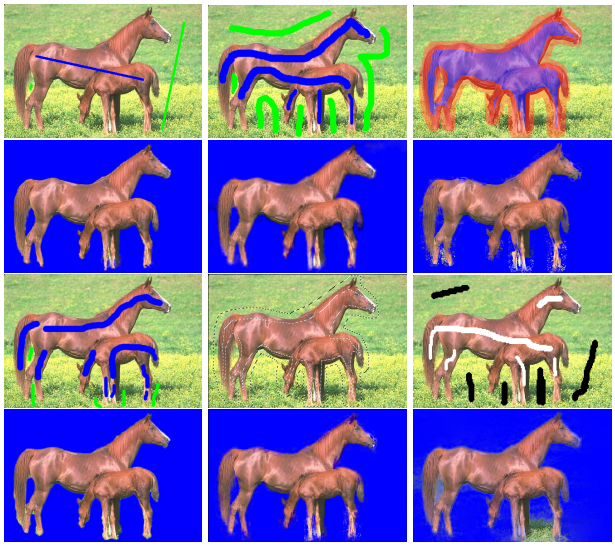


Figure 4. *Comparison of our results (first two rows, first column) with [20] (first two rows, second column), Photoshop Extract Filter [1] (first two rows, third column), Photoshop CS3 Quick Selection & Refine Edge tools [3] (last two rows, first column), Corel Knockout2 [2] (last two rows, second column), and Spectral Matting [10] (last two rows, last column). The first and third rows are the user inputs. The second and last rows are the corresponding results on blue background. ([1] and [2] require complete trimaps.)*

method with respect to the exact placement of the scribbles, see [23]). If the user marks a circle of $\mathcal{B}$ scribble around the object, all the exterior region will be classified as background. However, this is no longer guaranteed in the 3D spatio-temporal case. Consider the simple scenario in Figure 5. Two objects with similar color/feature distributions move towards each other, cross, and split apart. The inside of the tube has low distances to the $\mathcal{F}$ scribble (shown in red). The $\mathcal{F}$ scribble in object A propagates to the frames with occlusions, and then backwards to object B (B refers now to the second object in Figure 5 and not to the background value). Although the user might intend to separate object A as foreground in the initial frame, object B is mistakenly cut out because of the connectivity in 3D space (such connectivity doesn't occur in still images). This phenomenon happens when undesired objects in the background touch the foreground in a certain frame, and the error spreads temporally throughout all frames.

We address this problem with very limited extra computation. To eliminate the branch formed by the undesired object before occlusion, we simply constrain the propagation to be temporally non-decreasing, and Equation (2) is replaced by:

$$d(s_1, s_2) := \min_{C_{s_1,s_2}} \int_0^1 W dp, \ s.t.\ t_1 \leq t_2 \ if \ p_1 \leq p_2, \tag{6}$$

where $p_1, p_2 \in [0, 1]$ indicate any two positions on $C_{s_1,s_2}(p)$ and t_1, t_2 are their corresponding time coordinates. In other words, $d(s_1, s_2)$ is minimized among the paths that temporally go forwards. Of course we can also constrain the distance function in the opposite direction. However, it becomes the same definition if we let the video play reversely.

In the discrete scenario, the temporal links (the links that connect temporal neighbors) are replaced by directed links, i.e., the weight of going backwards in time is set to be infinity. This simple modification leads to the correct segmentation before the occlusion, but confusion might still exists after the occlusion (Figure 5(c)). We can further remove the wrong branch using the same approach, but now in the opposite direction. This can be done by specifying a point in the desired tube at a latter time, letting it propagate backwards within the tubes, constrained to move only backwards. Figure 5 illustrates the process. As a result, the ambiguity is removed in frames where the objects are disconnected within the frame.

Figure 6 shows the example of two people walking. The user desired to segment the person initially on the right. The two people are merged as a single object when they cross each other (since they share the features that are used to compute the weighted distance). The second column shows the results using the distance function without the constraint. The wrong segment appears in every frame (again, see Figure 5(a)). The third column shows the result by the constrained distance function. We can see that the error is removed before and after the intersection. Adding scribbles in the intersection frames will manage to separate them also there, see below, but this is left without in this figure to illustrate the power of the "tubing" effect just described.

4.2. Interactive refinement

For individual frames where occlusion actually happens and can not be fixed by the "tubing" approach described above, the user simply provides extra scribbles to segment the object. Since the color distribution might be inadequate to differentiate the objects (this is what led to their merge in the first case), we switch to another contrast sensitive weight to be used for the geodesic distance computation in Equation (2). This shows the power of the framework, features can be adapted to the problem at hand. For discrete images, the new feature is defined as $W_{pq} := \| I_p - I_q \|$, where p and q are two adjacent pixels and I is the color vector in Luv space. Figure 7 shows how the user separates the two persons using the new weights.

5. Additional video experimental results

We test our algorithm on three videos of 71, 79 and 78 frames respectively. We mark scribbles on two frames for

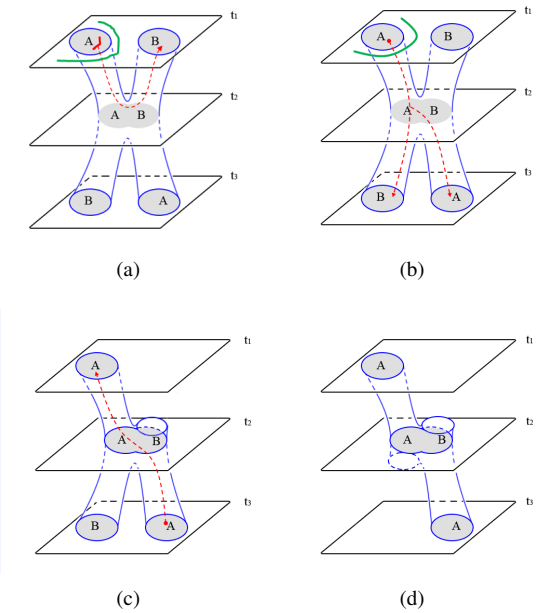


Figure 5. *Tubes in 3D space, where $t_1 < t_2 < t_3$ (a) Although the scribbles in the first frame intend to separate A, the $\mathcal{F}$ scribble (red) reaches the object B by a path in 3D space where both objects A and B overlap. (b) The scribble propagation is constrained to move forward and the branch between t_1 and t_2 is eliminated. (c) The user specifies a pixel in A at t_3 and lets it propagate backwards. The branch of B between t_2 and t_3 is removed. (d) Result with the proper separation of the object A.*

the video in Figure 8 and just a single frame for videos in figures 9 and 10. The results are shown in figures 8, 9 and 10 as image sequences sampled every few frames (please see the videos uploaded with the supplementary material to appreciate the moving camera and dynamic background). The columns correspond to the original frames, alpha matte, composites on a white background, and composites on a new movie.

Finally, we compare our approach with the rotoscoping algorithm in [4] for the video in Figure 8 (we only refer to the segmentation/tracking part, which is the contribution of our paper, and not the very nice special effects they show after the segmentation is obtained). Our approach has a number of advantages over this work: *(a)* We need significantly less user interaction. In [4] the user basically needs to draw the boundaries for all keyframes by hand (about every 10 frames for this video), while our method only requires very few rough scribbles, see Figure 8. *(b)* We explicitly compute the alpha matte, while [4] gives spline approximations of the detected boundaries (explicit computation of the matte was not in the original goals of [4] for their applications). *(c)* Our method can adapt to a wide variety of motions while the algorithm in [4] easily loses track of the object, especially when part of the object moves out of the

Figure 6. *A video example of two people crossing. Left column: original video. Scribbles drawn on the first frame. Middle column: Segmented results using unconstrained distance function. Right column: Segmented results using constrained distance function. See text for details.*

Figure 7. *(a) Original segmentation obtained by gradients of the PDF. (b) The user adds new scribbles. (c) Segmentation results obtained with the new geodesic distance.*

frame, requiring further user intervention. To better illustrate the comparison, we generate the boundaries by thresholding and dilating the alpha matte obtained by our method. A few frames are shown in Figure 11.

Figure 8. *Video example 1. (a total of 71 frames)*

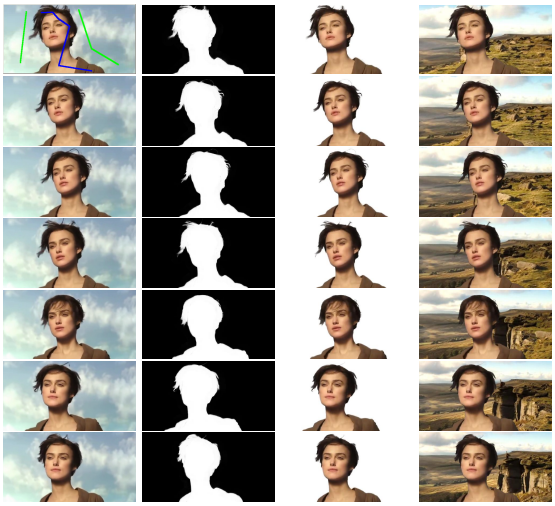


Figure 9. *Video example 2. (a total of 79 frames)*

6. Conclusions and future work

We presented a geodesics-based algorithm for (interactive) natural image and video segmentation and matting. We introduced the framework for still images and extended it to video segmentation and matting. We added constraints to the distance function in order to handle objects that cross each other in the video temporal domain. We showed examples illustrating the application of this framework to very different images and videos, including videos with dynamic background and moving cameras. Another application of our approach is to speed up available image matting algorithms (e.g. [20]). A narrow band trimap is quickly generated from a few scribbles, and then a different matting algorithm is applied. Figure 12 shows our method working in conjunction with [20].

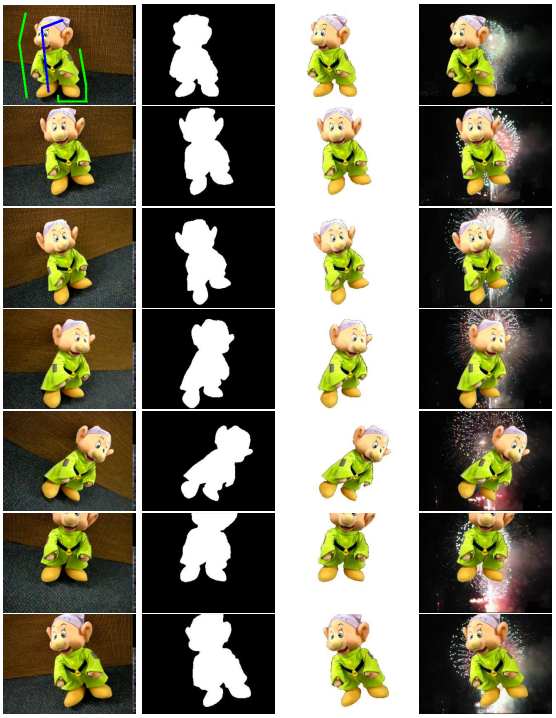


Figure 10. *Video example 3. (a total of 78 frames)*

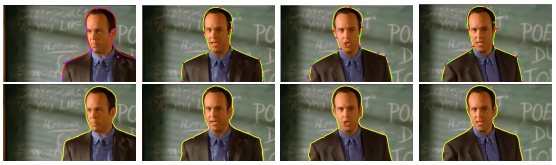


Figure 11. *Comparison with the rotoscoping algorithm in [4]. The curves indicate the boundaries. Top row: A few frames for the work in [4], obtained by their provided interface. The small squares are the control points of the splines. Bottom row: Results from our approach, obtaining similar segmentation with significantly less user intervention (see Figure 8).*

Although the proposed framework is general, we mainly exploited weights in the geodesic computation that depend on the pixel value distributions. As such, in this form the algorithm works best when these distributions do not significantly overlap. In principle, this can be solved with enough user interactions, but could be tedious, and would be better to solve this by enhancing the features used in deriving the weights. Our current efforts are concentrated on enhancing the features we currently use for weighting the geodesic. Also, we are investigating how to naturally add a regularization term into the model, without having to perform this as a post-processing step as currently done. Results in these directions will be reported elsewhere.

References

[1] *Adobe Photoshop User Guide*. ADOBE SYSTEMS

(a) (b) (c) (d)

Figure 12. *Our approach for speeding up [20]. (a) User inputs. (b) Boundary quickly found by our algorithm with the very few scribbles. (c) Trimap as input of [20]. (d) Foreground extracted by [20].*

INCORP, 2002. 4, 5

[2] *Knockout User Guide*. COREL CORPORATION, 2002. 4, 5

[3] *Adobe Photoshop CS3 New Features*. http://www.adobe.com/products/photoshop/photoshop, 2007. 4, 5

[4] A. Agarwala, A. Hertzmann, D. Salesin, and S. Seitz. Keyframe-based tracking for rotoscoping and animation. *Proceedings of SIGGRAPH'04*, 2004. 1, 6, 7

[5] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *IEEE ICCV 2001*, 01:105, 2001. 2

[6] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski. Video matting of complex scenes. In *SIGGRAPH '02*, pages 243–248, 2002. 2

[7] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271, December 2001. 2

[8] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *Proceedings of IEEE CVPR 2006*, pages 53–60, 2006. 2

[9] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006. 2

[10] A. Levin, A. R. Acha, and D. Lischinski. Spectral matting. In *Proceedings of IEEE CVPR 2007*, June 2007. 2, 4, 5

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

[12] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *Proceedings of IEEE CVPR 2006*, pages 61–68, 2006. 2

[13] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Trans. Image Processing*, 16:1046–1057, 2007. 3

[14] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *SIGGRAPH'04*, 2004. 2

[15] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *SIGGRAPH'04*, pages 315–321, 2004. 2

[16] J. Sun, J. Sun, S. Kang, Z. Xu, X. Tang, and H.-Y. Shum. Flash cut: Foreground extraction with flash and no-flash image pairs. In *Proceedings of IEEE CVPR 2007*, June 2007. 3

[17] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: An interactive tool for realtime high quality matting. In *SIGGRAPH'07*, 2007. 3

[18] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *SIGGRAPH'05*, 24(3):585–594, 2005. 2

[19] J. Wang and M. Cohen. Optimized color sampling for robust matting. In *Proceedings of IEEE CVPR 2007*, June 2007. 3

[20] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proceedings of IEEE ICCV 2005*, pages 936–943, 2005. 2, 4, 5, 7, 8

[21] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *Proceedings IEEE ICCV 2003, Nice, France*, pages 464–471, 2003. 3

[22] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(n) implementation of the fast marching algorithm. *Journal of Computational Physics*, 212:393–399, 2006. 3

[23] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Trans. on Image Processing*, 15:5:1120–1129, 2006. 3, 5