Interactive Image Segmentation via Adaptive Weighted Distances

Alexis Protiere and Guillermo Sapiro, *Member, IEEE*

***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.**

***Index Terms*—Adaptive weights, distance functions, interactive segmentation, linear complexity, natural images.**

I. Introduction

IMAGE segmentation consists of 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.

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 [21] 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 [28], 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 [23]. The authors of [32] propose user-assisted segmentation as a particular example of clustering with side information. Grady *et al.* [10] 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 [4], [14], [22]. Although these were particularly developed for foreground/background separation (see also [2] and [3]), 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., [8] and [16]. When compared with scribbles-based techniques, these algorithms have often been found not to be as robust and to require more user interaction [22]. Active contours have also been extensively developed to segment objects of interest. In particular, they also have been extended to deal with texture; see, for example, [12], [18], [24], and references therein. These are based on Gabor-type filters, [11], [25] (see Section III-A). The philosophy behind active contours is not necessarily to develop interactive system. Interactive systems help in cases where active contours have been shown to converge to (wrong) local minima and when the user is the one to (subjectively) define the foreground/background objects. We should also point out that weighted distances, which are at the core of the framework developed here, have been found to be very useful in image processing and other disciplines; see, for example, [1], [7], [17], [27], and references therein. 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., [29].

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 in the following, the core

Manuscript received August 27, 2006; revised December 8, 2006. This work was supported in part by the National Science Foundation, in part by the Office of Naval Research, in part by the National Geospatial-Intelligence Agency, and in part by DARPA. A. Protiere performed part of this work while visiting the Department of Electrical and Computer Engineering, University of Minnesota. G. Sapiro performed part of this work while on leave from the IMA. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitri Van De Ville.

A. Protiere is with the Ecole Polytechnique, Paris, France (e-mail: alexis.protiere@polytechnique.edu).

G. Sapiro is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: guille@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2007.891796

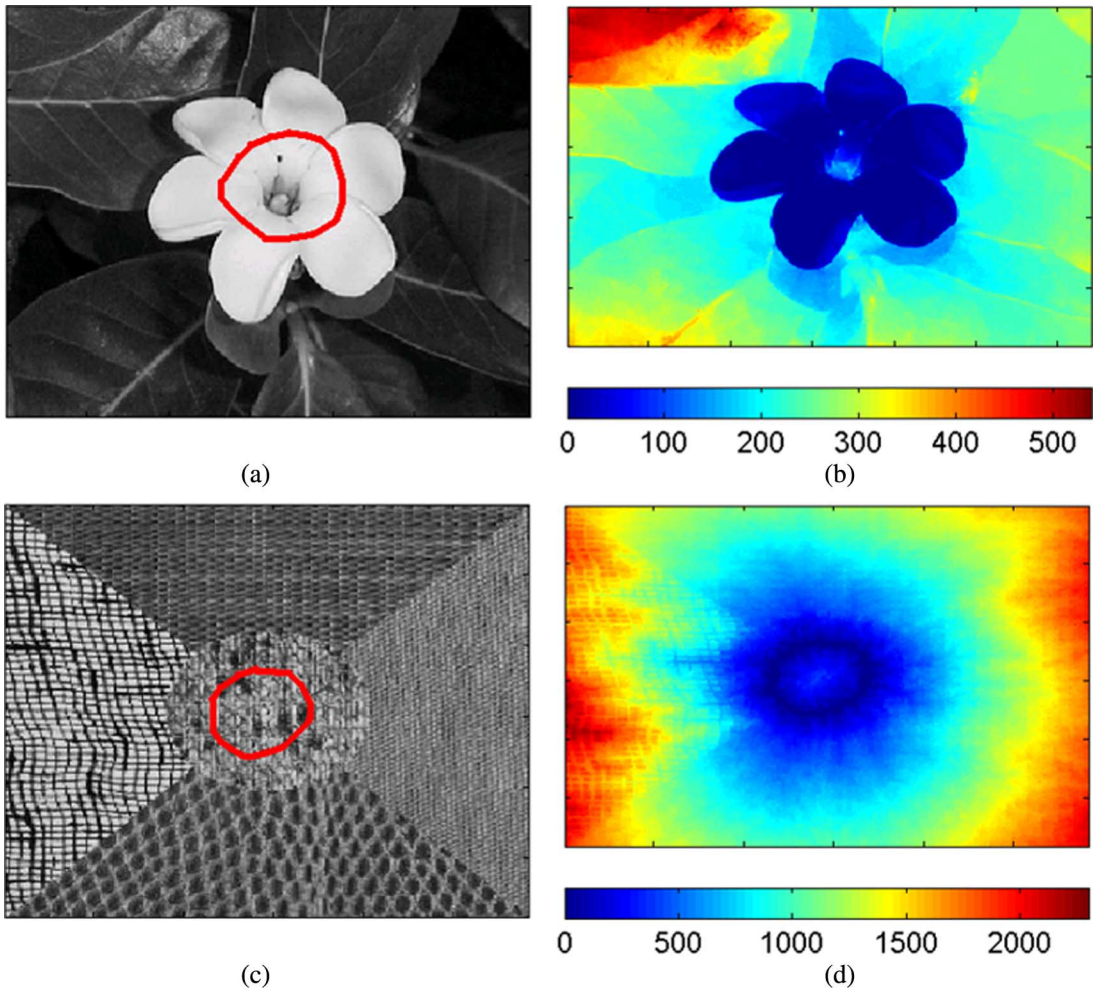

Fig. 1. (b) [resp. (d)] Geodesic distance from each pixel to the red label for the image in (a) [resp. (c)]. Only luminance gradients, following (1), are used in this case. While this geodesic computation is sufficient to segment the image in (a), we notice in (d) that the geodesic distance does not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

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 [5]), 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 [31], where the goal is to add color (or other special effects) to a given mono-chromatic image. In this work, following [13], 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

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

where p stands for the Euclidean arc-length. This distance is basically accumulating the gradient ∇Y along the path $C_{s,t}$, and computing the minimum accumulated gradient along all possible curves connecting s and t (other weights, replacing the gradient, will be proposed in this work for image segmentation). Note that, if we ignore the gradient component, (1) is simply computing the regular Euclidean distance between s and t along the path $C_{s,t}$ (which will be a straight line for the minimum).

This distance (1) can be efficiently computed in linear time [30], and in contrast with work such as the one in [10] and [20], 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/labels l_i, $i \in [1, N_l]$, with color indications in [31] (later on, for segmentation, these scribbles will correspond to region labels). Then, the distance from a pixel t to a single label $l_i, i \in [1, N_l]$, where each label represents a color or a segment, is

$$d_i(t) = \min_{s \in \Omega_c : label(s) = l_i} d(s,t) \tag{2}$$

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

$$\Pr(t \in l_i) = \frac{d_i(t)^{-1}}{\sum_{j \in [1, N_l]} d_j(t)^{-1}}. \tag{3}$$

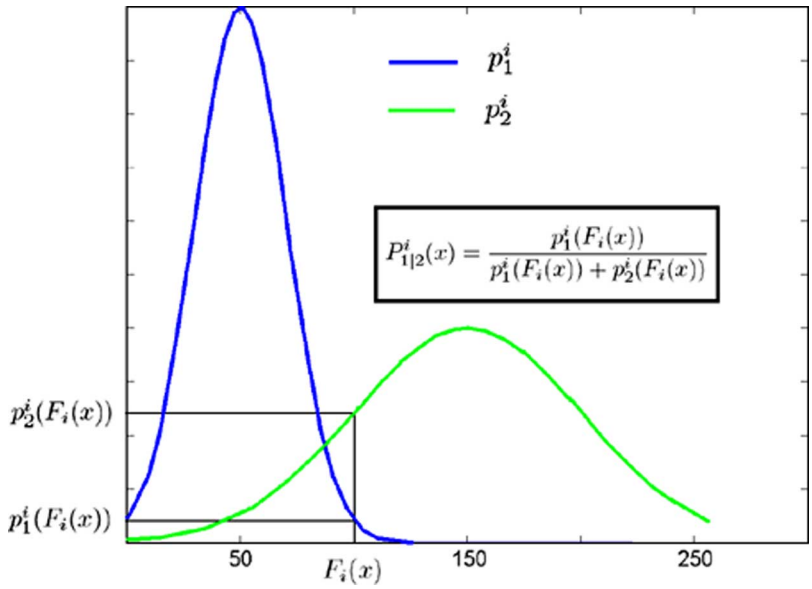

Fig. 2. Expression of the probability for the pixel x to be in the region 1 with respect to p_1^i and p_2^i. In this case, x is much more likely to be in region 2 than in region 1.

This probability is used to weight the amount of color the pixel t will receive from the color in the scribble (label) l_i; see [31] for details. In addition to its use for colorization and other special effects as presented in [31], 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, [30], in order to compute the geodesic distance in (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 first generalize and learn the weights to be used for the geodesic distance in (1), going beyond simple gradients, and thereby permitting to handle significantly more complicated data. The weight provided by the gradient in (1) is replaced by weights automatically learned and adapted to the image. We then compute the distances from every pixel to the scribbles (region labels provided by the user), following (1) and (2), keeping the low computational cost of the geodesic computation [30]. Finally, from these weighted distances, the probability of a pixel to belong to the region corresponding to every user-provided scribble is given by (3).

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 (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.

We conclude this introduction by stating the key contribution of this work. We propose a framework for interactive image segmentation which is based on fast (linear time) weighted distance computations. The weighted distances are computed from the user provided labels. The weights are automatically learned to optimize the discrimination between the different regions labelled by the user, further exploiting the valuable information resulting from the labels. The proposed technique efficiently handles rich images, with multiple different regions and textures and provides a soft segmentation useful for follow-up applications such as compositing.

II. General Framework Description

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 Fig. 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 (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 [30] and [31]. 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. Finally, note that the use of the linear complexity technique in [30] avoids the classical metrication errors of Dijkstra and graph-cut algorithms that operate on such graphs, thereby providing more accurate results.

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, $(F_i)_{i=1...N_c}$, and use the information contained in this bank to define the weights

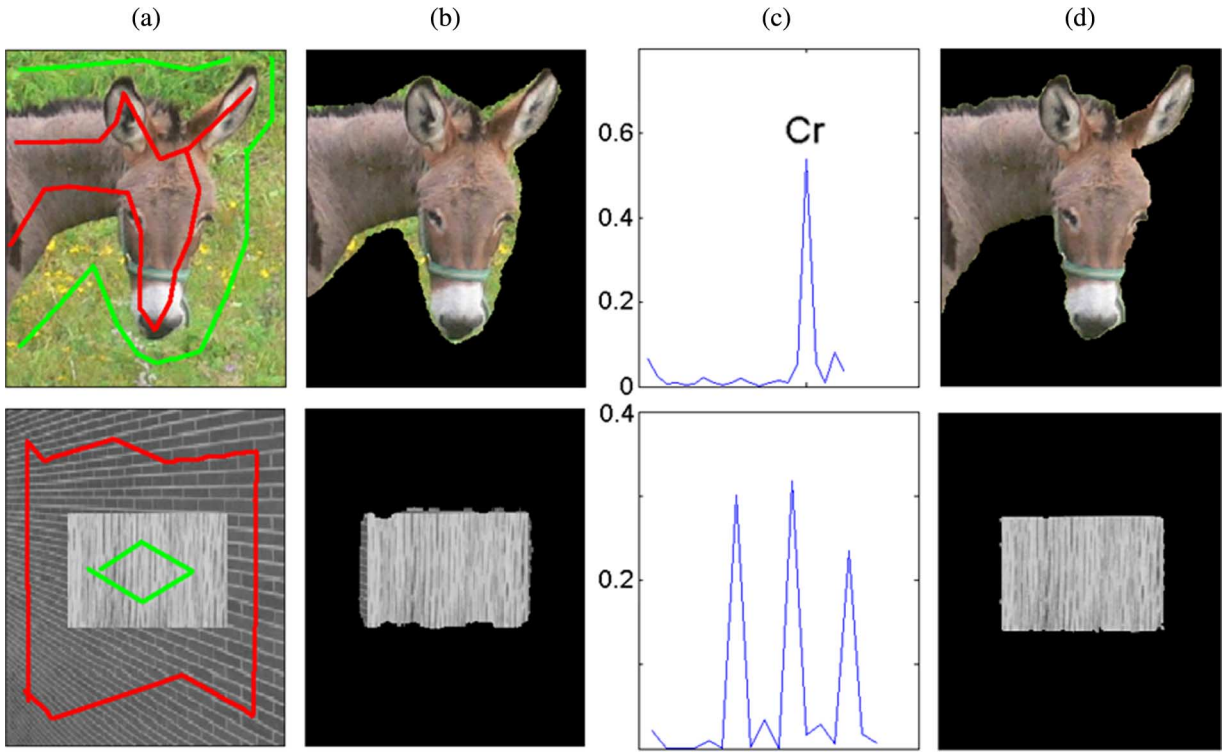

Fig. 3. Two examples of segmentation into two uniform regions, showing the importance of adaptive weights. (a) The user-scribbled image. (b) The segmented image using all equal weights, note the significant errors in the segmentation. (c) The automatically computed weights for each channel.The horizontal 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. First image: 281×391 pixels; second image: 200×200 pixels.

for the geodesic definition. 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

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

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 (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 for the region of interest labelled with l_i and high otherwise (this will lead to small weighted distances to l_i for the correct region and large for the rest). Then, we can efficiently compute the weighted distance maps d_i for $i \in \{1, \ldots, N_l\}$ [see (2)] 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.

A remark on computational complexity comes to place. 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 induced by this step, a different subset of filters can be used for each pre-established image class. For example, if the original color channels are sufficient for a given class, then there is no computational cost for this filtering stage. Note also that the filtering is done only once (e.g., while loading the data), while then the user can interactively segment the image. Following this filtering step, the proposed use of fast geodesic computations is optimal, since it is linear, [30], independently of the number of used channels since they are all combined into one metric. The weighted distances have to be computed for each user-provided scribble (this can be done in parallel for all scribbles), although they do not have to be computed for the entire image for every single scribble, e.g., distance computations do not cross scribbles and do not necessarily have to continue over pixels that have already received lower values from other scribbles; see [30] and [31] for additional details.

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 subset 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

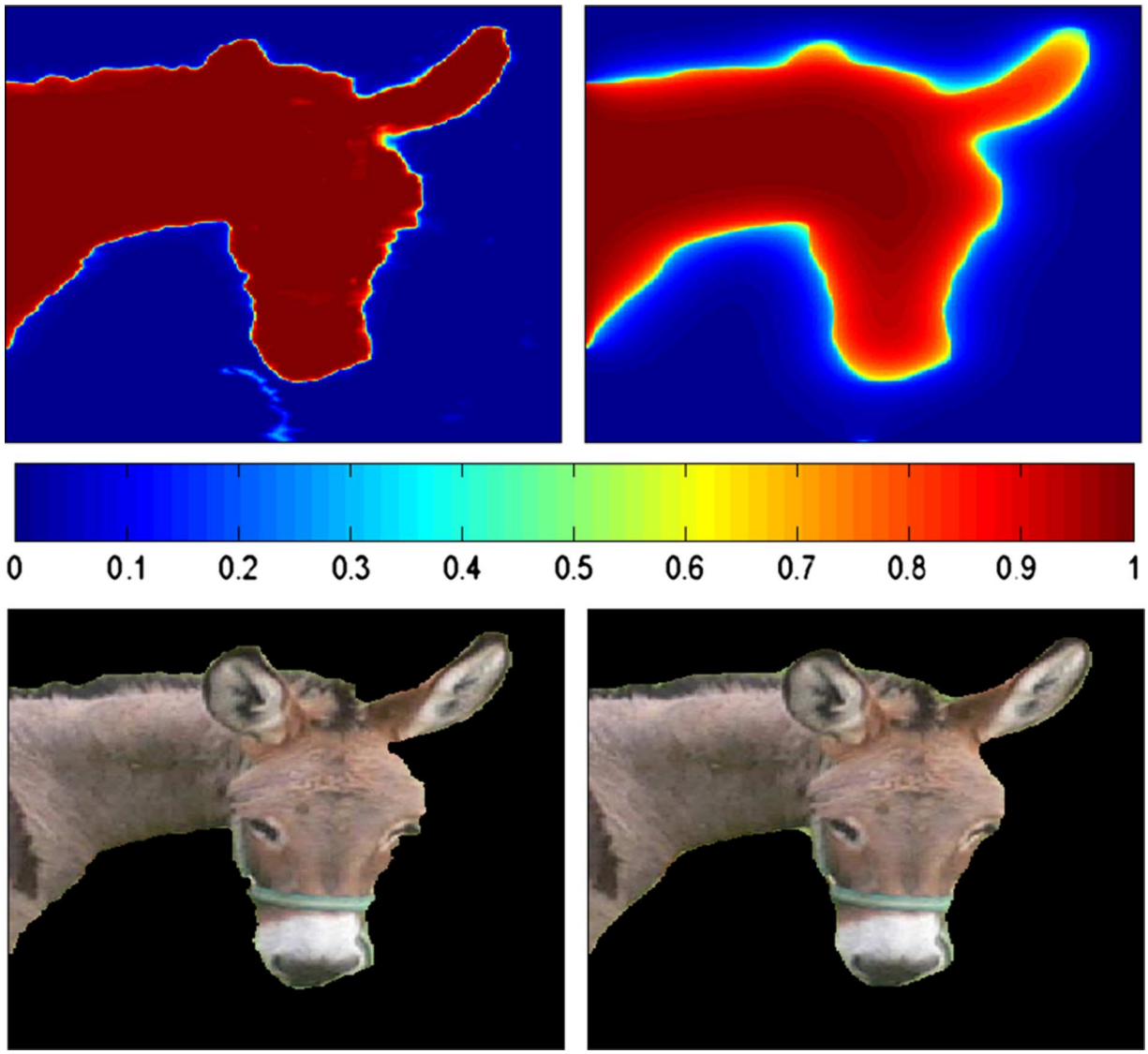

Fig. 4. Regularization effects on (top row) the probability distribution followed by (bottom row) its effects on the region boundary. Original images on the left; regularized ones on the right. Although the whole probability is regularized for illustration purposes here, only the region of border-line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

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 (four scales and four 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., [11] and [25]. The basic 2-D 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 (ω) and width (σ_x, σ_y) of the filter. These parameters are linked in order to optimize the coverage of the frequency spectrum. The filter response for an orientation θ and a frequency $a\omega$ is obtained by an appropriate rotation and dilatation of g

$$g'(x,y) = ag(X,Y)$$

where $X := a(x\cos\theta + y\sin\theta)$ and $Y := a(-x\sin\theta + y\cos\theta)$. The 16 filters are obtained using four scales (w = 1/2, 1/4, 1/8, 1/16) with four orientations ($\theta = 0, \pi/4, \pi/2, 3\pi/2$) for each scale. The idea is that, locally, these filters express the scale and orientation of a texture. This has been found in the literature to be fundamental for characterizing rich texture families. For details about the mathematical properties of the Gabor functions, we refer the interested reader to [15]. For speedup, for example, we could replace these filters by the steerable pyramid [26]. The

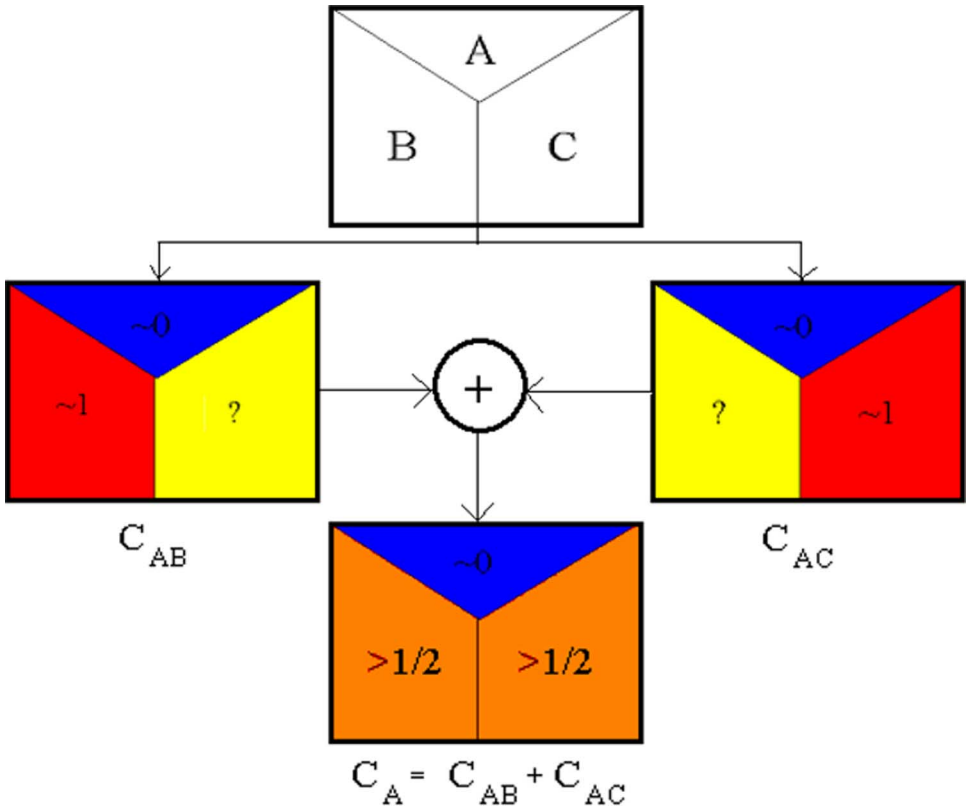

Fig. 5. Principle for the design of a weight function in the configuration of three to be segmented regions A, B, and C.

framework here developed is independent of the exact type of filters used to define the channels.

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 nonlinear transformation. Then, with an averaging operator, we smooth the variations. Therefore, if G_i is the response of a Gabor filter applied to Y (here assumed to have zero mean for simplicity), Ω the domain of the image, and $\Omega_{x,y}$ an $N \times N$ window around the pixel (x, y), we consider the channel F_i given by

$$\forall (x,y) \in \Omega : F_i(x,y) := \frac{1}{N^2} \int\int_{\Omega_{x,y}} \tanh\left(\alpha \frac{G_i(u,v)}{\sigma(G_i)}\right) dudv$$

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

One of the goals of this nonlinear transform is to make the different channels comparable, varying in the same interval and with similar standard deviations. The selection of $\alpha = 0.25$ corresponds to saturating values larger than $4\sigma(\cdot)$, which are rare in a Gaussian distribution. If we choose $\alpha \gg 0.25$, a lot of relevant information will be eliminated, while selecting $\alpha \ll 0.25$, the channel will vary in a very small interval, and we will loose information when quantizing. In conclusion, $\alpha = 0.25$ is a good compromise that also has been found to perform well in practice. Regarding the selection of N, we observed the results to be stable in the interval [3, 9], which are the values commonly used in the literature.

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 *uniform* regions, meaning that the regions can be discriminated by the set of computed channels. Let Ω_1 and Ω_2 be the set of prelabelled 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 [4] and [22]). Although other fitting functions might be more appropriate, we found this sufficient for the very good results here reported. 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 Fig. 2). Similarly, we compute $P^i_{2|1}$.

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

$$P_{1|2}(x) := \Pr(x \in l_1) = \sum_{i=1}^{N_c} w^i P^i_{1|2}(x) \tag{4}$$

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

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

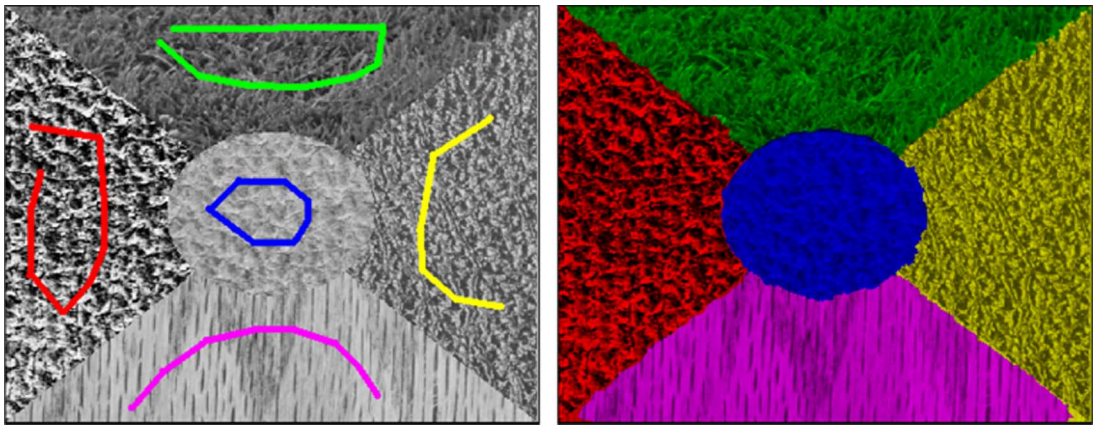

Fig. 6. Segmentation example with more than two regions. Image size: 512 × 512 pixels.

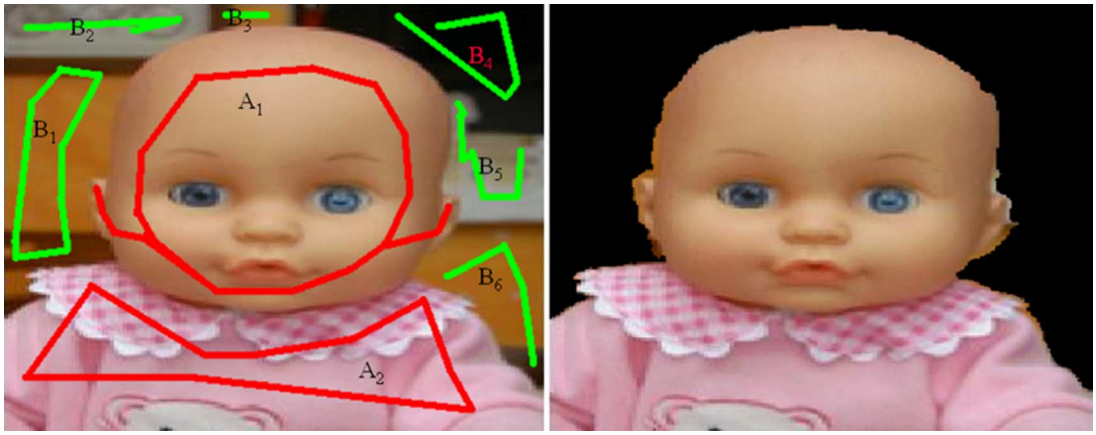

Fig. 7. Example of a segmentation with nonuniform regions. Green and red scribbles do not compete among themselves, only against each other. Image size: 253× 283 pixels.

Fig. 8. Example showing the importance of using all the channels F_i (color and Gabor filtering), with weights automatically learned, as here proposed. Following the original image with the user-provided scribbles, the result of our technique is presented in the middle, showing a much better segmentation than the one obtained while using only the color channels (right). Image size: 481 × 321.

W_2 is similarly obtained (see Fig. 3). In fact, we normalize each W_i by dividing by its standard deviation in order to make them comparable.

As in [4], 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. As an example, consider the case where there are $N_c \gg 1$ 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. We need then to find the relevant channels, relying on the user provided scribbles/labels. For this, 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.

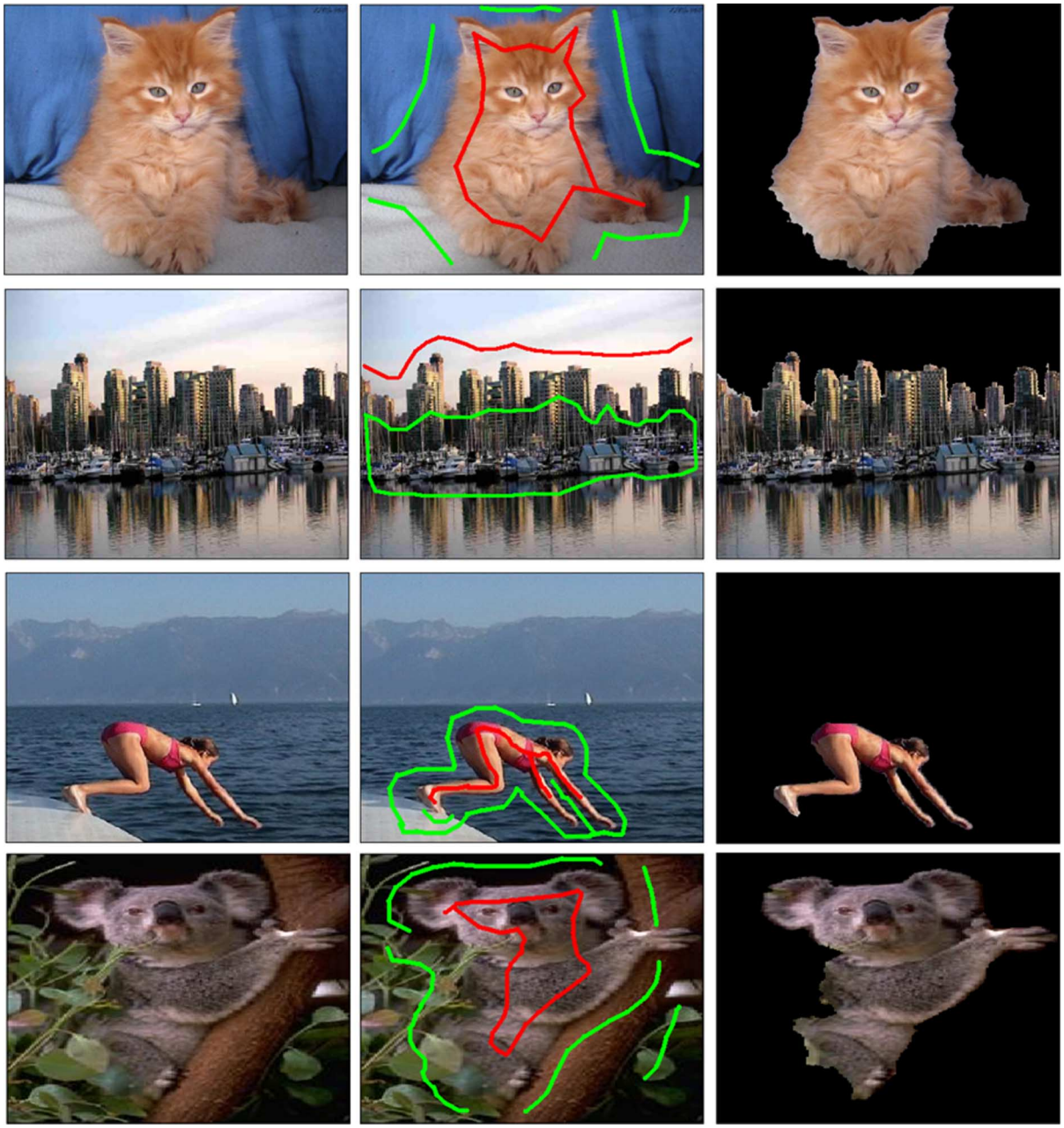

Fig. 9. Additional segmentation examples. In the image with the cat, the background is decomposed into two sublabels. The koala image also uses sublabels for the background. Cat image: 400 × 300 pixels; skyline image: 728 × 423 pixels; diver: 345 × 346 pixels; koala: 161 × 240 pixels.

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

$$\begin{aligned} P_i = &\Pr(s \in l_1)\Pr(s \to l_2 | s \in l_1)) \\ &+ \Pr(s \in l_2)\Pr(s \to l_1 | s \in l_2)) \\ P_i = &\frac{1}{2} \int\limits_{\{x: p_1^i(x) > p_2^i(x)\}} p_1^i(x)dx + \frac{1}{2} \int\limits_{\{x: p_2^i(x) > p_1^i(x)\}} p_2^i(x)dx. \end{aligned}$$

Then, we deduce that

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

This quantity has been shown by Dunn and Higgin [9] to be a good criteria for channel selection. From this, we deduce directly the weights for each one of the channels [to be used in (4)]

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

Fig. 3 shows that, depending on the type of image and the regions we want to segment, the weights are concentrated on

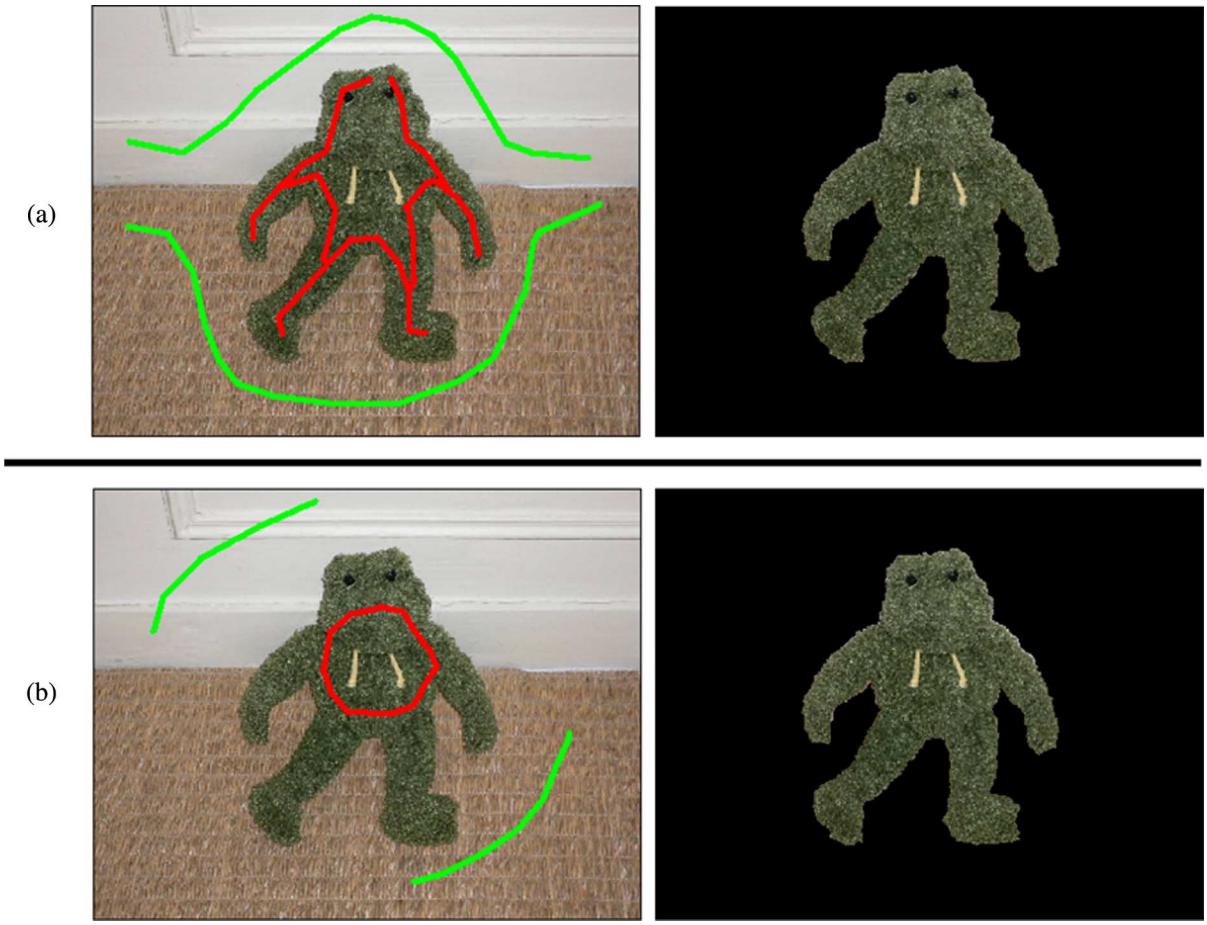

Fig. 10. (a), (b) Two segmentation results of the same image using different user scribbling. It shows the robustness of the algorithm, it is not necessary to carefully scribble in order to obtain very good results. Image size: 640 × 480 pixels.

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 (each pixel is assigned to the region corresponding to the scribble/label with shorter weighted distance). 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., [19], before hard assignment. This will, for example, regularize the contours (see Fig. 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. Note that the filtering is only done once, to obtain all the desired channels. The geodesic computations are also done once. What is done for all pairs of regions is the computation of the weighting of the channels, which are then combined as detailed below to produce a single weight (and thereby a single metric for a single geodesic computation from the provided scribbles). There is thereby no relevant increase in the computational complexity of the algorithm.

Let $\{l_1, \ldots, l_{N_l}\}$ be the set of labels and $W_{i|j}$ the weight function for the label l_i when competing only with l_j. $W_{i|j}$ is computed as in the previous section, as if there were only two labels, l_i and l_j. We want the global weight function of l_i to be very low inside its corresponding region and very high outside, and then we can define W_i as

$$W_i = \sum_{j \neq i} W_{i|j}. \tag{5}$$

This simple combination of the pairwise weights permits, for every region, to pick the best channel weights to discriminate

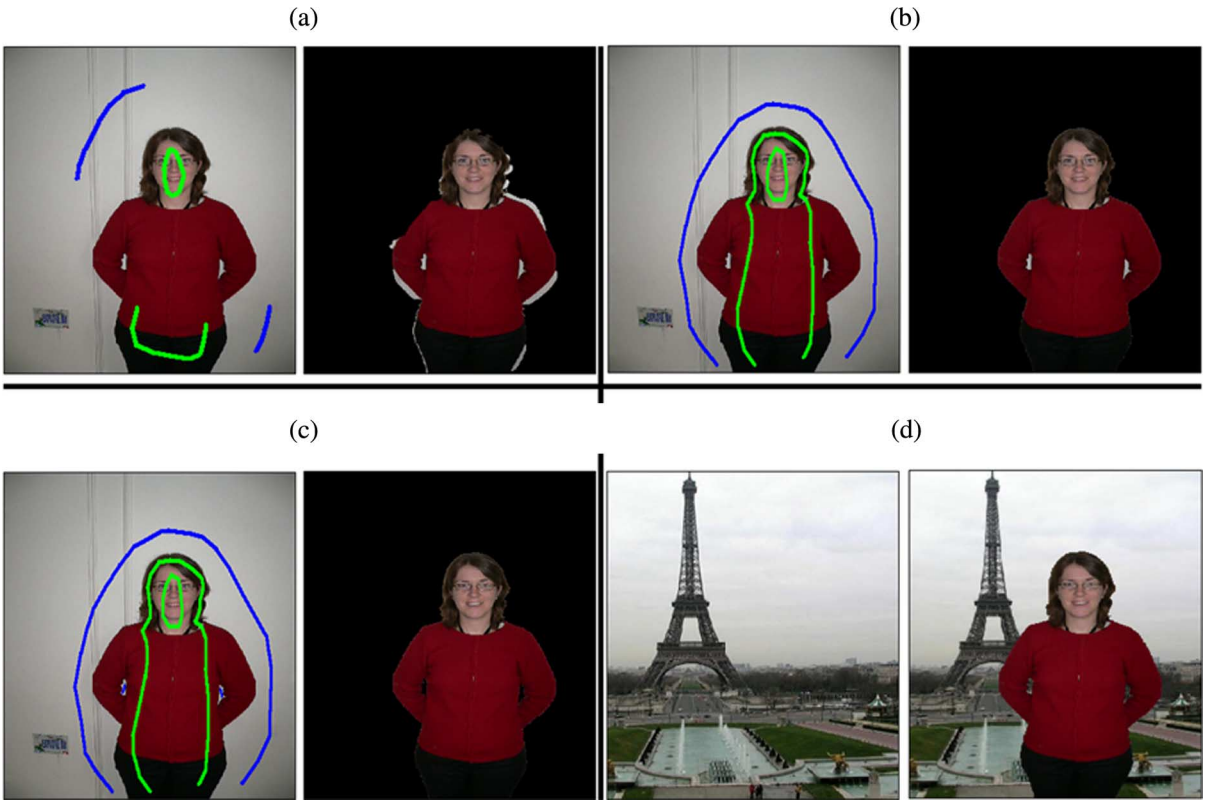

Fig. 11. Progressive segmentation. (a) The user starts with a few minor scribbles, obtaining only a partial desired segmentation. (b) The user adds scribbles, further improving the segmentation. (c) The user completes the segmentation by marking under the arms. (d) Example of a typical application of this type of foreground/background segmentation: image composition. Note that simple cut-and-paste has been used. Image size: 480 × 640 pixels.

between it and the given region i. Fig. 5 graphically shows how this method builds a good weight function. An example is presented in Fig. 6.

E. Nonuniform 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, [22]). Continuing with our philosophy of letting the user help, 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 subregions and to scribble the region using several components (see Fig. 7). Then, we consider each component as an independent labels (we call them sublabels), and we are back to the previous problem of multiple uniform regions, Section III-D. The only difference is that we do not need to make two components of the same label compete against each other, we do not care to discriminate between them. In other words, and considering as an example the separation between just two classes (foreground/background), while each user-marked label is propagated as described in Section III-D, all the labels that the user marked as background labels compete only with the labels the user marked as foreground labels.

Formally, let l_i^j be the component j of the label l_i. Then, W_i^j, its weight function, is given by

$$W_i^j = \sum_{k \neq i} \sum_{l} W_{l_i^j | l_k^l}. \tag{6}$$

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

IV. Additional Examples

The code for all the examples in this paper has been written in Matlab. We now present additional examples of the proposed framework for interactive natural image segmentation. First, in Fig. 8, we demonstrate once again the importance of using the stack of channels F_i, with automatically learned weights. Next, in Fig. 9, we present four very different and diverse images, showing the generality of our proposed framework. Fig. 10 exemplifies the robustness of the algorithm with respect to the positions of the user-provided 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. In Fig. 11, we first simulate the use of our

Fig. 12. Additional example of interactive progressive segmentation. First, (left image) the user roughly scribbles the image, and (second image) a part of the background is wrongly associated with the foreground. The user then adds some scribbles (third image) to correct the segmentation and gets (fourth image) the desired result. Image size: 481 × 321 pixels.

framework in a real interactive application, where the user progressively adds scribbles to achieve the desired segmentation result (see also [4] and [22]). Finally, in Fig. 11(d), we show the use of this result for image composition. The foreground from Fig. 11(c) has been cut and pasted on top of the background in Fig. 11(d). A simple cut-and-paste has been used, with no blending. For a real application, simple blending needs to be added; see, for example, [6], [22], and [29]. For this, the natural soft segmentation here obtained is very useful. An additional example of progressive segmentation is provided in Fig. 12.

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 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. Although the user can keep adding and removing scribbles until satisfied, 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. This can be achieved with models that go beyond simple Gaussians. 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. The fast classification of the image in order to reduce the number of filters that need to be tested might help to improve even further the computational complexity of the framework. In addition, metrics that take into account reflections, shadows, and other physical "artifacts" present in natural images will further improve the results.

The use of the framework here proposed for other disciplines such as medical imaging and geo-data should be pursued. The key challenge there is to define the appropriate weights, and thereby the appropriate bank to channels F_i, that better fit the given image modality. We are also extending this work to video, and results on this will be reported elsewhere.

References

[1] S. Aksoy, R. M. Haralick, F. A. Cheikh, and M. Gabbouj, "A weighted distance approach to relevance feedback," in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, vol. 4, pp. 812–815.

[2] B. Appleton and H. Talbot, "Globally minimal surfaces by continuous maximal flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 106–118, Jan. 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 Process.*, vol. 14, no. 9, pp. 1314–1323, Sep. 2005.

[4] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. Int. Conf. Computer Vision*, 2001, vol. I, pp. 105–112.

[5] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[6] P. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graph.*, vol. 2, pp. 217–236, 1983.

[7] V. Caselles, R. Kimmel, and G. Sapiro, "Geometric active contours for image segmentation," in *Hanbook of Image and Video Processing*, A. Bovik, Ed. New York: Academic, 2005.

[8] L. Cohen and R. Kimmel, "Global minimum for active contours models: A minimal path approach," *Int. J. Comput. Vis.*, vol. 24, pp. 57–78, 1997.

[9] D. Dunn and W. E. Higgin, "Optimal gabor filters for texture segmentation," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 947–964, Jul. 1995.

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

[11] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognit.*, vol. 24, pp. 1167–1186, 1991.

[12] W. E. L. Grimson, R. Keriven, L. M. Lorigo, O. Faugeras, and R. Kikinis, "Segmentation of bone in clinical knee mri using texture-based geodesic active contours," in *Proc. Medical Image Computing and Computer-Assisted Intervention*, pp. 1195–1204.

[13] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[14] Y. Li, J. Sun, C. K. Tang, and H. Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, pp. 303–308, 2004.

[15] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

[16] E. N. Mortensen and W. Barrett, "Intelligent scissors for image composition," *ACM Trans. Graph.*, pp. 191–198, 1995.

[17] K. Mueller, R. Yagel, and J. F. Cornhill, "The weighted distance scheme: A globally optimizing projection ordering method for art," *Trans. Med. Imag.*, vol. 16, no. 2, pp. 223–230, 1997.

[18] N. Paragios and R. Deriche, "Geodesic active regions for supervised texture segmentation," in *Proc. Int. Conf. Computer Vision*, Corfu, Greece, Sep. 1999.

[19] A. Pardo and G. Sapiro, "Vector probability diffusion," *IEEE Signal Process. Lett.*, vol. 8, no. 4, pp. 106–109, Apr. 2001.

[20] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 2, pp. 313–318, 2003.

[21] Y. Qu, T. Wong, and P. A. Heng, "Manga colorization," *ACM Trans. Graph.*, vol. 25, pp. 1214–1220, 2006.
[22] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, to be published.
[23] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
[24] G. Sapiro, "Color snakes," *Comput. Vis. Image Understand.*, vol. 68, no. 2, pp. 247–253, 1997.
[25] T. Shioyama, H. Wu, and S. Mitani, "Segmentation and object detection with gabor filters and cumulative histograms," in *Proc. Int. Conf. Pattern Recognition*, 2000, vol. 1, pp. 704–707.
[26] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multi-scale transforms," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, Mar. 1992.
[27] I.-M. Sintorn and G. Borgefors, "Weighted distance transforms for volume images digitized in elongated voxel grids," *Pattern Recognit. Lett.*, vol. 25, pp. 571–580, 2004.
[28] F. Wang, J. Wang, C. Zhang, and H. C. Shen, "Semi-supervised classification using linear neighborhood propagation," in *Proc. IEEE CVPR*, New York, 2006, pp. 160–167.
[29] J. Wang and M. F. Cohen, "An iterative optimization approach for unified image segmentation and matting," in *Proc. Int. Conf. Computer Vision*, Beijing, China, 2005, pp. 936–943.
[30] L. Yatziv, A. Bartesaghi, and G. Sapiro, "O(n) implementation of the fast marching algorithm," *J. Comput. Phys.*, vol. 212, pp. 393–399, 2006.
[31] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, May 2006.
[32] R. Zass and A. Shashua, "A unifying approach to hard and probabilistic clustering," presented at the Int. Conf. Computer Vision, Beijing, China, Oct. 2005.

Alexis Protiere, photograph and biography not available at the time of publication.

Guillermo Sapiro (M'94) was born in Montevideo, Uruguay, on April 3, 1966. He received the B.Sc. (summa cum laude), M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, The Technion–Israel Institute of Technology, Haifa, in 1989, 1991, and 1993, respectively.

After postdoctoral research at the Massachusetts Institute of Techology, Cambridge, he became a Member of Technical Staff at the research facilities of HP Labs, Palo Alto, CA. He is currently with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, where he holds the position of Distinguished McKnight University Professor. He works on differential geometry and geometric partial differential equations, both in theory and applications in computer vision, computer graphics, medical imaging, and image analysis. He has authored and coauthored numerous papers in these areas and has written the book *Geometric Partial Differential Equations and Image Processing* (Cambridge University Press, 2001).

Dr. Sapiro is a member of SIAM. He has coedited special issues of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the *Journal of Visual Communication and Image Representation*. He was awarded the Gutwirth Scholarship for Special Excellence in Graduate Studies in 1991, the Ollendorff Fellowship for Excellence in Vision and Image Understanding Work in 1992, the Rothschild Fellowship for Post-Doctoral Studies in 1993, the Office of Naval Research Young Investigator Award in 1998, the Presidential Early Career Awards for Scientist and Engineers (PECASE) in 1998, and the National Science Foundation Career Award in 1999.