Chapter 9

Skeletonization via Distance Maps and Level Sets

The medial axis transform (MAT) of a shape, better known as its skeleton, is frequently used in shape analysis and related areas. In this chapter a new approach for determining the skeleton of an object, is presented [80]. The boundary is segmented at points of maximal positive curvature and a distance map from each of the segments is calculated. The skeleton is then located by applying simple rules to the zero sets of distance maps differences.

A framework is proposed for numerical approximation of distance maps that is consistent with the continuous case, hence does not suffer from digitization bias due to metrication errors of the implementation on the grid. Subpixel accuracy in distance map calculation is obtained by using gray level information along the boundary of the shape in the numerical scheme. The accuracy of the resulting efficient skeletonization algorithm is demonstrated by several examples.

9.1 Chapter Introduction

In this chapter we present a new approach to shape skeletonization. The importance of the skeleton, or medial axis, was discussed by Blum in [7, 8] with motivation from visual perception analysis. Mathematical properties of the MAT were investigated over the years in many papers starting with [100, 101], and are still of great interest to the computer vision community [104, 142, 11]. Skeletons are frequently used as shape descriptors, *e.g.* in character recognition.

In [100] Montanari gives four (equivalent) definitions for the skeleton of shapes in the real plane. The first is the prairie fire model, the skeleton points being the locations where the propagating wavefront initiated on the shape boundary "intersects itself." This model of wavefront propagation uses the Hüygens principle and shock capturing as in optics and fluid dynamics. The boundary propagates with a constant velocity along its normal direction and the skeleton points are the shocks that appear during the propagation. Shocks are the collisions of "ignition lines" [126], as seen in Figure 9.1a. The propagating wavefront carries information about the boundary. Thus, using the prairie fire point of view, the skeleton points may be regarded as points in which shape boundary information vanishes. The second approach defines the skeleton as the set of "ridges" on the distance map surface measured from the boundary, see Figure 9.1b. The third model considers the skeleton as the geometric location of centers of maximal disks. A *maximal disk* is a disk contained in the shape, for which there exists no other disk in the shape that contains it, see Figure 9.1c. The last model defines a skeleton as the set of points that do not belong to any straight line segment connecting other points to their respective closest boundary points, see Figure 9.1d.

The four definitions mentioned above motivated a wealth of discrete implementations on binary images over the last two decades [3, 4, 37, 89, 97]. A skeletonization approach that has recently gained attention is the use of the Voronoi diagrams, *e.g.* [12, 104, 136].

Roughly speaking the skeletonization algorithms may be divided into three classes. Most algorithms operate on binary pixel images [3, 4, 37, 89, 97]. Other algorithms operate on polygons that are assumed to be provided by some polygonal approximation of the boundary [9, 86, 100]. Only few algorithms try to deal with the general problem of skeletonization for smooth shapes [49, 129, 130]. The algorithm proposed in this chapter belongs to all three classes. Motivated by global properties of the boundary and skeleton curves, we present a discrete implementation that is consistent with the continuous case.

In principle, each skeleton point corresponds to at least two boundary points. Those points are the boundary points whose Hüygens wavefronts met at the skeleton point, or alternatively, the boundary points touching the maximal disk centered on the skeleton point. The skeleton point is located at the intersection of the normals to the two boundary points, and the distance between the skeleton point to both boundary points is equal. Therefore, if we segment the boundary into small segments, and find a distance map from each of those segments, the skeleton will be located at places where at least two distance maps share the same value. Those locations could be equivalently defined as zero sets of distance maps differences.

The above description is based on the implicit assumption that we know how to segment the boundary so that no pair of boundary points corresponding to the same skeleton point belong to the same segment. This would be trivially accomplished by a very fine boundary segmentation which would lead to high computational complexity. Indeed, if we segment the boundary at each boundary pixel, we arrive at Voronoi diagram based skeletonization [86, 12, 104].

In order to solve the boundary segmentation problem, we study the relation between positive curvature maxima along the boundary and the skeleton. In particular, we show that a skeleton point is never generated by single curve segment defined between successive positive curvature maxima. The symmetry–curvature duality was previously explored by Leyton in [91]. In [116] that duality was used in constructing the "curvature skeleton" of binary images.

The structure of this chapter is as follows: In Section 9.2 we present the skeletonization algorithm for the continuous (pre-image) case. A discrete approximation of the proposed

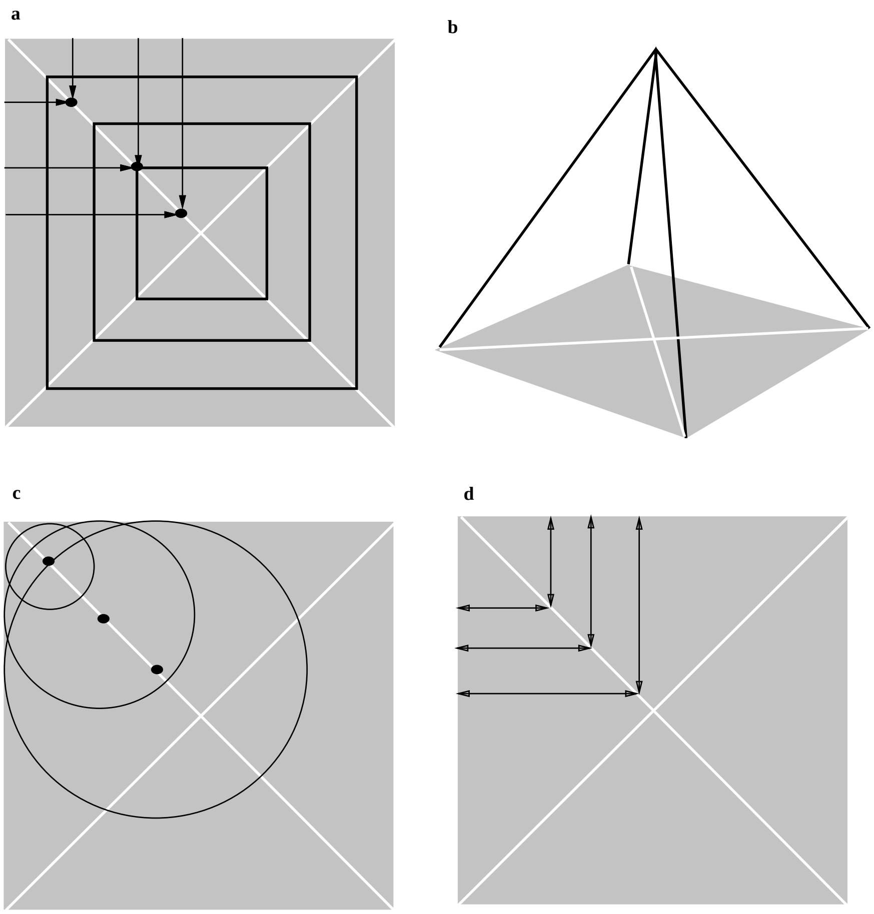

Figure 9.1: Illustration of the four points of view on the skeleton (white) of an object (gray). **a.** The black curves are offsets of the boundary obtained by constant velocity propagation. The set of shocks (self intersection points) of the propagating curves generate the skeleton. The arrows from the boundary to skeleton points represent ignition lines. **b.** The skeleton is the planar projection of the ridges of the distance map from the boundary. The ridges are shown as black curves in 3-D, where the third dimension represents distance. **c.** The skeleton is the set of centers of maximal disks. **d.** The skeleton is considered as the set of points that do not belong to any straight line connecting other interior points to their respective closest boundary points.

algorithm is introduced in Section 9.3, employing gray level information along the boundary to find the curvature along an implicit sub-pixel representation of the object boundary. Examples of applying the algorithm to image data are presented in Section 9.4. In [80] we state and prove a version of the four vertex theorem[38]. It is used in Section 9.2 to prove the correctness of the skeletonization algorithm construction.

9.2 The Skeleton and Curvature Maxima

Let the boundary of an object be given by a simple closed planar curve $\mathcal{C}(s) \in \mathbb{R}^2$, where s is the arclength parameterization along the curve. The curvature $\kappa(s)$ is defined, [38, 135], as the magnitude of the second derivative of the curve with respect to its arclength, a positive sign being assigned at convex points and a negative sign at concave points.

As explained in the introduction, a skeleton point generally corresponds to at least two boundary points that touch the maximal disk. We refer to those points as the boundary points "generating" the skeleton point. In order to justify our new skeletonization approach we prove the following property of the skeleton–boundary relation.

Theorem 9.1 *A boundary segment connecting two points that generate a skeleton point, contains at least one internal point having a positive curvature maximum.*

Proof: Given a boundary segment connecting two points that generate a skeleton point, the maximal disk corresponding to the skeleton point is tangent to the boundary at the end points of the boundary segment. Those two points segment the disk boundary into two circular segments. Consider a new shape defined by the boundary segment and the "outer" or opposite circle segment of the maximal disk. We now apply a version of the well known Four Vertex Theorem [38], which is stated and proven in [80]. According to the four vertex theorem, the boundary of the new shape has at least two distinct points at which the curvature is both positive and locally maximal. The circle segment on the boundary may account for only one such point. The other has to be found within the segment of the original boundary. ■

This theorem provides the basis for the required boundary segmentation

Corollary 9.1 *If we segment the shape boundary at points of positive maximal curvature, points generating its skeleton always belong to different boundary segments.*

An algorithm for skeletonization follows from this result. It's outline is:

1. Find the curvature along the boundary curve and split the boundary into segments at points of maximal positive curvature. The arclength parameters of the segmentation points, are referred to by s_i, where $i \in \{0, 1, ..., N-1\}$, $(s_N \equiv s_0)$. N being the number of local positive curvature maxima.

2. For each segment calculate the Euclidean distance map over the whole image domain $\Omega \subseteq \mathbb{R}^2$. For the i-th segment of the boundary curve, $\mathcal{C}(s) : s \in [s_i, s_{i+1}]$, the distance map $\mathcal{D}_i : (x, y) \in \Omega \to \mathbb{R}^+$, is defined as

$$\mathcal{D}_i(x, y) = \inf_{s \in [s_i, s_{i+1}]} \{ \ d(\mathcal{C}(s), (x, y)) \ \},$$

where $d(p, q)$ is the Euclidean distance between the two points p and q.

3. Find a preliminary skeleton as the location of the zero-level sets of all distance map differences.

$$\hat{\mathcal{S}} = \{(x, y) : \forall i, j; \ i \neq j, (\mathcal{D}_i - \mathcal{D}_j = 0) \cap (\cap_{k \neq i,j} \mathcal{D}_i < \mathcal{D}_k)\}, \tag{9.1}$$

where $i, j, k \in \{0, 1, .., N-1\}$. $\hat{\mathcal{S}}$ is clearly the Voronoi diagram of the curve segments.

4. Eliminate all exterior points (background points), and all points located on branches starting at points $\mathcal{C}(s_i)$ whose distance from $\mathcal{C}(s_i)$ is less than $1/\kappa(s_i)$. The elimination process is performed sequentially starting with the branch corresponding to the largest curvature maximum and ending with the smallest positive curvature maximum. False branches corresponding to larger curvature maxima are deleted first so that they will not interfere with the deletion process of false branches corresponding to smaller curvature maxima that are connected to them.

Note that for the segmentation of a closed curve, we need to assign at least two partition points. Assigning a partition point at points of positive maximal curvature results in a legal segmentation of the boundary, due to the four vertex theorem, which ensures the existence of at least two such points.

The skeletonization algorithm described above operates well on all shapes unless they have a specifically shaped hole in them, see Figures 9.2. The reason for this problem is that the four vertex theorem does not apply to boundaries of holes in shapes, or in other words, a shape (the shape of the hole) does not necessarily have two boundary points with *negative* curvature minima. However, if a shape has a single negative curvature minimum, the boundary necessarily contains a convex segment whose tangent's turn angle is larger than 2π, see Figure 9.2. Therefore, to make this algorithm general, we add an arbitrary segmentation point to every hole boundary whose turn angle is less then -2π.

In the next section a discrete version of the algorithm is devised. Gray level information along the object boundary is used to achieve sub-pixel accuracy in the numerical implementation.

9.3 Discrete Approximation

The numerical approximation used for implementing the proposed skeletonization algorithm considers the gray level information at the pixels along the object's boundary for interpolating

Figure 9.2: The white hole in the black object has a boundary segment along which the total curvature is less than -2π.

the boundary curve and its corresponding curvature. One possible model of the relation between the image pixels and the continuous pre-image, describes the object's boundary between the grid points by first thresholding the object from the background and then interpolating the boundary between the pixels. We shall consider the boundary to be a level set of the image gray level function, or alternatively, without loss of generality, we may consider the boundary to be the zero level set of the gray level minus a given threshold, as in [68]. We also assume a local linear behavior of the gray level function near that zero set. The planar curvature of any level set $f^{-1}(\mathbf{c})$, where $\mathbf{c}$ stands for a constant, of the function $f : \mathbb{R}^2 \to \mathbb{R}$ is given by (see Chapter 2 for proof)

$$\kappa = \frac{f_{xx}f_y^2 - 2f_xf_yf_{xy} + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{\frac{3}{2}}}. \tag{9.2}$$

First, the location of the boundary is obtained by linear interpolation (see [128] for a useful method for finding the boundary at a cell between four neighboring pixels). The curvature of the boundary at its intersections with the grid lines is then obtained by linear interpolation of the radius of curvature of level sets that pass through the two nearest grid points. Consider for example a grid line parallel to the x axis. Let the gray level at pixel i be $f_i \equiv f(i\Delta x) < 0$ and the gray level at the next pixel be $f_{i+1} > 0$. Then the zero set is interpolated to be at $\Delta x(i + |f_i|/(f_{i+1} - f_i))$, where Δx is the distance between the pixels. It is possible to calculate κ_i, the curvature of the f_i level set at pixel i using the simple central difference approximation for the partial derivatives in (9.2). The curvature at the zero crossing is interpolated as follows. For $\kappa_i\kappa_{i+1} > 0$, linear relation of the radius of curvature between successive level sets can be assumed, while if $\kappa_i\kappa_{i+1} < 0$ interpolation of the curvature itself at the zero crossing is more appropriate. Thus, κ at the zero crossing is interpolated by

$$\kappa = \begin{cases} \frac{|f_i|+|f_{i+1}|}{|f_{i+1}|\kappa_i^{-1}+|f_i|\kappa_{i+1}^{-1}} & \kappa_i\kappa_{i+1} \geq 0 \\ \\ \frac{|f_i|\kappa_i+|f_{i+1}|\kappa_{i+1}}{|f_{i+1}|+|f_i|} & \kappa_i\kappa_{i+1} < 0 \end{cases} \tag{9.3}$$

The above scheme provides the curvature values at the intersection points of the boundary and the grid lines.

Now the local positive maxima of the curvature can be detected and used for boundary segmentation. For each boundary segment the distance map is calculated. Two alternative techniques for computing sub-pixel distance maps in a way that is consistent with the continuous case are presented in [68] and in [76, 137], see Chapter 3. Both methods use the gray level information near the boundary as initial conditions. Since the image is not binarized, the subpixel boundary information that is implicitly contained in the gray levels is preserved. Having computed the N distance maps, the zero level sets of the distance map differences yield the preliminary skeleton points (Equation (9.1)).

The examples in the next section show synthetic objects created by first generating a binary image and then performing geometric smoothing to simulate the integration effect of a CCD. Some smoothing is usually necessary even with real gray level images in order to reduce the effects of noise in the boundary on the skeleton. The geometric smoothing we used considers each gray level set as a curve, and propagates all the level sets simultaneously using a geometric heat flow [120]. The Osher-Sethian algorithm [111] for curvature based flows is the basis for the numerical implementation of:

$$\frac{\partial f}{\partial t} = |\nabla f|\kappa, \tag{9.4}$$

where κ is the curvature defined at each level set of the gray level image f as in Equation (9.2). The above differential equation was numerically implemented using central difference approximation in space and forward difference approximation in time.

Consider a certain level set to be given as a parameterized curve $\mathcal{C}(s) : [0, L] \to \mathbb{R}^2$, where s is the arclength parameterization of $\mathcal{C}$. If f evolves under (9.4), then $\mathcal{C}$, embedded as a level set in f, propagates in time according to

$$\mathcal{C}_t = \mathcal{C}_{ss},$$

or equivalently

$$\mathcal{C}_t = \kappa\mathcal{N},$$

where $\mathcal{N}$ is the unit normal to the curve $\mathcal{C}$.

We smooth the boundary by propagating the image f according to Equation (9.4) with $\Delta t \simeq \Delta x$. Next, the curvature is interpolated between the grid points along the linearly interpolated boundary, via Equation (3). If there is more than one connected component like in case of holes in the object or several objects, several curvature lists are generated. For each of the curvature lists, the curvature function is first smoothed using a moving average rectangular window. Then, local positive maxima are detected using a finite window on the curvature function. Both the moving average window and the local-maxima-detection window are used to eliminate noise effects.

In step 2 of the algorithm the boundary was segmented at points of local positive curvature maxima. Gray level values in a 4 pixel width envelope around each segment are then used as initial conditions for the distance map calculation procedure. For computing the distance from each segment, we used the iterative procedure of [69, 76, 137]. That procedure is a special case of the shape from shading algorithm of Rouy and Tourin [117]. In the continuous domain, let $\mathcal{D}_0 : \Omega(\subset \mathbb{R}^2) \rightarrow \mathbb{R}^+$ be equal to the gray level values in an envelope $\mathcal{X} \subset \Omega$ along the given segment and $+\infty$ elsewhere: $\Omega \setminus \mathcal{X}$. The distance map of the segment is calculated by:

$$\mathcal{D}_t = 1 - |\nabla \mathcal{D}|,$$

for $\mathcal{D} \in \Omega \setminus \mathcal{X}$. The following differential approximation of the above equation was proven in [117] to converge to the proper solution (in our case the distance map). Using the discrete notation $\mathcal{D}^n_{i,j} = \mathcal{D}(n\Delta t, i\Delta x, j\Delta y)$, first define

$$\begin{aligned} a &\equiv D^x_- \mathcal{D}_{i,j} = (\mathcal{D}_{i,j} - \mathcal{D}_{i-1,j})/\Delta x \\ b &\equiv D^x_+ \mathcal{D}_{i,j} = (\mathcal{D}_{i+1,j} - \mathcal{D}_{i,j})/\Delta x \\ c &\equiv D^y_- \mathcal{D}_{i,j} = (\mathcal{D}_{i,j} - \mathcal{D}_{i,j-1})/\Delta y \\ d &\equiv D^y_+ \mathcal{D}_{i,j} = (\mathcal{D}_{i,j+1} - \mathcal{D}_{i,j})/\Delta y \end{aligned}$$

Then, the following numerical scheme is used to update $\mathcal{D}$.

$$\mathcal{D}^{n+1}_{i,j} = \mathcal{D}^n_{i,j} - \Delta t(\sqrt{\max[(a^+)^2, (b^-)^2] + \max[(c^+)^2, (d^-)^2]} - 1),$$

where, $\alpha^+ \equiv \max(\alpha, 0)$ and $\alpha^- \equiv \min(\alpha, 0)$. The L^1 norm of the change between two successive iterations is used to set the stopping condition, see [137, 117].

The distance map from each segment and the values of the local positive curvature at the segments edges are used to produce the final skeleton. Subpixel accuracy is achieved by interpolating the skeleton between the grid points, making use of the values of the distance maps at each grid point. It is possible to handle junctions separately and to correct the "cutting" effect.

9.4 Examples

We have presented an algorithm for locating the skeleton of a given object in a picture with subpixel accuracy. A geometric property relating the local positive curvature maxima along the boundary and the skeleton points was presented. It is the basis for boundary segmentation that ensures that a skeleton point may not be generated by two points on the same boundary segment. The boundary segments were used to construct the skeleton via zero sets of distance map differences. The suggested algorithm readily lends itself to parallelization. The following examples demonstrate the operation of the algorithm.

Consider the object shown (after geometric smoothing) at the upper left frame in Figure 9.3. The interpolated smoothed curvature along the outer and inner boundaries of the object are shown in the upper right frame. The lower left frame shows the pixels that correspond to the Voronoi diagram (before pruning). The branches to be pruned are the light gray pixel chains from the positive local curvature maxima along the boundary. The skeleton, interpolated to subpixel accuracy, is shown in the lower right frame. Additional objects and their skeletons are presented in Figures 9.4 and 9.5.

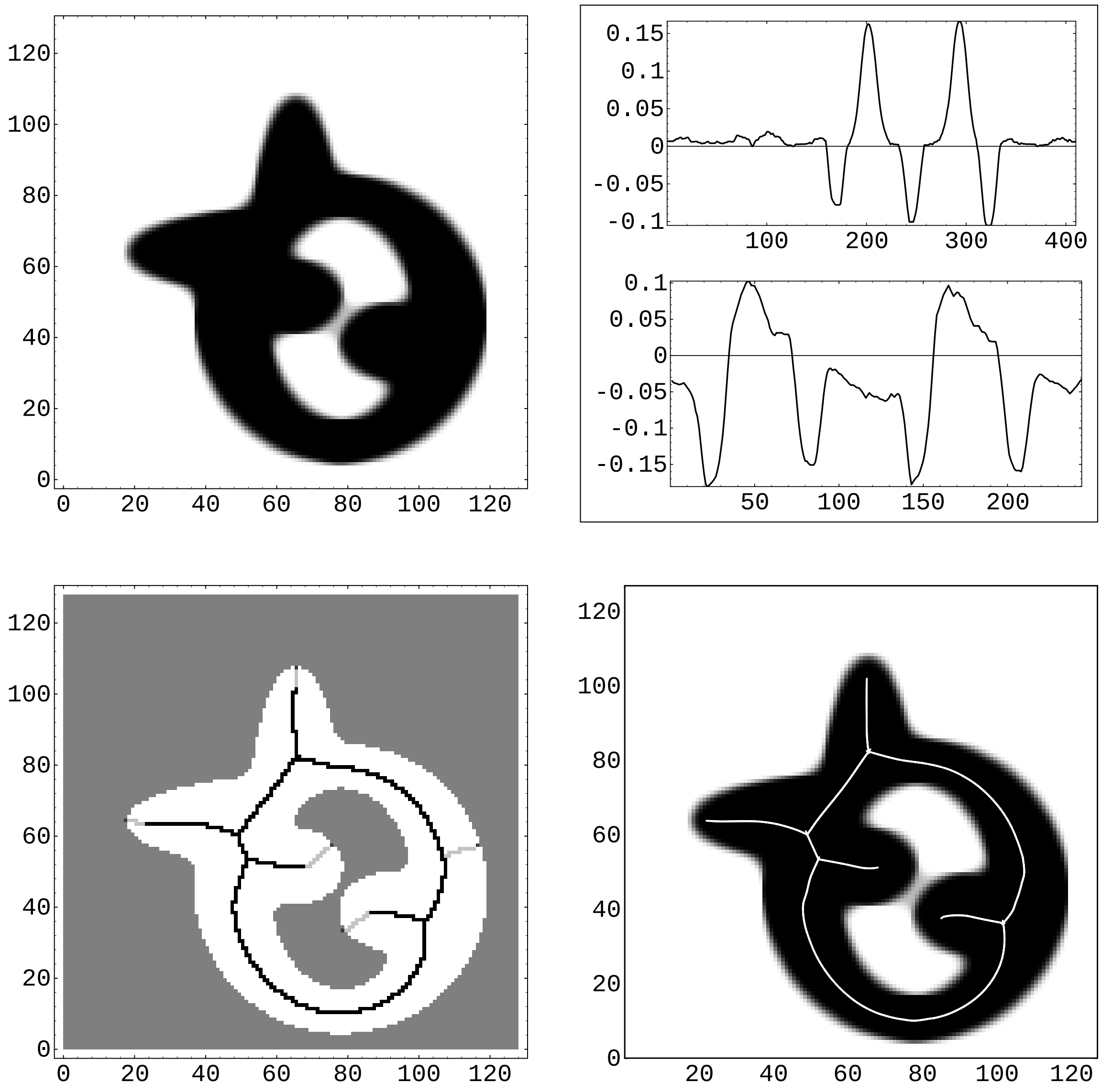

Figure 9.3: **Upper left:** An object after geometric smoothing. **Upper right:** Interpolated curvature function along the outer boundary (upper graph) and inner boundary (lower graph). **Lower left:** Voronoi diagram of the segments (preliminary skeleton) in which the branches to be pruned are in light gray. **Lower right:** The skeleton is the white curve, interpolated to subpixel accuracy, shown on the object.

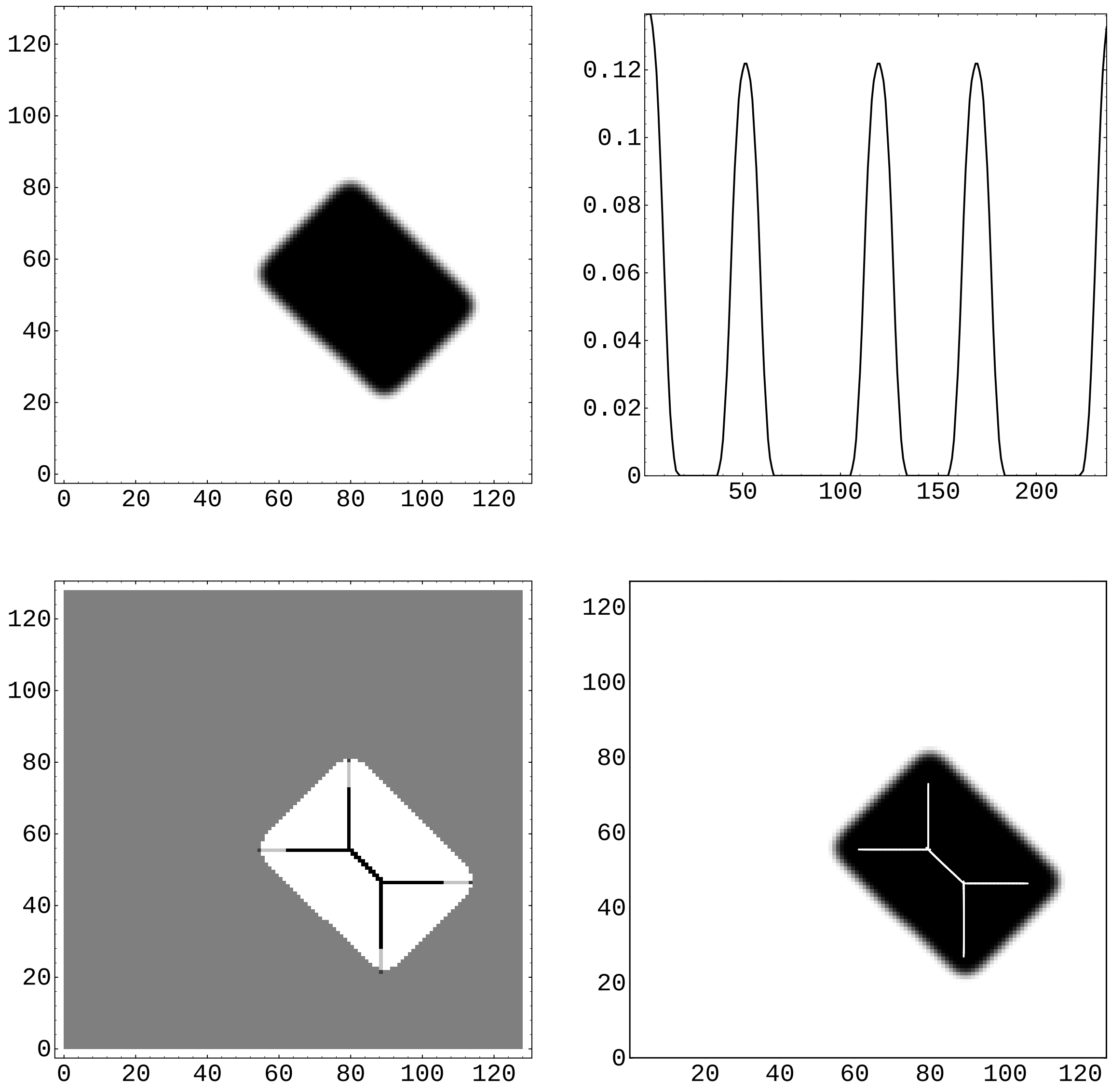

Figure 9.4: **Upper left:** A tilted 'box'–object after geometric smoothing. **Upper right:** Interpolated curvature function along the boundary. **Lower left:** Voronoi diagram of the segments (preliminary skeleton) in which the branches to be pruned are in light gray. **Lower right:** The skeleton is the white curve, interpolated to subpixel accuracy, shown on the object.

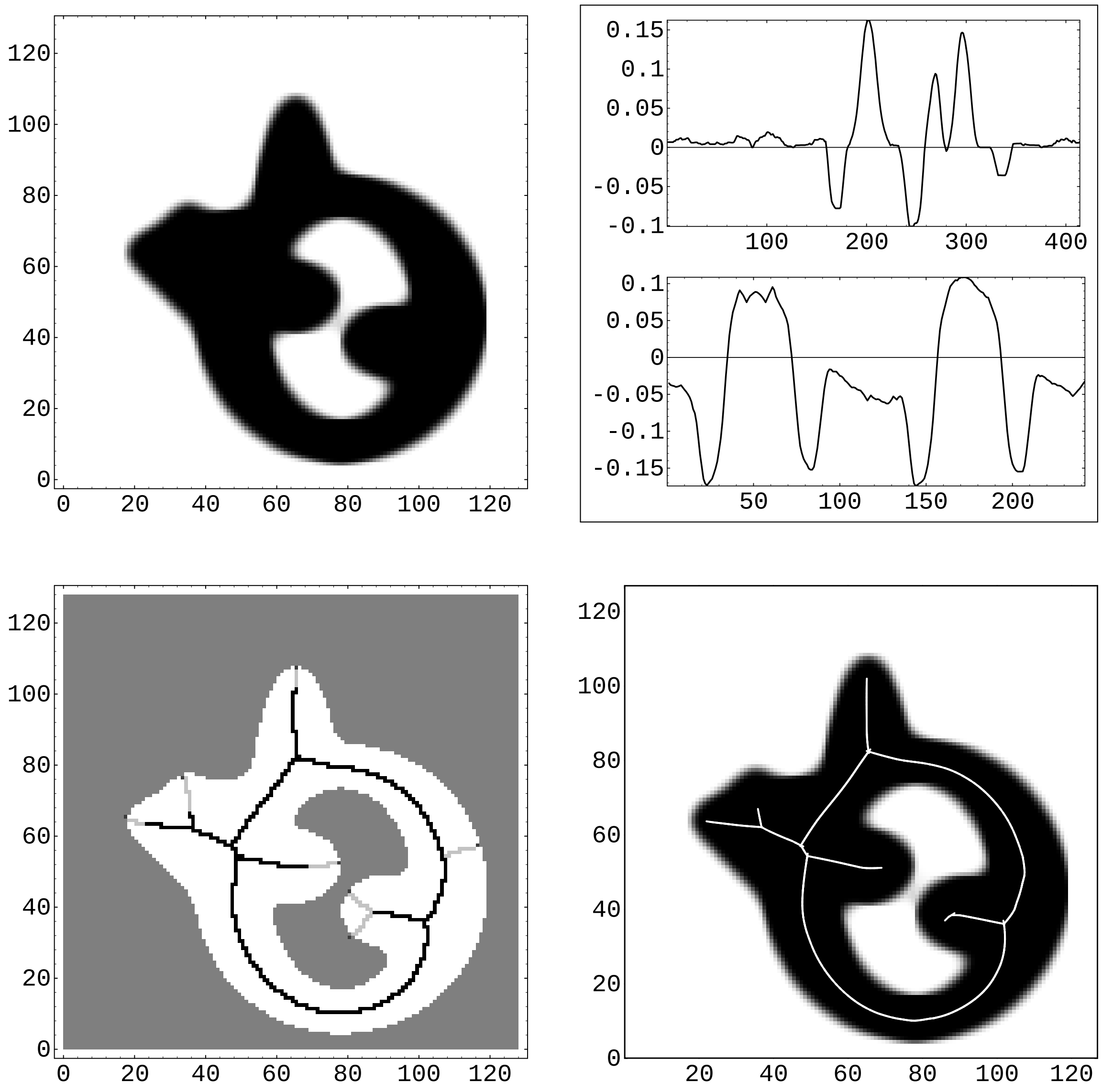

Figure 9.5: **Upper left:** An object after geometric smoothing. **Upper right:** Interpolated curvature function along the outer boundary (upper graph) and inner boundary (lower graph). **Lower left:** Voronoi diagram of the segments (preliminary skeleton) in which the branches to be pruned are in light gray. **Lower right:** The skeleton is the white curve, interpolated to subpixel accuracy, shown on the object.

Chapter 10

Geodesic Active Contours

In this chapter a novel scheme for the detection of object boundaries is presented [23, 22]. The technique is based on active contours deforming according to intrinsic geometric measures of the image. The evolving contours naturally split and merge, allowing the simultaneous detection of several objects and both interior and exterior boundaries. The proposed approach is based on the relation between active contours and the computation of geodesics or minimal distance curves. The minimal distance curve lays in a Riemannian space whose metric is defined by the image content. This geodesic approach for object segmentation allows to connect classical "snakes" based on energy minimization and geometric active contours based on the theory of curve evolution. Previous models of geometric active contours are improved as showed by a number of examples. Formal results concerning existence, uniqueness, stability, and correctness of the evolution are presented as well, see [22] for more details.

10.1 Chapter Introduction

Since original work by Kass *et al.* [61, 140], extensive research was done on "snakes" or active contour models for boundary detection. The classical approach is based on deforming an initial contour $\mathcal{C}_0$ towards the boundary of the object to be detected. The deformation is obtained by trying to minimize a functional. This energy model is not capable of handling changes in the topology of the evolving contour and in many cases heuristic topology handling procedures must be used [138]. The approach is also non-intrinsic.

Recently, novel geometric models of active contours were simultaneously proposed by Caselles *et al.* [21] and by Malladi *et al.* [95]. These models are based on the theory of curve evolution and geometric flows, which has received a large amount of attention from the image analysis community in recent years [45, 72, 119, 120, 121, 122]. The approach allows automatic changes in the topology when implemented using the level-sets based numerical algorithm [111], as presented in Chapter 2.

In this chapter, a case of the classical energy snakes model is proved to be equivalent to finding a geodesic curve in a Riemannian space with a metric derived from the image content. This means that in a certain framework, boundary detection can be considered equivalent to finding a curve of minimal weighted length. This gives a new approach for boundary detection via geodesic or local minimal distance computations. Then, assuming that this *geodesic active contour* is represented as the zero level-set of a $3D$ function, the geodesic computation is reduced to a geometric flow similar to the one obtained in the curve evolution approaches mentioned above. However, this geodesic flow includes a new velocity component based on image information, that improves those models. In [22] we also show that the solution to the geodesic flow exists in the viscosity framework, and is unique, stable and consistent.

The approach here presented has therefore the following main properties: 1- Describes the connection between energy and curve evolution approaches of active contours. 2- Presents active contours for object detection as a geodesic computation approach. 3- Improves existing curve evolution models. 4- Allows simultaneous detection of interior and exterior boundaries in several objects without special contour tracking procedures. 5- Holds formal existence, uniqueness, stability, and consistency results. 6- Does not require special stopping conditions.

10.2 Geodesic Active Contours

In this section we discuss the connection between energy based active contours (snakes) and the computation of geodesics or minimal distance curves in a Riemmanian space derived from the image. From this geodesic model for object detection, we derive a novel geometric partial differential equation for active contours that improves previous curve evolution models. See [22] for more details.

10.2.1 Energy based active contours

Let $\mathcal{C}(q) : [0,1] \to \mathbb{R}^2$ be a parametrized planar curve and let $I : [0,a] \times [0,b] \to \mathbb{R}^+$ be a given image in which we want to detect the objects boundaries. The classical snakes approach [61] associates the curve $\mathcal{C}$ with an energy given by $(\alpha, \beta, \lambda \in \mathbb{R}^+)$

$$E(\mathcal{C}) = \int_0^1 \alpha|\mathcal{C}'|^2 + \beta|\mathcal{C}''|^2 dq - \lambda \int_0^1 |\nabla I(\mathcal{C})| dq. \tag{10.1}$$

The first two terms control the smoothness of the contours to be detected (internal energy), while the third term is responsible for attracting the contour towards the object in the image (external energy). Solving the problem of snakes amounts to finding, for a given set of constants, the curve $\mathcal{C}$ that minimizes E. Note that the approach is not topology independent. When considering more than one object in the image special topology-handling procedures must be added.

Another possible problem of the energy based models is the need to select the parameters. Let us consider a particular class where $\beta = 0$. Two main reasons motivate this selections: First, this selection will allow us to derive the relation between this energy based active contours and geometric curve evolution ones. Second, as we will see in sections 10.2.2 and 10.2.4, the regularization effect on the geodesic active contours comes from curvature based curve flows, obtained only from the other terms in (10.1). This allows to achieve smooth curves in the proposed approach without having the high order derivatives given by $\beta \neq 0$. The use of the curvature driven curve motions for smoothing was proved to be very efficient in previous literature [2, 21, 95, 120], and is also supported by our experiments in Section 10.3. Therefore, curve smoothing will be obtained also with $\beta = 0$, having only the first regularization term. Assuming this, and replacing the edge detector $|\nabla I|$ by a general function $g(|\nabla I|)^2$ of the gradient such that $g(r) \to 0$ as $r \to \infty$, we obtain,

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'(q)|^2 dq + \lambda \int_0^1 g(|\nabla I(\mathcal{C}(q))|)^2 dq. \tag{10.2}$$

The geodesic active contours will be derived from (10.2).

The functional in (10.2) is not intrinsic since it depends on the parametrization q that until now is arbitrary. This is an undesirable property, since parametrizations are not related to the geometry of the curve (or object boundary), but only to the velocity they are traveled. A new parametrization of the curve in (10.2) can change the energy in any arbitrary form [22].

10.2.2 The geodesic curve flow

We now proceed and show that the solution of the particular energy snakes model (10.2) is given by a geodesic curve in a Riemannian space induced from the image I. Let us define $\mathcal{U}(\mathcal{C}) := -\lambda g(|\nabla I(\mathcal{C})|)^2$, and $\alpha := m/2$. Therefore,

$$E(\mathcal{C}) = \int_0^1 \mathcal{L}(\mathcal{C}(q)) dq,$$

where $\mathcal{L}$ is the Lagrangian given by $\mathcal{L}(\mathcal{C}) := \frac{m}{2}|\mathcal{C}'|^2 - \mathcal{U}(\mathcal{C})$. The Hamiltonian [40] is then given by $H = \frac{p^2}{2m} + \mathcal{U}(\mathcal{C})$, where $p := m\mathcal{C}'$. We now have the following:

Theorem 10.1 (Maupertuis' Principle [40]) *Curves $\mathcal{C}(q)$ in Euclidean space which are extremal corresponding to the Hamiltonian H, and have a fixed energy level E_0 (law of conservation of energy), are geodesics, with non-natural parameter, with respect to the new metric $g_{ij} = 2m(E_0 - \mathcal{U}(\mathcal{C}))\delta_{ij}$ $(i, j = 1, 2)$.*

This classical Theorem explains, among other things, when the solution to the energy problem is given by a curve of minimal "weighted distance" between given points. Minimizing $E(\mathcal{C})$ in (10.2) with $H = E_0$ (conservation of energy) is equivalent to minimizing ($\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$)

$$\int_0^1 \sqrt{g_{11}\mathcal{C}'^2_1 + 2g_{12}\mathcal{C}'_1\mathcal{C}'_2 + g_{22}\mathcal{C}'^2_2}\, dq.$$

In order to have a completely closed form for boundary detection, we have to select E_0. [1] Note that (10.2) has also a free parameter, λ/α. Selecting E_0 is equivalent to fixing this free parameter and the parametrization q. The value of E_0 is arbitrarily selected to be zero, where the corresponding formulas for $E_0 \neq 0$ are given in [22]. This selection simplifies the notation (see Appendix A in [22]), and clarifies the relation of Theorem 10.1 and energy-snakes with curve evolution active contours. Therefore, with $E_0 = 0$, and $g_{ij} = 2m\lambda g(|\nabla I(\mathcal{C})|)^2\delta_{ij}$, we obtain

$$\text{Min} \int_0^1 \sqrt{2m\lambda}\, g(|\nabla I(\mathcal{C}(q))|)|\mathcal{C}'(q)|dq.$$

Without loss of generality we can now set $2\lambda m = 1$ to obtain

$$\text{Min} \int_0^1 g(|\nabla I(\mathcal{C}(q))|)|\mathcal{C}'(q)|dq. \tag{10.3}$$

We have transformed the problem of minimizing (10.2) into a problem of geodesic computation in a Riemannian space, according to a new metric.

The Euclidean length of the curve $\mathcal{C}$ is given by $L := \oint |C'(q)|dq = \oint ds$, where s is the Euclidean arc-length (or Euclidean metric). In (10.3) a new length definition L_R, in a different Riemannian space, is given. Since $|\mathcal{C}'(q)|dq = ds$, from (10.3) we obtain

$$L_R := \int_0^{L(\mathcal{C})} g(|\nabla I(\mathcal{C}(q))|)ds.$$

Comparing this with the classical length definition L, we observe that the new length is obtained by weighting the Euclidean element of length ds by $g(|\nabla I(\mathcal{C}(q))|)$, which contains information regarding the boundary of the object. Therefore, when trying to detect an object, we are not just interested in finding the path of minimal classical length but the one that minimizes a new length definition which takes into account image characteristics.

In order to minimize (10.3) (or L_R) we search for the gradient descent direction of (10.3). Thus, according to the steepest-descent method, and the Euler-Lagrange of L_R as computed in [22], to deform the initial curve $\mathcal{C}(0) = \mathcal{C}_0$ towards a (local) minima of L_R, we should follow the curve evolution equation

$$\frac{\partial \mathcal{C}(t)}{\partial t} = g(I)\,\kappa\mathcal{N} - \langle \nabla g, \mathcal{N}\rangle \mathcal{N}, \tag{10.4}$$

where κ is the Euclidean curvature, $\mathcal{N}$ is the unit inward normal, and the right hand side of the equation is given by the Euler-Lagrange of (10.3).

To summarize, equation (10.4) presents a curve evolution flow that minimizes the weighted distance L_R, derived from the classical snakes case (10.2) via Maupertuis' Principle.

[1] The interpretation of the snakes model (10.2) for object detection as a geodesic computation is valid for any value of E_0.

10.2.3 The level-sets geodesic flow

The geodesic flow (10.4) is represented using the level-sets approach [111], as introduced in Chapter 2. Assume that the curve $\mathcal{C}$ is a level-set of a function $\phi : [0,a] \times [0,b] \to \mathbb{R}$. This representation is parameter free, then intrinsic. It is also topology free since different topologies of the zero level-set do not imply different topologies of ϕ [111]. By embedding the evolution of $\mathcal{C}$ in that of ϕ, topological changes of $\mathcal{C}(t)$ are handled automatically and accuracy and stability are achieved using the proper numerical algorithm [111]. This representation was formally analyzed in [25, 44, 134]. Based on (10.3) and embedding (10.4) in ϕ, we obtain that solving the geodesic problem is equivalent to searching for the steady state solution of the evolution ($\phi(0,\mathcal{C}) = \phi_0(\mathcal{C})$)

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \operatorname{div}\left(g(I)\frac{\nabla \phi}{|\nabla \phi|}\right) = g|\nabla\phi|\kappa + \langle \nabla g, \nabla \phi \rangle, \tag{10.5}$$

where the right hand of the flow is the Euler-Lagrange of (10.3) with $\mathcal{C}$ represented by a level-set of ϕ, and κ is computed on the level-sets of ϕ ($\kappa = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$). That means, (10.5) is obtained embedding (10.4) for $\beta = g(I)\kappa - \langle \nabla g, \mathcal{N}\rangle$. Equation (10.5) is the main part of the proposed active contours model.

10.2.4 Boundary detection

Let us proceed and explore the geometric interpretation of the geodesic active contours equation (10.5) from the point of view of object segmentation, as well as its relation to other geometric curve evolution approaches to active contours. In [21, 95], the authors proposed the following model for boundary detection:

$$\frac{\partial \phi}{\partial t} = g(I)(c+\kappa)|\nabla\phi|, \tag{10.6}$$

where $c \in \mathbb{R}^+$. Following [21, 95], equation (10.6) can be interpreted as follows: First, the flow $\phi_t = (c+\kappa)|\nabla\phi|$, means that each one of the level-sets $\mathcal{C}$ of ϕ is evolving according to $\mathcal{C}_t = (c+\kappa)\mathcal{N}$ [22, 111]. This equation was first proposed in [111], were extensive numerical research on it was performed. The flow $\mathcal{C}_t = \kappa\mathcal{N}$, denoted also as *Euclidean heat flow*, is well known for its very satisfactory geometric smoothing properties [48, 51]. (It was extended in [120, 121] for the affine group and in [121] for others.) The flow decreases the total curvature as well as the number of zero-crossings and the value of maxima/minima curvature. This flow also moves the curve in the gradient direction of its length functional L [51]. Therefore, it has the properties of "shortening" as well as "smoothing." This shows that having only the first regularization component in (10.1), $\alpha \neq 0$ and $\beta = 0$, is enough to obtain smooth active contours as explained in Section 10.2.1 when the selection $\beta = 0$ was done. The constant velocity $c\mathcal{N}$, which is related with classical mathematical morphology [119] and shape offsetting in CAD [68], is similar to the balloon force introduced in [32, 31]. This velocity is crucial in the above model in order to capture non-convex shapes since a convex curve remains convex when evolving according to the Euclidean heat flow [48]. Of course, c must be specified a priori in order to make the object detection algorithm automatic, being this a non trivial issue [21].

The external image dependent force is given by the stopping function $g(I)$. The goal of $g(I)$ is to stop the evolving curve when it arrives to the objects boundaries. In [21, 95], the authors choose

$$g = \frac{1}{1 + |\nabla \hat{I}|^p},$$

where $\hat{I}$ is a smoothed version of I and $p = 1$ or 2. For an ideal edge, $\nabla \hat{I} = \delta$, $g = 0$, and the curve stops ($\phi_t = 0$). The boundary is then given by the set $\phi = 0$.

In contrast with classical energy models of snakes, the curve evolution model (10.6) is topology independent due to the level-set representation [111]. This allows to detect any number of objects in the image, without knowing their exact number and without topology tracking procedures.

Comparing Equation (10.5) to (10.6), we see that the term $\langle \nabla g, \nabla \phi \rangle$, naturally incorporated via the geodesic framework, is missing in the old model. This term attracts the curve to the boundaries of the objects (∇g points toward the middle of the boundaries. See Figure 10.1).

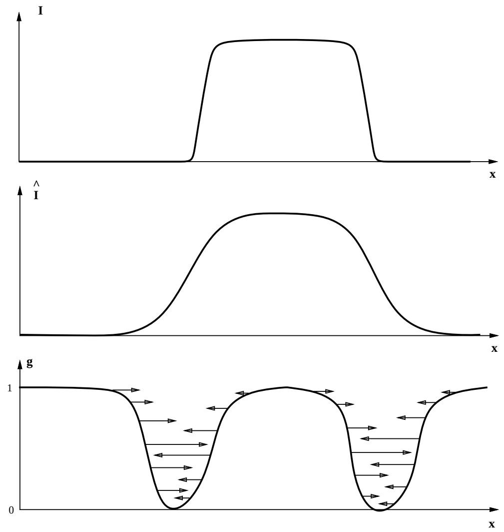

Figure 10.1: Geometric interpretation of the attraction force in 1D. The original edge signal I, its smoothed version $\hat{I}$, and the derived stopping function g are given. The evolving contour is attracted to the valley created by $\nabla g \cdot \nabla u$.

Note that in the old model, the curve stops when $g = 0$. This only happens at an ideal edge. In cases where there are different gradient values along the edge, as often happens in real images, g gets different values at different locations along the boundaries. It is necessary to restrict the g values, as well as possible gaps in the boundary, so that the propagating curve is guaranteed to stop. This makes the geometric model (10.6) inappropriate for the detection of boundaries with (un-known) high variations of the gradients. In the proposed model, the curve is attracted towards the boundary by the new gradient term. This new force increases the attraction of the deforming contour towards the boundary [22], being of special help when this boundary has high variations, including gaps, on its gradient values. The second advantage of this new term is that we partially remove the necessity of the constant velocity given by c. This constant velocity, that mainly allows the detection of non-convex objects, introduces an extra parameter to the model, that in most cases is an undesirable property. In our case, the new term will allow the detection of non-convex objects as well. In case we wish to add this constant velocity, in order for example to increase the speed of convergence, we can consider the term $cg(I)|\nabla\phi|$ like a "constraint" to the geodesic problem (10.3), obtaining

$$\frac{\partial\phi}{\partial t} = |\nabla\phi|\text{div}\left(g(I)\frac{\nabla\phi}{|\nabla\phi|}\right) + cg(I)|\nabla\phi|. \tag{10.7}$$

This equation of course means that the level-sets move according to $\mathcal{C}_t = g(I)(c+\kappa)\mathcal{N} - \langle\nabla g, \mathcal{N}\rangle\mathcal{N}$. Equation (10.7), which is the level-sets representation of the modified solution of the geodesic problem (10.3) derived from the energy (10.2), constitutes the general *geodesic active contour* model we propose. The solution to the object detection problem is then given by the zero level-set of the steady state. In [22] we present theoretical results, based on the theory of viscosity solutions, proving existence and uniqueness, stability, and consistency.

10.3 Experimental Results

The numerical implementation of (10.7) is based on the algorithm for curve evolution via level-sets developed by Osher and Sethian [111]. See [22, 111] and Chapter 2 for details.

The deforming contour ($\phi = 0$) is represented by a gray contour, and the final one (the geodesic) by a white one. For inward motion, the original curve surrounds all the objects. Figure 10.2 presents two wrenches separated by a few pixels. We applied the geodesic model (10.7) (inward motion) to the image, and indeed, both objects are detected. The original connected curve splits in order to detect both objects. The geodesic contours also manages not to be stopped by the shadows (false contours), due to the stronger attraction force provided by the term $\langle\nabla g, \nabla\phi\rangle$ towards the real boundaries. Figure 10.3 presents an outward flow. The original curves are the two small circles, one inside each of the objects. Note that both curves deform simultaneously and independently. In the case of energy approaches, disjoint curves must be tracked to ensure that they contribute to different energy functionals. This tracking is not necessary in our model. Both interior and exterior boundaries are both detected. The initial curves manage to automatically split and detect all the contours in both objects.

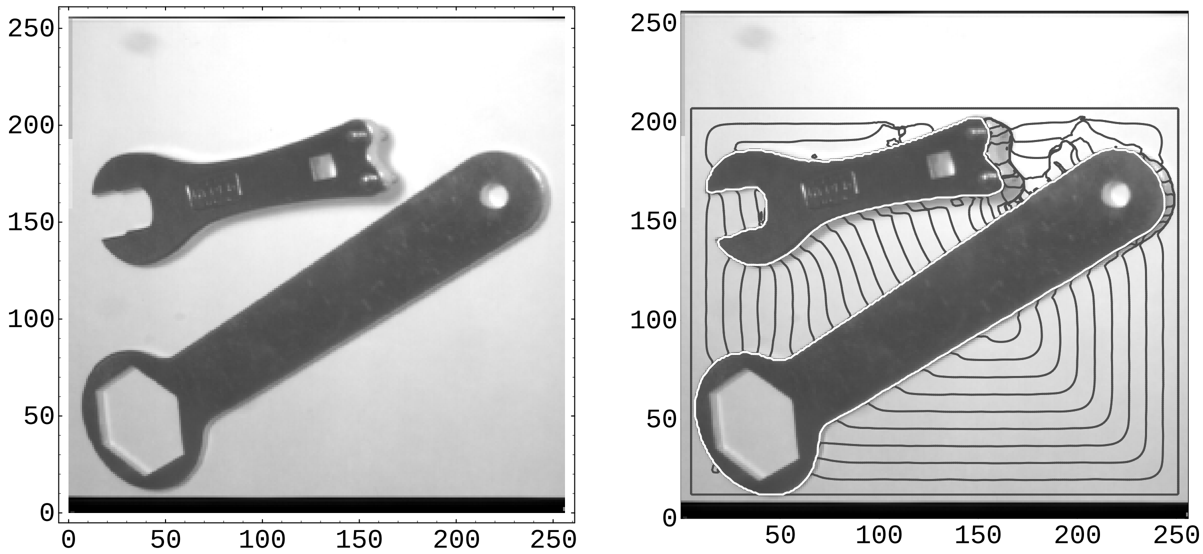

Figure 10.2: Inward motion to detect two objects, separated by only a few pixels. The original image is given on the left and the one with the deforming contours on the right. The deforming contour ($u = 0$) is represented by a green contour, and the final one (the geodesic) by a red one. The initial contour is given by the frame of the image, surrounding both objects.

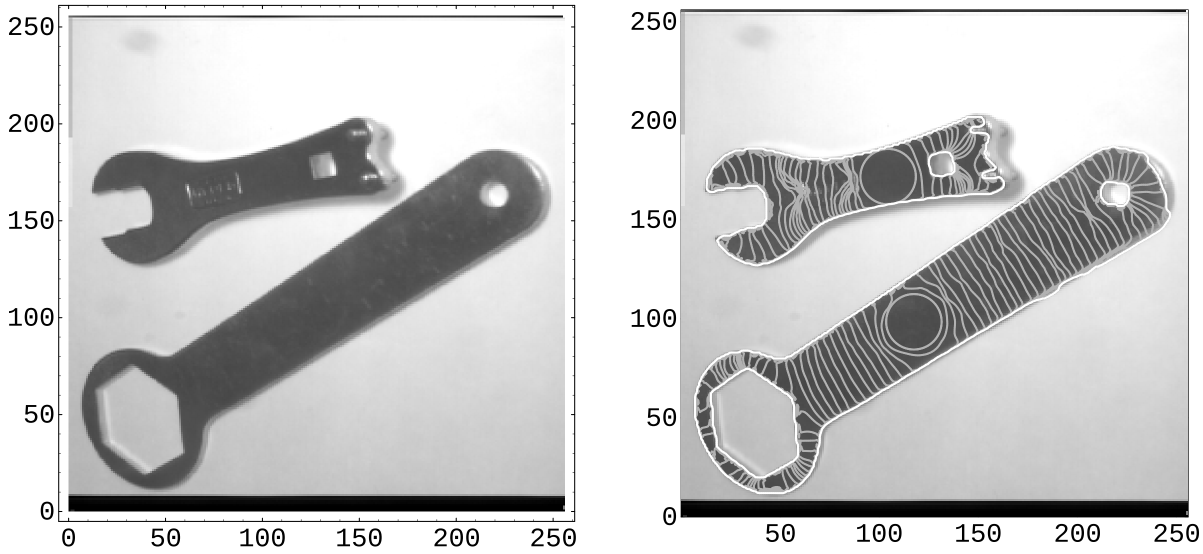

Figure 10.3: An example of outward flow. The original curves are given as two small circles inside the objects. Note that both curves deform simultaneously and independently. Both interior and exterior boundaries are detected. The initial curves manage to split and detect all the contours in both objects. This splitting is automatic.