International Journal of Computer Vision, ??, 1–26 (??)

Global Minimum for Active Contour Models: A Minimal Path approach

LAURENT D. COHEN
Cohen@ceremade.dauphine.fr
CEREMADE, U.R.A. CNRS 749, Université Paris IX - Dauphine.
Place du Marechal de Lattre de Tassigny 75775 Paris CEDEX 16, France

RON KIMMEL*
Ron@csr.lbl.gov
Lawrence Berkeley National Laboratory, Berkeley,
Mailstop 50A-2152, LBNL, Berkeley, CA 94720, USA

Received ??. Revised ??.
Editor:

Abstract. A new boundary detection approach for shape modeling is presented. It detects the global minimum of an active contour model's energy between two points. Initialization is made easier and the curve is not trapped at a local minimum by spurious edges. We modify the "snake" energy by including the internal regularization term in the external potential term. Our method is based on finding a path of minimal length in a Riemannian metric. We then make use of a new efficient numerical method to find this shortest path.

It is shown that the proposed energy, though based only on a potential integrated along the curve, imposes a regularization effect like snakes. We explore the relation between the maximum curvature along the resulting contour and the potential generated from the image.

The method is capable to close contours, given only one point on the objects' boundary by using a topology–based saddle search routine.

We show examples of our method applied to real aerial and medical images.

Keywords: Shape modeling, Deformable Models–Snakes, Path of minimal cost, Level Sets, Segmentation, Feature Extraction, Energy Minimization, Partial Differential Equations, Curve Evolution.

1. Introduction

An active contour model for boundary integration and features extraction, introduced in (Kass et al. 1988), has been considerably used and studied during the last decade. Most of the approaches that were introduced since then try to overcome the main drawbacks of this model: *initialization, minimization and topology changes.*

The model requires the user to input an initial curve close to the goal. It has to be a very precise polygon approximation and it may be fastidious to use when we deal with a large number of images. In a sequence of images where there are small changes between two successive images, once initialization is made for the first image, it is possible to use the resulting contour of the first image as initial condition for the second and so on, as proposed in (Kass et al. 1988). Using the balloon model (Cohen 1991) allows a less demanding initialization since any initial closed curve inside an object may be used to obtain its complete boundary. In some cases, it enables a completely automatic initialization. For example, in (Cohen 1991) preprocessing is used

* This work was supported in part by the Applied Mathematics Subprogram of the Office of Energy Research under DE-AC03-76SF00098, and by the ONR grant N00014-96-1-0381.

to get an initial guess that has to be inside the desired area. The same property can be realized using the geometric model introduced in (Caselles et al. 1993), (Malladi et al. 1994) and recently improved in (Caselles et al. 1995). In (Neuenschwander et al. 1994), only two end points on the boundary are needed to follow the contour. Also, based on simulated annealing, a minimal path between two points is obtained in (Grzeszczuk and Levin 1994).

Although the smoothing effect of the snakes may overcome small defaults in the data, spurious edges generated by noise or in a complex image may stop the evolution of the curve so that it might be trapped by an insignificant local minimum of the energy. The inflation or expansion force (Cohen 1991) may help the contour to avoid isolated edges that may trap it into a local minimum. A region based approach introduced in (Cohen et al. 1993) also makes the solution less sensitive to local minima and initialization. It considers a mixed energy including a snake like term on the boundary and an homogeneous value constraint inside the region.

For segmenting several objects simultaneously or an object with holes, it is possible (Caselles et al. 1993), (Malladi et al. 1994) to model the contour as a level set of a surface, allowing it to change its topology in a natural way. These approaches have motivated many other recent works like (Caselles et al. 1995), (Caselles et al. 1996), (Whitaker 1995), (Kichenassamy et al. 1995), (Tek and Kimia 1995) for 2D and 3D implicit deformable models. Other models that can handle topology changes have also been used for curves (McInerney and Terzopoulos 1995) or surfaces (Leitner and Cinquin 1991), (Szeliski and Tonnesen 1992), (Szeliski et al. 1993).

In this paper we present a new approach for finding the global minimum of energy minimizing curves given only one or two end points. Our goal is to help the user to solve the boundary detection problem by mapping it into a single minimum problem. The proposed method contributes to the improvement of the first two items above, *initialization* and *minimization* which are obviously related. Only end points are needed as an easy initialization and we are guaranteed that the global minimum is found between these points.

We modify the snake energy in a way that makes it 'intrinsic' or free of the parameterization. Most of the classical snake models are non-intrinsic models. Therefore, different parameterizations of the same initial curve (i.e. having the same geometric shape), could lead to different solutions. The modification we follow enables us to include the internal regularization term in the external potential term in a natural way. The snake energy now depends only on the location of the point and not on the geometry of the curve at this point.

We use an evolution scheme that computes at each image pixel the energy along the path of minimal integrated energy joining that pixel to the given start point. We use Sethian's Fast Marching Method (Sethian 1996), (Sethian 1995), (Adalsteinsson et al. 1996). The search for a minimal path is then done efficiently. While this path is restricted to connect two given points, we also present a topology–based saddle search that helps in automatically closing contours by clicking on a single point along the boundary. We stress the fact that the proposed algorithm is based on a search for the minimal path and may therefore lead to meaningless classifications in some cases. Yet, since the whole process is controlled by the user, such pathological cases may be easily avoided.

An upper bound for the curvature along the minimal path is introduced. It enables a direct control over the final result by simple changes of the potential function. This justifies the fact that although our approach is a path integration, it also incorporates the regularization of the path like a "snake" model. Qualitatively, the relation between the potential and the smoothness of the result was understood and used in (Fischler et al. 1981), long before the age of snakes. Here, we introduce a quantitative bound expressing the connection between the curvature and the generated potential. This bound is useful in many applications.

The structure of the paper is as follows: Section 2 explores the relation of deformable models to the proposed model. Section 3 gives a formal definition of our edge integration procedure for the shape modeling problem, and a description of two numerical methods, leading to Sethian's Fast Marching Method. In Section 5, we explore the relation between the smoothing properties of our model and the potential. Section 6 presents an extension of our minimal path approach to find a closed boundary given a single point. Section 7 presents results of applying the proposed procedure to real images.

2. Deformable Contours

The inherent difficulty in active contour models is that searching for a minimum over a non convex functional is possible only under predefined limitations that lead to the desired solution. One possibility is to allow the user to specify an initial guess that is close to a local minimum. Starting from the user selection, like an initial given contour, a minimization scheme refines the initial guess to fit it to the given image data. The global minimum of the given functional does not necessarily make sense and initial and boundary conditions are important in the process of locating the desired local minimum. A global minimum is meaningless in the case of free end points or closed curves, since in both cases, the curve can vanish into a single point at a global minimum of the potential (which is then 0). In other cases, where some points known to be part of a contour are given as fixed end points or as a constraint for a closed curve to pass through, it is more sensible to search for the minimal path between the end points. Roughly speaking, we can distinguish between "good" and "bad" local minima for snakes. The bad local minima that we would like to avoid are those that trap the curve in noisy areas as shown in Figure 8. The desired solution is usually found by active contours with an adequate initialization. It is also a local minimum when it has free ends, and actually each boundary in the image corresponds to such a minimum. However, when the curve is forced to pass through some given points along the same boundary, we may assume that the global minimum is the desired solution, since the potential should be the smallest along the path that joins the end points[1]. Our approach gives the global minimum path between two end points, and thereby simplifies the initialization process in this case.

To motivate the proposed solution let us explore its relation to the classical active contour model.

Since the introduction of "snakes" (Kass et al. 1988), deformable models have been often used to integrate boundaries and extract features from images. The extraction of local features is specified by initial conditions that lead to the selection of one of the local minima. Snakes are a special case of deformable models as presented in (Terzopoulos 1987). The deformable contour model is a mapping:

$$\begin{aligned} \mathcal{C}(v) : \Omega &\longrightarrow \quad \mathbb{R}^2 \\ v &\longmapsto (x(v), y(v)), \end{aligned} \tag{1}$$

where $\Omega = [0, 1]$ is the parameterization interval. In some cases v is chosen to be the arc-length parameter, and then $\Omega = [0, L]$ where L is the length of the curve[2]. In some other cases, like periodic closed curves, $\Omega = S^1$ is the unit circle (in this case the parameter v is a mapping from the unit circle to the curve). The deformable model is a space of admissible paths or deformations $\mathcal{A}$ and a functional E. This functional represents the energy of the model which will be minimized on $\mathcal{A}$ and has the following form:

$$\begin{aligned} E: \quad & \mathcal{A} \to \mathbb{R} \\ & \mathcal{C} \mapsto E(\mathcal{C}) \ = \end{aligned} \tag{2}$$

$$\int_\Omega \frac{w_1}{2}\|\mathcal{C}'(v)\|^2 + \frac{w_2}{2}\|\mathcal{C}''(v)\|^2 + P(\mathcal{C}(v))dv,$$

where $\mathcal{C}'$ and $\mathcal{C}''$ are the first and second derivatives of $\mathcal{C}$ with respect to v, and P is the potential associated to the external forces. The potential is computed as a function of the image data according to the desired goal. If, for example, we want the snake to be attracted by edges, the potential should depend on the image gradient. For the problem to be well-posed, the space of admissible deformations $\mathcal{A}$ is restricted by boundary conditions. These may be free boundaries, as in the original snakes (Kass et al. 1988), cyclic boundaries by using periodic closed curves (Terzopoulos 1987), or fixed end points by giving $\mathcal{C}(0)$, $\mathcal{C}'(0)$, $\mathcal{C}(1)$ and $\mathcal{C}'(1)$

(Cohen 1991), (Cohen and Cohen 1993). The mechanical properties of the model are controlled by the functions or constants w_j.

If $\mathcal{C}$ is a local minimum of E, it satisfies the associated Euler-Lagrange equation:

$$\begin{cases} -(w_1\mathcal{C}')' + (w_2\mathcal{C}'')'' + \nabla P(\mathcal{C}) = 0 \\ \text{given boundary conditions.} \end{cases} \tag{3}$$

In this formulation each term appears as a force acting on the curve. A solution can be viewed either as satisfying the equilibrium of the forces in the Euler Lagrange equation or as reaching a minimum of the energy. Thus the curve is under the control of two kinds of forces:

- The internal forces (the first two terms) which impose the regularity on the curve. The choice of constants w_1 and w_2 determines the elasticity and rigidity of the curve.
- The image force (the potential term) pushes the curve to the significant lines which correspond to the desired attributes. It is defined by a potential of the form $\int_0^1 P(\mathcal{C}(v))dv$ where for example

  $$P(\mathcal{C}) = g(\|\nabla I(\mathcal{C})\|). \tag{4}$$

  Here, I denotes the image and $g(\cdot)$ is a decreasing function. In the classical snakes (Kass et al. 1988), we have $g(x) = -x^2$. The curve is then attracted by the local minima of the potential, i.e. edges (see (Fua and Leclerc 1990) for a more complete discussion of the relationship between minimizing the energy and locating contours). Other forces can be added to impose constraints defined by the user. As introduced in (Cohen 1991), previous local edge detection might be taken into account as data for defining the potential.

A geometric approach for deformable models was recently introduced in (Caselles et al. 1993), (Malladi et al. 1995). A level set approach for curve evolution (Osher and Sethian 1988), (Sethian 1989) is used to implement a planar curve evolution of the form:

$$\frac{\partial \mathcal{C}(s,\tau)}{\partial \tau} = P(\mathcal{C})(\frac{\partial^2 \mathcal{C}}{\partial s^2} + w\ \vec{n}), \tag{5}$$

where s is the arc-length parameter of the curve $\mathcal{C}$ in this case. Therefore, $\frac{\partial^2 \mathcal{C}}{\partial s^2} \equiv \kappa \vec{n}$ is the curvature vector ($\vec{n}$ is the unit normal), and w is some predefined constant. This constant term is thus similar to the pressure force introduced for the balloon model (Cohen 1991). It is also related to the dilatation transform in mathematical morphology and the grass-fire transform (Leymarie and Levine 1993).

It was shown that the geometric snakes model performs better than the classical snakes in some cases like topology changes when implemented by the implicit embedding function technique proposed by Osher and Sethian (Osher and Sethian 1988). It was recently proven that introducing the 'gradient of potential' (∇P) term of the classical energy minimization snakes (Kass et al. 1988), (Cohen and Cohen 1993), (Leymarie and Levine 1993) into the geometric snakes (Caselles et al. 1993), (Malladi et al. 1995), (Malladi et al. 1994) is based on geometrical as well as energy minimization reasoning, leading to the "geodesic active contour" (Caselles et al. 1995).

The basic idea of the geometric model is that the curve follows an evolution by expansion in the normal direction, with lower speed when $P(\mathcal{C})$ is small. Yet, it never comes to a complete stop, and heuristic stopping procedures are used to switch off the evolution process when an edge is reached. The 'gradient of potential' term added in the geometric model forces it to stop at the boundary similar to the image force in the classical active contours.

The geodesic active contours (Caselles et al. 1995) were shown to 'behave' better than both its 'ancestors' since they enjoy the advantages of both. Given an initial curve $\mathcal{C}(s,0)$, the geodesic active contours is based on the planar evolution equation

$$\frac{\partial \mathcal{C}(s,\tau)}{\partial \tau} = P(\mathcal{C})\frac{\partial^2 \mathcal{C}}{\partial s^2} - \langle \nabla P, \vec{n} \rangle \vec{n}, \tag{6}$$

where s is the arclength. There is a major difference between (5) and (6). In (5), the geometric snake evolution is slower when the potential is small but the curve does not necessary stop completely at the boundary. It may reduce its speed but keep on propagating since it never reaches an equilibrium. This might be a drawback when part of the initial curve is close to the boundary and part of it is far. When the further part of the curve has reached the boundary, the closer part may already have passed through. In (6), the curve reaches an equilibrium which is similar to the classical snakes. The ∇P term is a projection of the attraction force $-\nabla P$ on the normal to the curve. This force balances the other term close to the boundary and causes the curve to stop there.

It is shown in (Caselles et al. 1995) that (6) is a result of minimizing the functional

$$E(\mathcal{C}) = \int_{\Omega} P(\mathcal{C}(s))ds, \tag{7}$$

where s is the arclength (or $E(\mathcal{C}) = \int_{\Omega} P(\mathcal{C}(v))||\mathcal{C}'(v)||dv$, for the arbitrary parameter v). The curve evolution equation is then reformulated and implemented using the Osher-Sethian numerical algorithm (Osher and Sethian 1988). Similar geometric models were also introduced in (Kichenassamy et al. 1995), (Whitaker 1995), (Shah 1996) and extended to color and texture in (Sapiro 1996).

Although our work is related to (Caselles et al. 1995), it is a totally independent approach. Actually, the geodesic active contours may be considered as a natural refinement procedure to the proposed approach. We note that following the formulation of (Caselles et al. 1995), the minimization of the classical energy (2) may be modified into the problem of finding *local geodesics* in a Riemannian metric computed from the image, where we propose to find the *minimal geodesics* in a similar Riemannian metric (see Equation (8) in the following section). Although it is shown in (Caselles et al. 1995) that the solution of active contour models is closely related to geodesics, no method is proposed to find the minimal ones. In general, active contours models search for a local minimum that is close to the initial guess, while we propose a method to find the minimal path (minimal geodesic) of the same energy between two points.

3. Paths of Minimal Action

Given some potential P that takes lower values near the edges or features, our goal is to find a single contour that best fits the boundary of a given object or a line of interest. This 'best fit' question leads to algorithms that seek for the minimal path, i.e. paths along which the integration over P is minimal. As mentioned earlier, snakes start from a path close to the solution and converge to a local minimum of the energy. Given only the end points, our goal is to find the minimal path between these points, thereby simplifying the initialization process and avoiding erroneous local minima. At first glance, this limits the problem to the type of boundary conditions with fixed end points, however, as we will see in Section 6, the proposed approach may also be used for closed contours. Motivated by the ideas put forward in (Kimmel et al. 1995), (Kimmel et al. 1996) we develop an efficient and consistent method to find the path of minimal cost between two points, using the surface of minimal action (Rutovitz 1968), (Kimmel et al. 1996), (Verbeek and Verwer 1990) and the fact that operating on a given potential (cost) function helps in finding the solution for our path of minimal action (also known as minimal geodesic, or path of minimal potential). Thereby, we are able to isolate the boundary of a given object in the image.

3.1. *Problem Formulation*

The minimization problem we are trying to solve is slightly different from the deformable models, though there is much in common. One may still differ between "internal" and "external" forces, yet now all terms are geometric, which means a result of an intrinsic energy functional. Contrary to the classical snake energy, here s represents the arc-length parameter, which means that $||\frac{\partial \mathcal{C}}{\partial s}(s)|| = 1$. The reason we modified the energy is that we now have an expression in which the internal regularization energy is included in the potential term in a natural way. We can then solve the energy minimization in a similar way to that of finding the shortest path on a surface using the method developed in (Kimmel et al. 1995). The fact that the energy integral is now intrinsic will also help us to explore the relation between the smoothness of the result and the potential. The energy of the new model has the following form:

$$\begin{aligned}\mathcal{A}_{p_0,p_1} &\rightarrow I\!R \\ \mathcal{C} &\mapsto E(\mathcal{C}) = \int_\Omega w\|\frac{\partial\mathcal{C}}{\partial s}(s)\|^2 + P(\mathcal{C}(s))ds \\ = wL(\mathcal{C}) &+ \int_\Omega P(\mathcal{C}(s))ds = \int_\Omega \tilde{P}(\mathcal{C}(s))ds,\end{aligned} \tag{8}$$

with

$$\tilde{P}(p) = w + P(p). \tag{9}$$

Here $\mathcal{A}_{p_0,p_1}$ is the space of all curves connecting two given points (restriction by boundary conditions): $\mathcal{C}(0) = p_0$ and $\mathcal{C}(L) = p_1$, where L is the length of the curve. Contrary to the classical snake energy, here s represents the arc-length parameter. So, Eq. (8) should actually be read as

$$\mathcal{C} \mapsto E(\mathcal{C}) = \int_\Omega (w + P(\mathcal{C}(v)))\, \|C'(v)\| dv, \tag{10}$$

for an arbitrary parameter v. This makes the energy depend only on the geometric curve and not on the parameterization. The regularization term multiplied by the constant w, now measures the length of the curve. We note that a similar regularization effect may be also achieved by smoothing the potential P (Fischler et al. 1981). Section 5 gives more details about the smoothing effects of the energy.

Having the above minimization problem in mind, we first search for the *surface of minimal action* U_0 that starts at $p_0 = \mathcal{C}(0)$. At each point p of the image plane, the value of this surface U_0 corresponds to the minimal energy integrated along a path that starts at p_0 and ends at p.

$$U_0(p) = \inf_{\mathcal{C}(L)=p} \left\{ \int_\mathcal{C} \tilde{P} ds \right\} = \inf_{\mathcal{A}_{p_0,p}} E(\mathcal{C}), \tag{11}$$

where s is the arclength parameter.

We next show how to determine the value of U_0 everywhere in the image domain.

3.2. ***Shortest Paths As a Set***

Following (Kimmel et al. 1995), (Kimmel 1995), given the minimal action surfaces U_0 to p_0 and U_1 to p_1, then the minimal geodesic between p_0 and p_1 is exactly the set of coordinate points p_g that satisfy

$$U_0(p_g) + U_1(p_g) = \inf_{p \in I\!R^2} \{U_0(p) + U_1(p)\}. \tag{12}$$

Usually, the set of points p_g needs to be refined from a given "fat" set of points into a curve. Since we operate on a discrete data, in order to keep the two end points connected by the minimal set we need to threshold the function $U_0 + U_1$ using a value larger than its infimum. This operation results in a fat set. In (Kimmel et al. 1996) a thinning algorithm was applied. In our case, a natural refinement of this set is to select any curve in the set connecting the two points, and apply a local minimization based on the Euler-Lagrange equations minimizing the same functional. The geodesic active contours without the constant term and fixed end points is the right flow for this case. Observe that it should operate only within the "fat" set, which can be considered as a fixed narrow band (Adalsteinsson and Sethian 1995), thereby reducing drastically the computational complexity of this refining.

When there are two or more minimal paths, as we will see in Section 6, the destination point p_1 is a saddle point and each path can be obtained from one of the decreasing directions at p_1. By using the sum of the two distances, one can simultaneously obtain all minimal paths.

The above is a global way for extracting the global minimum. In our experiments we have preferred to use a back propagation procedure that results in a single curve (see Section 4.5.)

3.3. ***Minimal Action Level Sets Evolution***

In what follows, we assume that $P \geq 0$. For the minimization of our energy (8), let us first formulate a partial differential evolution equation for the set of equal energy contours $\mathcal{L}$ in 'time', where

t is in fact the value of the energy. These are the level sets of U_0 defined by Equation (11). In the evolution equation, t represents the height of the level set of U_0:

$$\frac{\partial \mathcal{L}(v,t)}{\partial t} = \frac{1}{\tilde{P}}\vec{n}(v,t), \tag{13}$$

where $\tilde{P} = P + w$ and $\vec{n}(v,t)$ is the normal to the closed curve $\mathcal{L}(.,t) : S^1 \to \mathbb{R}^2$. The motivation for this evolution is that we need to propagate with a velocity that is proportional to the inverse of the penalty. So that at 'low cost' area the velocity is high while at a 'high cost' area the velocity is low.

The curve $\mathcal{L}(.,t)$ corresponds here to the set of points p for which the minimal energy $U_0(p)$ is t:

$$\{\mathcal{L}(v,t), v \in S^1\} \;=\; \{p \in \mathbb{R}^2 \mid U_0(p) = t\}. \tag{14}$$

This evolution equation is initialized by a curve $\mathcal{L}(.,0)$ which is a small circle around the point p_0. It corresponds to a null energy. It then evolves according to Equation (13), similar to a balloon evolution (Cohen 1991) with an inflation force depending on the potential. Considering the (x,y,t)-space, the family of curves $\mathcal{L}(.,t)$ constructs the level sets of the surface $U(x,y) : \mathbb{R}^2 \to \mathbb{R}^+$ defined in (11). The t level set of U is exactly the curve $\mathcal{L}(.,t)$. Although a rigorous proof of this statement can be found in (Bruckstein 1988), it can be understood simply by the following geometric interpretation. Observe that when we add to a path that ends at a point of $\mathcal{L}(.,t)$ a small segment in the normal direction to $\mathcal{L}(.,t)$ and of length $\frac{1}{\tilde{P}}dt$, we add to the accumulated energy of (8) a contribution of $\tilde{P}\frac{1}{\tilde{P}}dt = dt$. This means that the new point is on the level $t + dt$, that is on the curve $\mathcal{L}(.,t+dt)$. Figures 13 and 14 presents such a surface U and its corresponding level sets.

It is possible to compute the surface U in several ways. We shall describe three of them that are consistent with the continuous case while implemented on a rectangular grid. It is, however, possible to implement a simple approximation like the shading from shape algorithm introduced in (Verbeek and Verwer 1990), or even graph search based algorithms (see Section 4.1), if consistency with the continuous case is not important, see also (Rutovitz 1968).

4. Numerical Implementation

The numerical schemes we propose are consistent with the continuous propagation rule. The consistency condition guarantees that the solution converges to the true one as the grid is refined. This is known **not** to be the case in general graph search algorithms that suffer from *digitization bias* due to the *metrication error* when implemented on a grid (Mitchell et al. 1987), (Kiryati and Székely 1993). This gives a clear advantage to our method over minimal path estimation using graph search. Before we introduce the proposed method, let us review the graph search based methods that try to minimize the energy given in (8).

4.1. Graph Search Algorithms and Metrication Error

To evaluate and minimize the snake energy (2), the "internal" terms can be evaluated only from the shape of the curve, leading to curve deformation and evolution schemes from an initial curve. Based on the new energy definition (8), we are able to compute the final path without evolving an initial contour, by using the surface of minimal action. To find the surface of minimal action, graph search and dynamic programming techniques were often used, where the image pixels serve as vertices in a graph (Montanari 1971), (Fischler et al. 1981), (Chandran et al. 1991).

A description of A^* and F^* algorithms, applied to road detection, can be found in (Fischler et al. 1981). The distance image is initialized with value ∞ everywhere except at a start point with value zero. At each iteration, the A^* algorithm expands to a neighbor pixel a previously obtained minimal path ending at the vertex with smallest current cost value. Since at each iteration one pixel gets a final value, and a search for the minimal vertex to update is performed, the algorithm complexity is $O(NlogN)$ where N is the number of pixels in the image. Our approach solves a continuous version of the problem.

Sethian Fast Marching Method (Sethian 1996), described in section 4.4, has a similar complexity, yet it is consistent!

The A^* algorithm has to search among all vertices, the one to expand at each iteration. This is why the F^* algorithm was preferred in several applications. The F^* algorithm (so called in (Fischler et al. 1981)) computes the distance with a sequential update of the pixels. It is similar in spirit to the algorithm used in Section 4.3 (see also (Dupuis and Oliensis 1994)), except that Equation (18) is again consistent. Using the F^*, the global minimum is reached only after the image is scanned iteratively top to bottom, row by row, left to right followed by right to left, and then bottom to top. The number of such passes depends on the shape of the minimal path, which is usually unknown in advance. If that path expands from the starting point monotonically with respect to the row index, one pass is sufficient. However, if it has a spiral shape from the starting point, it needs as much iterations as turns in the path, to propagate the information from the start point to the end point. The resulting complexity is of $O(N \int \kappa ds)$, where the integral is along the longest path and κ is the curvature ($\int \kappa ds / 2\pi =$ number of loops of a planar curve). In practice, the iterations are stopped either when there is no more change in the process (this has to happen in a fixed number of iterations) or after a given number of passes. This kind of approach was used to compute distance maps in (Borgefors 1984), (Danielsson 1980). It was also used for road detection in (Merlet and Zerubia 1993), (Merlet and Zerubia 1994), using some improvements in the potential definition. The authors also add some constraints on the curvature by taking into account sets of three vertices instead of two in the graph search to update the distance. In their algorithm, they find that 8 passes are sufficient for their applications.

Such an algorithm was used by (Geiger et al. 1995) for interactive boundary drawing giving a sequence of points on the boundary and finding the path between two successive data points. A similar approach is used in an interactive tool called live-wire (Mortensen and Barrett 1995).

A simplified F^* algorithm is used in (Chandran et al. 1991) to minimize a snake energy. It assumes the path expands from the starting point only in a restricted range of directions and makes only one pass. Thus, it only finds the global minimum among all paths restricted by this condition. This is a problem for non monotonic paths. In case there are gaps in the potential that can lead the expansion of the path in a wrong direction, then the algorithm has no way to correct itself and to find the right path. Although these last authors generalize their approach to the continuous case, they solve it only for the discrete graph approach and their approach, as well as other graph search algorithms, is also subject to metrication error.

A completely different approach related to dynamic programming for detection of salient boundaries was introduced in (Shaashua and Ullman 1988). It defines iteratively at each pixel of the image a value of the maximal energy of a path passing through this pixel. Then high valued pixels are grouped to get salient curves. The context is different there since each pixel or vertex is considered as a start point and the algorithm should find simultaneously all interesting feature curves.

Dynamic programming has also been used for snakes, starting with (Amini et al. 1990). Although a complete theoretical description of continuous dynamic programming is reviewed, the proposed application to active contours is different from the ones above and our approach. The dynamic programming minimization is not applied there to find a minimal path between two points but to find the local deformation from an initial curve that gives the best energy descent. This is applied iteratively from an initial curve, exactly as in the classical curve evolution scheme for snakes. However, instead of using gradient descent, it finds at each iteration the global minimization among all possible local deformations, i.e. paths obtained by giving each node of the curve the ability to move in a small neighborhood (3×3 pixels usually). This reduces considerably the size of the graph, since the vertices are the nodes on the curve and the possible values for these are only the eight neighbors of the initial vertex. In (Fujimura et al. 1992), the range of possi-

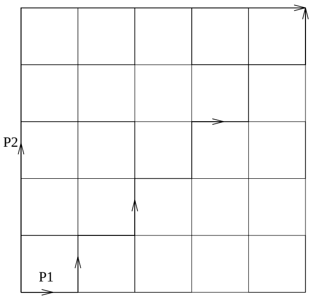

Fig. 1. An L^1 norm causes the shortest path to suffer from errors of up to 41%. In this case both P_1 and P_2 are optimal, and will stay optimal no matter how much we refine the (4-neighboring) grid.

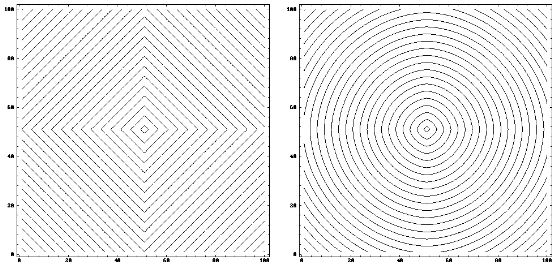

Fig. 2. Illustration of metrication error for computation of the distance map to a single point, showing level sets of the distance. On the left: a graph search-like discrete distance computation gives squares; On the right: the distance is obtained by our approach, giving circles.

ble local deformations is broadened using a multiscale dynamic programming algorithm. However, in both approaches, this kind of graph search does not avoid undesirable local minima of the energy, and the solution remains very sensitive to the initialization, as in classical snakes. Also, like classical snakes, it is non intrinsic and the same looking two initial contours with different control points may lead to completely different solutions.

One may argue that using previously mentioned graph search algorithms like the A^*, Dijkstra (Dijkstra 1959), (Sedgewick 1988), or F^* as proposed in (Fischler et al. 1981) for road tracking, might be sufficient. These algorithms are indeed efficient, yet suffer from 'metrication errors'. The graph based algorithms consider the image as a graph in which each pixel is a node, and the 4 (or 8) connections to the neighboring pixels are the vertices of the graph. The weights along these vertices are usually taken as the average of the potential at the two end pixels, multiplied by the length of the L^1 "city block" distance between these pixels (1 for horizontal and vertical connections). However, it is clear that measuring length of the shortest path between the lower left and the upper right corners of the graph in Figure 1 this way, the length of P_1 is equal to that of P_2. It does not matter how fine the grid gets, P_2 is still an optimal path. Our goal is to get the diagonal connection as the optimal path with the 'right' Euclidean distance measure (L^2) in this simple case. Our problem is that in graph search algorithms we are obligated to the distance measure imposed by the graph (L^1 in Figure 1).

Of course the result of the graph-search could be improved by taking a larger neighborhood as structuring element, giving better approximations of the distance in some directions (like $\sqrt{2}$ for the diagonals) (Borgefors 1984), (Thiel and Montanvert 1992).

These give a different polygonal approximation of the circle, but there will always be an error in some direction that will be invariant to the grid resolution, which is not the case in the approach we use. Also, some fixes that minimize the average error by modifying the weights along the connections between the pixels were proposed in (Kiryati and Székely 1993) and used in (Kimmel and Kiryati 1994). We show the example of Figures 1 and 2 for the simplest case of graph search, to clarify the metrication error effect.

Our philosophy here is different. We propose to deal with the continuous problem as long as possible. In that, we follow the numerical analysis community, by first analyzing the underlying problem in the continuous domain. Then, at the last stage which involves numerical implementation we will consider the image given as a grid of pixels, compute optimal paths and the surface of minimum action in a relatively efficient way, while at the same time enjoy the 'consistency' property of converging to the desired continuous solution as the grid is refined. The main reason is obviously accuracy which is important for example in medical applications. As an illustration, Figure 2 shows the isodistance curves using a graph-search approach and the continuous level-set approach. These curves are squares in the first case, not depending of the size of the grid, while in our case, the curves are getting closer to a perfect circle when the size of the grid is refined.

4.2. *Front Propagation Approach*

According to this first continuous approach, the curve evolution $\mathcal{L}(.,t)$ of Equation (13) is reformulated into an evolution of an implicit representation of the curve defined by an evolving surface $\phi : \mathbb{R}^2 \times [0,T) \to \mathbb{R}$, where for each value of t, $\mathcal{L} = \phi^{-1}(0)$. This means that curve $\mathcal{L}(.,t)$ is the zero level set of $\phi(t) : \mathbb{R}^2 \to \mathbb{R}$. This Eulerian formulation for curve evolution was introduced by Osher and Sethian (Osher and Sethian 1988), (Sethian 1989) to overcome numerical difficulties and handle topological changes. As initialization for $\mathcal{L}(.,0)$, we start with an infinitesimal circle around the start point p. We mean a small one for practical implementation. The function ϕ is initialized at $t = 0$ to be negative in the interior and positive in the exterior of the curve $\mathcal{L}(.,0)$. This is obtained by setting one pixel to -1 and the rest to $+1$. The evolution rule of ϕ is then given by:

$$\frac{\partial \phi}{\partial t} = -\frac{1}{\tilde{P}}\|\nabla\phi\|. \tag{15}$$

It was this same idea of considering the curve as the zero level set of an evolving surface that initiated the geometric snake approach (Caselles et al. 1993), (Malladi et al. 1995) described in the end of Section 2. For a fast implementation, of order $O(M\frac{\int ds}{\Delta t})$ where M is the number of points in a narrow band around the front and Δt is the time step of the scheme, of the above approach we refer to (Adalsteinsson and Sethian 1995).

4.3. *Shape from Shading Approach*

The second approach is based on Rouy-Tourin shape from shading method (Rouy and Tourin 1992), (Dupuis and Oliensis 1994) and searches for the surface U itself instead of tracking its level sets as in the previous approach. In this case the surface may be found as solution of the Eikonal equation

$$\|\nabla U\| = \tilde{P}, \tag{16}$$

with $U(p_0) = 0$ at the start point. The solution U is obtained as the steady state of $\mathcal{U}(p,\tau)$ when τ is large, where $\mathcal{U}(p,\tau)$ satisfies the following evolution equation:

$$\frac{\partial \mathcal{U}}{\partial \tau} = \tilde{P} - \|\nabla\mathcal{U}\|, \tag{17}$$

given $\mathcal{U}(p_0,\tau) = 0$ at the start point as boundary condition. The limit value $U = \mathcal{U}_\infty$ is solution of the Eikonal equation (16).

We can again give a geometric interpretation that relates (13) to (16). The gradient of U is normal to its level sets $\mathcal{L}(.,t)$, and the gradient norm is thus the value of the spatial directional derivative in the normal direction. As U increases by dt, the normal displacement of the level set $\mathcal{L}(.,t)$ is $\frac{dt}{\tilde{P}}$ from (13). So the derivative $\frac{\partial U}{\partial \vec{n}} = \langle \nabla U, \vec{n}\rangle = \|\nabla U\|$ is equal to $dt/\frac{dt}{\tilde{P}} = \tilde{P}$. A rigorous proof of this idea can be found for example in (Bruckstein 1988). (see also Bellman (Bellman and Kalaba 1965) for a nice proof on the orthogonality of the wave fronts and the geodesics). Here, boundary conditions are given in the form of fixing the point $\mathcal{C}(0) = p_0$, i.e. $\mathcal{U}(p_0,\tau) = 0$ for all τ. Authors of (Rouy and Tourin 1992) also presented a direct numerical approach to solve (16) and gave a convergence proof to that minimization procedure in the viscosity solutions framework (Crandall et al. 1992). We shall discuss this method and its discretization in more details in the following section. The method we recommend is presented in the following section. It is in some sense a hybrid of both methods just described.

4.4. *Sethian Fast Marching Method*

In his recent report (Sethian 1996), Sethian presents a fast and efficient method for solving Equation (16). It is based on a clever way for propagating the information on the grid. Motivated by the two methods above, this method uses the proposed numerical scheme in (Osher and Sethian 1988), (Rouy and Tourin 1992). However, by marching in an ordered way, the problem is solved after a finite number of steps, and by that contradicting Remark 5 in (Rouy and Tourin 1992). We recommend this method for any real time application.

Given the potential values $P_{i,j} = P(i\Delta x, j\Delta y)$ on a grid (e.g. the pixel grid), the numerical method approximating $U_{i,j}$ in Eq. (16) is given by

$$\begin{aligned}&(\max\{u - U_{i-1,j}, u - U_{i+1,j}, 0\})^2 \\ &+ (\max\{u - U_{i,j-1}, u - U_{i,j+1}, 0\})^2 = P_{i,j}^2,\end{aligned} \quad (18)$$

where, for simplicity, we assume $\Delta x = \Delta y = 1$. In (Rouy and Tourin 1992) the numerical viscosity solution was obtained by solving the above equation at each grid point, selecting for $U_{i,j}$ the largest u that satisfies Eq. (18). The grid points were selected in an arbitrary way, and thus it was claimed that convergence is obtained after infinite number of such iterations. Where each iteration involves an arbitrarily selection of a grid point (i, j), and updating the value of $U_{i,j}$ at that point. In practice, it means many passes on the image.

The Fast Marching Method introduces order in the selection of the grid points. It is based on the fact that information is propagating from the source point 'outwards'. It needs only one pass on the image (see beginning of Section 4.1). Following (Sethian 1996), the method goes as follows:

- Initialization:
  - For each point in the grid, let $U_{i,j} = \infty$ (large positive value). Label all points as *far*.
  - Set the start point $(i, j) = p_0$ to be zero: $U_{p_0} = 0$, and label it *trial*.
- Marching Loop:
  - Let (i_{min}, j_{min}) be the *trial* point with the smallest U value.
  - Label the point (i_{min}, j_{min}) as *alive*, and remove it from the *trial* list.
  - For each of the 4 neighboring grid points (k, l) of (i_{min}, j_{min}):
    - If (k, l) is labeled *far*, then label it *trial*.
    - If (k, l) is not *alive*, then compute $U_{k,l}$ according to Eq. (18), selecting the largest solution to the quadratic equation, which is the only valid solution. *i.e.* solve with respect to u

$$\begin{aligned}&(\max\{u - \min\{U_{k-1,l}, U_{k+1,l}\}, 0\})^2 \\ &+ (\max\{u - \min\{U_{k,l-1}, U_{k,l+1}\}, 0\})^2 = P_{k,l}^2,\end{aligned} \quad (19)$$

and let $U_{k,l} = u$.

The algorithm is based on the fact that solving equation (19), the value at a pixel (k, l) depends only on those neighboring pixels that have lower value than $U_{k,l}$. For efficiency, the *trial* list is kept as min heap structure. We refer to (Sethian 1996), (Sethian 1995), (Adalsteinsson et al. 1996), (Kimmel and Sethian 1996) for further details on the above algorithm, as well as a proof of correct construction. Using a min-heap structure for the *trial* list, the algorithm computational complexity is $O(NlogN)$ where N is the number of grid points. It has similar complexity to that of graph search based algorithms like the A^* or Dijkstra (Dijkstra 1959), (Sedgewick 1988). For example on a SPARC 1000, it took a second to compute the U surface of a 256×256 image. This is a first order numerical scheme. As an example for accuracy we should note that the Euclidean distance ($P_{i,j} = 1$) from a straight line is accurate with sub pixel accuracy (error = 0). In general, the consistency condition guarantees that as the grid is refined, the solution converges to the true continuous one.

4.5. *Global Snake Minimization Between Two End Points*

Shortest path between p_0 and p_1. Using the approach of (Kimmel et al. 1995) described in Section 3.2, the shortest path between a start point p_0 and a destination point p_1, according to the energy minimization is the set of points $p_m = (x_m, y_m)$ that satisfy:

$$(U_0 + U_1)(x_m, y_m) = \inf_{(x,y)}\{(U_0 + U_1)(x, y)\}, \quad (20)$$

where U_0 and U_1 correspond to the minimal action obtained in the previous section with paths starting at p_0 and p_1 respectively. A natural combination is to use the above method in order to locate the minimal set, and then let the model defined in (Caselles et al. 1995) take over and refine the result. However, we recommend an easier way to compute the path by back propagation.

Back propagation from p_1. In order to determine the minimal path between p_0 and p_1, we need only to calculate U_0 and then slide back on the surface U_0 from $(p_1, U_0(p_1))$ to $(p_0, 0)$. The surface of minimal action U_0 has a convex like behavior in the sense that starting from any point $(q, U_0(q))$ on the surface, and following the gradient descent direction, we will always converge to p_0. It means that U_0 has only one local minimum that is of course the global minimum and is reached at p_0 with value zero. We show in Figures 13 and 14 an example of 3D representation of the $U_0(x, y)$ surface and a level set image of the same U_0. Given the point p_1, the path of minimal action connecting p_0 (the minimal point in U_0, $U(p_0) = 0$) and p_1 is the curve $\tilde{\mathcal{C}}(\sigma)$ starting at p_1 and following the opposite gradient direction on U_0:

$$\frac{\partial \tilde{\mathcal{C}}}{\partial \sigma} = -\nabla U_0, \quad \tilde{\mathcal{C}}(0) = p_1 \tag{21}$$

Then the solution $\mathcal{C}(s)$ is obtained by arclength parameterization of $\tilde{\mathcal{C}}(-\sigma)$ with $\mathcal{C}(0) = p_0$ and $\mathcal{C}(L) = p_1$. The minimal path can be obtained this way since ∇U is tangent to the geodesic. This is a consequence of the results in (Bellman and Kalaba 1965) that show that the light rays (geodesics, constant parameter curves) are orthogonal to the wave fronts (equal cost contours). The gradient of U is therefore orthogonal to the wave fronts since these are its level sets.

The back propagation procedure is a simple steepest gradient descent. It is possible to make a simple implementation on a rectangular grid: given a point $q = (i, j)$, the next point in the chain connecting q to p is selected to be the grid neighbor (k, l) for which $U(k, l)$ is the minimal, and so forth. Of course, a better tracking can be obtained using a more precise estimation of the gradient of U. In our examples we have chosen the discrete steepest descent just described, because of its simplicity, and the fact that it is used only for presentation purpose. Being a local operation, back propagation suffers from angular error accumulation. See (Kimmel and Sethian 1996) for a more sophisticated high order ODE integrator developed for other purposes, it is used in the examples of Figure 11.

We back track the path of minimal action connecting the two points, which is the global minimum of the snake energy defined in Eq. (8).

Using back propagation following the gradient of U, once the surface U is available, the minimal path between the start point p_0 and any other point p can be obtained without additional computation. This approach is used for example to simultaneously track four roads in the same image, as shown in Figure 9. Notice that in (Merlet and Zerubia 1993), a back propagation is made from all pixels along the image boundary. We could also apply the same idea with our approach, and then deal with non the meaningless parts of the paths.

Remark that if at some point along the path, we have $\nabla U_0 = 0$, there may be more than one path that reaches the global minimum. This will be the case in Section 6 where we find two minimal paths from a saddle point. In this case, we perform back propagation with two initial opposite directions. These directions can be found either directly, as the steepest descents on the discrete grid, or from the eigenvectors of the second differential operator.

5. Discussion on the Potential term

5.1. *Regularization properties*

We now show how the constant w and the potential P in Eq. (8) control the smoothness of the solution. A qualitative understanding of a similar control was used in (Fischler et al. 1981). Here, we first introduce quantitative results in the form of geometric bounds on the curvature of the final contour.

We shall make use of the following lemmas to introduce an upper bound on the curvature along the resulting contour $\mathcal{C}(s)$ by controlling the potential P. We also assume that the potential is given as a positive function.

Lemma 1 *Given a potential $P > 0$, the curvature magnitude $|\kappa| = \|\frac{\partial^2 \mathcal{C}}{\partial s^2}\|$ along the geodesics minimizing*

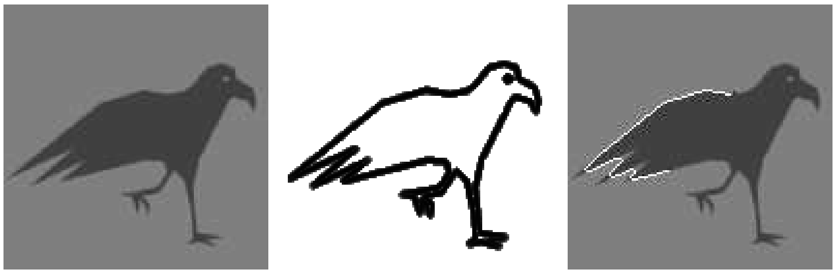

Fig. 3. Bird image: original on the left, potential in the middle and minimal path on the right.

$$\int_{\Omega} P(\mathcal{C}(s))ds, \tag{22}$$

where s is the arclength parameter, or $\int P(\mathcal{C}(v))||\mathcal{C}'(v)||dv$ for an arbitrary parameter v, is bounded by

$$|\kappa| \leq \sup_{\Omega} \left\{ \frac{||\nabla P||}{P} \right\}. \tag{23}$$

Proof: Following (Caselles et al. 1995) (see also (Dubrovin et al. 1984)), the Euler-Lagrange equation of (22) is given by

$$P\kappa\vec{n} - \langle \nabla P, \vec{n} \rangle \vec{n} = 0. \tag{24}$$

It indicates the curve's behavior at the minima of (22). This yields the following expression for the curvature along the geodesics of P:

$$\kappa = \frac{\langle \nabla P, \vec{n} \rangle}{P}. \tag{25}$$

Since $\vec{n}$ is a unit vector, the numerator is a projection on a unit vector operation. Thus, we can conclude that along any geodesic path minimizing (22) the curvature magnitude is bounded by Equation (23). □

Using Lemma 1, an *a priori* bound of the curvature magnitude may be obtained by evaluation of sup and inf over the image domain $\mathcal{D}$ instead of the curve domain Ω in (23). We readily have the following result which applies to our case with the energy of (8):

Lemma 2 *Given a potential $P \geq 0$ defined on the image domain $\mathcal{D}$, and let $\tilde{P} = w + P$, the curvature magnitude $|\kappa|$ along the geodesics minimizing the energy of (8) is bounded by*

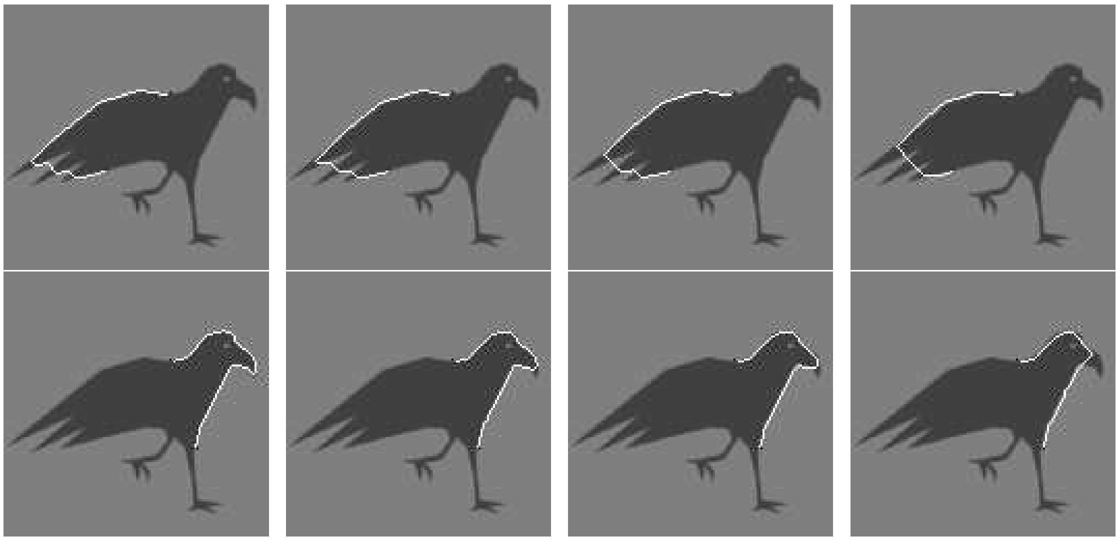

Fig. 4. Regularization effect by increasing the coefficient w from left to right.

$$|\kappa| \leq \frac{\sup_{\mathcal{D}}\{\|\nabla P\|\}}{w}. \tag{26}$$

Proof: Since $P \geq 0$ we have that $\inf_\Omega\{\tilde{P}\} \geq w$. Using this relation and Equation (23) we have:

$$\begin{aligned} |\kappa| &\leq \sup_\Omega \left\{ \frac{\|\nabla \tilde{P}\|}{\tilde{P}} \right\} = \sup_\Omega \left\{ \frac{\|\nabla P\|}{P+w} \right\} \\ &\leq \sup_{\mathcal{D}} \left\{ \frac{\|\nabla P\|}{P+w} \right\} \leq \frac{\sup_{\mathcal{D}}\{\|\nabla P\|\}}{w + \inf_{\mathcal{D}}\{P\}} \\ &\leq \frac{\sup_{\mathcal{D}}\{\|\nabla P\|\}}{w}. \end{aligned} \tag{27}$$

□

Equation (26) enables us to control the behavior of any geodesic minimizing (8), and especially the minimal geodesics that interest us. Lemma 1 also gives a nice interpretation of the connection between the curvature of the resulting contour, and the ratio between the gradient magnitude and the value of the potential P. When the curve's normal is orthogonal to the slope of P, so that the curve is directed towards the valley, then the curvature is zero implying a straight line. While if the curve travels along a contour of equal height in P, then the normal $\vec{n}$ coincides with the slope of P and the curvature increases causing the curve to bend and direct the curve to flow into the valley, where the potential is lower.

The conclusion is that to decrease the limit of the curvature magnitude of the geodesics in Equation (26), and thereby lead to a smoothing effect on the resulting contour, we have two alternatives:

- Smoothing the potential (or the image) to decrease $\sup_{\mathcal{D}}\{\|\nabla P\|\}$.
- Increasing the constant w added to P, increases the denominator without affecting $\sup_{\mathcal{D}}\{\|\nabla P\|\}$. This gives a justification for referring to w as a regularization parameter in Section 3.1.

Figure 4 shows the effect of changing w on the solution (it varies between 0.04 and 0.4). The potential shown in Figure 3 is based on the image gradient like in (4) (the range of P and ∇P is normalized between 0 and 1).

A possible application of the bound in Equation (26) would be to limit the domain in which the curve lies and thus reduce computation of the minimal action only to this area of the image.

5.2. *Attraction Potential*

As noted in Section 2, it is useful in some cases to define a potential from an edge image. These edge points may be extracted from the original image using an edge detection operator or given as a set of data points. This kind of potential is often used in the literature (see (Cohen 1995) for several possibilities of selecting such potential functions). Choosing this potential function is useful when the edge detection operation produces most of the edge points but has gaps in the contours, as shown in Figure 5. The distance based potential considers the distance from the detected edge points to be the penalty. In this case the gradient of the potential points towards the closest detected point.

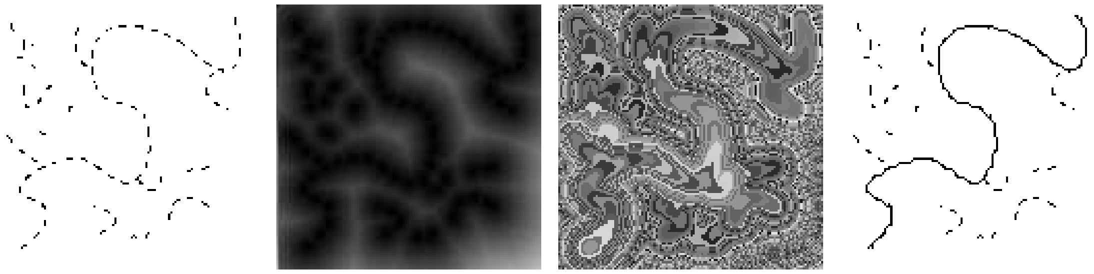

Fig. 5. Line image. From left to right: original, potential, minimal action (random look up table to show the level set propagation starting from the bottom left), minimal path between bottom left and top right.

Also, the use of such a potential may avoid node concentration at some high gradient points. Indeed, since the gradient norm usually changes its values along a boundary contour, this operation assigns an equal attraction weight along the boundary. Several approaches of generating 'attraction potentials' from such data for various reconstruction methods were surveyed in (Cohen and Cohen 1993) and a 'physical' interpretation was given as weak springs linking the curve to data points.

Let $I(x,y) : \mathcal{D} \subset I\!R^2 \rightarrow I\!R^+$ be a given gray level image. Applying a standard edge detector to I results in a set of points in the image domain ($\mathcal{D}$) some of which correspond to true edge points. These points are scattered over the image domain and serve as the key points in generating a single boundary contour. Finding such a contour is usually referred to as 'shape modeling' that is used for object segmentation and classification (Malladi et al. 1995), (Malladi et al. 1994), (Malladi and Sethian 1994). The difficulty here is that there is no order in the set of points and that it is unknown in advance which points belong to the boundary. This is defined as implicit constraints in (Cohen 1996).

Denote by $\mathcal{E}(x,y) : \mathcal{D} \rightarrow \{0,1\}$ a binary function representing the result of applying a standard edge detector on the image I, where 1 corresponds to a detected edge point. One possible way of defining a potential $P : \mathcal{D} \rightarrow I\!R^+$ is as a function of the distance map (Cohen and Cohen 1993), where each point p is assigned with a value representing the shortest Euclidean distance to an edge point:

$$d_{\mathcal{E}}(p) = \inf_{\mathcal{E}(q)=1} \{dist(p,q)\}, \qquad (28)$$
$$\text{and} \quad P(p) = f(d_{\mathcal{E}}(p))$$

where $dist(p,q)$ is the Euclidean distance between the two points p and q and f is an increasing function. An example of distance map is shown in Figure 5. Consistent numerical approximations of (28) for the computation of $d_{\mathcal{E}}$ on a sequential computer may again be implemented by using the fast marching method (Sethian 1996). Quick sequential algorithms (Borgefors 1984), (Danielsson 1980) were used for defining the attraction potential in (Cohen and Cohen 1993). Sub-pixel estimation of the distance using a parallel algorithm was presented in (Kimmel et al. 1996). It gives a high *sub-pixel* precision of the distance. This is one possible application of shortest path estimation (Kimmel et al. 1995), (Sussman et al. 1994) presented briefly in Section 3.2. Note also that the distance potential selection P may be also considered as the normalized force introduced in (Cohen 1991) for stabilizing the results (*i.e.* for $P = d_{\mathcal{E}}$ we have $\|\nabla P\| = \frac{\nabla P}{\|\nabla P\|}$) since $\|\nabla d_{\mathcal{E}}\| = 1$ almost everywhere. The motivation for choosing such a potential is that the penalty grows as a function of the distance from the edge points.

This last equality is useful in the context of the previous section to obtain an estimation of the curvature's bound when $\tilde{P} = w + d_{\mathcal{E}}$. From Equation (26), we have:

$$|\kappa| \leq \frac{1}{w}, \qquad (29)$$

i.e. w is the minimum curvature radius along the final contour. In the case $\tilde{P} = w + f(d_{\mathcal{E}})$, the upper bound becomes

$$|\kappa| \leq \sup_d \frac{f'(d)}{w+d}. \qquad (30)$$

where d ranges from 0 to the maximal distance in the image. The bound in (30) can be easily found for the functions $f(d) = \alpha d^2$ or $f(d) = 1 - e^{-\alpha d^2}$ which corresponds to robust statistics (see (Cohen 1996)).

A synthetic example is presented in Figure 5 where the potential used is obtained from a distance map to the edge points. Observe the way the level curves propagate faster along the line.

6. Closed Boundary Extraction from a Single Point

It is often needed to detect a closed contour. Our previous approach of finding a minimal path between two given end points, detects the two paths that complete a closed contour only if both ways correspond to a global minimum. In the general case of selecting the second point, it is clear that although both ways are local minima, only one is a global minimum. Assuming only one start point p_0 is given on the closed contour, let us compute the minimal action U from this start point. We should then find a second point p_1 that is located on the unknown contour, from which the two geodesics have the same energy. This means

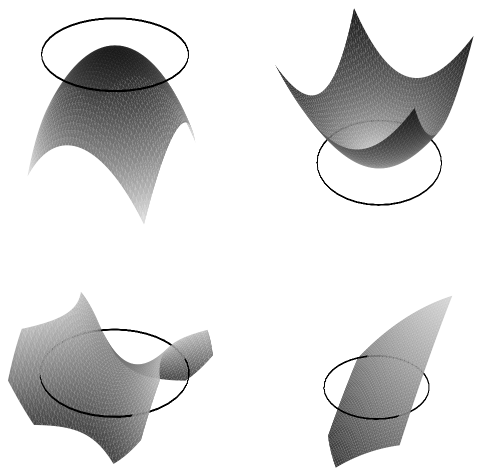

Fig. 7. Illustrating the number of level crossings. At the top, a maximum and a minimum points give 0, at the bottom left, a saddle point gives 4, and at the bottom right, other points give 2 level crossings.

we have to find a point p_1 from which there is more than one minimal curve connecting it to the source p_0. These special points are the saddles of U.

6.1. ***Justification of the search for saddle points***

A saddle point is a surface point at which there are two descents and two ascents. The descents indicate reachability by two minimal geodesics in our case. Assuming that all the points at the boundary of a closed shape, belong to one of two ("left" or "right") geodesics connecting it to the start point. There is only one point p_1 at which the "left" and "right" geodesics have the same length, *i.e.* their meeting point, which is a saddle of U.

Since U is maximal at p_1 along both ways ("left" or "right"), the derivative of U along the direction tangent to the path is zero. As mentioned before, the two minimal paths are orthogonal to the level sets of U. Thus, the derivative of U along the normal to the path is also zero, which means that $Du(p_1) = 0$. Since U is maximal at p_1 along the path, U has a negative second derivative in that direction. Since $w > 0$, at any point of the image there has to be a direction in which U increases, and for which the second directional derivative has to be positive. As a consequence, $D^2u(p_1)$ has to have opposite sign eigenvalues, that is one definition of a saddle point.

The saddle points may serve as clues in closing contours of objects that are contained within the image domain. When the user searches for a closed contour from p_0, an automatic search for saddle points on U is performed. Back propagating from a saddle point p_1 to both directions will connect the saddle to the source point p_0 by two curves. (see end of Section 4.5). Alternatively, computing the minimal action surface from the saddle point and searching for the minimal set of the sum of both action surfaces, yields the desired result as a set of points (to be refined). Thereby, a closed contour is formed representing the complete boundary of an object.

6.2. ***Saddle points characterization***

As mentioned, to detect such a saddle point, we can compute the gradient $|\nabla U|$ and the Gaussian curvature $(\kappa_1\kappa_2)$, and check for $|\nabla U| < \epsilon$ and $\kappa_1\kappa_2 < 0$.

Another possibility to isolate the saddle points on U is to use a simple test to determine the number of *level crossings*. Consider a small radius circle centered at a candidate point q and embedded in the horizontal plane $(x, y, U(q))$. Denote the number of level crossings to be the number of points this circle intersects with the surface $(x, y, U(x, y))$. It is shown in Figure 7 that this number at a saddle point is equal to four, while for most surface points it is two, and at maximum and minimum points there are no level crossings. In our implementation of the number of level crossings, for each point (i, j) in the pixels grid, we simply count the number of sign changes in $U(k, l) - U(i, j)$ while traveling around the 8 neighbors (k, l) of the point.

6.3. *Saddle points filtering*

Although there are only few saddle points in U (see Figure 14 for example), finding the level crossing for every point q in the domain is not enough. It is necessary to filter out the insignificant saddles that have a relatively large value of P or U.

This usually reduces the number of candidates to a relatively small number (only two remain after simple filtering of the saddles in Figure 14). In a favorable case where there are not many gaps in the boundary contour, another criteria that will do the work is to consider only those saddle points that are close to edge points, since it is obvious that the contour should pass close to an edge point. Selecting the right regularization constant w will obviously filter out most of the saddles that are formed due to noise, yet will obviously introduce further constraints on w. According to our experience, selecting the right w for a smoothing effect reduces the number of saddles to the only interesting ones. Since we are dealing with a user interactive procedure, it is possible to paint the candidate saddle points on the image and let the user pick the right saddle among the filtered saddle points. Selecting the right saddle point will close the contour and segment the object.

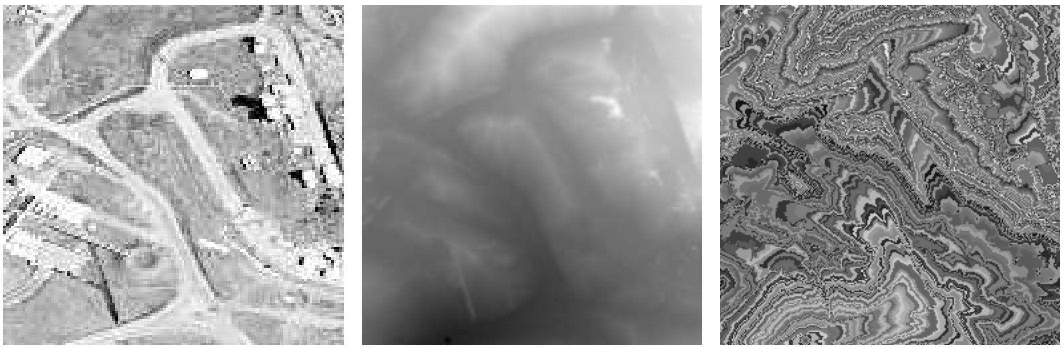

Fig. 6. Road Image. Original on the left. Minimal action U from bottom left start point: in the middle, black corresponds to lower values of U, on the right a random look up table is used to render the level curves of U.

7. Examples and Results

We demonstrate the performance of the proposed algorithm (using the minimal action algorithm described in section 4.4) by applying it to several real images. The images were scaled to 128×128 pixels, and the gray levels for P were normalized between 0 and 1. Parameter w is usually of the order of 0.1.

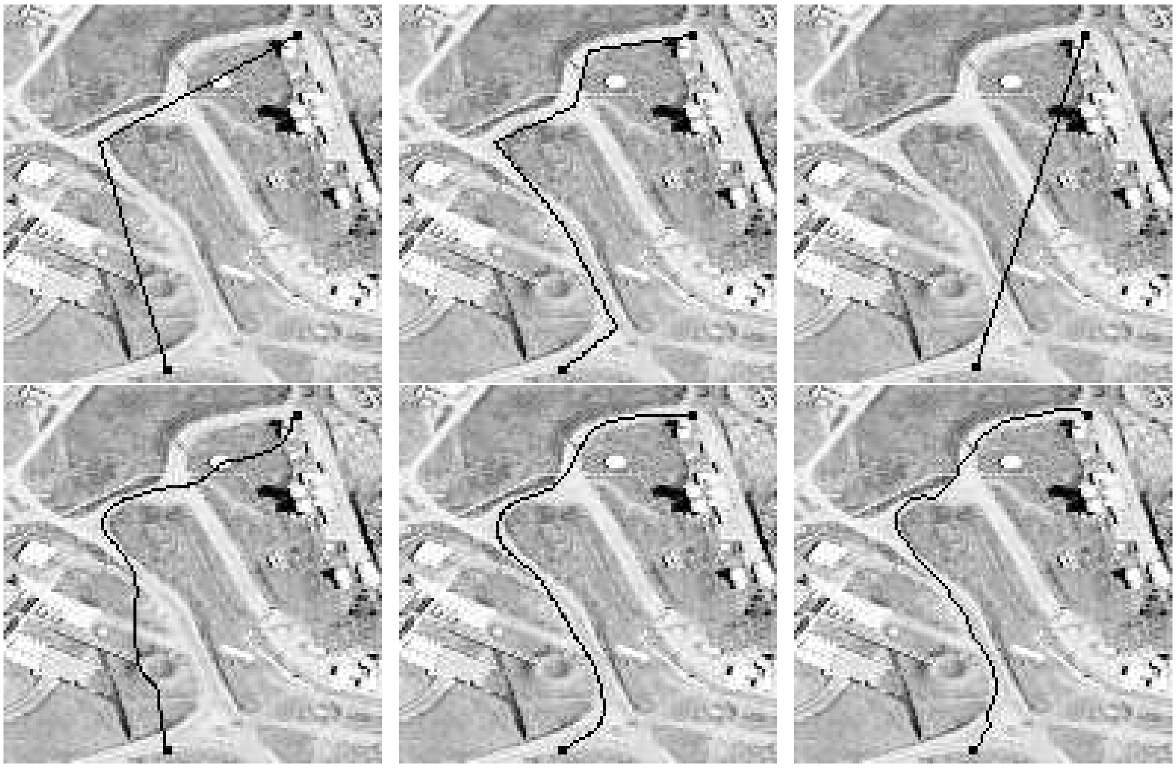

Fig. 8. Local and global Minimum. The initial data is shown at the top and the result at the bottom. The left and middle columns show the results of two different initializations of the classical snakes. The right example shows our path of minimal action connecting the two black points as start and end points.

7.1. ***Open contour: Road and Medical Image***

In the first example, we are interested in a road detection between two points in the image of Figure 6. Road areas are brighter and correspond to higher gray levels. The potential function P was thus selected to be the opposite of the gray level image itself: $P = 1 - I$. Minimizing this potential along a curve yields a path that follows the middle of the road. This example illustrates the efficiency of our approach compared to classical snakes. We do not claim that this is a road detection algorithm as one can find for example in (Fischler et al. 1981), (Geman and Jedynak 1996). For such an application, if the two edges of the road are needed rather than the middle way, our result could be refined using either ribbon snakes (Neuenschwander et al. 1994) or a thick contour potential like in (Davatzikos and Prince 1993).

Given a start point p_0 on the bottom left, the image of minimal action $U(x, y)$ from this point is shown in Figure 6. Observe the way the level curves propagate faster along the road. At the left of Figure 8, we show how a bad initialization for classical snakes leads to a wrong local minimum and it requires a very accurate initial guess, as in the middle example, to guarantee convergence to the desired solution. It is shown that given two end points, the proposed procedure detects the path of minimal action along the right road.

Note, that using a completely different approach based on classical snakes, the authors of (Neuenschwander et al. 1994) have also found a way to solve efficiently the snake problem with fixed end points. Although their method behaves better than classical snakes, it does not ensure to converge to the global minimum and may be trapped in a bad local minimum solution as we illustrate in the following example. Using the same road image, Figure 10 presents two examples for which their method leads to a local minimum. At the top, taking the same end points as in Figure 8, the part of curve close to the upper right end point is trapped by the white building below it, like in the left example of Figure 8. At the bottom, if the end point is slightly shifted, the curve follows the road correctly from both ends but at some point it prefers a short-cut. Note, that in both examples we do not present the final curve position but its position at some intermediate time from which it is not possible to return back to the correct road. The interactive tool for outlining roads in aerial or medical images presented in (Neuenschwander et al. 1994) could also make use of our method between fewer constraint points or key-points to solve some cases in which there are many erroneous local minima.

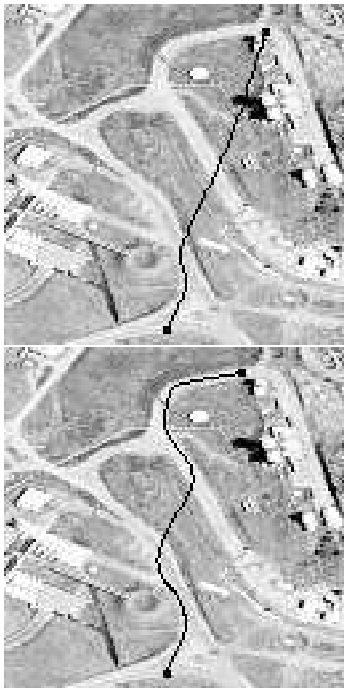

Fig. 10. Two examples of applying the approach of (Neuenschwander et al. 1994) with two slightly different initializations. In both cases the curve is trapped by a local minimum (see text).

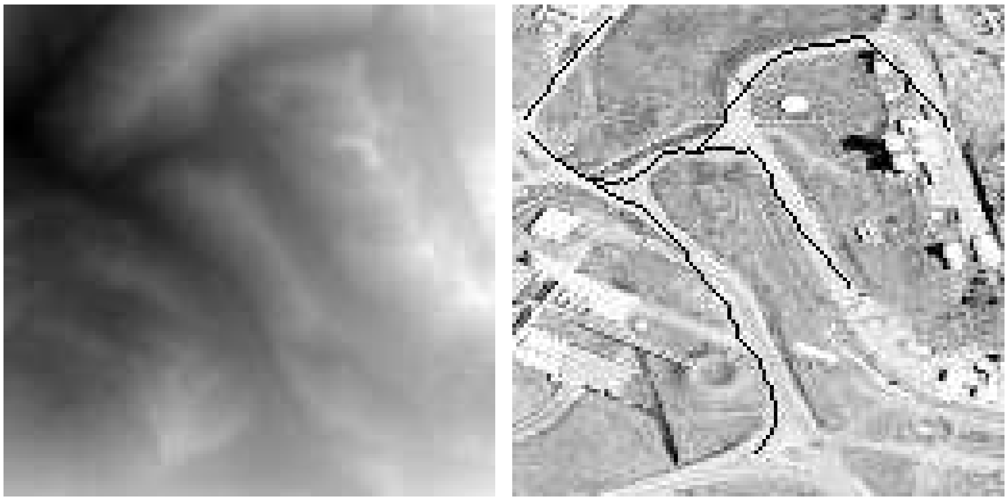

Fig. 9. Many paths are obtained simultaneously connecting the start point on the upper left to four other points. The minimal action is shown on the left.

Our approach can be used for the minimization of many paths emerging from the same point in one single calculation of the minimal action. Figure 9 shows an application of this operation for the road image. Given a start point in the upper left area, the path achieving the global minimum of the energy is found between this point and four other given points to determine the roads graph in our previous image.

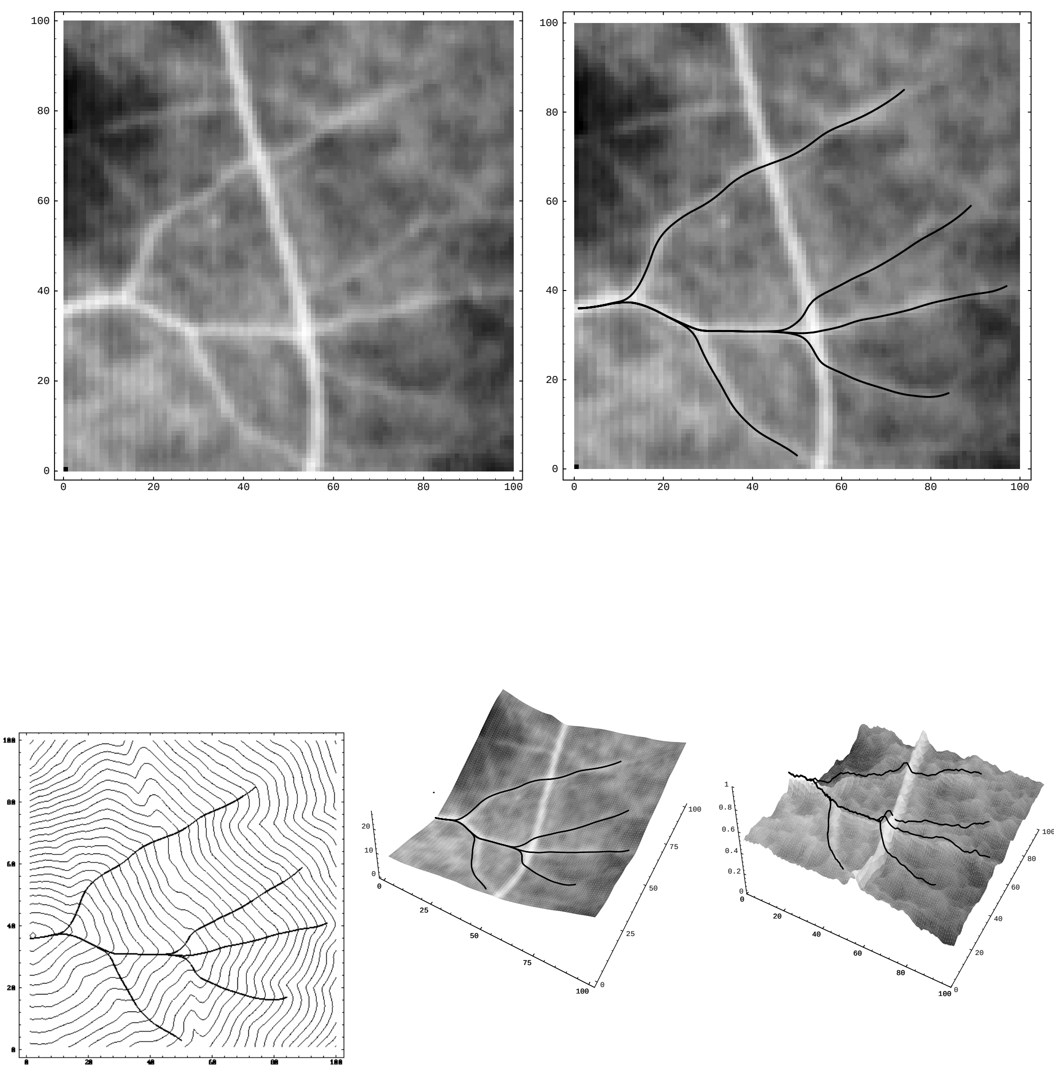

Fig. 11. Finding vessels in a medical angiographic image of the eye fundus; At the top, the original image and multiple path detection with a start point on the left of the image. At the bottom, the level sets of U are shown on the left and the paths are superimposed on the surfaces obtained with elevation U on the middle and I on the right.

In Figure 11, we show an application to the detection of blood vessels in a medical angiographic image of the eye fundus. Here also, the potential is obtained from the image itself to detect higher gray levels. These results make use of high order ODE integrators for the back propagation as described in (Kimmel and Sethian 1996).

Note that our method is very efficient in finding boundaries in a static image. In a sequence of images, it may be faster to use it once for the first image. For the subsequent images, the boundary found in the previous image is usually a good initialization for classical or geodesic active contours (Kass et al. 1988), (Caselles et al. 1995).

7.2. Closed contour: Medical Image

In this third example, we want to extract the left ventricle in an MR image of the heart. The potential is a function of the distance to the closest edge in a Canny (Canny 1986) edge detection image (see Figure 12). Since it is a closed contour we use the saddle points classification in closing the boundaries of a single object in the heart image (see Figures 13 and 14). Given a single point, saddle point classification is used to find the second end point. The closed contour is formed of the two minimal paths joining the start and end points.

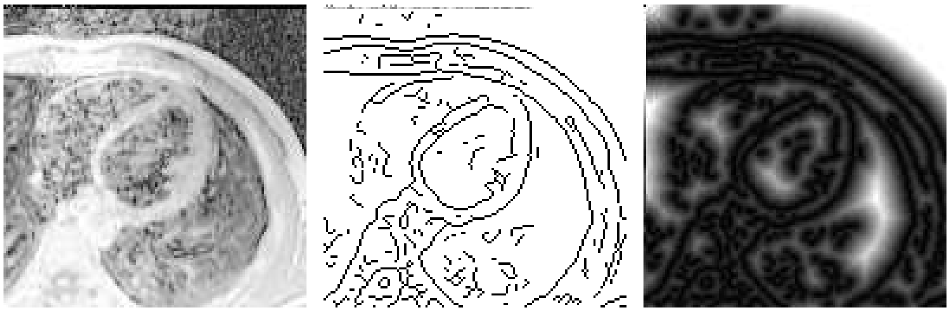

Fig. 12. MRI heart image: Original image on the left, edge image in the middle, distance map on the right.

8. Concluding Remarks

In this paper we presented a method for integrating objects boundaries by searching for the path of minimal action connecting two points. The search for the global minimum makes sense only after the two end points are determined, and the 'action' or 'potential' is generated from the image data. The proposed approach makes snake initialization an easier task that requires only one or two end points and overcomes one of the fundamental problems of the active contour model, that is being trapped by an insignificant local minimum. Applying the proposed procedure to real images gave very promising results that were compared to the results obtained by other approaches that search for local solutions.

An upper bound over the curvature magnitude of the final contour was obtained by the ratio of gradient magnitude and the value of the potential. It was shown that controlling the smoothness of the final contour is possible by adding a regularization term to the potential function, thereby decreasing this bound.

The result of the proposed procedure may be considered either as the solution or as initial condition for classical snake models, or even more naturally for geodesic active contours for further refinement. In the later case, refinement to the proper solution should be almost immediate.

Acknowledgments

We thank Prof. James Sethian, Dr. David Adalsteinsson, and Dr. Ravikanth Malladi for intriguing discussions on the fast marching method, and the authors of (Neuenschwander et al. 1994) for providing the roads image and apologize for squaring it for easier presentation. We also thank the Arc-en-Ciel Keshet program, who helped us starting working together. We would like to thank the anonymous reviewers for their detailed remarks that helped us improve the presentation of this paper.

Notes

1. Our contribution here does not deal with the way to find the ideal potential, but only to find the global minimum given this potential. We give examples with intensity, gradient or distance map potentials.
2. We shall refer to arc-length parameter as s, to differ from an arbitrary parameter v

References

Adalsteinsson, D., Kimmel, R., Malladi,R. and Sethian, J. A. 1996. Fast marching method for computing solutions to static Hamilton-Jacobi equations. Technical report, LBL UC Berkeley, CA 94720. submitted for publication.

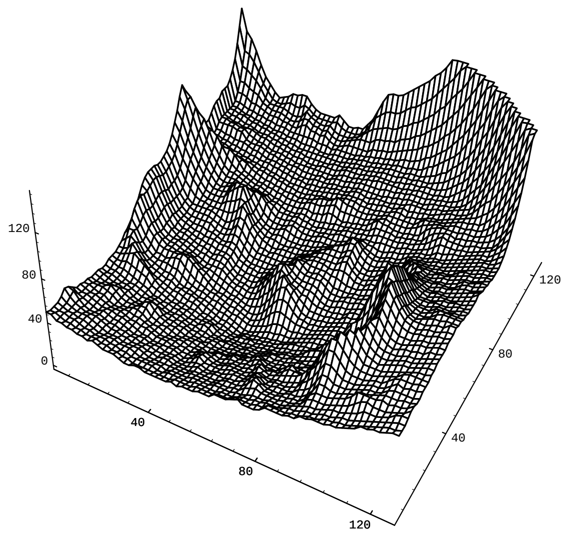

Fig. 13. MRI heart image: minimal action U represented as a graph surface. The start point is the lower point ($U = 0$) located on the bottom left of the ventricle at pixel (48,44).

Adalsteinsson, D. and Sethian, J. A. 1995. A fast level set method for propagating interfaces. *J. of Comp. Phys.*, 118:269–277.

Amini, A.A., Weymouth, T.E. and Jain, R.C. 1990. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(9):855–867.

Bellman, R. and Kalaba, R. 1965. *Dynamic Programming and modern control theory.* London mathematical society monographs, London.

Borgefors, G. 1984. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345.

Bruckstein, A. M. 1988. On shape from shading. *Computer Vision, Graphics, and Image Processing*, 44:139–154.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Caselles, V., Catté, F., Coll, T. and Dibos, F. 1993. A geometric model for active contours. *Numerische Mathematik*, 66:1–31.

Caselles, V., Kimmel, R. and Sapiro, G. 1995. Geodesic active contours. In *Proc. Fifth IEEE International Conference on Computer Vision (ICCV'95)*, pages 694–699, Cambridge, USA. Long version to appear in *International Journal of Computer Vision.*

Caselles, V., Kimmel, R., Sapiro, G. and Sbert, C. 1996. Three dimensional object modeling via minimal surfaces. In *Proc. Third European Conference on Computer Vision (ECCV'96)*, Cambridge, U. K.

Chandran, S., Meajima, T. and Miyazaki, S. 1991. Global minima via dynamic programming: Energy minimizing active contours. In *Proc. SPIE Vol. 1570 Geometric Methods in Computer Vision*, pages 391–402, San Diego.

Cohen, Laurent D. 1991. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218.

Cohen, Laurent D. 1995. *Variational Methods for Image Processing (in French).* Université Paris Dauphine. "Mémoire d'Habilitation à diriger des recherches" Presented together with 10 main publications (in English) during 1988-1995.

Cohen, Laurent D. 1996. Auxiliary variables and two-step iterative algorithms in computer vision problems. *Journal of Mathematical Imaging and Vision*, 6(1):61–86. See also ICCV'95.

Cohen, Laurent D., Bardinet, E. and Ayache, N. 1993. Surface reconstruction using active contour models. In *Proceedings SPIE 93 Conference on Geometric Methods in Computer Vision*, San Diego, CA. See also INRIA TR 1824, December 1992.

Cohen, Laurent D. and Cohen, Isaac. 1993. Finite element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-15(11).

Crandall, M. G., Ishii, H. and Lions, P. L. 1992. User's guide to viscosity solutions of second order partial linear differential equations. *Bulletin of the American Math. Society*, 27:1–67.

Danielsson, P. E. 1980. Euclidean distance mapping. *Computer Vision, Graphics, and Image Processing*, 14:227–248.

Davatzikos, C. and Prince, J. 1993. Adaptive active contour algorithms for extracting and mapping thick curves. In *Proc. 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 524–529, New York.

Dijkstra, E. W. 1959. A note on two problems in connection with graphs. *Numerische Mathematic*, 1:269–271.

Dubrovin, B. A., Fomenko, A. T. and Novikov, S. P. 1984. *Modern Geometry - Methods and Applications I.* Springer-Verlag, New York.

Dupuis, P. and Oliensis, J. . 1994. An optimal control formulation and related numerical methods for a problem in shape reconstruction. *Annals of Applied Probability*, 4(2):287–346.

Fischler, M.A., Tenenbaum, J.M. and Wolf, H.C. 1981. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–223.

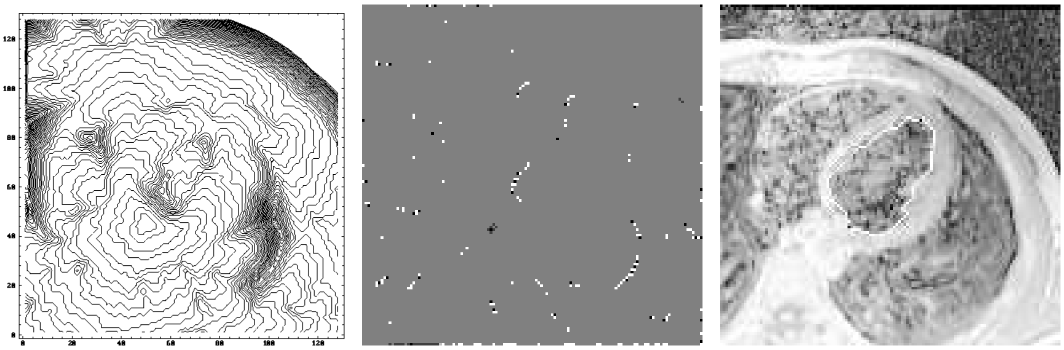

Fig. 14. MRI heart image: From left to right, level set curves of the minimal action from previous figure; Heart ventricle detection: To find the second end point, saddle point classification is used on the middle image (after filtering, only two of these points remain). The number of level crossings appears in black for 0 (maximum and minimum), gray for 2 (most of the points) and white for 4 (saddle). After filtering the white pixels, the selected saddle point is used to find the two half contours on the right. The contour is white and the two end points are the two black pixels. The start point is on the lower left and the other one is the detected saddle.

Fua, P. and Leclerc, Y. G. 1990. Model driven edge detection. *Machine Vision and Applications*, 3:45–56.

Fujimura, K., Yokoya, N. and Yamamoto, K. 1992. Motion tracking of deformable objects based on energy minimization using multiscale dynamic programming. In *Proc. IEEE International Conference on Pattern Recognition*, volume I, pages 83–86, The Hague.

Geiger, D. Gupta, A. Costa, L. and Vlontzos, J. 1995. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3).

Geman, D. and Jedynak, B. 1996. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Grzeszczuk, R. P. and Levin, D. N. 1994. Brownian strings: segmenting images with stochastically deformable contours. In *Proceedings of Visualization in Biomedical Computing (VBC'94)*, pages 72–89.

Kass, M., Witkin, A. and Terzopoulos, D. 1988. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A. and Yezzi, A. 1995. Gradient flows and geometric active contour models. In *Proc. Fifth IEEE International Conference on Computer Vision (ICCV'95)*, pages 810–815, Cambridge, USA.

Kimmel, R., Amir, A. and Bruckstein, A. 1995. Finding shortest paths on surfaces using level sets propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-17(6):635–640.

Kimmel, R. and Kiryati, N. 1994. Finding shortest paths on surfaces by fast global approximation and precise local refinement. In *Proceedings of SPIE Vision Geometry III*, volume 2356, pages 198–209, Boston, Massachusetts. Long version to appear in *International Journal of Pattern Recognition and Artificial Intelligence*.

Kimmel, R., Kiryati, N. and Bruckstein, A. M. 1996. Distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision*. Special Issue on Topology and Geometry in Computer Vision.

Kimmel, R. and Sethian, J. A. 1996. Fast marching methods for computing distance maps and shortest paths. Technical report, LBL, Univ. of California, Berkeley.

Kimmel, R. 1995. *Curve Evolution on Surfaces*. PhD thesis, Technion, Israel Institute of Technology, Haifa, Israel.

Kiryati, N. and Székely, G. 1993. Estimating shortest paths and minimal distances on digitized three dimensional surfaces. *Pattern Recognition*, 26(11):1623–1637.

Leitner, F. and Cinquin, P. 1991. Dynamic segmentation: Detecting complex topology 3D-object. In *Proceedings of International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 295–296, Orlando, Florida.

Leymarie, F. and Levine, M. D. 1993. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634.

Malladi, R. and Sethian, J. A. 1994. A unified framework for shape segmentation representation, and recognition. Technical Report LBL-36069 UC-405, Dept. of Mathematics, Univ. of California, Berkeley.

Malladi, R., Sethian, J. A. and Vemuri, B. C. 1994. Evolutionary fronts for topology-independent shape modeling and recovery. In *Proc. Third European Conference on Computer Vision*, pages 3–13, Stockholm, Sweden.

Malladi, R. Sethian, J. A. and Vemuri, B. C. 1995. Shape modeling with front propagation: A level set approach. *IEEE Trans. on PAMI*, February.

McInerney, T. and Terzopoulos, D. 1995. Medical image segmentation using topologically adaptable snakes. In Springer, editor, *Proceedings of the First International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine CVRMed'95*, Nice, France. See also ICCV'95.

Merlet, N. and Zerubia, J. 1993. A curvature dependent energy function for detecting lines in satellite images. In *Proc. of 8th SCIA*, Tromso.

Merlet, N. and Zerubia, J. 1994. New prospects in line detection for remote sensing images. In *Proc. of IEEE ICASSP*, Adelaide.

Mitchell, J. S. B., Payton, D. and Keirsey, D. 1987. Planning and reasoning for autonomous vehicle control. *International Journal of Intelligent Systems*, 2:129–198.

Montanari, U. 1971. On the optimal detection of curves in noisy pictures. *Communications of the ACM*, 14(5):335–345.

Mortensen, E. and Barrett, W. 1995. Intelligent scissors for image composition. In *Computer Graphics (Proceedings of SIGGRAPH'95)*, pages 191–198, Los Angeles.

Neuenschwander, W. Fua, P. Szekely, G. and Kubler, O. 1994. Making snakes converge from minimal initialization. In *Proc. 12th IEEE International Conference on Pattern Recognition*, pages A–613–615, Jerusalem, Israel.

Osher, S. J. and Sethian, J. A. 1988. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49.

Rouy, E. and Tourin, A. 1992. A viscosity solutions approach to shape-from-shading. *SIAM. J. Numer. Analy.*, 29:867–884.

Rutovitz, D. 1968. Data structures for operations on digital images. In G.C. Cheng, R.S. Ledley, D.K. Pollock, and A. Rosenfeld eds, editors, *Pictorial pattern recognition*, pages 105–133. Thompson Book, Washington.

Sapiro, G. 1996. Vector-valued active contours. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, USA.

Sedgewick, R. 1988. *Algorithms*. Addison-Wesley.

Sethian, J. A. 1989. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent flows. *J. Differential Geometry*, 31:131–161.

Sethian, J. A. 1995. A review of the theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*.

Sethian, J. A. 1996. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.*, 93(4).

Shaashua, A. and Ullman, S. 1988. Structural saliency: The detection of globally salient structures using a locally connected network. In *Proc. Second IEEE International*

Conference on Computer Vision (ICCV'88), pages 321–327.

Shah, J. 1996. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, USA.

Sussman, M. Smereka ,P. and Osher, S. 1994. A level set approach for computing solutions to incompressible two-phase flow. *J. of Computational Physics*, 114:146–159.

Szeliski, R. and Tonnesen, D. 1992. Surface modeling with oriented particle systems. In *SIGGRAPH'92 Conference Proceedings Computer Graphics*, pages 185–194, Chicago, IL.

Szeliski, R. Tonnessen, D. and Terzopoulos, D. 1993. Curvature and continuity control in particle-based surface models. In *Proceedings SPIE 93 Conference on Geometric Methods in Computer Vision*, San Diego, CA.

Tek, H. and Kimia, B. 1995. Image segmentation by reaction-diffusion bubbles. In *Proc. Fifth IEEE International Conference on Computer Vision (ICCV'95)*, pages 156–162, Cambridge, USA.

Terzopoulos, D. 1987. On matching deformable models to images: Direct and iterative solutions. In *Topical meeting on machine vision, Technical Digest Series*, volume 12, pages 160–167. Optical Society of America.

Thiel, E. and Montanvert, A. 1992. Chamfer masks: discrete distance functions, geometrical properties and optimization. In *IEEE ICPR*, volume C, pages 244–247, The Hague.

Verbeek, P. W. and Verwer, B. J. H. 1990. Shading from shape, the eikonal equation solved by gray-weighted distance transform. *Pattern Recognition Letters*, 11:681–690.

Whitaker, R. 1995. Algorithms for implicit deformable models. In *Proc. Fifth IEEE International Conference on Computer Vision (ICCV'95)*, pages 822–827, Cambridge, USA.