HAL
open science

Combination of piecewise-geodesic curves for interactive image segmentation

Julien Mille, Sébastien Bougleux, Laurent D. Cohen

To cite this version:

Julien Mille, Sébastien Bougleux, Laurent D. Cohen. Combination of piecewise-geodesic curves for interactive image segmentation. 8th international Conference on Curves and Surfaces, Jun 2014, paris, France. hal-01100406

HAL Id: hal-01100406

https://hal.science/hal-01100406v1

Submitted on 27 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Combination of piecewise-geodesic paths for interactive segmentation

Julien Mille · Sébastien Bougleux · Laurent D. Cohen

Received: date / Accepted: date

Abstract Minimum cost paths have been extensively studied theoretical tools for interactive image segmentation. The existing geodesically linked active contour (GLAC) model, which basically consists of a set of vertices connected by paths of minimal cost, blends the benefits of minimal paths and region-based active contours. This results in a closed piecewise-smooth curve, over which an edge or region energy functional can be formulated. As an important shortcoming, the GLAC in its initial formulation does not guarantee the curve to be simple, consistent with respect to the purpose of segmentation. In this paper, we draw our inspiration from the GLAC and other boundary-based interactive segmentation algorithms, in the sense that we aim to extract a contour given a set of user-provided points, by connecting these points using paths. The key idea is to select a combination among a set of possible paths, such that the resulting structure represents a relevant closed curve. Instead of considering minimal paths only, we switch to a more general formulation, which we refer to as *admissible* paths. These basically correspond to the roads travelling along the bottom of distinct valleys between given endpoints. We introduce a novel term to favor the simplicity of the generated contour, as well as a local search method to choose the best combination among possible paths.

J. Mille
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
F-69622, Villeurbanne, France
E-mail: julien.mille@liris.cnrs.fr

S. Bougleux
Université de Caen-Basse Normandie, GREYC, UMR6072
F-14050, Caen, France
E-mail: sebastien.bougleux@unicaen.fr

L. Cohen
Université Paris-Dauphine, CEREMADE, UMR7534
F-75016, Paris, France
E-mail: cohen@ceremade.dauphine.fr

1 Introduction

1.1 Related work

Methods addressing the problem of two-phase segmentation based on energy minimization techniques provide a solid mathematical background, and have proven to find suitable solutions in many practical situations. Among them, many methods are interactive, in the sense that they require user-defined information, provided prior to the optimization process or during the different steps of segmentation. For instance, in the case of local optimization methods, such as active contours or level sets driven by gradient descent of the energy [11], the interaction serves to provide an initialization that one hopes to be relatively close to the final solution. We especially focus on prior interactions, which may be roughly classified into two types. They may be region-based, in which case the user quickly draws scribbles that will be used as region seeds. Particular graph cut-based methods [6, 43], random walks [20] or the lazy snapping algorithm [27] fall into this category. Level sets also lend themselves to region-based user interactions, which were used as hard constraints [18] or soft contraints to guide the segmentation [4, 14, 22]. Interactions can also be boundary-based, in which case the user draws a curve, e.g for parametric active contours [21, 23], or provides a set of landmark points that will be linked by curves satisfying, among others, smoothness conditions and fitting to the image data. Significant examples include

the intelligent scissors (or *live-wire*) [34] and their *on-the-fly* extension [17], or the riverbed algorithm [33] based on the image foresting transform [16].

Interactive segmentation methods constrained by user-provided landmark points very often rely on minimum cost paths, which may be expressed either within a discrete or a continuous framework. In the discrete setting, the image domain is structured as a graph following the grid connectivity. Depending on the model, nodes can be pixels [34] or pixel vertices [17] - also referred to as *pointels*, for point element - so that arcs are edges between pairs of pixels or pixel boundaries - *linels*, for line element - respectively. In any case, relevant image contours are sought as paths, i.e. ordered loopless sequences of adjacent nodes, minimizing some cost functional. The most natural case is the additive cost function, allowing boundaries to be computed thanks to Dijkstra's shortest path algorithm, as in [17, 34], but more general cost functions have also been studied [33].

In the continuous setting, many boundary-based interactive segmentation methods fall into the broad domain of curve optimization. Optimal cost boundaries were addressed under the framework of minimal paths introduced by Cohen and Kimmel [12]. They proposed to find a global minimum of the geodesic active contour functional [9], provided that one or two points of the target object boundary are initially supplied by the user. The resulting geodesic curve, which can be respectively closed or open, is efficiently computed through the solution of the Eikonal equation, numerically solved using the Fast Marching method [39, 41]. Another significant contribution to globally minimal curves was made in [2] with a different approach relying on continuous maximal flows. Unlike graph-based methods, which are discrete by essence, the minimal path method is a discretization of a continuous problem, and hence does not suffer from metrication errors [12]. To some extent, in a different trend than convex relaxation and dual minimization schemes [10, 7], the minimal path approach addresses the issue of sensitivity to local minima of "old-fashioned" edge-based active contours and level sets [9, 23, 28], which are driven by gradient descent of the Euler-Lagrange equation derived from the energy. Extensions of the minimal path algorithm include implementations dedicated to interactive segmentation using multiple endpoints [19] and more recently to complex curves with arbitrary topologies [24]. Whether discrete or continuous, optimality is a major advantage of path-based methods, since in most cases, a minimum cost path can be efficiently found as the global solution of the corresponding minimization problem. In order to fit paths to image contours, the cost function is most often chosen as a decreasing function of the image gradient magnitude. The reader will notice that *path* denotes both an ordered sequence of connected nodes in a graph or an open continuous regular curve between two endpoints.

In their basic formulations, minimum cost path algorithms extract open curves between two fixed endpoints. For practical segmentation, i.e. extraction of closed contours, one needs to modify and/or constrain the original frameworks. On the one hand, a class of methods require minimal interaction, i.e. a single user-provided point on the object boundary. For instance, the construction of a closed curve from a single point was considered in [12]. Thanks to the Fast Marching algorithm, the minimal action map, or geodesic distance map, is propagated from the origin point. The first point where the front collides with itself, namely a *saddle point*, is detected and taken as the origin of two opposite backpropagations, both sides apart from the colliding location, so that the two resulting paths can be assembled in order to form a closed contour. Benmansour and Cohen [5] proposed to recursively detect intermediate keypoints during the propagation, with respect to a stopping criterion based on the Euclidean length of minimal paths. Each detected keypoint is taken as the origin of a new propagation, until the first point is reached again, thus making a closed contour. On the other hand, closed contours may be constructed by linking together n ordered user-provided landmark points with minimum cost paths. The resulting contour can thus be referred to as *piecewise-geodesic*. In the same way that at least three vertices are necessary to define a polygon, it is naturally assumed that $n \geq 3$. Significant examples include the Riverbed algorithm by Miranda *et al* [33].

Whether based on a single or several user-provided point(s), the previous approaches were not designed to consider the regions inside and outside the contour in the optimization process. Indeed, they fall into the "edge-based only" category of segmentation methods, the edge-fitting criterion being indirectly contained in the definition of minimum cost paths. However, adding region homogeneity terms into the energy functional has proven to go beyond limitations of "edge-based only" energies, and may lead to more accurate segmentation, e.g. in case of images exhibiting noise, lack of sharp contours or cluttered background. Global statistical data computed over the entire regions is a well established technique to improve the behavior of variational segmentation methods. Widely used region homogeneity terms are particular instances of the Mumford-Shah functional [35] - like the piecewise

constant intensity model by Chan and Vese [11], the piecewise smooth model [8], localized versions of these models [26] or extensions to higher level features such as texture descriptors [36, 38, 42] - or rely on color probability distributions - like the maximum-entropy energy [25] or the maximum discrepancy between inner and outer color distributions [29]. Region-based variational methods usually perform an iterative evolution of the contour - on a parametric curve or a level set function - in order to minimize the energy. However, if one desires to integrate region terms into segmentation methods relying on minimal paths, one quickly gets faced with a "chicken and egg" dilemna: inner and outer regions are well defined only when the contour is closed. Thus, region homogeneity can be expressed and optimized, even locally, only when all minimal paths are available. However, if one wishes to compute minimal paths exactly and in agreement to some region homogeneity criterion, the regions should already exist. Hence, minimal paths and region-based energies cannot be held together in a classical variational framework.

The geodesically linked active contour (GLAC) model was introduced by Mille and Cohen [32] as a heuristic approach for combining the benefits of boundaries defined by minimal paths and the use of region terms in the energy functional. It basically consists in evolving a set of n variable points in discrete local search windows so that, at every step of the evolution, the piecewise-geodesic curve linking the vertices together minimizes a combination of edge and region terms. Each point is iteratively moved to the location for which the corresponding piecewise-geodesic curve locally minimizes the energy. This model may be viewed as a classical active contour, with control points remaining linked by minimal paths. Generally, a requirement on the recovered object boundary is that it should be simple, i.e. without double point. While the GLAC is relatively robust to local minima, it can fail to construct a simple curve, as several minimal paths emanating from different points may partially overlap. This can occur at the initialization step, or during the evolution, particularly when few control points are used. This heavily diminishes the benefits of the approach, as the small amount of required input points is supposed to be one of the desirable features of the GLAC. Building upon this model, the purpose of the current work is to define a closed boundary, assembled from several piecewise-geodesic curves derived from the minimal path approach, which is able to satisfy a region homogeneity term while avoiding self-overlap and self-intersection issues.

1.2 Overview of the proposed method

When linking two given endpoints with a path, the key idea is that we should not limit ourselves to the minimal one. We propose to generate several relevant paths between landmark points and to select the combination of paths generating the best closed contour. To this end, we introduce an energy functional, combining contour and region terms with a novel term favoring the simplicity of the curve, penalizing self-overlap and self-intersections. Our contribution is threefold:

- *Definition and construction of admissible paths*: a path between two endpoints may be optimal according to the edge-based cost function, but might turn out irrelevant when assembled with paths emanating from other endpoints to build a closed contour. Given two endpoints, other good paths can be built in addition to the shortest one. We will define what "potentially good" paths are, and how they are generated.
- *Design of the simplicity energy*: we should define how to measure the consistency of a combination of paths with respect to the segmentation purpose. This combination should be located on actual image edges, split the image into regions matching some homogeneity criterion, and be simple from a geometrical point of view. In other words, we should penalize contours that self-overlap or intersect.
- *Heuristic optimization*: exhaustive enumeration of all possible combination of paths leads to an exponential complexity. We define a greedy search method to determine a relevant combination of paths in reasonable time, following the natural evolution of an active contour.

First, the concepts which the proposed approach rely on are presented. The minimal path approach and the GLAC are detailed. Secondly, we focus on the study of valleys on the geodesic distance map, leading to the definition of admissible paths. Then, the various energy terms are formulated. Among others, we present the novel simplicity term. The local search method to find a good enough combination of paths is explained, followed by remarks on discretization and implementation. Finally, experimental results demonstrate the ability of our model to perform accurate object segmentation on a well-established dataset. We show the influence of the various energy terms, and provide comparisons against related existing algorithms. The current document is an extended version of a previous shorter paper by the same authors [31]. Appendix B, which describes implementation details, is provided in a supplemental document available at `http://liris.cnrs.fr/~jmille`.

2 Background

2.1 Minimal paths

To extract structures in a given image $I:\mathcal{D}\to\mathbb{R}$, Cohen and Kimmel [12] proposed to find curves of minimal length according to a heterogeneous isotropic metric defined from a potential $P:\mathcal{D}\to\mathbb{R}^+$. This potential, which is chosen to take lower values on the structure of interest, allows to measure the length of piecewise smooth curves $\mathcal{C}:[0,1]\to\mathcal{D}$ as follows [1]:

$$L[\mathcal{C}] = \int_0^1 P(\mathcal{C}(u))\,\|\mathcal{C}'(u)\|\,\mathrm{d}u. \tag{1}$$

In the context of contour extraction, curves should be located along edges. The potential is thus defined as $P(\boldsymbol{x}) = g(\boldsymbol{x}) + w$, where $g:\mathcal{D}\to\mathbb{R}^+$ is a decreasing function of the gradient magnitude of the image at some scale s, usually obtained by convolution with the gradient of a Gaussian kernel K_s with standard deviation s and $w\in\mathbb{R}^{*+}$ is the weight of the Euclidean distance component, acting as a regularizing constant. A commonly chosen edge detecting function is:

$$g(\boldsymbol{x}) = \frac{1}{1+\|\nabla K_s * I(\boldsymbol{x})\|}. \tag{2}$$

The target image structure is then extracted by finding a path of minimal length among all paths connecting two given points $\boldsymbol{a}$ and $\boldsymbol{b}$ located on the structure

$$\gamma_{\boldsymbol{a},\boldsymbol{b}} = \underset{\mathcal{C}\subset\mathcal{D}}{\operatorname{argmin}}\,\{L[\mathcal{C}]\} \quad \text{s.t.} \begin{cases}\mathcal{C}(0)=\boldsymbol{a}\\ \mathcal{C}(1)=\boldsymbol{b}.\end{cases} \tag{3}$$

Such a globally defined minimal path is called a geodesic. The solution of minimization problem (3) can be obtained by considering the geodesic distance map, also referred to as the *minimal action map*, $U_{\boldsymbol{a}}:\mathcal{D}\to\mathbb{R}^+$ which assigns, to each point $\boldsymbol{x}\in\mathcal{D}$, the length of the minimal path connecting $\boldsymbol{x}$ to a given point $\boldsymbol{a}\in\mathcal{D}$:

$$U_{\boldsymbol{a}}(\boldsymbol{x}) = \inf_{\mathcal{C}}\,\{L[\mathcal{C}]\} \quad \text{s.t.} \begin{cases}\mathcal{C}(0)=\boldsymbol{a},\\ \mathcal{C}(1)=\boldsymbol{x}.\end{cases} \tag{4}$$

This map is the unique viscosity solution of the Eikonal equation

$$\begin{cases}\|\nabla U_{\boldsymbol{a}}(\boldsymbol{x})\| = P(\boldsymbol{x}), \quad \forall \boldsymbol{x}\in\mathcal{D}\setminus\{\boldsymbol{a}\},\\ U_{\boldsymbol{a}}(\boldsymbol{a}) = 0,\end{cases} \tag{5}$$

see for instance [13]. This allows to replace optimization problem (4) by a partial differential equation. Its discrete version, on a cartesian grid, can be efficiently solved by the Fast Marching (FM) method in $O(N\log N)$ operations, where N is the number of grid points [41, 39, 40]. Once the distance map has been numerically computed, the minimal path from any point $\boldsymbol{x}$ of $\mathcal{D}$ to $\boldsymbol{a}$ can be extracted by a gradient descent on $U_{\boldsymbol{a}}$,

$$\begin{cases}{\gamma_{\boldsymbol{x},\boldsymbol{a}}}'(u) = -\dfrac{\nabla U_{\boldsymbol{a}}(\gamma_{\boldsymbol{x},\boldsymbol{a}}(u))}{\|\nabla U_{\boldsymbol{a}}(\gamma_{\boldsymbol{x},\boldsymbol{a}}(u))\|},\\ \gamma_{\boldsymbol{x},\boldsymbol{a}}(0) = \boldsymbol{x},\end{cases} \tag{6}$$

where $'$ is the derivative with respect to u. This corresponds to a back-propagation from $\boldsymbol{x}$ to $\boldsymbol{a}$. The potential being isotropic, and thus independent from the direction, $\gamma_{\boldsymbol{a},\boldsymbol{x}}$ can be obtained by simply reversing $\gamma_{\boldsymbol{x},\boldsymbol{a}}$. In practice, since the FM is a monotonically-advancing front propagation method, finding the minimal path between two points does not require to compute the distance on the whole domain $\mathcal{D}$. Starting from one point, the FM can be stopped when the second point is reached, ensuring that the minimal path can be extracted with (6).

The minimal path approach is not restricted to extract an open curve, provided its endpoints. In particular, in the context of object extraction, it is able to find a closed curve, given only one point on the target object boundary. The closed curve is obtained by detecting a saddle point of the distance map and then by performing two back-propagations, in opposite directions, starting from this saddle point [12]. At this stage, we do not provide a precise mathematical definition of saddle points yet. As they are used in the current proposed approach, a formal definition is given in Section 3. Whether the curve is closed or open, the minimal path approach can fail to extract the desired curve. As depicted in Fig. 1(d), some portions of the minimal path do not follow the desired curve. This happens for instance when P is too noisy or not contrasted enough, when the length of the target curve is too important, or when the regularization constant w is too high. This undesirable behaviour hides a sampling problem, that is one or two points may be insufficient to capture the whole desired curve.

2.2 The geodesically linked active contour model

In order to extract closed contours using minimal paths, Mille and Cohen [32] proposed the geodesically linked active contour (GLAC) model, represented by a closed piecewise-smooth curve, allowing initialization inside the object or around the object boundary. The optimal contour is searched for among a space of piecewise-defined curves. These ones are generated by concatenating geodesic paths, i.e. joining them end-to-end, which are built from a set of vertices $\mathcal{V} = \{\mathbf{v}_i\}_{1\le i\le n}$,

[1] Note that, in the entire paper, curves will be assumed to be defined over the normalized range $[0,1]$

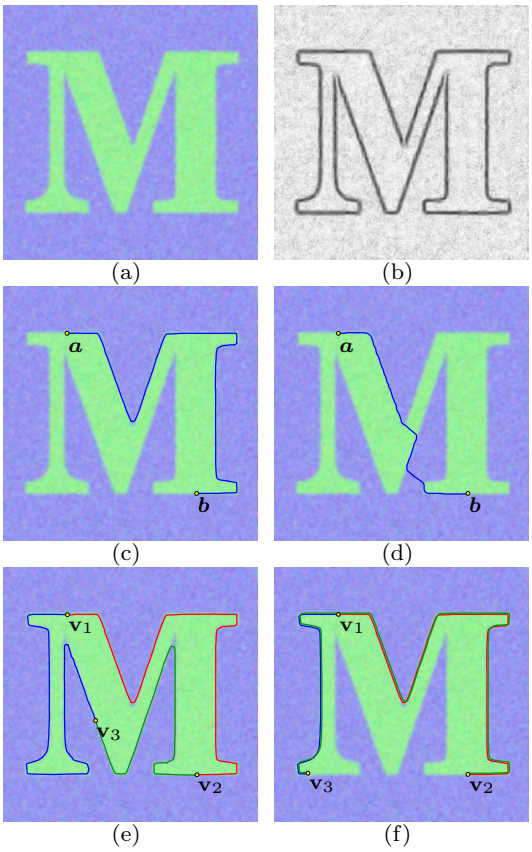


Fig. 1 (a) Input image. (b) Potential P. (c) With a sufficiently low regularization weight w, the geodesic between two given points follows the object contour. (d) Due to an excessive regularization weight w, the geodesic makes an undesirable shortcut. (e) Relevant piecewise-geodesic curve with evenly spaced vertices. (f) Undesirable overlapping with unevenly spaced vertices

with $n \geq 3$. Let Γ be the closed contour resulting from the assembly of geodesic paths $\gamma_{\mathbf{v}_i,\mathbf{v}_{i+1}}$. It is important to keep in mind that a concatenation of geodesics is not a geodesic itself, thus it is relevant to refer to Γ as *piecewise-geodesic*.

The GLAC model implements an evolution method which enables to move the vertices so as to minimize the energy. Indeed, the authors handled the case in which vertices could be located either outside or inside the object boundary. Since we focus on interactive segmentation with a set of fixed landmark points, we believe this deformation step to be beyond the scope of this paper and will not describe it. While the GLAC allows to blend the benefits of minimal paths and region-based terms, it turns out to have a significant drawback, as its initial state is not necessarily a simple closed curve. As depicted in Fig. 1(f), this can occur when the initial vertices are unevenly distributed around the target boundary. In this case, geodesics gather on particular sides of the target boundary, as $\gamma_{\mathbf{v}_2,\mathbf{v}_3}$ takes a way opposed to the expected one, and overlaps $\gamma_{\mathbf{v}_1,\mathbf{v}_2}$ and $\gamma_{\mathbf{v}_3,\mathbf{v}_1}$. The reason is that each geodesic is generated independently of the others, such that the obtained piecewise-geodesic curve does not depend on the visiting order of pairs of adjacent vertices. This undesirable phenomenon may occur either as soon as the GLAC is initialized, or after several evolution steps on a previously well initialized contour.

As in Section 2.1, this problem can be seen as a sampling one. Intuitively, one could think of imposing evenly spaced vertices, as depicted in Fig. 1(e), or adding vertices near the parts of the target boundary which are not covered by the piecewise geodesic curve. In the considered context, such sampling criteria are difficult to express, since the target boundary is unknown and applications usually need minimal user interaction. Otherwise, one could think of imposing hard constraints on the overlap between paths or penalizing paths enclosing a region with excessively small area, but the independent construction of paths, which allows parallel implementation, prevents such constraints to be implemented.

2.3 The geodesically linked active contour model with multiple disjoint paths

To overcome the drawbacks of the GLAC, the authors studied in [30] a more relevant contour construction preserving the advantages of piecewise geodesic curves. Assuming that several possible paths linking successive vertices are available, the key idea consists in selecting the paths generating the most relevant boundary curve. For each pair of successive vertices $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$, a set $\mathcal{A}_i$ of K *short* paths is considered:

$$\mathcal{A}_i = \{\gamma_{i,j}\}_{1 \leq j \leq K}.$$

Paths in $\mathcal{A}_i$ are disjoint and sorted by cost in ascending order, so that $\gamma_{i,1}$ actually corresponds to the minimal path between $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ whereas the remaining curves $\gamma_{i,j}$, $2 \leq j \leq K$, are only short paths of increasing cost. They are built by successive deletion of already existing paths from the potential map, hence imposing paths to be disjoint. Curve $\gamma_{i,1}$ is the minimal path between $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ in the space endowed by the initial potential $P_1 = P$. Once the minimal path $\gamma_{i,1}$ has been computed, the second admissible path $\gamma_{i,2}$ is sought under the constraint that it should not pass through points belonging to $\gamma_{i,1}$. Hence, $\gamma_{i,2}$ is not a geodesic in the space induced by potential P, but in the space induced by a modified potential P_2. The deletion of $\gamma_{i,1}$ in the modified potential map is achieved by setting the

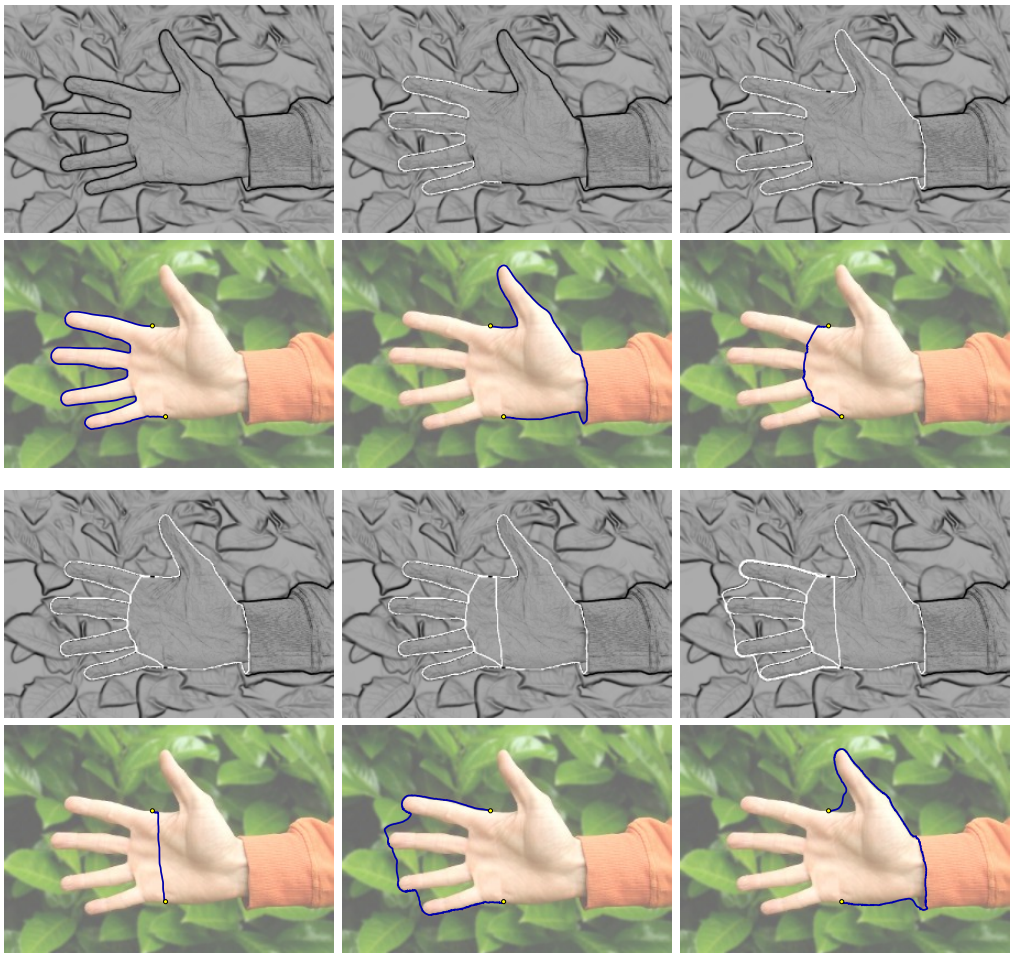


Fig. 2 Generation of $K = 6$ disjoint paths for a given pair of endpoints. (Rows 1 and 3) Successive potential maps P_j, where the black-to-gray scale corresponds to range $[0, 1]$ and white corresponds to $+\infty$. (Rows 2 and 4) Corresponding paths $\gamma_{\cdot,j}$, with $j = 1...6$.

potential to $+\infty$ at all points of the geodesic. Extending this principle to the construction of the jth admissible path $\gamma_{i,j}$, a recursive definition of potential functions can be written as:

$$P_j(\boldsymbol{x}) = \begin{cases} +\infty & \text{if } \boldsymbol{x} \in \gamma_{i,j-1} \\ P_{j-1}(\boldsymbol{x}) & \text{otherwise.} \end{cases} \tag{7}$$

This approach should not be mistaken for the so-called *K shortest paths* problem [45, 15], which, in its basic formulation, does not impose paths to be disjoint. In the present case, the non-overlap constraint simplifies the generation of several paths between a pair of vertices, as they can be found by running several instances of the shortest path algorithm, after removal of vertices and incident edges belonging to already found paths. An example of a set of successive disjoint paths $\gamma_{\cdot,j}$ and corresponding potential maps P_j is depicted in Fig. 2. The computation of a closed contour Γ consists in selecting one path out of each set $\mathcal{A}_i$, such that the contour resulting from the concatenation of selected paths minimizes an energy, made up of an edge-fitting and region homogeneity terms as well as a term penalizing self-overlap of the curve. In [30], the best combination of paths was determined by an exhaustive search among all possible combinations.

2.4 Limitations of the previous models

Our contribution partially relies on the GLAC [32], as we aim to build a piecewise-regular closed curve given a set of input vertices, while reusing the idea of generating several relevant paths for each pair of vertices and selecting the best combination of paths, as in [30]. The improvements brought by the current work are justified in a first time by highlighting several shortcomings of [30], particularly the generation of disjoint paths. When computing K disjoint paths, preventing each path to pass through already visited points arbitrarily removes sections of contours that could be taken

by other relevant paths. Moreover, in case of unsharp contours, low potential areas may get undesirably thick, so that two paths may agglomerate, i.e. take the same contour sections, the second path being most often an offset counterpart of the first one. For instance, this is the case for the 2nd and 6th of Fig. 2, respectively. This behavior, referred to as the *path agglomeration* issue, is further discussed in Section 3.3 once the proposed construction of path is presented.

Moreover, imposing a constant number K of admissible paths per pair of vertices may be unsuitable in many situations, and raises the issue of choosing K appropriately. When the object and background are complex between vertices $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$, many relevant paths may be found and excessively small values of K would restrict their generation. Conversely, when the object and background are simple and well separated from each other, there may be a single obvious path and one would like to avoid computing supplemental paths which would turn out useless for the generation of the final contour.

Finally, we should point out the shortcomings related to the computational cost. Since the potential map is modified after the construction of one path, the action map should be propagated again so that the next path can be extracted. This implies running K instances of the Fast Marching algorithm. As regards the computation of the combination minimizing the energy, exhaustive search, which has an exponential complexity $O(K^n)$, limits the practical usability of the approach. For obvious reasons, search methods with lower algorithmic complexity, even heuristical approaches, are preferable.

3 Generation of admissible paths

As in the GLAC, let us consider a set of vertices $\mathcal{V} = \{\mathbf{v}_i\}_{1\leq i\leq n}$, with $n \geq 2$. As an additional constraint, vertices are assumed to be distributed in a clockwise order around the object of interest. Hence, any simple curve $\mathcal{C}_{i,i+1}$ linking $\mathbf{v}_i$ to $\mathbf{v}_{i+1}$ is positively oriented, such that the unit normal vector

$$\mathbf{n}_{\mathcal{C}_{i,i+1}}(u) = \frac{\mathcal{C}_{i,i+1}{}'(u)^{\perp}}{\left\|\mathcal{C}_{i,i+1}{}'(u)\right\|}$$

points towards the interior of the object, at any position u along $\mathcal{C}_{i,i+1}$. The clockwise order is a requirement for the computation of region integrals involved in the various energy terms. We introduce the admissible set $\mathcal{A}_i$, as a non-empty set of K_i admissible paths:

$$\mathcal{A}_i = \{\gamma_{i,j}\}_{1\leq j\leq K_i}.$$

As in the previous approach, we assume that these admissible sets are organized such that $\gamma_{i,1}$ is the geodesic path from $\mathbf{v}_i$ to $\mathbf{v}_{i+1}$ and remaining curves $\gamma_{i,j}$, $2 \leq j \leq K_i$, are other interesting paths. Let $\Cup$ be the curve concatenation operator. The curve resulting from the concatenation of $\mathcal{C}_1$ and $\mathcal{C}_2$, which is valid only if $\mathcal{C}_1(1) = \mathcal{C}_2(0)$, is defined as follows:

$$(\mathcal{C}_1 \Cup \mathcal{C}_2)(u) = \begin{cases} \mathcal{C}_1(2u) & \text{if } 0 \leq u \leq \frac{1}{2} \\ \mathcal{C}_2(2u-1) & \text{if } \frac{1}{2} < u \leq 1. \end{cases} \tag{8}$$

Selecting the x_i^{th} path in each admissible set $\mathcal{A}_i$, the closed curve resulting from the concatenation of selected admissible paths

$$\Gamma = \gamma_{\mathbf{v}_1,x_1} \Cup \gamma_{\mathbf{v}_2,x_2} \Cup \ldots \Cup \gamma_{\mathbf{v}_{n-1},x_{n-1}} \Cup \gamma_{\mathbf{v}_n,x_n},$$

should match the boundaries of the object of interest. To compute the admissible paths, we propose an approach based on the extraction of *saddle points*.

3.1 Valleys and saddle points

Let us consider two distinct points $\boldsymbol{a}$ and $\boldsymbol{b}$ and the edge-based potential P. The geodesic linking $\boldsymbol{a}$ and $\boldsymbol{b}$ may be actually extracted in three different ways:

- Propagate the minimal action from $\boldsymbol{a}$, stop when $\boldsymbol{b}$ is reached, and perform gradient descent from $\boldsymbol{b}$
- Propagate the minimal action from $\boldsymbol{b}$, stop when $\boldsymbol{a}$ is reached, and perform gradient descent from $\boldsymbol{a}$
- Propagate **simultaneously** from $\boldsymbol{a}$ and $\boldsymbol{b}$, stop at the first location where the two fronts collide, perform two gradient descents both sides apart from the meeting location, and assemble the two obtained paths adequately

The third principle is the basis for the generation of multiple paths. When the action map is propagated from two source points $\boldsymbol{a}$ and $\boldsymbol{b}$ - which generates what we refer to as the *combined action map* $U_{\boldsymbol{a},\boldsymbol{b}}$ -, the two propagation fronts meet at the **saddle points** of $U_{\boldsymbol{a},\boldsymbol{b}}$ (see Fig 3). If one intuitively thinks of the action as the height in a mountainous area, the saddle points are the highest points on the different roads travelling from $\boldsymbol{a}$ to $\boldsymbol{b}$. These roads may either be disjoint or share common sections, but generally each road lies in the bottom of a particular valley. A formal definition of saddle points is given by first introducing the medial set $\mathcal{M}_{\boldsymbol{a},\boldsymbol{b}}$, made up of points geodesically equidistant from $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$\mathcal{M}_{\boldsymbol{a},\boldsymbol{b}} = \{\boldsymbol{x} \mid U_{\boldsymbol{a}}(\boldsymbol{x}) = U_{\boldsymbol{b}}(\boldsymbol{x})\}.$$

The medial set $\mathcal{M}_{\boldsymbol{a},\boldsymbol{b}}$ forms a crest on the combined action map and is thus a set of critical points of $U_{\boldsymbol{a},\boldsymbol{b}}$, put another way $\nabla U_{\boldsymbol{a},\boldsymbol{b}}$ is not defined over $\mathcal{M}_{\boldsymbol{a},\boldsymbol{b}}$. However, set $\mathcal{M}_{\boldsymbol{a},\boldsymbol{b}}$ may be parameterized by a simple curve

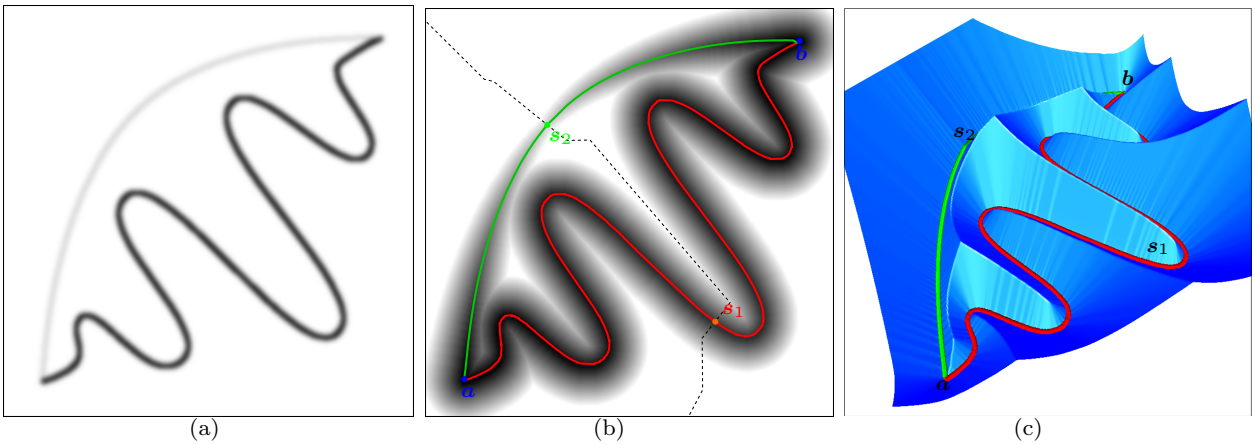


Fig. 3 Saddle points as a basis for determining relevant admissible paths between two given endpoints $\boldsymbol{a}$ and $\boldsymbol{b}$: (a) Potential highlights two possible distinctive paths. (b) Action map with two admissible paths with their respective saddle points located halfway and medial curve in dashed line. (c) Corresponding 3D plot.

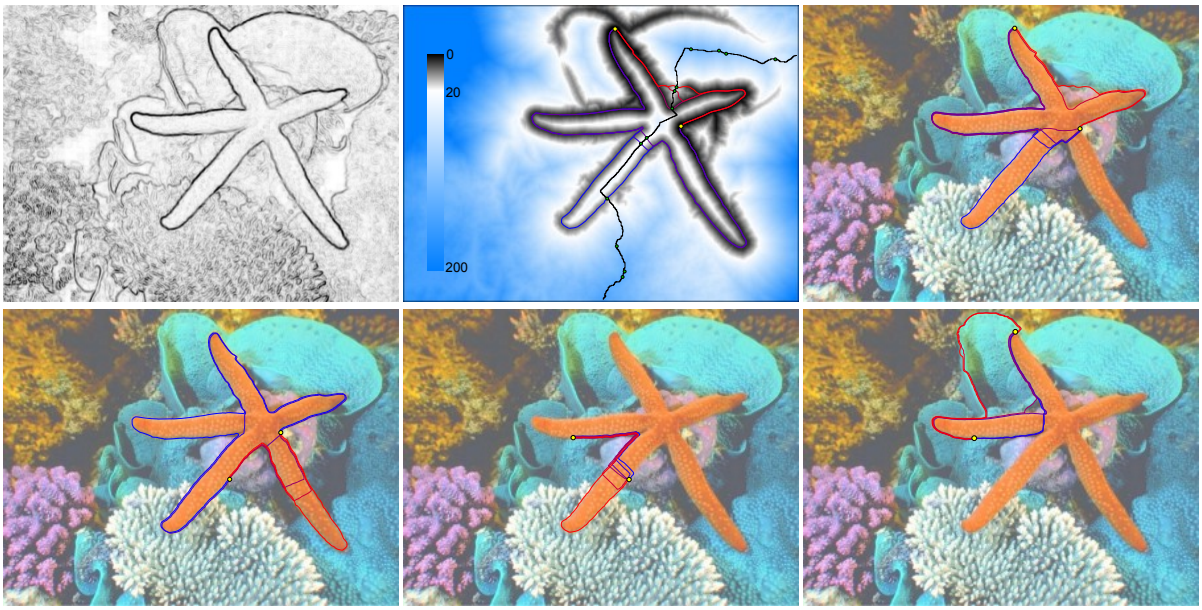


Fig. 4 Admissible paths for every pair of successive input vertices. (Top row, left) Potential P and (Top row, middle) Medial curve (black) and saddle points (green) with corresponding paths drawn over combined action map (Top row, right, and bottom row) Admissible set for each pair of successive vertices with $n = 4$. Paths are sorted (blue to red) according to their exteriority.

- which may be closed or open, depending on the locations of the endpoints and the potential function - and the combined action map may be differentiated along it. Let $\boldsymbol{m}_{\boldsymbol{a},\boldsymbol{b}} : [0,1] \longrightarrow \mathcal{D}$ be the *medial curve*, sweeping along the equidistant points. The saddle points on the combined action map between $\boldsymbol{a}$ and $\boldsymbol{b}$ are the local minima of $U_{\boldsymbol{a},\boldsymbol{b}}$ along $\boldsymbol{m}_{\boldsymbol{a},\boldsymbol{b}}$.

3.2 Admissible paths as piecewise-geodesic curves

The generation of admissible paths is the first stage of the proposed algorithm, summarized in Algorithm 1. For each pair of successive vertices $(\mathbf{v}_i, \mathbf{v}_{i+1})$, a combined action map $U_{\mathbf{v}_i,\mathbf{v}_{i+1}}$ is propagated, generating two monotonically advancing fronts from $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$. The locations where the two fronts collide are detected and stored until the combined action map has been propagated over the entire image domain. Only locations that are local minima of $U_{\mathbf{v}_i,\mathbf{v}_{i+1}}$ in the direction orthogonal to the propagation direction, i.e. along the medial curve, are kept as the K_i selected saddle points. Two gradient descents are performed on both sides of each saddle point $\boldsymbol{s}_j$, one path going down to $\mathbf{v}_i$ and the other one to $\mathbf{v}_{i+1}$. In other words, we take as starting

Algorithm 1 Summary algorithm of the entire segmentation process

Input : image I, sequence of n vertices $\{\mathbf{v}_i\}_{1\leq 1\leq n}$

// Generate sets of admissible paths
Compute potential P
foreach vertex pair $(\mathbf{v}_i, \mathbf{v}_{i+1})$

Propagate combined action map $U_{\mathbf{v}_i,\mathbf{v}_{i+1}}$ using Fast Marching
Extract saddle points of $U_{\mathbf{v}_i,\mathbf{v}_{i+1}}$
Build admissible set $\mathcal{A}_i$ by performing gradient descents from saddle points
Sort paths in $\mathcal{A}_i$ with respect to their exteriority measure in Eq. (17)

endfor

// Local search optimization
Set current combination of admissible paths to $(1, ...1)$
while at least one index in the combination can be increased

Compute energy of current combination using Eq. (12)
Compute energies of at most n candidate combinations using Eq. (12)
Select the candidate combination with lowest energy as the current combination for the next iteration

endwhile

points for the gradient descents

$$\boldsymbol{m}(u_j) + \epsilon \frac{\boldsymbol{m}'^{\perp}(u_j)}{\|\boldsymbol{m}'(u_j)\|} \quad \text{and } \boldsymbol{m}(u_j) - \epsilon \frac{\boldsymbol{m}'^{\perp}(u_j)}{\|\boldsymbol{m}'(u_j)\|},$$

where $\boldsymbol{m}$ stands for $\boldsymbol{m}_{\mathbf{v}_i,\mathbf{v}_{i+1}}$, u_j is the curve parameter such that $\boldsymbol{m}(u_j) = \boldsymbol{s}_j$ and ϵ is an infinitesimal offset. These two points are necessarily on different slopes of the mountain, their respective gradient vectors $\nabla U_{\mathbf{v}_i,\mathbf{v}_{i+1}}$ being almost opposite. The two gradient descents generate paths $\gamma_{\boldsymbol{s}_j,\mathbf{v}_i}$ and $\gamma_{\boldsymbol{s}_j,\mathbf{v}_{i+1}}$ respectively. The jth admissible path in $\mathcal{A}_i$ is obtained by reversing the first path and concatenating it with the other path.

$$\gamma_{i,j} = \gamma_{\mathbf{v}_i,\boldsymbol{s}_j} \Cup \gamma_{\boldsymbol{s}_j,\mathbf{v}_{i+1}}.$$

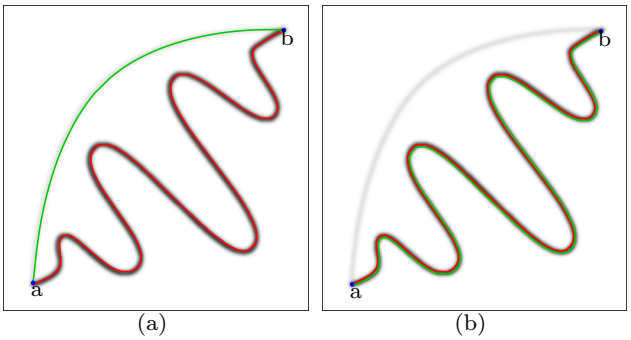
Among all saddle points, the one with the lowest action, which is also the first colliding location of fronts,

$$\boldsymbol{s}_{\min} = \underset{\boldsymbol{s}\in\mathcal{S}_{\mathbf{v}_i,\mathbf{v}_{i+1}}}{\operatorname{argmin}} \{U_{\mathbf{v}_i,\mathbf{v}_{i+1}}(\boldsymbol{s})\},$$

is located halfway, in the geodesic sense, on the minimal path between $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$:

$$\gamma_{\mathbf{v}_i,\boldsymbol{s}_{\min}} \Cup \gamma_{\boldsymbol{s}_{\min},\mathbf{v}_{i+1}} = \gamma_{\mathbf{v}_i,\mathbf{v}_{i+1}}.$$

The same relation does not hold for all higher saddle points $\boldsymbol{s}_j$. Paths from $\mathbf{v}_i$ to $\mathbf{v}_{i+1}$ passing through these saddle points may be only referred to as *piecewise-geodesic*. Details about the discrete implementation of Fast Marching and the extraction of saddle points are given in Appendix B.1.



(a) (b)

Fig. 5 Saddle point-based paths versus disjoint paths on smooth contours. (a) The second path (green) extracted with the higher saddle point follows a contour distinct from the one followed by $\gamma_{\boldsymbol{a},\boldsymbol{b}}$ (b) The second shortest path (green) constrained to be disjoint from $\gamma_{\boldsymbol{a},\boldsymbol{b}}$ turns out to be an offset curve of $\gamma_{\boldsymbol{a},\boldsymbol{b}}$.

3.3 Advantages over disjoint paths

We believe that considering saddle point-based paths, i.e. valley roads, instead of disjoint paths generates a more relevant and less redundant set of candidate paths. Indeed, different valleys may correspond to different sections of the boundary of an object, whereas different *short paths* may run through the same valley and thus be redundant. If the image contours were infinitesimally thick, only a single path could run through each valley, thus the notion of valley roads and disjoint shortest paths would be nearly equivalent. However, in real images where contours may be arbitrarily smooth, areas with low potential lead to large valleys on the action map. In continuous space, there is an infinite number of "almost minimal paths" passing along the bottom of the lowest valley.
Consider the example in Fig. 3(a) again, where the combined action map gives rise to two saddle points. Let $\boldsymbol{s}_1$ be the saddle point on the low and curvy valley, and $\boldsymbol{s}_2$ the saddle point on the higher and straighter valley. Let $\gamma^1_{\boldsymbol{a},\boldsymbol{b}}$ and $\gamma^2_{\boldsymbol{a},\boldsymbol{b}}$ be the paths passing through these respective saddle points, drawn in Fig. 3(b). Regardless of the Euclidean length, $\gamma^1_{\boldsymbol{a},\boldsymbol{b}}$ is the shortest path in the geodesic sense: $\gamma^1_{\boldsymbol{a},\boldsymbol{b}} = \gamma_{\boldsymbol{a},\boldsymbol{b}}$. Should the lower valley be large enough, i.e. the potential would be low over a thick band, the second shortest disjoint path would be very likely an offset curve of $\gamma_{\boldsymbol{a},\boldsymbol{b}}$, as depicted in Fig. 5. This illustrates the *path agglomeration* issue mentioned in Section 2.4. In the disjoint paths approach, this issue is dependent on the choice of the thickness of paths. A curve being by essence a mathematical object with infinitesimal thickness, setting P to $+\infty$ according to Eq. (7) has rigorously no effect since it does not prevent subsequent paths to take the same way, up to an infinitesimal offset. In practice, one has to add an arti-

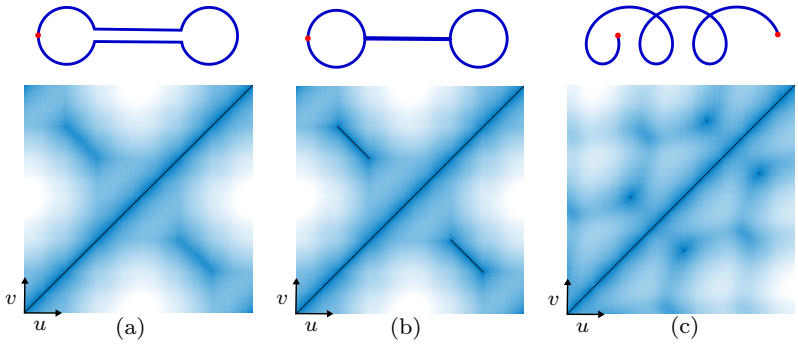


Fig. 6 Distances $\phi_{\mathcal{C}}$ plotted in (u, v)-space for several types of curve: (a) On a simple closed curve, $\phi_{\mathcal{C}}$ vanishes only on the graph diagonal. (b) On a curve with a section of self-tangency, $\phi_{\mathcal{C}}$ additionally exhibits two symetrical zero lines. (c) On a curve with self-intersections, $\phi_{\mathcal{C}}$ additionally exhibits isolated zeros.

ficial thickness to the curve, i.e. choose a bandwidth B over which subsequent paths will not be allowed to pass. Accounting for this consideration, a practical redefinition of Eq. (7) would be

$$P_j(\boldsymbol{x}) = \begin{cases} +\infty & \text{if } \exists u \in [0,1] \text{ s.t. } \|\boldsymbol{x} - \gamma_{i,j-1}(u)\| \leq \frac{B}{2} \\ P_{j-1}(\boldsymbol{x}) & \text{otherwise.} \end{cases}$$

From an implementation point of view, since the potential is discretized over the image grid, it needs to be set to $+\infty$ at all integer-coordinates pixels within a ball of radius $B/2$ centered at each path point. B may be referred to as the *erasure thickness* and adds a supplemental parameter, which is needed both for defining the correct mathematical notion of *path disjointness* and for effective implementation. Unless the erasure thickness B is chosen as large as the thickness of low-potential valleys, subsequent paths tend to be offset curves of the geodesic curve. The thickness of valleys is an intractable property, depending on the sharpness of contours and varying both between valleys and along each valley, which prevents from choosing an appropriate value of B without prior knowledge. Conversely, in the proposed approach, we endeavour to develop a more mathematically sound framework, which is not influenced by an artificial erasure thickness added to curves. The proposed saddle point-based approach is not affected by the geometry of valleys, as it depends only on the number of local minima of the action map. The action map having no flat plateau, the number of local minima along the medial curve is finite. Hence, the number of saddle point-based admissible paths is also finite, and varies with respect to the underlying image structures. In images containing smooth regions with sharp contours, the action maps exhibit few local minima along the medial curves and the resulting sets of admissible paths are advantageously small. In presence of cluttered or noisy data, spurious structures can lead to unwanted local minima along the medial curves if no smoothing operation is applied. In any case, provided that regularization is wisely applied, the risk of generating useless candidate paths is reduced. Conversely, there may be an infinite set of short disjoint paths, which has to be truncated by fixing a threshold K, regardless of the considered pair of vertices. On the one hand, in case of images with smooth regions, redundant paths may be generated if K is too large, increasing the computational time of the optimization step. On the other hand, in case of cluttered images, relevant contours can be missed if K is too small.
Eventually, in the current approach, one propagation is sufficient per pair of successive vertices, whereas disjoint paths require K instances of the Fast Marching algorithm. Even if a $O(KN \log N)$ complexity is reasonable, extracting all paths from the same action map generated by a single propagation in $O(N \log N)$ is more convenient. Each admissible path is computed on the combined action map regardless of other paths, which allows to compute admissible paths in parallel. Notice that, despite this possibility, computation times reported in Section 6 are obtained without any parallel implementation.

4 Defects of non-simple curves

Let us now assume that several admissible paths are concatenated one behind the other, yielding an assembled closed contour Γ. One of the desirable properties of Γ is that it should be simple, i.e. with no multiple point. Instead of imposing simplicity as a hard constraint, which might exclude relevant contours, it is encouraged by an additional energy. Indeed, it is reasonable to allow a certain degree of non-simplicity, e.g. when vertices are located far from the target bound-

aries, which might cause several admissible paths to have common sections before splitting up. Dealing with the geometrical and topological properties of the obtained curve, we wish to measure to what extent the curve is not simple.

If a curve $\mathcal{C}$ is non-simple, it has a number of *multiple points* that should be studied. We only consider *double points* [2], which may be of two kinds: self-tangencies and self-intersections. The proposed simplicity term is based on a distinction between these two aspects.

4.1 Self-tangency

Let $(u, v) \in [0, 1]^2$ s.t. $u \neq v$ be the pair of curve positions identifying a double point: $\mathcal{C}(u) = \mathcal{C}(v)$. If (u, v) corresponds to a point of self-tangency, velocity vectors $\mathcal{C}'(u)$ and $\mathcal{C}'(v)$ are colinear:

$$\left| \frac{\mathcal{C}'(u) \cdot \mathcal{C}'(v)}{\|\mathcal{C}'(u)\| \, \|\mathcal{C}'(u)\|} \right| = 1.$$

On the other hand, if (u, v) corresponds to a self-intersection - also known as an *ordinary double point* or *crunode* - velocity vectors point towards different directions, making the curve cross itself. This distinction allows to address separately two different defects on curves, which are not necessarily related. A curve with points of self-tangencies will exhibit self-overlapping segments, as depicted in Fig. 6(b), whereas a curve with self-intersections shown in Fig. 6(c) will split the image domain into more than two connected regions.

As regards the first kind of double points, the amount of self-tangency is quantified by measuring the length of overlapping curve segments. Considering function $\phi_{\mathcal{C}}$ measuring the squared Euclidean distance between two points on curve $\mathcal{C}$,

$$\phi_{\mathcal{C}}(u, v) = \|\mathcal{C}(u) - \mathcal{C}(v)\|^2 ,$$

the zero level set of $\phi_{\mathcal{C}}$, $\mathcal{Z}_{\mathcal{C}} = \{(u, v) \mid \mathcal{C}(u) = \mathcal{C}(v)\}$, is the set of pairs of positions giving equal points. Trivially, this set is never empty, since it contains at least all pairs (u, u). The length of $\mathcal{Z}_{\mathcal{C}}$ in the (u, v)-space corresponds to the total length of the overlapping segments:

$$\begin{aligned} |\mathcal{Z}_{\mathcal{C}}| &= \int_0^1 \int_0^1 \|\nabla H(\phi_{\mathcal{C}}(u, v))\| \, du dv \\ &= \int_0^1 \int_0^1 \delta(\phi_{\mathcal{C}}(u, v)) \, \|\nabla \phi_{\mathcal{C}}(u, v)\| \, du dv, \end{aligned} \tag{9}$$

where ∇ is the gradient operator in the $(u, v)-$space, H is the Heaviside step function and δ is the Dirac delta distribution.

It can be proven that for any simple regular curve $\mathcal{C}$, the amount of self-tangency $|\mathcal{Z}_{\mathcal{C}}|$ is $\sqrt{2}$ (a detailed proof with a more general distance function and parameterization is given in Appendix A.1). It can be intuitively understood by observing the plots of ϕ in the curve parameter space in Fig. 6, where dark lines correspond to self-overlapping segments. If the curve is non-simple, $|\mathcal{Z}_{\mathcal{C}}| \geq \sqrt{2}$. Moreover, this term is advantageously intrinsic, i.e. independent of parameterization, and invariant to scaling. Note that intersection points may be viewed as overlapping segments of length zero, and thus have no contribution in $|\mathcal{Z}_{\mathcal{C}}|$. Details about the discretization and computation of Eq. (9) are given in Appendix B.2.

[2] In our framework, curves with points of multiplicity> 2 are detected and excluded from the search

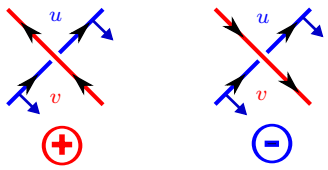


Fig. 7 Positive and negative crossings defined by ordered pair of positions (u, v). $\mathcal{C}(u)$ and $\mathcal{C}(v)$ are the intersected and intersecting sections in blue and red respectively.

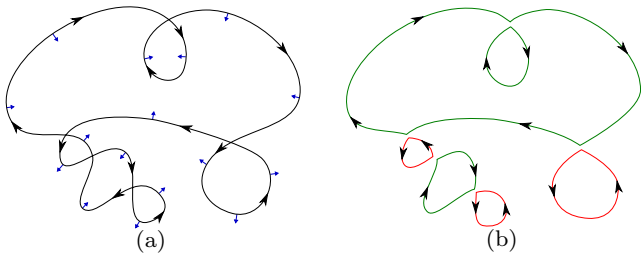


Fig. 8 Self-intersecting curve and inverted segments: (a) A nonsimple loop with ordinary double points which could be transformed into a set of (b) disjoint simple loops using a sequence of *uncrossing moves*. Resulting badly-oriented (counterclockwise) simple loops are drawn in red.

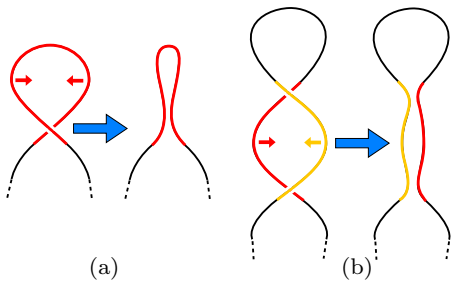


Fig. 9 Untwisting simple and double loops by (a) Reidemeister move of type I and (b) Reidemeister move of type II. Only colored parts of the curve are moved.

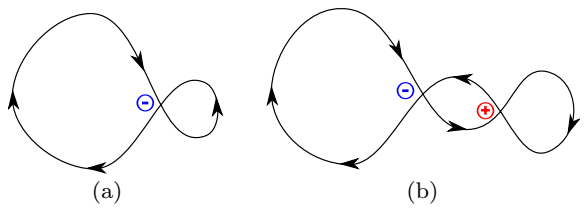


Fig. 10 Simple and double loops making inverted curve segments. (a) Simple loop made by a single negative crossing and (b) double loop made by a couple of positive and negative crossings.

4.2 Twisting

While tangent double points are used to measure overlapping, ordinary double points will serve as a basis for measuring the amount of *twisting* of the curve. Self-crossings are pairs of curve positions (u, v) where tangent vectors are not colinear:

$$\left| \frac{\mathcal{C}'(u) \cdot \mathcal{C}'(v)}{\|\mathcal{C}'(u)\| \, \|\mathcal{C}'(v)\|} \right| \neq 1.$$

Pairs of positions corresponding to crossing points are ordered, so that u is the position where the curve is *intersected* and v is the position where the curve is *intersecting*. Whether a position on the curve is intersected or intersecting only depends on the order in which these positions are met while one travels along the curve, so it follows that $u < v$. Whitney [44] studied crossing points of closed regular curves and divided them into two categories with respect to their orientation. Let us recall the orientation of curves, set up in the standard computer left-handed Cartesian coordinate system. In this setting, we choose the natural orientation of curves as the clockwise one, such that when one travels along the curve, looking forward, the interior of the curve is on the right. In such case, $\mathcal{C}'^{\perp}$ is the inward normal. The crossing will be *positive* if the intersecting part of the curve $\mathcal{C}(v)$ goes from right to left, or *negative* if it goes from left to right (see Fig. 7). In a positive (resp. negative) crossing, the intersecting section arrives from the interior (resp. exterior) of the intersected section. It follows that:

$\mathcal{C}'(u)^{\perp} \cdot \mathcal{C}'(v) < 0$ for a positive crossing
$\mathcal{C}'(u)^{\perp} \cdot \mathcal{C}'(v) > 0$ for a negative crossing.

Consider the self-intersecting curve shown in Fig. 8(a). It splits the image domain into disjoints subdomains, some of which are demarcated by *inverted segments*, i.e. portions of curves along which the normal vector points outward. If one decomposes the curve using *uncrossing moves* - replacing each couple of crossing sections by two new non-crossing sections - one obtains a collection of disjoint simple loops, as in Fig. 8(b), also known as *Seifert circles* [1, p. 94]. While some simple loops are well oriented (clockwise), others are inverted (counter-clockwise). We propose to quantify the twisting of $\mathcal{C}$ as the proportion of area demarcated by inverted segments to the total area of $\mathcal{C}$, or equivalently, the area inside the counter-clockwise oriented loops one would obtain when decomposing the curve. To some extent, this proportion is related to the energy that one should exert to untwist the curve.

A self-intersecting curve can be thought of as a planar projection of a knot, i.e. an embedding of the unit circle into Euclidean $\mathbb{R}^3$ space. Following the theory of knots [1, p. 13] [3], we consider twisted configurations of curves in which corresponding knots could be unknotted by Reidemeister moves of type I and II (see Fig. 9). The simple loop, described by a single inverted segment and involving a single intersection, is related to the Reidemeister move of type I, whereas the double loop, consisting of two segments and involving two intersections, is related to the Reidemeister move of type II. The two types of crossing points, that were previously presented, are used to detect such loops. A simple inverted loop will be detected thanks to a negative crossing point (u, v) such that there is no other crossing from u to v, as in Fig. 10(a). A double inverted loop will be detected thanks to a negative crossing (u, v) such that the path from u to v contains a positive crossing, as in Fig. 10(b). Actually, we do not explicitly split the curve into simple loops, as in Fig. 8(b). We rather detect self-crossings, determine their orientations, and extract inverted loops on the initial curve.

Let $\mathrm{SL}(\mathcal{C}) \subset [0,1]^2$ be the set of ordered pairs of curve positions (u, v) s.t. $u < v$, describing intersections involved in single inverted loops, and $\mathrm{DL}(\mathcal{C}) \subset [0,1]^2 \times [0,1]^2$ be the set of double ordered pairs $((u_1, v_1), (u_2, v_2))$ s.t. $u_1 < u_2$ and $v_1 > v_2$ describing the couples of intersections involved in double inverted loops. Sets SL and DL are extracted by the loop detection algorithm detailed in Appendix B.2. When a portion of curve $\mathcal{C}$ from s to t, denoted $\mathcal{C}|_{s \to t}$, is closed and simple, the signed area inside $\mathcal{C}|_{s \to t}$ can be expressed using Green's theorem:

$$\begin{aligned} \int_{\Omega_{\mathrm{in}}(\mathcal{C}|_{s \to t})} \mathrm{d}\boldsymbol{x} &= \frac{1}{2} \int_s^t x(u)y'(u) - x'(u)y(u) \mathrm{d}u \\ &= \int_s^t \frac{\mathcal{C}(u)^{\perp} \cdot \mathcal{C}'(u)}{2} \mathrm{d}u. \end{aligned} \tag{10}$$

Eventually, the total area of inverted loops of $\mathcal{C}$, denoted by $\mathcal{I}[\mathcal{C}]$, is expressed by considering all simple

and double inverted loops in SL and DL:

$$\mathcal{I}[\mathcal{C}] = -\sum_{(s,t)\in \mathrm{SL}(\mathcal{C})} \int_s^t \frac{\mathcal{C}^\perp \cdot \mathcal{C}'}{2} \mathrm{d}u - \sum_{\substack{((s_1,t_1),(s_2,t_2)) \\ \in \mathrm{DL}(\mathcal{C})}} \int_{s_1}^{s_2} \frac{\mathcal{C}^\perp \cdot \mathcal{C}'}{2} \mathrm{d}u + \int_{t_2}^{t_1} \frac{\mathcal{C}^\perp \cdot \mathcal{C}'}{2} \mathrm{d}u. \tag{11}$$

Note that $\mathcal{I}[\mathcal{C}]$ is positive, as the signed area of every inverted loop is negative.

5 Combination of admissible paths using the simplicity energy

5.1 Energy terms

The proposed energy functional extends the energy of the GLAC. It is designed to penalize contours exhibiting strongly overlapping sections or self-intersections, poorly fitting to image edges or enclosing regions with indistinct color distributions:

$$E[\Gamma] = E_{\mathrm{simplicity}}[\Gamma] + \omega_{\mathrm{edge}} E_{\mathrm{edge}}[\Gamma] \ + \ \omega_{\mathrm{region}} E_{\mathrm{region}}[\Gamma]. \tag{12}$$

Weights ω_{edge} and ω_{region} are user-defined parameters controlling the relative significance of the data terms over the simplicity term. The latter involves the self-tangency and twisting measures defined in Eqs. (9) and (11),

$$E_{\mathrm{simplicity}}[\Gamma] = \frac{|\mathcal{Z}_\Gamma| - \sqrt{2}}{\sqrt{2}} + \frac{1}{|\Omega_{\mathrm{in}}(\Gamma)|} \mathcal{I}[\Gamma]. \tag{13}$$

Note that the two terms in this simplicity term are normalized each within range $[0,1]$. In order to penalize equally self-tangency and twisting, the normalized terms are added without any additional weight. Energy $E_{\mathrm{simplicity}}[\Gamma]$ vanishes when Γ is a simple curve. As regards image-based terms, the same edge energy is kept as in the GLAC, so E_{edge} integrates the edge indicator function g (Eq. (2)) along the contour:

$$E_{\mathrm{edge}}[\Gamma] = \frac{1}{|\Gamma|} \int_0^1 g(\Gamma(u)) \, \|\Gamma'(u)\| \, \mathrm{d}u. \tag{14}$$

In order not to penalize lengthy contours, it is normalized by the Euclidean length $|\Gamma|$. One may note that the edge indicator g is used instead of the potential P so that the Euclidean component of the curve length is not taken into account. This ensures that short curves, which could be undesirable shortcuts, are not preferred over longer ones. For the region term, instead of using the piecewise-constant model as in the GLAC, which limits the segmentation to relatively homogeneous objects and backgrounds, we use the Bhattacharyya coefficient between the color probability distributions inside and outside Γ, following [29]:

$$E_{\mathrm{region}}[\Gamma] = \int_Q \sqrt{\mathrm{p}_{\mathrm{in}}(\Gamma, \boldsymbol{q}) \mathrm{p}_{\mathrm{out}}(\Gamma, \boldsymbol{q})} \mathrm{d}\boldsymbol{q}, \tag{15}$$

where Q is the color space. Probability distribution functions (PDF), for a given color $\boldsymbol{q}$, are estimated using a Gaussian kernel-based histogram:

$$\begin{aligned} \mathrm{p}_{\mathrm{in}}(\Gamma, \boldsymbol{q}) &= \frac{1}{|\Omega_{\mathrm{in}}(\Gamma)|} \int_{\Omega_{\mathrm{in}}(\Gamma)} G_\sigma(\boldsymbol{q} - I(\boldsymbol{x})) \mathrm{d}\boldsymbol{x} \\ \mathrm{p}_{\mathrm{out}}(\Gamma, \boldsymbol{q}) &= \frac{1}{|\Omega_{\mathrm{out}}(\Gamma)|} \int_{\Omega_{\mathrm{out}}(\Gamma)} G_\sigma(\boldsymbol{q} - I(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{16}$$

where Ω_{in} and Ω_{out} are the regions inside and outside Γ, respectively.

5.2 Local search method

The computation of the final contour can be formulated as determining the sequence of labels $(x_1, x_2, \ldots, x_n)$ minimizing energy (12), where label x_i corresponds to the chosen admissible path in set $\mathcal{A}_i$:

$$\min_{\substack{(x_1,x_2,\ldots,x_n)\in \\ [1..K_1]\times[1..K_2]\times\cdots\times[1..K_n]}} E\left[\gamma_{1,x_1} \Cup \gamma_{2,x_2} \Cup \ldots \Cup \gamma_{n,x_n}\right],$$

Determining the best combination is the second stage of the proposed approach summarized in Algorithm 1. Let $K_{\max}$ be the maximum allowed number of admissible paths over all pairs of successive vertices:

$$K_{\max} = \max_{1\le i\le n} K_i$$

To determine the best sequence of labels $(x_1, \ldots, x_n)$, a brute-force search would yield an exponential time-complexity upper-bounded by $O(K_{\max}^n)$. To avoid testing all possible configurations, we propose a greedy search in $O(n^2 K_{\max})$ relying on a specific ordering of paths. In each admissible set $\mathcal{A}_i$, paths are sorted according to increasing *exteriority* $\mathcal{X}$, i.e. the signed area, calculated with Green's theorem, formed by a given path $\mathcal{C}$ and the line segment from $\mathcal{C}(1)$ returning to $\mathcal{C}(0)$:

$$\mathcal{X}[\mathcal{C}] = \frac{1}{2} \int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u + \frac{1}{2} \mathcal{C}(1)^\perp \cdot \mathcal{C}(0). \tag{17}$$

If the straight line from $\mathcal{C}(0)$ to $\mathcal{C}(1)$ is taken as a reference horizontal axis, the exteriority is negative (resp. positive) if $\mathcal{C}$ is predominantly below (resp. above) the axis. The vertices being located clockwise, admissible paths are sorted from the innermost to the outermost

(see Fig. 4). In a given admissible set $\mathcal{A}_i$, $\gamma_{i,1}$ is now the innermost path whereas γ_{i,K_i} is the outermost one. The optimization procedure starts with the initial labelling $(1,\ldots,1)$, which corresponds to the most interior configuration

$$\gamma_{1,1} \Cup \gamma_{2,1} \Cup \ldots \Cup \gamma_{n-1,1} \Cup \gamma_{n,1}.$$

Labels are then changed according to a local search, by iteratively testing candidate labellings. At each iteration, given the current sequence of labels $\mathcal{S}$, candidate sequences are tested that differ from a single label from $\mathcal{S}$, by increasing labels only. Obviously, at the current iteration, there are at most n candidate sequences. For instance, if $(2,3,1)$ is the current sequence, candidate sequences will be $(3,3,1)$, $(2,4,1)$ and $(2,3,2)$. Testing a candidate sequence $(x_1, x_2, \ldots x, x_{n-1}, x_n)$ implies computing energy

$$E\left[\gamma_{1,x_1} \Cup \gamma_{2,x_2} \Cup \ldots \Cup \gamma_{n-1,x_{n-1}} \Cup \gamma_{n,x_n}\right]$$

according to Eq. (12). The candidate sequence leading to the smallest energy, regardless of the current energy, is chosen as the base sequence for the next iteration. The best sequence found since the beginning of the procedure is updated as well. Since increasing labels correspond to paths of increasing exteriority, generating candidate sequences by solely increasing labels makes the contour expand monotonically. To some extent, the greedy search is in accordance with the evolution of a classical active contour or level set with a balloon-based inflating speed. The algorithmic details of the optimization procedure are given in Appendix B.3.

6 Experiments and discussion

We demonstrate the ability of the model to recover closed boundaries of objects in natural color images, given a variable number of user-provided points along the target boundary. Most of the tests are carried out on the Grabcut dataset [37]. These experiments include an independent study of our algorithm, as well as comparisons against the original GLAC method [32] (without deformation, i.e. the piecewise-geodesic curve only) and the Riverbed algorithm [33], in order to show the benefits brought by the use of admissible paths and additional energy terms.

The experiments are limited to simply connected objects, as every tested method is designed to recover a single closed outer boundary of one object. One may note that it would be possible to extend the model such that the set of initial vertices would be separated into subsets: one subset containing the vertices on the outer boundary and several subsets containing vertices on inner boundaries. However, we chose to restrict ourselves to objects without inner boundaries.

As regards the selection of parameters, each method is assessed in the most favorable configuration. Parameters such as the regularization weight w (for the GLAC and our approach) or the energy weights ω_{edge} and ω_{region} (specifically for the proposed algorithm) are tuned separately each time, in order to achieve the most relevant segmentation. The appropriate color space is also chosen for each image, i.e. RGB or the more perceptually uniform Lab, which affects potential P and the color PDFs involved in the region energy (15). In order for the reader to get an accurate idea of the importance of the regularization weight w in balance with the gradient magnitude in potential (18), notice that color components of $\boldsymbol{I}$ are normalized between 0 and 1. The maximum number $K_{\max}$ of admissible paths per set $\mathcal{A}_i$ is typically chosen between 3 and 10.

6.1 Potential for color images

The potential used in the minimal path approach in Section 2.1 was described for scalar-valued images. As an implementation detail, in order to avoid any ambiguity, one should note that the potential $P(\boldsymbol{x}) = w + g(\boldsymbol{x})$ actually used is based on color edge filtering. Hence we consider a vector-valued input image $\boldsymbol{I} : \mathcal{D} \to \mathbb{R}^3$ in the chosen color space. Moreover, instead of a hyperbolic decrease of potential with respect to edge magnitude, as in Eq. (2), we found that clipped linear decrease led to better contour extraction, so that the potential used is

$$P(\boldsymbol{x}) = w + \max\left(0, 1 - \alpha \left\|(\nabla K_s * \boldsymbol{I})(\boldsymbol{x})\right\|\right) \tag{18}$$

where $\nabla K_s * \boldsymbol{I}$ should be understood as the 2×3 Jacobian matrix made up of Gaussian-smoothed x and y-derivatives of each component of $\boldsymbol{I}$. In this case, $\|.\|$ is the *entrywise* Euclidean norm (strictly speaking, the Frobenius norm). Parameter α controls the decreasing slope with respect to edge magnitude.

6.2 Influence of edge, region and simplicity terms

One of the main benefits of the proposed approach over usual minimal path-based segmentation algorithms is the ability to formulate a region-based criterion, as in classical active contours. It allows to formulate the simplicity term as well. Fig. 11 illustrates the interest of using the region and simplicity terms, in addition to the edge term. The LEAF image is processed with regularization weight w set to 0.01, and $K_{\max} = 5$. Chosen

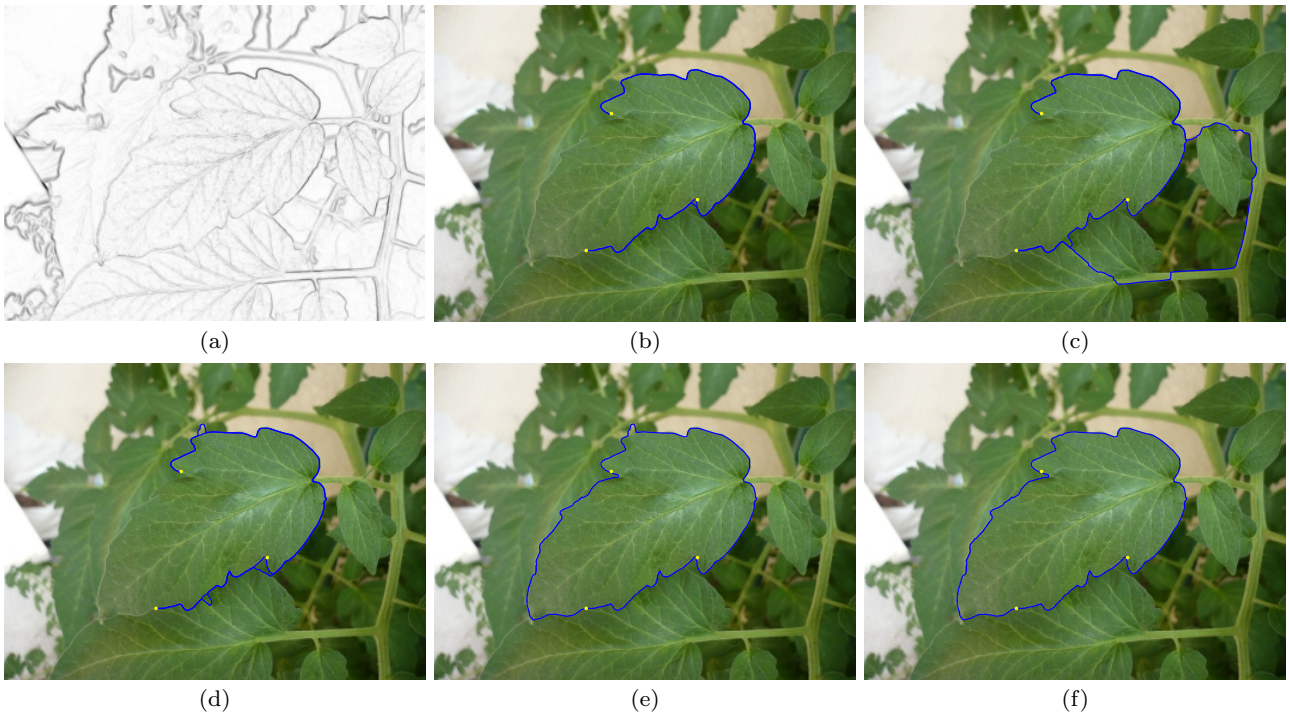


Fig. 11 Influence of energy terms on the final segmentation: (a) Potential, (b) Comparison against GLAC without deformation (piecewise-geodesic curve), Combination of paths with (c) edge term only (d) edge and region terms (e) edge and simplicity terms and (f) edge, region and simplicity terms.

color space is RGB. When corresponding energy terms are used, weights are set to $\omega_{\text{edge}} = 1$ and $\omega_{\text{region}} = 2$, 0 otherwise. The undesirable overlapping phenomenon yielded by the piecewise-geodesic curve is shown in Fig. 11(b). With the edge term only (Fig. 11(c)), our algorithm selects the strongest contours, leading to the lowest potential, regardless of their lengths. Hence, it tends to capture any boundary in the image which is more salient than the actual object boundaries. On this particular image, where the object and background have strongly overlapping color distributions, adding the region term (Fig. 11(d)) yields a contour including an almost empty interior region. The inner color PDF is thus very compact, which limits the overlapping with the outer color PDF and thus minimizes the region term. A satisfactory segmentation is achieved as soon as the simplicity term is added, as depicted in Figs. 11(e) and 11(f). The combination of region and simplicity terms allows to remove the small remaining background areas that were mistakenly included into the object.

6.3 Overview of the Riverbed algorithm

We give a summary of the recent Riverbed algorithm by Miranda *et al*, so that the reader can get the essence of this segmentation method without going into the specifics of the implementation. We especially focus on the properties which make this algorithm suitable for comparison. For further details, we refer the reader to [33].

First, the Riverbed algorithm rests upon a discrete framework, as the image domain is modeled as a directed 4-connected grid graph. Miranda *et al* developed their method with a generic edge cost function c, leaving the choice depending on the selected features of interest in the image. In order to use a cost function similar to the potential P used in our approach, we chose the edge cost $c(\boldsymbol{x}, \boldsymbol{y})$ from node (pixel) $\boldsymbol{x}$ to node $\boldsymbol{y}$ as

$$c(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{y}).$$

Hence, as in our approach, favoring low-cost edges is equivalent to following high-gradient areas. A sequence of n ordered vertices is placed along the object contour. Minimum cost paths between these vertices are extracted thanks to the Image Foresting Transform (IFT) [16], which may be considered as a heuristic generalization of Dijkstra's shortest path algorithm. In addition to computing the distance from a set of seed nodes to every other node in the grid, it generates a predecessor map, which links every node to its preceding node in the corresponding path. Let $\pi_{\boldsymbol{x}}$ be a path ending at node $\boldsymbol{x}$. Assuming that $\boldsymbol{y}$ is a successor node of $\boldsymbol{x}$,

we denote by $\langle \pi_{\boldsymbol{x}}, \boldsymbol{y} \rangle$ the extension of $\pi_{\boldsymbol{x}}$ by arc $(\boldsymbol{x}, \boldsymbol{y})$. Instead of considering an additive cost function only,

$$C\left(\langle \pi_{\boldsymbol{x}}, \boldsymbol{y} \rangle\right) = C(\pi_{\boldsymbol{x}}) + c(\boldsymbol{x}, \boldsymbol{y}),$$

the IFT computes the distances and predecessors for arbitrary path cost functions. If the cost function satisfies smoothness conditions, it is proven that the distance and predecessor map are optimal. Otherwise, they correspond to a local solution. This is the case with the Riverbed approach, which uses the following cost function:

$$C_{\text{riverbed}}\left(\langle \pi_{\boldsymbol{x}}, \boldsymbol{y} \rangle\right) = c(\boldsymbol{x}, \boldsymbol{y}).$$

Therefore, at any time, the IFT propagation algorithm decides to take the most interesting arc, regardless of the past history. As a summary of the justifications stated in [33], it allows not to favor shortest distance paths and to avoid undesirable shortcuts. Given n initial vertices, n instances of the IFT are run. The ith instance takes vertex $\mathbf{v}_i$ as the single seed node and generates a path to $\mathbf{v}_{i+1}$. Between two instances of the IFT, nodes belonging to previously-built paths are "frozen", so that subsequent paths cannot pass through them. Hence, by construction, the Riverbed forbids overlap between paths as a hard constraint. Unlike our approach, the resulting contour depends on the order in which pairs of successive vertices are considered.

6.4 Qualitative comparison

To keep a critical look at our contribution, Fig. 12 depicts typical cases of comparison between the current approach, the GLAC and the Riverbed algorithm, on a subset of the Grabcut dataset [37]. The combination of paths may lead to strong global improvements, or slight localized improvements, or no improvement when the edge strength and locations of vertices are favorable to all segmentation algorithms. For each image, the same set of initial vertices is used.

The BANANA image (row 1) depicts a situation where our approach does not improve segmentation over existing algorithms, for this particular configuration of initial points. Despite the complex background and object containing many inner edges, boundaries are well defined and vertices are evenly distributed along the boundary so that the GLAC and Riverbed manage to extract the object. The FLOWER (row 2) and DOLL (row 3) images are cases where the GLAC exhibits strong overlap between geodesics when few vertices are provided, although the vertices are reasonably well distributed along the object boundary. Since it does not have any non-overlapping constraint, the GLAC systematically favors portions of contours with the lowest potential. Hence, smooth boundary segments are ignored, not because of their length, but because they may contain sparsely weak edges making the potential increase in small parts of the contour. Conversely, these boundary segments, despite from not being part of the minimal path, create valleys in the distance map and are very likely to be considered as parts of admissible paths by our approach. This proves the proposed method to be inherently less sensitive to weak edges. As regards the Riverbed approach, it does not generate self-overlapping contours, but tends to favor the sharpest edges in the neighboring structures of the object.

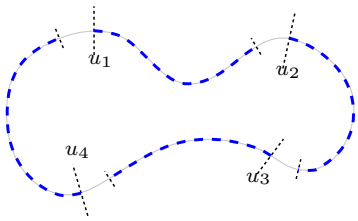


Fig. 13 Splitting the ground truth contour into segments of equal length for random locations of initial vertices: example with $n = 4$.

The CERAMIC (row 4) and TEDDY (row 5) images depict situations where the GLAC takes *shortcuts* through the object, due to the presence of inner edges stronger than the actual boundaries. Both simplicity and region energies contribute to solve this issue in our algorithm. The former prevents overlapping while the latter favors high discrepancy between inner and outer color distributions, hence avoiding to select the undesirable shortcuts, which would yield less distinct color histograms than the actual boundaries would. Finally, the MUSHROOM (row 6) combines the issues of inner shortcut and strong self-overlapping.

6.5 Quantitative comparison with random locations of vertices

The qualitative study with user-provided initial points cannot give alone an insight into the robustness of the compared segmentation algorithms. The experienced user has the essence of the algorithms in mind, and thus tends to place initial points in favorable configurations, such as even spacing of vertices along the boundary, or dense sampling along blurry edges and sharp turns of the boundary.

In order to provide an evaluation less dependent on the user-induced bias, we assess the robustness of our

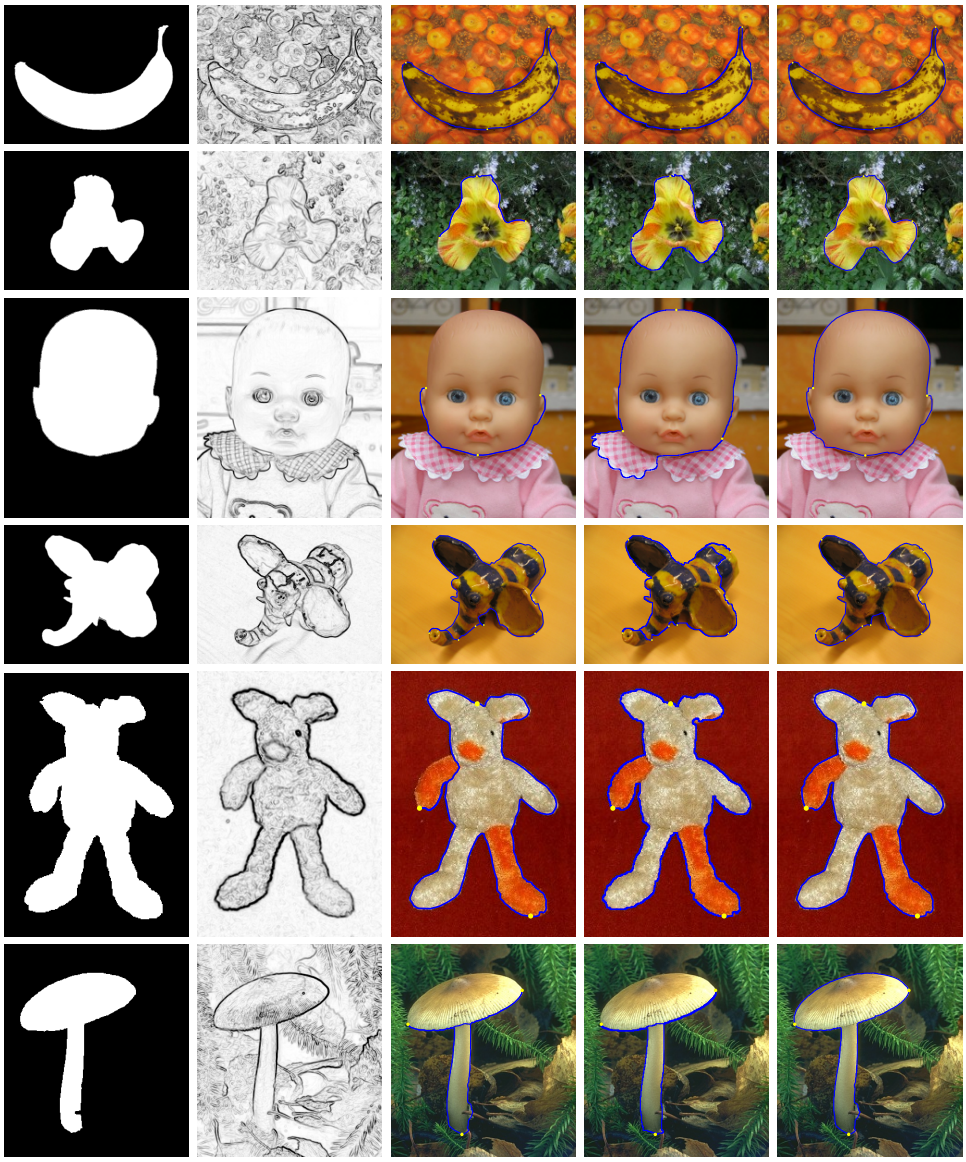


Fig. 12 Qualitative comparison with the GLAC and Riverbed on a sample of the Grabcut dataset. Column 1: ground truth segmentation, Column 2: potential, Column 3: GLAC without deformation (piecewise-geodesic curve), Column 4: Riverbed, Column 5: combination of paths.

method by running several tests on a subset of the Grabcut dataset, where initial vertices are selected randomly along the contour. The actual object boundary is extracted from the available binary ground truth image and a random start position u_1 along the curve is generated. The number of vertices n being fixed for each image, the ground truth contour is split into n segments of equal length, starting from u_1. Each segment is subsequently split into two subsegments: a free subsegment, covering the largest part of the segment, and a safety margin of a few pixels (typically, in the order of 10 pixels). Each vertex $\mathbf{v}_i$ is randomly generated within the ith free subsegment, the safety margins preventing two successive vertices from getting excessively close. In this way, the distribution of vertices roughly covers the entire contour, without being necessarily evenly spaced. Fig. 13 illustrates the splitting of a contour with $n = 4$. Segments are separated by long dashed lines normal to the curve and free subsegments are highlighted with dashed blue lines. For each image, we generate 20 random configurations of vertices. and the same random configuration is used for initializing the GLAC, Riverbed and the proposed method. For each test, segmentation accuracy is assessed by measuring the overlap between regions, using the Jaccard index

$$J(S, G) = \frac{|S \cap G|}{|S \cup G|},$$

where S and G are the segmented and ground truth regions, respectively. The mininum, maximum, average and standard deviation values of the Jaccard segmentation accuracy index, computed over the 20 runs per image, are listed in Table 1. The best segmentation for each image, yielding the maximal Jaccard score, is shown in Fig. 14. Notice the distribution of vertices, which may be unevenly spaced along the object boundary, thus lending itself to assess the robustness of the algorithm with respect to unfavorable initial conditions.
If one considers the 'Min' columns in Table 1, it appears that segmentation accuracy can be almost null in the least favorable cases, for every algorithm. For a start, there is no image for which the poorest segmentation obtained with our algorithm is significantly worse than the poorest segmentations obtained with the GLAC and Riverbed algorithm. On the contrary, the combination of paths even guarantees a higher standard, since there is an important proportion of images where the worst segmentation is improved by our algorithm. As can be seen by considering simultaneously the 'Max' columns in Table 1 and the contours in Fig. 14, Jaccard percentage values around 80% correspond to segmentations where important parts of the object are missing or background areas are mistakenly included into the object. As a general remark regarding the maximum Jaccard percentages, they are most often obtained with Riverbed or our algorithm. The improvement in the maximum quality can be seen on a small subset of images, such as the BANANA3 and CERAMIC data, but cannot be generalized to the entire dataset. One may get an idea of the most significant improvement brought by our approach by considering the average Jaccard percentages, both on each image independently and on the global score. The combination of paths leads to a better average score than the GLAC and Riverbed for all tested images, which conveys the overall robustness of our approach with respect to initial conditions.

6.6 Local search versus brute-force search

The search method described in Section 5.2 minimizes the energy over combinations of admissible paths in a heuristic way. In order to assess the efficiency of the local search against exhaustive search, we should determine whether the local minimum obtained is significantly worse than the actual optimal solution found by a brute-force algorithm. As in the previous experiments, we perform 20 runs per image with randomly selected initial vertices, reporting the Jaccard index, energy and computation time obtained with both search methods. Statistics of Jaccard coefficients are listed in Table 2. Notice that the Jaccard coefficients for the combination of paths with local search are not equal to the ones appearing in Table 1, as they correspond to another sequence of runs, with different randomly localized vertices. Statistics on Jaccard coefficients show that the local search does not significantly damage segmentation quality over the exhaustive search.
As regards combinations of paths and their energies, on the same dataset, checking final combinations shows that the local search finds the same combination of admissible paths as the brute-force method in 65% of the cases. Obviously, for the remaining 35% of cases, the combinations determined by the local search have greater energies than the ones found by the brute-force algorithm. However, it happens that the local search leads to a better segmentation quality. This is a phenomenon that arises in many optimization-based segmentation methods, such as active contours, as the global minimum of the energy does not necessarily correspond to the best expected segmentation.
As regards computation time, the generation of admissible paths (i.e. propagating using Fast Marching, detecting saddle points, building paths and sorting them with respect to their exteriority) is relatively fast. The major part of the computational cost lies in the search for the best combination of admissible paths, which highly depends on the number of initial vertices n and the upper bound $K_{\max}$ on the number of admissible paths. This significantly favors the heuristic local search over the brute-force algorithm. Reported execution times are obtained with a C++ implementation running on a standard Intel Core2 Duo 2.8GHz architecture with 4Gb RAM. The MUSHROOM image, with $n = 3$ and $K_{\max} = 5$, was fully processed in 3s300ms with the local search method and 12s with the brute-force search (both including 400ms to generate admissible paths). The CERAMIC image, with $n = 7$ and $K_{\max} = 5$, was fully processed in 17s with the local search method and 1h10mn23s with the brute-force search (both including 2s to generate admissible paths).

7 Conclusion

The main contributions of the current work, namely the construction of admissible paths as well as the introduction of the simplicity term, solve important shortcomings of existing interactive path-based segmentation algorithms. Admissible paths generalize the notion of geodesic path, and allow to capture relevant image structures in addition to the most salient contour. By searching the best paths configuration among sets of admissible paths, given an energy functional combining edge and region data terms with a novel term favoring the simplicity of the curve, we addressed the

	GLAC				Riverbed				Combination of paths			
Image	Min	Max	Avg	Std	Min	Max	Avg	Std	Min	Max	Avg	Std
BANANA1	1.0	84.0	23.3	0.21	5.9	88.8	53.5	0.21	17.0	89.3	71.7	0.18
BANANA2	0.8	55.5	11.4	0.14	1.8	84.6	39.5	0.20	0.1	82.9	44.6	0.28
BANANA3	0.2	76.6	32.6	0.25	2.7	72.2	43.3	0.18	1.2	81.5	55.7	0.26
CERAMIC	77.1	89.0	82.2	0.03	59.2	78.8	64.2	0.05	85.3	93.3	88.4	0.02
DOLL	0.3	87.6	45.0	0.40	0.7	90.7	62.0	0.40	78.9	87.6	81.0	0.03
FLOWER	1.0	98.2	20.9	0.39	2.0	98.1	80.3	0.34	1.0	98.3	82.8	0.34
MUSHROOM	1.4	65.3	15.0	0.25	1.5	91.5	37.7	0.33	22.0	91.1	66.4	0.21
MUSIC	0.7	98.6	83.9	0.35	0.9	98.5	47.7	0.47	97.3	98.6	97.9	0.01
SHEEP	1.6	83.9	11.5	0.24	36.5	90.7	54.0	0.14	55.9	89.3	78.2	0.09
TEDDY	2.0	89.2	55.8	0.34	88.5	96.3	90.5	0.03	51.9	96.9	79.3	0.12
Overall	0.2	98.6	38.1	0.38	0.17	98.5	57.3	0.31	0.1	98.6	74.6	0.24

Table 1 Quantitative comparison between the GLAC without deformation (piecewise-geodesic curve), Riverbed and the proposed algorithm: minimum, maximum, average and standard deviation values of the Jaccard segmentation accuracy percentage computed over 20 runs per image.

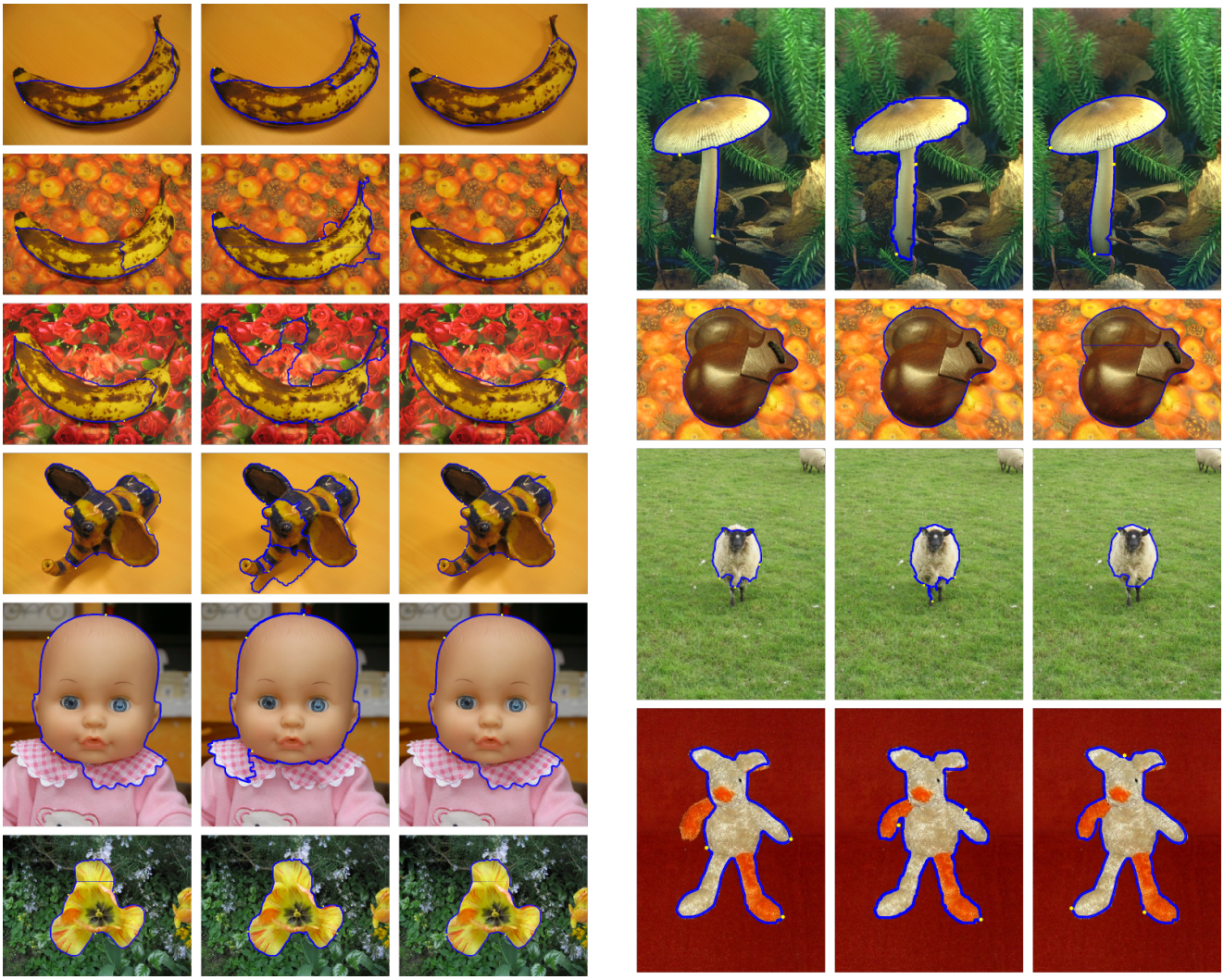


Fig. 14 Best segmentations (leading to the maximum Jaccard segmentation accuracy index) over 20 runs per image, with randomly-located initial vertices, for the GLAC without deformation (piecewise-geodesic curve, left), Riverbed (center) and the proposed algorithm (right).

Image	Combination of paths Local search Min	Max	Avg	Std	Combination of paths Brute-force search Min	Max	Avg	Std
BANANA1	20.0	89.1	60.4	0.20	20.0	89.1	59.2	0.20
BANANA2	0.7	88.3	47.3	0.25	0.7	91.1	52.2	0.31
BANANA3	26.1	86.6	62.5	0.15	26.1	86.6	59.7	0.18
CERAMIC	74.8	89.8	85.6	0.03	74.4	88.1	85.0	0.05
DOLL	72.5	87.7	80.8	0.04	78.9	87.7	82.6	0.07
FLOWER	1.4	98.2	88.1	0.29	1.3	98.2	88.4	0.29
MUSHROOM	33.0	91.1	61.3	0.17	25.3	86.3	56.6	0.19
MUSIC	97.3	98.6	97.8	0.01	97.3	98.6	97.9	0.05
SHEEP	4.5	90.2	77.0	0.18	2.9	90.2	76.2	0.19
TEDDY	17.6	96.7	74.9	0.17	17.6	96.7	74.9	0.17
Overall	0.7	98.6	73.5	0.23	0.7	98.6	73.3	0.24

Table 2 Quantitative comparison between the proposed algorithm (with local search) and a modified version with brute-force search: minimum, maximum, average and standard deviation values of the Jaccard segmentation accuracy percentage computed over 20 runs per image.

issues raised by minimal paths, namely shortcuts and self-overlapping or intersecting contours. To some extent, the proposed approach blends the benefits of minimal paths and region-based active contours. Comparison against the geodesically linked active contour model and the Riverbed algorithm, which have similar inputs and purposes, demonstrated the advantages of the approach.

A Mathematical derivations of overlap and exteriority terms

A.1 Overlap term

Let $\mathcal{C}$ be a regular curve parameterized over $[0, L]$. Let ϕ be a C^1 function defined over $[0, L]^2$ representing the distance between two positions on the curve:

$$\phi(u, v) = \|\mathcal{C}(u) - \mathcal{C}(v)\|^p$$

where $p > 1$ is a real exponent. The length of the zero level set of $\phi_{\mathcal{C}}$,

$$|\mathcal{Z}_{\mathcal{C}}| = \int_0^L \int_0^L \delta(\phi(u, v)) \, \|\nabla\phi(u, v)\| \, \mathrm{d}u\mathrm{d}v, \tag{19}$$

quantifies the self-overlap of $\mathcal{C}$.

Proposition:
If $\mathcal{C}$ is simple, i.e. without self-intersection and self-tangency, then $|\mathcal{Z}_{\mathcal{C}}| = L\sqrt{2}$.

Proof:
As a preliminary calculation, let us express the gradient of ϕ (partial derivatives are written using the indexed notation):

$$\begin{aligned}\nabla\phi(u, v) &= [\phi_u(u, v) \quad \phi_v(u, v)]^T \\ &= p\,\|\mathcal{C}(u) - \mathcal{C}(v)\|^{p-2} \begin{bmatrix} \mathcal{C}'(u) \cdot (\mathcal{C}(u) - \mathcal{C}(v)) \\ -\mathcal{C}'(v) \cdot (\mathcal{C}(u) - \mathcal{C}(v)) \end{bmatrix}\end{aligned}$$

If $\mathcal{C}$ is regular and simple, varying with respect to u in range $[0, L]$, $\phi(u, v)$ is nowhere zero except when $u = v$. Hence, for a fixed v, we have:

$$\delta(\phi(u, v)) = \frac{\delta(u - v)}{|\phi_u(v, v)|} \tag{20}$$

Integrating (20) into (19) and applying the definition of measure δ:

$$\begin{aligned}|\mathcal{Z}_{\mathcal{C}}| &= \int_0^L \int_0^L \delta(\phi(u, v)) \, \|\nabla\phi(u, v)\| \, \mathrm{d}u\mathrm{d}v \\ &= \int_0^L \int_0^L \frac{\delta(u - v)}{|\phi_u(v, v)|} \, \|\nabla\phi(u, v)\| \, \mathrm{d}u\mathrm{d}v \\ &= \int_0^L \frac{\|\nabla\phi(v, v)\|}{|\phi_u(v, v)|} \mathrm{d}v\end{aligned}$$

Expanding the gradient gives:

$$\begin{aligned}|\mathcal{Z}_{\mathcal{C}}| &= \int_0^L \left\{ \frac{p\,\|\mathcal{C}(v) - \mathcal{C}(v)\|^{p-2}}{p\,\|\mathcal{C}(v) - \mathcal{C}(v)\|^{p-2}} \right. \\ &\qquad \left. \frac{\sqrt{2(\mathcal{C}'(v) \cdot (\mathcal{C}(v) - \mathcal{C}(v)))^2}}{\left|\mathcal{C}'(v) \cdot (\mathcal{C}(v) - \mathcal{C}(v))\right|} \right\} \mathrm{d}v \\ &= \int_0^L \sqrt{2} \, \mathrm{d}v \\ &= L\sqrt{2}\end{aligned}$$

A.2 Exteriority term

Let $\mathcal{C}$ be a piecewise-smooth regular curve parameterized over $[0, 1]$. If it is simple and positively oriented such that normal vector $\mathcal{C}'^{\perp}$ points inward, its inner area may be expressed using Green's theorem:

$$\begin{aligned}|\Omega_{\mathrm{in}}(\mathcal{C})| &= \frac{1}{2} \int_0^1 \mathcal{C}^{\perp}(u) \cdot \mathcal{C}'(u) \, \mathrm{d}u \\ &= \frac{1}{2} \int_0^1 x(u)y'(u) - x'(u)y(u) \, \mathrm{d}u\end{aligned}$$

When one calculates the previous expression on a non-simple closed curve, one gets the signed area, in which positively

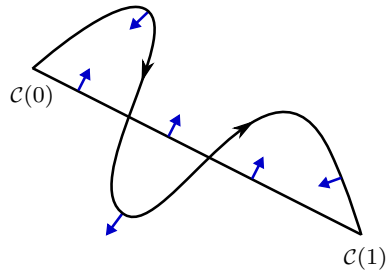


Fig. 15 The exteriority of an open curve is measured as the signed area of the multiple connected region that it forms with the line segment joining its two endpoints.

and negatively oriented connected components have positive and negative contributions, respectively.

Proposition:
The signed area formed by an open curve $\mathcal{C}$ over $[0,1]$ and the line segment from $\mathcal{C}(1)$ returning to $\mathcal{C}(0)$ (see Fig. 15), which we use to as the *exteriority* measure in Section 5.2, may be expressed as:

$$\mathcal{X}[\mathcal{C}] = \frac{1}{2}\int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u + \frac{1}{2}\mathcal{C}^\perp(1)\cdot\mathcal{C}(0)$$

Proof:
Let S be the parametrization of the line segment joining $\mathcal{C}(1)$ and $\mathcal{C}(0)$, over $[0,1]$:

$$S(u) = (1-u)\mathcal{C}(1) + u\mathcal{C}(0)$$

The signed area is then obtained by applying Green's theorem on a piecewise basis:

$$\begin{aligned}
\mathcal{X}[\mathcal{C}] &= \frac{1}{2}\int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u + \frac{1}{2}\int_0^1 S^\perp(u)\cdot S'(u)\mathrm{d}u \\
&= \frac{1}{2}\int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u \\
&\quad + \frac{1}{2}\int_0^1 ((1-u)\mathcal{C}^\perp(1) + u\mathcal{C}^\perp(0))\cdot(\mathcal{C}(0)-\mathcal{C}(1))\mathrm{d}u \\
&= \frac{1}{2}\int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u \\
&\quad + \frac{1}{2}\int_0^1 (1-u)\mathcal{C}^\perp(1)\cdot\mathcal{C}(0) + u\mathcal{C}^\perp(1)\cdot\mathcal{C}(0)\mathrm{d}u \\
&= \frac{1}{2}\int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u + \frac{1}{2}\int_0^1 \mathcal{C}^\perp(1)\cdot\mathcal{C}(0)\mathrm{d}u \\
&= \frac{1}{2}\int_0^1 \mathcal{C}^\perp \cdot \mathcal{C}' \mathrm{d}u + \frac{1}{2}\mathcal{C}^\perp(1)\cdot\mathcal{C}(0)
\end{aligned}$$

References

1. C.C. Adams. *The Knot Book: An elementary introduction to the mathematical theory of knots*. American Mathematical Society, 2004.
2. B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):106–118, 2006.
3. V.I. Arnold. Plane curves, their invariants, perestroikas and classification. *Advances in Soviet Mathematics: Singularities and Bifurcations*, 21:33–91, 1994.
4. N. Ben-Zadok, T. Riklin-Raviv, and N. Kiryati. Interactive level-set segmentation for image guided therapy. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, pages 1079–1082, Boston, USA, 2009.
5. F. Benmansour and L. Cohen. Fast object segmentation by growing minimal paths from a single point on 2D or 3D images. *Journal of Mathematical Imaging and Vision*, 33(2):209–221, 2009.
6. Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.
7. X. Bresson, S. Esedoglu, P. Vandergheynst, J-P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and vision*, 28(2):151–167, 2007.
8. T. Brox and D. Cremers. On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional. *International Journal of Computer Vision*, 84(2):184–193, 2009.
9. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
10. T. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.
11. T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.
12. L. Cohen and R. Kimmel. Global minimum for active contour models: a minimal path approach. *International Journal of Computer Vision*, 24(1):57–78, 1997.
13. M.G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27:1–67, 1992.
14. D. Cremers, O. Fluck, M. Rousson, and S. Aharon. A probabilistic level set formulation for interactive organ segmentation. In *SPIE Medical Imaging*, volume 6512, San Diego, USA, 2007.
15. D. Eppstein. Finding the k shortest paths. *SIAM Journal of Computing*, 28(2):652–673, 1998.
16. A. Falcão, J. Stolfi, and R. Lotufo. The Image Foresting Transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.
17. A. Falcão, J. Udupa, and F. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live Wire on the Fly. *IEEE Transactions on Medical Imaging*, 19(1):55–62, 2000.
18. Y. Gao, R. Kikinis, S. Bouix, M. Shenton, and A. Tannenbaum. A 3D interactive multi-object segmentation tool using local robust statistics driven active contours. *Medical Image Analysis*, 16(6):1216–1227, 2012.
19. O. Gérard, T. Deschamps, M. Greff, and L.D. Cohen. Real-time interactive path extraction with on-the-fly adaptation of the external forces. In *European Conference on Computer Vision (ECCV)*, volume 3, pages 807–821, Copenhagen, Denmark, 2002.
20. L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.
21. J. Ivins and J. Porrill. Active region models for segmenting textures and colours. *Image and Vision Computing*, 13(5):431–438, 1995.

22. P. Karasev, I. Kolesov, K. Fritscher, P. Vela, P. Mitchell, and A. Tannenbaum. Interactive medical image segmentation using PDE control of active contours. *IEEE Transactions on Medical Imaging*, 32(11):2127–2139, 2013.
23. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
24. V. Kaul, A. Yezzi, and Y. Tsai. Detection of curves with unknown endpoints and arbitrary topology using minimal paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1952–1965, 2012.
25. J. Kim, J.W. Fisher, A. Yezzi, M. Çetin, and A.S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005.
26. S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008.
27. Y. Li, J. Sun, C-K. Tang, and H-Y. Shum. Lazy snapping. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2004*, 23(3):303–308, 2006.
28. R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.
29. O. Michailovich, Y. Rathi, and A. Tannenbaum. Image segmentation using active contours driven by the Bhattacharyya gradient flow. *IEEE Transactions on Image Processing*, 16(11):2787–2801, 2007.
30. J. Mille, S. Bougleux, and L. Cohen. Minimally overlapping paths sets for closed contour extraction. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, Rome, Italy, 2012.
31. J. Mille, S. Bougleux, and L. Cohen. Combination of paths for interactive segmentation. In *British Machine Vision Conference (BMVC)*, Bristol, UK, 2013.
32. J. Mille and L. Cohen. Geodesically linked active contours: evolution strategy based on minimal paths. In 2nd *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 5567 of *LNCS*, pages 163–174, Voss, Norway, 2009. Springer.
33. P. Miranda, A. Falcão, and T. Spina. Riverbed: A novel user-steered image segmentation method based on optimum boundary tracking. *IEEE Transactions on Image Processing*, 21(6):3042–3052, 2012.
34. E. Mortensen and W. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998.
35. D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
36. N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.
37. C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.
38. C. Sagiv, N. Sochen, and Y. Zeevi. Integrated active contours for texture segmentation. *IEEE Transactions on Image Processing*, 15(6):1633–1646, 2006.
39. J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Science*, 93(4):1591–1595, 1996.
40. J.A. Sethian. *Level Sets Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.
41. J.N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.
42. G. Unal, A. Yezzi, and H. Krim. Information-theoretic active polygons for unsupervised texture segmentation. *International Journal of Computer Vision*, 62(3):199–220, 2005.
43. S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, USA, 2008.
44. H. Whitney. On regular closed curves in the plane. *Compositio Mathematica*, 4:276–284, 1937.
45. J.Y. Yen. Finding the K shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.