INTERACTIVE IMAGE SEGMENTATION VIA ADAPTIVE WEIGHTED DISTANCES

By

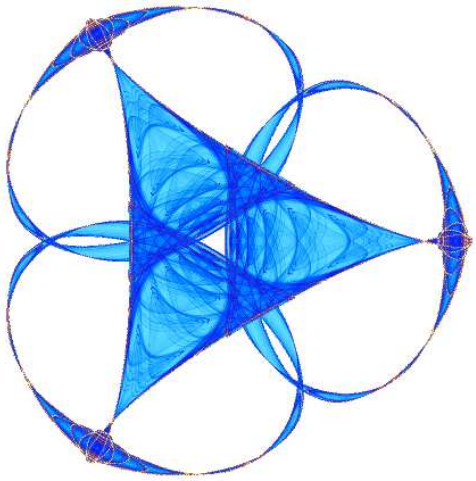
Alexis Protiere

and

Guillermo Sapiro

IMA Preprint Series # 2132

(August 2006)



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

Interactive Image Segmentation via Adaptive Weighted Distances

Alexis Protiere
Ecole Polytechnique, Paris
alexis.protiere@polytechnique.edu

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

***Abstract*— An interactive algorithm for soft segmentation of natural images is presented in this paper. The user first roughly scribbles different regions of interest, and from them the whole image is automatically segmented. This soft segmentation is obtained via fast, linear complexity, computation of weighted distances to the user-provided scribbles. The adaptive weights are obtained from a series of Gabor filters, and are automatically computed according to the ability of each single filter to discriminate between the selected regions of interest. We present the underlying framework and examples showing the capability of the algorithm to segment diverse images.**

I. Introduction

Image segmentation consists in separating an image into different regions, and is one of the most widely studied problems in image processing. There are three main segmentation categories: fully automatic methods, semi-automatic methods, and (almost) completely manual ones. The framework here proposed falls in the semi-automatic category. In particular, the segmentation is obtained after the user has provided rough scribbles labelling the regions of interests. This type of user intervention can help to segment particularly difficult images. Moreover, it is often imperative to mark the regions of interest, which completely depend on the user and the application. For example, the user might be interested in separating a selected object (foreground) from the rest of the image (background), independently of how complicated this background is.

Work supported by the National Science Foundation, the Office of Naval Research, the National Geospatial-Intelligence Agency, and DARPA. AP performed part of this work while visiting ECE at the University of Minnesota. GS performed part of this work while on leave at the IMA.

A number of very inspiring and pioneering user-assisted segmentation-type algorithms of the style here presented have been recently introduced in the literature. The level-set method proposed in [16] for cartoon colorization initializes a curve at the user-provided scribble and evolves it until it finds boundaries of the region of interest. The speed of the moving front depends on local features and global properties of the image. In [21], the authors present a segmentation algorithm based on the assumption that if a pixel is a linear combination of its neighbors, then its label will be the same linear combination of its neighbors' labels. In this way, the user-provided labels (scribbles) are propagated. This is an extension of the learning algorithm developed in [18]. The authors of [25] propose user-assisted segmentation as a particular example of clustering with side information. Grady *et al.*, [8], have also proposed a user-assisted segmentation algorithm. The image is seen as a graph, whose nodes are the pixels and the edges join neighboring pixels. Then they propose to compute the probability for a random walker starting from a unlabelled pixel to reach the user provided scribbles (labels), and assign the pixel to the label with the highest probability. The random walk is biased by weights on the edges, these being a function of the gradient of the intensity. Minimum-cut type of energy algorithms were proposed in [3], [11], [17]. Although these were particularly developed for foreground/background separation (see also [1], [2]), they could in principle be extended to multiple objects, as here addressed (additional relationships between these approaches and ours will be presented throughout the text). Other interactive algorithms are not based on scribbles but on the user helping to trace the boundary of the objects of interest, e.g., [6], [13]. When compared with scribbles based techniques, these

algorithms have often been found not to be as robust and to require more user interaction [17]. Finally, and in particular because we obtain a soft segmentation, we should note that the framework here introduced is also related and can be used for matting, that is, soft separation of foreground from background, e.g., [22].

The pioneering interactive image segmentation approaches just mentioned are mostly based on image gray (or color) values, thereby limiting their use for example for textured data. To address this we introduce the use of an adaptive set of Gabor-based features. User-assisted image segmentation must be fast, preferably on real time. As detailed below, the core computational effort in our framework is linear on the number of pixels. The graph-cut based approaches have also reported very reasonable running times (see also [4]), although not linear.

In order to address the above mentioned key challenges (work fast and for a large class of images), we present an interactive image segmentation approach inspired by the colorization work in [24], where the goal is to add color (or other special effects) to a given mono-chromatic image. In this work, following [10], the authors provide a series of color scribbles on a luminance-only image, and then use geodesic distances computed from the same luminance channel to compute the probability for a pixel to be assigned to a particular scribble. 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. Let also Y stand for the provided luminance channel. The geodesic distance between s and t is defined by:

$$d(s,t) := \min_{C_{s,t}} \int_0^1 |\nabla Y \cdot \dot{C}_{s,t}(p)| dp. \qquad (1)$$

This distance can be efficiently computed in linear time [23], and in contrast with work such as the one in [8] and [15], is related to solving a first order Hamilton-Jacobi equation and not a diffusion or Poisson one. Let Ω_c be the set of pixels labelled by the user, in other words, the user-provided scribbles, with color indications in this case (later on, for segmentation, these scribbles will correspond to region labels). Then, the distance from a pixel t to a label l_i,[1] $i \in [1, N_l]$, is

$$d_i(t) = \min_{s \in \Omega_c \,:\, label(s)=l_i} d(s,t),$$

and the probability for t to be assigned to the label l_i is given by:

$$\Pr(t \in \ i) = \frac{d_i(t)^{-1}}{\sum_{j \in label} d_j(t)^{-1}}.$$

[1]Each label represents a color or a segment.

This probability is used to weight the amount of color the pixel t will receive from the color in the scribble (label) l_i; see [24] for details. In addition to its use for colorization and other special effects as presented in [24], this probability assignment can also be seen as a first step towards soft segmentation, and this is exploited and extended in this work. Thanks to the use of a linear fast marching, [23], in order to compute the geodesic distance in Equation (1), the core algorithm has linear complexity in the number of pixels which is the best we can obtain, since we have to visit each pixel at least once.

Inspired by the ideas just described on colorization, in this paper we propose a semi-automatic algorithm for the segmentation of natural images. We generalize the weights for the geodesic distance, going beyond simple gradients, and thereby permitting to handle significantly more complicated data. We keep the low computational cost of the geodesic computation. The remainder of this paper is organized as follows: In Section II we present the general proposed framework for user-assisted segmentation. Then, in Section III we detail how we compute the weights to be used to replace the simple luminance gradient in Equation (1). Examples are provided all throughout the paper, while introducing the key concepts, and additional ones are presented in Section IV. Finally, in Section V we conclude the paper and present possible directions for future research.

II. General framework description

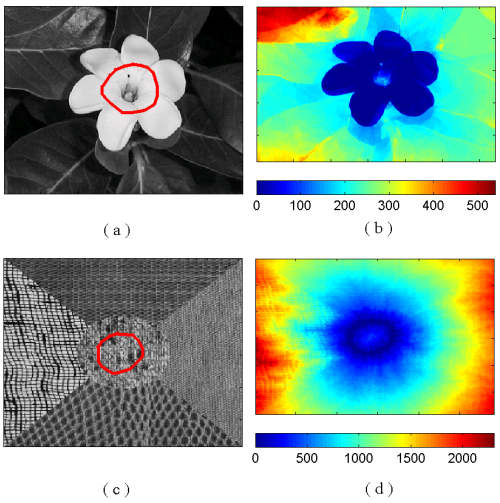


Fig. 1. *Figure (b) (resp. (d)) shows the geodesic distance from each pixel to the red label for the image (a) (resp. (c)). Only luminance gradients, following Equation (1), are used in this case. While this geodesic computation is sufficient to segment the image in Figure (a), in (d) we notice that the geodesic distance does not contain enough relevant information about the different textured regions. (This is a color figure.)*

As detailed above, following the colorization work, the use of fast geodesic computations is a very interesting way to perform semi-automatic segmentation starting from user provided labels. In its original form, this method assumes that the gradient of the intensity (or color) is low inside the region of interest and high at the boundaries. Although, there are a lot of images where this assumption is reasonable, it obviously fails for example for images containing textures, see Figure 1. While preserving the general idea of obtaining a soft segmentation by geodesic propagation of user-provided labels, we would like to use different weights in defining the distance in Equation (1). In other words, we propose to replace the ∇Y term representing the gradient of the luminance channel, by a more elaborated weighting function, and then still derive the soft segmentation via the fast geodesic computation following [23], [24]. We basically consider the image grid as a graph with pixels as nodes and edges connecting neighboring pixels. In this framework, the geodesic distance can be seen as the cost of the shortest path on this graph.[2]

A color image can be described by different combinations of channels. The most commons are (R, G, B) or (Y, C_b, C_r) (luminance/chrominance), but we can also build many additional channels, for example, by filtering the luminance Y. We then represent the image as a bank of N_c channels, $(F_i)_{i=1\ldots N_c}$, and use the information contained in this bank to define the weights for the geodesic definition.[3] When computing the weights, we can restrict the used information to the positions of the user-provided labels (scribbles). To summarize, the general expression for the weights W_i for each scribble (label) i of the N_l provided by the user is given by

$$W_i = f(F_1, \ldots, F_{N_c}, \Omega_c, i), \quad i = 1, \ldots, N_l$$

where again, W_i is the weight function on the graph for the fast geodesic computation associated with the label l_i (to replace the luminance gradient in Equation (1)), and as before, Ω_c is the set of pixels corresponding to the user-provided scribbles. The objective is to design W_i such that the weights are low inside of the region of interest labelled with l_i and high outside. Then, we can efficiently compute the weighted distance maps d_i for $i \in \{1, \ldots, N_l\}$, and assign each pixel to its closest label (or leave this as a soft clustering). In the next section, we show how to compute these weights W_i.

We should note that both the pixel values at the user-provided scribbles (see below), and their actual position, are explicitly used in the segmentation. This permits for example to avoid wrongly disconnected foreground and or background segments.

III. Design of the weight functions

The weight design includes several components. First, we have to define the set of images F_i. Then, we have to show how to adaptively select the relevant sub-set from them, or how to differently weight each channel F_i. This is critical, since when selecting a large number of channels, as needed to address a rich spectrum of data, the critical information for a particular region in a particular image is mostly in a few channels. If not explicitly addressed, this will be obscured by any metric comparing the whole set of channels, and the different regions will not be well separated. In this section, we first describe the selection made to create the channels F_i. We then consider the particular case where the goal is just to segment two different regions in the image, and show how to adaptively weight the different channels. Later we extend this to more than two regions.

A. Selecting the channels

In addition to the luminance and chrominance channels Y, C_b, C_r, we use a bank of 16 Gabor filters (4 scales and 4 orientations) on the channel Y ($N_c = 19$ in our experiments then). This type of filter has been frequently used in the literature to deal with texture, e.g., [9], [19]. The basic two dimensional Gabor function, $g(x, y)$, is a harmonic modulated by a Gaussian,

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j\omega x\right],$$

where the different standard parameters control the frequency and width of the filter. The 16 impulse filter responses are obtained by appropriate rotations and dilatations of this basic function. The idea is that, locally, these filters express the scale and orientation of a texture. For details about the mathematical properties of the Gabor functions, we refer the interested reader to [12]. For speed-up for example, we could replace these filters by the steerable pyramid [20].

Since natural texture can be complex and noisy, we regularize the outputs of the Gabor filters. First, we saturate the too high values by a non-linear transformation. Then, with an averaging operator, we smooth the variations. Therefore, if G_i is the response of a Gabor filter applied

[2]Note that the use of the linear complexity technique in [23] avoids the classical metrication errors of Dijkstra and graph-cuts algorithms that operate on such graphs, thereby providing more accurate results.

[3]The larger the class of images that we want to address with a single algorithm, the richer this bank of channels needs to be, thereby increasing the complexity of the proposed approach. This type of rich representation is needed by all techniques working with large image classes, and thereby this is a step intrinsic in all general segmentation algorithms. To reduce the complexity, a different sub-set of filters can be used for each pre-established image class.

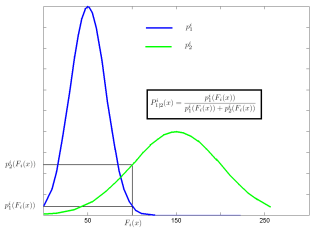


Fig. 2. *Expression of the probability for the pixel x to be in the region 1 with respect to p_1^i and p_2^i. In this case x is much more likely to be in region 2 than in region 1. (This is a color figure.)*

to Y, Ω the domain of the image, and Ω_x an $N \times N$ window around the pixel x, we consider the channel F_i given by:

$$\forall x \in \Omega : \quad F_i(x) := \frac{1}{N^2} \int_{\Omega_x} \tanh\left(\alpha \frac{G_i(y)}{\sigma(G_i)}\right) dy,$$

where α and N are parameters experimentally set to 0.25 and 5 respectively, and $\sigma(\cdot)$ stands for the standard deviation.

We have just defined the bank of channels F_i that represent the image. We now show how to adaptively weight them to define the global weight to be used in the geodesic computation.

B. Segmenting two uniform regions

Let us begin by the segmentation of two uniforms regions.[4] Let Ω_1 and Ω_2 be the set of pre-labelled pixels for the user-provided labels l_1 and l_2 respectively. On each one of the N_c channels, we first approximate the probability density function (PDF), with the samples on Ω_1 and Ω_2, by a Gaussian (see also [3], [17]).[5] We then compute the likelihood for a pixel x to be assigned to the label l_1 based on the channel F_i:

$$P^i_{1|2}(x) := \frac{p^i_1(F_i(x))}{p^i_1(F_i(x)) + p^i_2(F_i(x))},$$

where p^i_j is the PDF of Ω_j on F_i; see Figure 2. Similarly, we compute $P^i_{2|1}$.

Then, the probability for a pixel x to be assigned to l_1 is given by

$$P_{1|2}(x) := \Pr(x \in l_1) = \sum_{i=1}^{N_c} w^i P^i_{1|2}(x), \qquad (2)$$

where w_i are weights reflecting the ability of the channel $i \in N_c$ to discriminate between the two regions of interest (their computation will be explained below). Then, the weight associated with the geodesic computation for the label l_1 is given by

$$W_1 = W_{1|2} = 1 - P_{1|2}.$$

W_2 is similarly obtained; see Figure 3.[6]

As in [3], we could re-estimate the probability functions as the algorithm progresses. This has the advantage of creating richer representatives, at the cost of additional computations and the risk of including wrongly assigned pixels in the estimation.

C. Weighting of the channels

We have chosen to consider a number of channels because it permits to characterize a wide range of images. On the other hand, for one precise image, there are often very few channels which are relevant for the discrimination, and using the others will just mislead and hide the useful information.[7] We need then to find the relevant channels, relying on the user provided scribbles/labels.[8]

To compute the relevance of an individual channel for a given image/region, we assume that the PDFs of the

[4]Here, we use the word *uniform*, in the sense that the regions can be discriminated by the set of computed channels.

[5]Although other fitting functions might be more appropriate, we found this sufficient for the very good results here reported.

[6]In fact, we normalize each W_i by dividing by its standard deviation in order to make them comparable.

[7]As an example, consider the case where there are $N_c >> 1$ and $N_c - 1$ channels are identical for both regions, but one is very different. Using all N_c channels will in general lead to consider both regions the same.

[8]We assume that the user is not an adversary, and if he/she marked scribbles in different regions is because the data around the scribbles is different, and is useful to perform the segmentation.

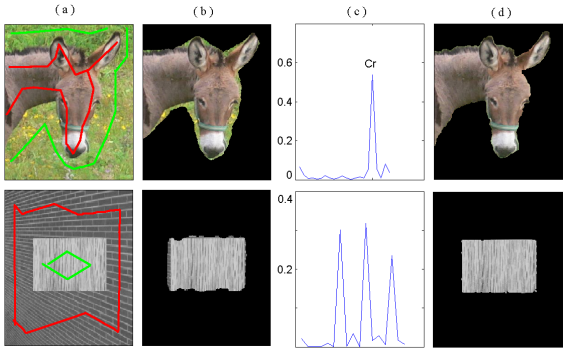


Fig. 3. *Two examples of segmentation into two uniform regions, showing the importance of adaptive weights. (a) The user-scribbled image. (b) The segmented image using all equal weights, note the significant errors in the segmentation. (c) The automatically computed weights for each channel.The horizontal axis indicates the different $N_c = 19$ channels, and the vertical their corresponding weights. While the top image mostly uses a chrominance channel, the bottom one strongly uses three of the Gabor channels obtained by filtering the luminance. (d) The segmented image, with automatically computed adaptive weights. (This is a color figure.)*

regions of interest are represented by the PDFs obtained using the information in the scribbles Ω_1 and Ω_2. With this in mind, we evaluate the probability for a random point s in the image to be assigned to the wrong label, as a function of the previously computed PDF's on the scribbles (p_1^i and p_2^i):

$$\begin{aligned} P_i &= \Pr(s \in 1)\Pr(s \to 2 | s \in 1)) + \\ &\quad \Pr(s \in 2)\Pr(s \to 1 | s \in 2)), \end{aligned}$$

$$\begin{aligned} P_i = &\frac{1}{2} \int_{\{x: p_1^i(x) > p_2^i(x)\}} p_1^i(x)dx + \\ &\frac{1}{2} \int_{\{x: p_2^i(x) > p_1^i(x)\}} p_2^i(x)dx. \end{aligned}$$

Then, we deduce that

$$P_i = \frac{1}{2} \int \min(p_1^i(x), p_2^i(x))dx.$$

This quantity has been shown by Dunn and Higgin, [7], to be a good criteria for channel selection. From it, we deduce directly the weights for each one of the channels (to be used in Equation (2)):

$$\forall i = 1 \dots N_c : \; w_i = \frac{(P_i)^{-1}}{\sum_{k=1}^{N_c} (P_k)^{-1}}.$$

Figure 3 shows that depending on the type of image and the regions we want to segment, the weights are concentrated on different channels, either the Gabor filters channels, or the luminance or chrominance channels. This is automatically computed with the technique just described. The figure also shows the critical importance of adapting the weights to the data.

We should mention that in the segmentation results presented in this paper, for simple visualization, we hard threshold the soft segmentation obtained from the framework just described. Recall that every pixel is assigned a probability of belonging to each one of the regions represented by the user-provided scribbles. Such probabilities can be regularized, e.g., [14], before hard assignment. This will for example regularize the contours, see Figure 4. Such regularization needs to be performed only around regions of border-line decisions, thereby not adding significant computational cost. In order to simplify the presentation and to concentrate on the novel contributions, for the rest of this paper, we visualize only the results of hard assignment and without any regularization.

D. Multiple uniform regions

In the previous section, based on the user-provided scribbles, we compare the properties of two regions in order to discriminate them. If we want to segment an image in more than two uniform regions, we might not be able to find one particular channel which discriminates well between one region and all the others. Therefore, we have to first compute an optimal weight function for each pair of regions and then combine them to build a global weight function for the region (to use in the geodesic computation), in such a way that it is low inside the region and high everywhere else.

Let $\{l_1, \dots, l_{N_l}\}$ be the set of labels and $W_{i|j}$ the weight function for the label l_i when competing only with l_j.[9] We want the global weight function of l_i to be very low inside its region and very high outside, and then we can define W_i as

$$W_i = \sum_{j \neq i} W_{i|j}. \tag{3}$$

Figure 5 graphically shows how this method builds a good weight function. An example is presented in Figure 6.

E. Non uniform regions

In many pictures, objects and background are not uniform (composed by many objects) and the PDFs for each label can not be modelled by a simple Gaussian as we previously did. We could of course use other models, such as mixture of Gaussians (see for example [17]). Continuing with our philosophy of letting the user help,

[9] $W_{i|j}$ is computed as in the previous section, as if there were only two labels, l_i and l_j

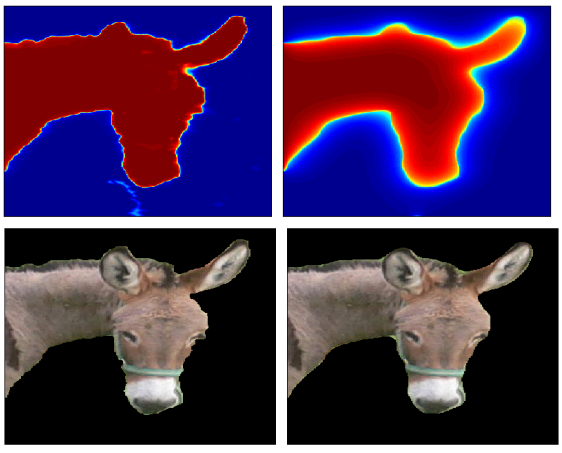


Fig. 4. *Regularization effects on the probability distribution, top row, followed by its effects on the region boundary, bottom row. Original images on the left and regularized ones on the right. Although here the whole probability is regularized for illustration purposes, only the region of border-line decision needs to be processed. (This is a color figure.)*

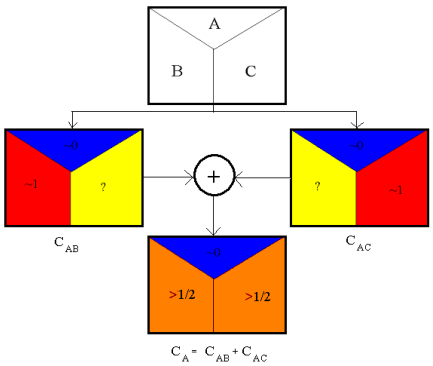


Fig. 5. *Principle for the design of a weight function in the configuration of three to be segmented regions A, B and C. (This is a color figure.)*

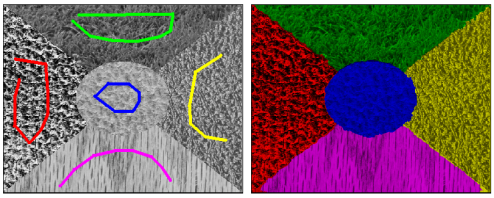


Fig. 6. *Segmentation example with more than 2 regions. (This is a color figure.)*

instead of computationally complicating the algorithm, we opt for a different approach. Although, for example, the background is not composed of a single region, often we can easily distinguish several uniform areas in the object. The choice we made here is to let the user decide about the uniform sub-regions and to scribble the region using several components (see Figure 7). Then, we consider each component as an independent labels (we call them sub-labels), and we are back to the previous problem of multiple uniform regions. The only difference is that we don't need to make two components of the same label compete against each other, we don't care to discriminate between them. Let l_i^j be the component j of the label l_i. Then W_i^j, its weight function, is given by:

$$W_i^j = \sum_{k \neq i} \sum_{l} W_{l_i^j | l_k^l}. \tag{4}$$

After segmentation, we merge the regions assigned to components coming from the same label.

IV. Additional examples

We now present additional examples of the proposed framework for interactive natural image segmentation. First, in Figure 8 we present two very different and diverse images, showing the generality of our proposed framework. Figure 9 exemplifies the robustness of the algorithm with respect to the positions of the user-provided

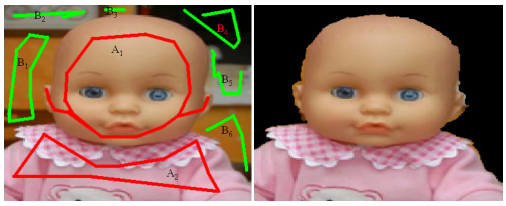


Fig. 7. *Example of a segmentation with non uniform regions. Green and red scribbles do not compete among themselves, only against each other. (This is a color figure.)*

scribbles. Since the segmentation is based on geodesic distances, we can explicitly apply the triangle inequality to study the algorithm robustness. Thereby, the error in the probability assignment of a given pixel is upper-bounded by the geodesic distance between the user-placed scribbles in the two different scenarios (assuming the PDFs for both scribbles are the same since they are in the same region, if not, this can be easily included in the bound). If in both cases the scribbles are placed inside the same region, as expected from a non-adversary user, this distance is small (ideally zero), and as such, the error is small. Finally, in Figure 10 we first simulate the use of our framework in a real interactive application, where the user progressively adds scribbles to achieve the desired segmentation result (see also [3], [17]). Finally, we show the use of this result for image composition. A simply cut-and-paste has been used, with no blending. For a real application, simple blending needs to be added, see for example [5], [17], [22]. For this, the natural soft segmentation here obtained is very useful.

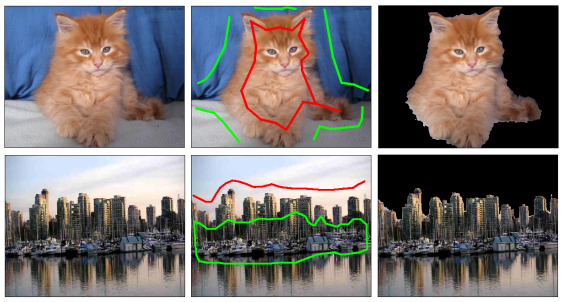


Fig. 8. *Additional segmentation examples. In the image with the cat, the background is decomposed into two sub-labels. (This is a color figure.)*

V. Concluding remarks

In this paper, we have proposed an interactive algorithm for soft image segmentation. The proposed technique is adaptable to a wide range of images thanks to the automatic weighting of the different channels involved in the segmentation. Based on the fast computation of

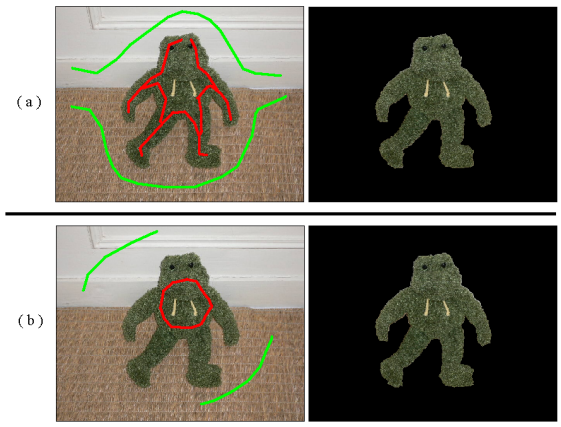


Fig. 9. *(a) and (b) are two segmentation results of the same image using different user scribbling. It shows the robustness of the algorithm, it is not necessary to carefully scribble in order to obtain very good results. (This is a color figure.)*

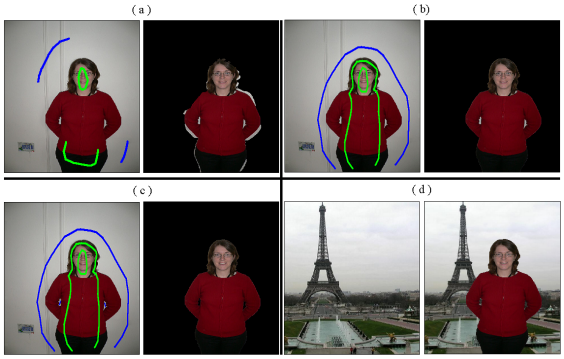


Fig. 10. *Progressive segmentation. (a) The user starts with a few minor scribbles, obtaining only a partial desired segmentation. (b) The user adds scribbles, further improving the segmentation. (c) The user completes the segmentation by marking under the arms. (d) Example of a typical application of this type of foreground/background segmentation: image composition. Note that simple cut-and-paste has been used. (This is a color figure.)*

geodesic curves, the core algorithm is linear in time and can be used for interactive image labelling.

There are several directions of research to pursue with the user-oriented segmentation framework here introduced. We still want as little as possible user work, and thereby helping the user to place the scribbles will be very helpful. For example, a simple edge detector can hint for good scribble locations. In addition, for segmenting rich images into just foreground and background, models that remain computationally simple but allow the user to provide just one scribble for the whole foreground and just one for the whole background, will be very helpful. In particular, and assuming that the foreground object is completely inside the image, it is interesting to study the use of the image borders as the background scribbles. We are also extending this work to video, and results on this will be reported elsewhere.

REFERENCES

[1] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *Digital Image Computing: Techniques and Applications*, 2003.

[2] 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.

[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *International Conference on Computer Vision (ICCV)*, I:105–112, 2001.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124–1137, 2004.

[5] P. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2:217–236, 1983.

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

[7] D. Dunn and W. E. Higgin. Optimal gabor filters for texture segmentation. *IEEE Trans. on Image Processing*, 4:947–964., 1995.

[8] 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.

[9] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[10] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, 2004.

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

[12] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:837–842, 1996.

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

[14] A. Pardo and G. Sapiro. Vector probability diffusion. *IEEE Signal Processing Letters*, 8:106–109, April 2001.

[15] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH'03)*, 2:313–318, 2003.

[16] Y. Qu, T. Wong, and P. A. Heng. Manga colorization. *ACM Trans. on Graphics*, 25:1214–1220, 2006.

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

[18] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[19] T. Shioyama, H. Wu, and S. Mitani. Segmentation and object detection with gabor filters and cumulative histograms. *Proceedings International Conference Pattern Recognition*, 1:704–707, 2000.

[20] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Trans Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.

[21] 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.

[22] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. *Proceedings of International Conference Computer Vision*, pages 936–943, Beijing, China, 2005.

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

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

[25] R. Zass and A. Shashua. A unifying approach to hard and probabilistic clustering. *Proceedings International Conference on Computer Vision (ICCV)*, Beijing, China, Oct., 2005.