Simulating the Grassfire Transform Using an Active Contour Model
Frédéric Leymarie and Martin D. Levine, Fellow, IEEE

Abstract—In this paper, we present a new method for shape description of planar objects that integrates both region and boundary features. Our method is an implementation of a 2-D dynamic grassfire that relies on a distance surface on which elastic contours minimize an energy function. A Euclidean distance transform combined with an active contour model, such as the snake, is used for this minimization process. Boundary information is integrated into the model by the extraction of curvature extrema and arcs of constant curvature. The use of an active contour on a field of grass, represented as a distance surface, combined with the curvature features of the boundary permits us to extract a Euclidean skeleton representation of the shape while bypassing many of the discretization problems found in other skeletonization algorithms. We propose a new concept for skeletal branch significance based on the notion of the local deformation introduced by symmetry points on the distance surface. We call this the ridge support. Furthermore, we show how the ridge support can be evaluated in terms of the velocity of a 2-D grassfire simulation.

Index Terms—Active contour model, contour and region shape features, deformable skeleton, distance surface, Euclidean skeleton, multiscale representation, ridge support, 2-D grassfire simulation.

I. INTRODUCTION

THIS PAPER presents a new method for shape description of planar objects in which both region and boundary features are extracted. We propose a new algorithm for shape skeletonization to achieve this.1 Skeletons are object representations that are particularly appropriate for the description of amorphous or biological shapes commonly found in nature for which other representational schemes based on ordinary geometry are inappropriate [18]. They can be used to encode visual cues such as symmetry, shape primitives, width information, and the process history of deformable objects. We argue that boundary information can be combined with a region-based representation such as the skeleton to produce a richer, more powerful and efficient shape representation. For example, extrema of curvature along the contour can be used to generate significant skeleton branches [32], [52].

Our method for shape description is an implementation of a 2-D dynamic grassfire [18] that relies on a distance surface [37] on which elastic contours minimize an energy function (Fig. 1). A Euclidean distance transform (EDT) combined with an active contour model, such as the snake [43], is used for this minimization process. Boundary information is integrated into the model by the extraction of curvature extrema and arcs of constant curvature.

A. Approach and Motivation

Following our previous research [32], we have tried to improve shape description algorithms based on the notion of the skeleton of a shape, that is, on its representation by "idealized thin lines" that retain the connectivity or topology of the original shape. The skeleton is believed to be the most powerful representation available at present for characterizing the shape deformations and evolution of shape subparts of nonrigid natural forms [32], [66]. The skeletonization of a shape, that is, the process by which the skeleton is obtained, permits us to explicitly relate significant boundary features to the internal structure of an object. In particular, we have used the grassfire transform as a skeletonization process; an object's boundary is taken as an initial fire front that propagates within the object's interior region. Points where the fire front folds or interacts with itself are retained as indicators of shape features such as symmetries, subparts, protrusions, and depressions. To simulate the fire propagation, we use the snake model [52] and show its advantages over previous approaches for shape skeletonization. However, our shape skeletonization method requires some initial processing of the object boundary to identify those particular curvature extrema where the fire front will collapse as soon as the fire is "ignited." To do so, we have proposed a new method for extracting contour features on the basis of morphological operators applied to the curvature function of the boundary [51].

B. Biological Basis for Shape Analysis

Within the context of this paper, we are principally concerned with the notion of biological shape or the shape of natural forms assumed by living organisms. How can we categorize this notion of shape? What do we mean by the shape of an object? Following the early ideas of Blum, we observe that biological shape is concerned with three kinds of problems [18]. The first is the "taxonomic" or descriptive problem
Fig. 1. Example of grassfire propagation on a blob-like shape. The potential surface on which the active contour is constrained to lie is shown at the top right corner. The dark lines superimposed on it represent the snake at its initial and final states. Such a potential surface is computed as a reversed distance right corner. The fire is ignited at time $t = 0$ and propagates outward. The complete set of skeletal points is recovered by those features that are emphasized by both the internal and external descriptors, namely, the curvature extrema and symmetric axes endings. C. Region or Boundary?

Are the two classes of descriptor, that is, boundary- and region-based, distinct or are they related? Can we express shape description by a continuum of information ranging from purely boundary descriptors to purely internal descriptors? Can we exploit both their descriptive powers and relate them in natural ways or is this dichotomy inevitable? Answers have been proposed in the past. For example, Leyton has shown that in the case of regular curves, there is a direct relation between boundary and region features [56], [57]. More precisely, to each extremum of curvature, there is a corresponding symmetric axis that emerges or terminates at it. Furthermore, Leyton has proposed that “shape [may be] understood as the outcome of processes that formed it.” The “process history is recovered” by those features that are emphasized by both the internal and external descriptors, namely, the curvature extrema and symmetric axes endings.

Axes of symmetry of an object may be defined in a number of ways in both the continuous and discrete domains. Essentially, a point in the interior of a shape is a symmetry point (part of a symmetric axis) if it is at an equal “distance” from two or more boundary points where the distance is any valid metric in the continuous or discrete domain. An equivalent definition is based on the law of propagating wavefronts where the boundary of the shape is taken as the source of a grassfire [17], [18]. The skeletal points are then defined as the locus of meeting wavefronts, here fire fronts, where the fire is quenched. The complete set of skeletal points forms the skeleton of the shape. With such definitions, the skeleton and the complete set of symmetric axes are equivalent [18].

D. A New Method for Shape Description

We are seeking a method of shape description that should answer three types of problems: developmental, perceptive, and descriptive. We know from psychophysics and neuropsychological experiments that shape in the visual cortex seems to be mainly feature based and that the curvature along a curve or contour is a key descriptor [33], [51]. From topological considerations, we know that contour-based descriptions alone are not sufficient to resolve biological shape. Furthermore, from psychophysical experiments, we know that any valid model for shape description should be invariant under certain image transformations, such as those arising from variations in illumination, color, rigid motion, and scale [11].

In order to emphasize the “pertinent” shape features, we will need to transform the figure-ground image into a useful representation. Thus, we assume that the binary image (object-nonobject) has been obtained from a preceding segmentation step. Emphasized shape features should reflect the characteristics of both the object’s outline and the structure of its interior. These features should also be invariant under the above-mentioned image transformations. In this context, the particular representation we propose to use will be expressed in terms of shape symmetries and their related curvature features.

1) Overview: In the subsequent sections of this paper, we will propose a method for biological shape and its imple-
mentation that should address these needs. First, however, we will briefly survey the different techniques used to obtain the skeleton representation (Section II) and briefly describe the snake model (Section III). Then, we will propose a novel implementation of the grassfire transform (Section IV). We have adopted this formulation for the extraction of shape symmetries because it explicitly emphasizes the relation between boundary features and region features. We will see how the formation of the skeleton is initiated at curvature extrema or centers of curvature where, in some sense, the "shape is concentrated." Here, the fire fronts start merging; in other words, the transforming shape, under the grassfire process, starts interacting with itself. The issue of integration of boundary information in our shape model will also be addressed (Section V). Following this, we will describe how a graph representation of the skeleton is easily obtained using the grassfire transform (Section VI). Such a hierarchical representation permits us to naturally keep only these branches of the skeleton that are significant. We will summarize the advantages of this model for biological shape; variations in the implementation of the algorithm will also be proposed (Section VII). Finally, we will conclude this paper by showing how the proposed model for biological shape can be used for the tracking and description of deformable objects such as cells (Section VIII).

2) Contributions: In this paper, a number of contributions are made. They are summarized below:

- We present a new method for representing and describing shape on the basis of an active contour model. This permits us to efficiently simulate the grassfire transform. For the first time, the fire fronts are considered as distinct entities with regard to the field of grass or the interior of a shape.

- We emphasize the relationship that exists between the grassfire transform and the notion of a "distance surface," which is a relation first mentioned by Kotelly [45] and Blum [18] and, for the most part, ignored since. We take advantage of this relationship to efficiently simulate the grassfire propagation by an active contour.

- The use of an active contour on a field of grass, represented as a distance surface, permits us to employ an Euclidean metric while bypassing discretization problems found in many other skeletonization algorithms also based on the Euclidean metric.

- Our method for shape explicitly relates boundary information to region information, which is a departure from most shape description techniques found in the literature.

- The use of both types of shape descriptors gives us a powerful tool for obtaining robust multiscale descriptions. In particular, this permits us to specify significance criteria and measures for each skeleton branch, thereby producing skeletons without spurious branches; this is a major advantage over most other skeletonization techniques.

- We propose a new concept for branch significance based on the notion of local deformation introduced by symmetry points on the distance surface. We call this the ridge support. Furthermore, we show how the ridge support can be evaluated in terms of the velocity of formation of a symmetric axis or in terms of the slope amplitude of a tangent to the symmetric axis.

- We propose a new concept (the deformable skeleton) that is useful for tracking deformable shapes. We refer to this as the dynamic skeleton.

II. SHAPE DESCRIPTION BY SKELETONIZATION

A skeleton is a representation of an object by idealized thin lines that retain the connectivity of the original shape [32], [39]. Skeletal points, when connected, form a skeleton. They can be defined in a number of ways in both the continuous and discrete domains. The earliest definition found in the literature had its roots in the concept of nearest-point mapping (or distance mapping) for a closed region or set $O$. In this case, the set of points in the background with more than one nearest point in $O$, that is, the set of skeletal points of the exterior of a shape, is considered important for characterizing the geometry of $O$ [23]. This concept was further explored by considering the locus of centers of osculating circles to $O$ (from the exterior) at more than one point to characterize the convexity or non-convexity of the set $O$ [64], [65]. Much later came Blum, who turned his attention to the interior of a region $O$ [15], [18]. Inside $O$, skeletal points can be defined as the locus of centers of maximal inscribed circles, that is, circles in $O$ not included in any other circle. Such circles touch the boundary of $O$ at more than one point. An equivalent definition is based on the law of propagating wavefronts, where the boundary of the shape is taken as the source of a grassfire [15]–[17]. The skeleton is then defined as the locus of meeting wavefronts. Brady and Asada define their skeleton as the locus of the chord midpoints of maximal inscribed circles [24]. They call this shape representation smoothed local symmetries (SLS). Leyton defines skeletal points as the locus of arc midpoints of maximal inscribed circles or minimal circumscribed circles [57]. He calls this representation the process inferring symmetric axis (PIASA).

The actual transform by which the complete set of skeletal points is recovered is referred to by different names such as the symmetric axis transform (SAT), the medial axis transform (MAT), the grassfire transform, shape skeletonization, and shape thinning.

A. Algorithms for Skeleton Computation

Over the years, since the first ideas of Blum emerged, various algorithms and implementations have been published. These can be classified into the following three different groups [52]:

1) Direct thinning of an object
2) Analytic computation of a skeleton based on an approximation of the object contour
3) Ridge following based on a distance mapping of the object.

Note that Blum also studied such a definition of skeletal points based on the locus of chord midpoints and called them symmetric chord coordinates [18].
We briefly mention the characteristics of each class of algorithm as well as its main drawbacks.

Most algorithms in the literature are based on the thinning concept [72], in which one iteratively peels off the contour of an object; this is an approximation of the grassfire process [39]. Sequential as well as parallel algorithms exist. These are iterative procedures whose computational time depends on the maximum width of the object. Their accuracy is limited since the flow of information when performing a peeling-off step is biased by the intrinsic connectivity of the digitized grid (four- or eight-connected grid in general).

Another class of algorithm uses a direct (analytic) computation of the symmetric axes based on the polygonal approximation of a shape. Here, the problem of accuracy is more critical for biological shape description since a polygonal approximation is often not sufficient. Objects with noisy boundaries would require very fragmented polygons with short sides, which could have the effect of making the results less accurate. The computation of symmetry points is generally more complex than for the other two classes of algorithms. In addition, objects with holes are generally difficult to process by such methods.

A third class of algorithm uses ridge following techniques based on distance transforms (DT's) applied to the object shape. Different DT's could be used: the city block DT [9], [71], the chessboard DT [5], the hexagonal DT, the chamfer DT's and other quasi-Euclidean DT's [34], [61], and the Euclidean DT [28], [41], [44], [52]. Skeletons based on the city block or chessboard DT can be computed very quickly and are assured to be connected within a fixed number of passes. This is due to the simplicity of the connectivity of skeletal pixels [6]-[8]. The drawback is that these two DT's are not accurate, yielding a 40 to 50% error in distance values with respect to the Euclidean distance. Furthermore, they are not as consistent as their Euclidean or quasi-Euclidean counterparts. In particular, they are not robust under planar rotation.

Methods based on hexagonal and quasi-Euclidean or Euclidean DT's rely on ridge following on the surface of a distance map to obtain the skeleton. They produce accurate and smoother results, but thickness and connectivity of the skeleton branches must be carefully checked [74]. For example, gaps often occur due to the spatial quantization of the digital grid and must be filled in, perhaps by using gradient ascent methods [28], [41], [44].

B. Advantages of Algorithms Based on Distance Mapping

The most attractive type of algorithm to date seems to be those based on distance transforms. The main advantages are as follows:

- Only simple computations such as mask convolutions are required. The number of passes is fixed and independent of the complexity of the object boundary.
- The skeletal pixels are labeled directly by the DT. The distance value from the background corresponds to the radius of a maximal osculating circle to the boundary.
- On a computational cost performance basis, these methods yield more accurate and smoother results than those of any other type of algorithm.

Consequently, we have decided to study this class of algorithm for shape skeletonization. We have attempted to improve performance in order to use the algorithm to track the shape of large numbers of cells in real time [66]. This has led to a new algorithm, which is presented in the following sections.

III. THE SNAKE MODEL: BASIC CONCEPTS

A snake is a model of a deformable curve or contour (if closed) composed of abstract elastic materials. This elastic contour in the model consists of two types of materials: strings and rods. The former makes the snake resistant to stretching, whereas the latter makes it resistant to bending. Such a deformable curve is activated by making it sensitive to the graph of a function of two variables embedded in R^3 (e.g., the graph of an image I(x,y) or the graph of a distance transform DT(x, y)). Such a graph can be seen as a 3-D surface H. The snake is constrained to lie on H under the action of some gravitational force g. In other words, a weight is assigned to the snake to make it fall down the slopes of the surface H.

Depending on the nature of the surface considered, the snake can be used for different purposes. In our case, we will treat the distance data as the surface H. The distance transform DT(x, y) can be represented as a 3-D surface with coordinates (x, y) used as the planar Cartesian reference system. Once H is determined, the idea is to have the snake lie on the surface, allowing it to deform according to the surface topography, while under the influence of g. Throughout this paper, we will fix the magnitude of this gravitational force to be constant, that is, \[|g| = g = mg\], where m is the constant mass of the snake, and g is the magnitude of a constant gravitational acceleration.

The snake is positioned on a surface H and, under the influence of gravity, will seek out valleys or ridges of the surface until it achieves some sort of equilibrium state. Such behavior is obtained by associating with the snake an energy function that depends on the height of the surface at the snake’s location (similar to a potential energy). The snake will then attempt to minimize its energy by seeking out local minima in height of the surface H.

The original snake model also provides us with external constraints or forces that will pull the snake toward significant features [43]. A complementary kind of external force is also provided, which permits the snake to be pushed away from undesirable features. It is worth noticing that this particular method of imposing external constraints is an indirect one because it relies on other external mechanisms, such as a
preanalysis of the DT data or user interaction to place these
constraints near the identified features. This process of impos-
ing external constraints will be clarified when considering the
integration of boundary information to the grassfire transform
(Section V).

In summary, we have an elastic contour positioned on a 3-D
surface on which external forces can be imposed. In addition,
a mass is assigned to the snake, which is then embedded in
an uniform gravitational field.7 Furthermore, the surface H on
which it lies can be seen to specify its gravitational potential
energy function. Hereafter, in this paper, we will refer to such
a surface by using the term potential surface to make clear
how it influences the snake motion. The gravitational field is
defined spatially by the surface coordinates (x, y), whereas the
gravitational potential energy values are given by the surface
height z = H(x, y). Depending on where a snake is positioned
on the potential surface, that is, at a certain height, it will have
given a gravitational potential energy term. Other influences
will also come from the previously mentioned internal (i.e.,
elasticity forces) and external constraints (Fig. 2).

We observe that a natural way of causing the snake to move
in order to reach a lower gravitational potential energy, by
seeking valleys or ridges in the potential surface, is to convert
potential energy to kinetic energy. The potential energy of
the snake Esnake is given by the sum of its gravitational
potential energy Efield and of the potential energy terms
obtained from the internal and external constraints acting on
it (Eint and Eext, respectively). Then, the kinetic energy is
dissipated by damping, resulting in the snake reaching a
new lower equilibrium, that is, lower in terms of potential
energy and, thereby, lower in terms of height. By using such
a natural description of the dynamics of an active contour,
the snake model was first described in terms of a Langrangian
formulation of the motion [77]. The dynamics of the snake
defined within such a formalism in the continuous domain
and the discretization of the equations of motion are briefly
described in Appendix A; a more detailed description can be
found in [49].

Given the Lagrangian formulation of the dynamics of the
snake, we observe that equilibrium is reached when a steady
state is achieved, that is, when the snake stops moving. This
proves to be a too strong a terminating criterion for the energy
minimization process. Instead, we have proposed to use a
different criterion called the steady-support criterion [49]. Its
basis consists of examining the data for which we are seeking
the features. For the snake model, we have the potential surface
H, and the features we are seeking are the valleys and folds
or ridges of H. Instead of seeking a minimum of the total
potential energy of the snake Esnake (see (A.2) in Appendix

7Gravitational field: A region of space in which a body (e.g., the earth)
exerts a force g on another body (e.g., the snake) through space.71 We
consider, in the case of the snake model, that the former body is much more
massive than the second one (the snake) and that it exerts a constant force of
magnitude g = mg on this snake.

8Gravitational potential energy: The work an object at height can do by
falling under the influence of a gravity force. The "gravitational potential
energy function" is then obtained as the possible states (or heights) this object
can assume in a gravitational field. In the case of the snake model, these states
are fixed by the surface H on which the snake is constrained to lie.

A) as a terminating condition of the optimization process, we
propose to search only for a minimum of the potential field
energy Efield (see (A.4) in Appendix A). In other words, the
stability criterion for the snake becomes one in which we seek
a minimum and stable height of the snake based directly on
the H topography and not on other snake constraints (i.e., Eint
and Eext). A simple implementation of this stability criterion,
which provides an adequate solution to finding this minimum
of Efield, consists of summing the local heights of the snake all
along its length and dividing this sum by the snake length. This
summation is computed for each snake element, or snaxel,
and each interval interpolating pairs of snaxels. Experimentally,
we have used simple straight lines between snaxels for the
interpolation, which is sufficient for most applications.

IV. SHAPE SKELETONIZATION BY WAVEFRONT PROPAGATION

We now introduce a new algorithm for computing skeletons.
This algorithm combines the advantages of algorithms based
on the Euclidean distance metric with certain important addi-
tional features: connectivity (which is implicitly ensured),
integration of both contour and region information, and a
multiscale description (which is immediately available).

A. The Grassfire Transform as a Potential Surface

Our method for computing a skeleton of a binary image is
based on an efficient implementation in the discrete domain
of the original grassfire algorithm. It incorporates a physical
analogy of grassfire propagation. This is achieved by igniting a
"fire" on the object's boundary points, which are the limits
of the field of grass. The fire then propagates at constant speed
within the object. Points at which the fronts of the fire merge
are retained as being on a symmetric axis. These correspond
to points that are at a maximal Euclidean distance from two
or more boundary points, or equivalently, at the center of the
maximal inscribed circle.

If we consider the grassfire to be a function of time, a space-
time graph known as a 2-D dynamic grassfire is obtained [18].
This generates a 3-D surface, as shown in Fig. 3, where the time variable is along the vertical axis \( \tilde{T} \) and where the \( \tilde{X} \) and \( \tilde{Y} \) axes correspond to the fire front coordinates.\(^9\) Each \((\tilde{X}, \tilde{Y})\) plane at a particular time \( t \) contains the fire front at that time.

We obtain a distance surface by replacing time by a distance coordinate.\(^10\) Such a surface has a maximal directional slope of one everywhere except along ridges, where it is undefined [37]. These ridges correspond to the locus of the skeleton (Fig. 3).

The resulting distance surface is equivalent to the distance map obtained by computing a distance transform (DT) on the initial figure-ground image [52]. A distance transformation on a binary image (object \( O \), nonobject \( O' \)) produces a mapping from a double-valued function \( f_1 \) in a space \( S_1 \) \((f_1(p) = 1 \text{ iff } p \in O \) and \( f_1(p) = 0 \text{ iff } p \in O' \)) to a multivalued function \( f_2 \) in a space \( S_2 \) of identical dimension. The mapping is applied to \( f_1 \) by minimizing a distance metric \( \text{dist}(p, p') \), where \( p \) and \( p' \) are two points in the space \( S_1 \). The distance values of \( f_2 \) are given by

\[
\begin{align*}
 f_2(p) &= \min_{p' \in O'} \text{dist}(p, p')
\end{align*}
\]

where \( p = p(\tilde{x}, \tilde{y}) \in O \). Therefore, the first step in our method consists of computing the distance transform of a discrete figure-ground image.\(^11\) We use a simple sequential algorithm for computing DT's based on Danielsson's work [28]. We have shown in [49] and [50] that this algorithm can be implemented in an efficient manner; this very fast implementation has a computational complexity comparable to the simple chessboard DT.

The second step involves simulating the grassfire propagation. To achieve this, we initiate an active contour (in our case, a snake) on the figure boundary. A snake consists of an elastic curve that tends to minimize its energy by sensing the potential surface on which it is located (see Section III). We take the potential surface \( H \) to be the negative of the discrete distance surface

\[
H(p) = \begin{cases} 
-f_2(p) < 0, & p = p(\tilde{x}, \tilde{y}) \in O, \\
0, & \text{elsewhere}.
\end{cases}
\]

This means that potential values correspond to null or negative distance values. It is necessary to invert the distance map in this way so that the snake will reduce its energy by falling down the potential surface. Ground is taken to be at potential 0, the figure contour corresponds to potential values of \(-1\), and inside the figure, the potential values vary from \(-1\) to the negative of the maximal possible distance (Fig. 4). We call the space containing 3-D potential surfaces \( H \), which is obtained from (2), the distance transform domain \( DT \).

Having initialized the snake on the locus defined by potential values of \(-1\), the snake is then activated by sensing the potential surface (ignition of the fire). This is accomplished by observing the first directional slopes of the potential surface \( H \) in the \( \tilde{X} \) and \( \tilde{Y} \) directions\(^12\) and using a gradient descent approach [49]. At each time step \( \Delta t \) of the iterative process, each snake segment is reevaluated based on the gradient, the elastic internal constraints, and certain other external constraints (see Section III). Points on the potential surface where the snake folds into thin lines, that is, where the fire fronts merge, are retained as the points of the skeleton. This occurs when the snake reaches an equilibrium under the "steady-support" criterion (see Section III) or stops moving. An example of the simulation of the grassfire transform using such a method based on an active contour model was given in Fig. 1.

B. Regions Containing Holes

When applying an active contour model to planar regions that are not topologically equivalent to a disc, such as 2-D ob-

\(\text{\textcopyright 1990 IEEE}\)
jects containing holes, a slightly more complex implementation of the grassfire transform is required.

Essentially, we must define as many fire fronts as there are boundaries on the object. For each internal contour of a region, that is, each contour delimiting a hole, we need to generate a supplementary active contour.\textsuperscript{13} Therefore, given a 2-D object limited by one external boundary and possibly $N$ internal contours, where $N$ is some fixed positive integer, we simply need to initialize $N$ supplementary snakes on the potential surface: one at each internal contour position. Each of the $N + 1$ snakes will simulate a particular fire front. We then activate each snake independently in parallel and use the same gradient descent approach to simulate the grassfire propagation, as for the case of a region without holes. Two types of skeletal points are then defined. First, as before, skeletal points are extracted where a given snake folds into a thin line. Second, skeletal points are also identified, where two or more different snakes merge or meet each other. The final stage of the grassfire propagation also occurs when the snakes reach a stable state, that is, when the "steady-support" criterion is satisfied or when they stop moving.

Since the snake's activation and inhibition, by the use of the "steady-support" criterion, are directly dependent only on the forces applied directly to the snake itself (elasticity and external constraints) and on the topography of the potential surface, but not on the presence or absence of other snakes in their vicinity, the actual simulation of the grassfire transform can be computed in a sequential fashion. This approach permits simpler implementations on traditional sequential computers. Furthermore, we can activate the $N + 1$ snakes in any order since they are created independently. An example of a sequential simulation of the grassfire transform for an annulus is given in Fig. 5.

V. INTEGRATION OF BOUNDARY INFORMATION

The second building block of our method, the first being the simulation of the grassfire process, is concerned with the integration of contour information within the skeleton computation. To do so, critical points of the boundary and, particularly, positive curvature extrema are extracted from the original object boundary. We refer to these positive curvature extrema by using the symbol $C$. Another type of critical point may exist within the interior of a shape, that is, the center of a maximal inscribed circle osculating $O$ at a minimum of one connected set of points along the boundary (i.e., an arc of circle). We will consider this second type of critical point in detail in Section V-A.

The snake is then attached to those critical points (the $C$'s) that will ultimately correspond to the branch end points of the skeleton. By "attached," we imply that a snaxel is pinned to the potential surface at the spatial position of the critical point. This may be implemented in two different ways: using an external force or partitioning the snake. The former is achieved using the so-called spring model (Appendix A). Furthermore,\textsuperscript{14}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig5.png}
\caption{Example of grassfire propagation on an object containing an annular region. In the first column, on the left, the fire propagation for a snake activated from the exterior boundary is shown. In the second column, the fire propagation for a snake activated from the interior boundary delimiting the hole is shown. In this second column, the result of the first fire propagation is also shown as a connected snake (line of smaller width). In this particular case, where the exterior and the interior contour do not delimit any protrusions, both snakes converge to the same solution.}
\end{figure}

\textsuperscript{13}Note that some preprocessing of the object may be required to eliminate insignificant holes of small size. For example, mathematical morphology operations could be used to fill in holes of a predetermined size [72].

\textsuperscript{14}The same idea applies to objects with holes. Every boundary delimiting a hole is processed to extract its significant positive curvature extrema at which a snake is attached.
A. Fire Front Propagation from Circular Arcs

There exists one case where the curvature extremum criterion is insufficient for determining the branch end node. This is the case of boundary segments that consist of arcs of circles that have their center of curvature situated within the interior of the object, which is also part of the skeleton (i.e., being a symmetry point), and is furthermore simultaneously an extremity of a ridge of the distance surface (i.e., being the end point of a skeleton branch). A simple procedure will be needed to stop fire propagation at the closest ridge point of the distance surface. Due to the uniformity of the maximal directional slope of the potential surface and of the gradient descent technique used to update snaxel positions, we are assured that snaxels ignited on an arc of a circle will reach the closest ridge points. This follows from the fact that snaxels crawl down the distance surface following the direction of maximal slope. It is easily shown that this direction is along a normal or a radial line to a boundary point. For all boundary points, such lines necessarily lead to the nearest ridge point due to the minimal distance constraint of (1). Blum called these lines of steepest descent, which lead necessarily to symmetry points, pannormals [18].

During fire propagation, it is necessary to check snaxels initially ignited on an arc of a circle to inhibit the fire propagation as soon as a ridge is reached. This is required to ensure that the end of the skeleton branch that corresponds to the center of curvature of an arc of a circle on the boundary is retained in the case of a ridge starting with an increasing directional slope, that is, a skeleton branch consisting of increasing absolute distance values with regard to the branch end. In this situation, if a snake were not fixed, it would continue to fall down along the ridge, as shown in Fig. 8. With the snake model, one way to check if a snaxel has reached a ridge is to examine the directional slope, which is used to activate the snake. If this slope is lower than one (i.e., the snaxel is on a ridge) or if it changes abruptly in direction (i.e., the snaxel went across a ridge), then we are in the vicinity of a ridge point of the potential surface (see Section B of Appendix B for more details).

However, an even simpler procedure can be used to efficiently detect a center of curvature that is also a symmetry point and an end of a skeleton branch. Let us first examine points that are simultaneously the center of curvature of a circular boundary arc and a symmetry point. We will refer to these points by using the symbol $C_A$. We note that the following two properties are shared by snaxels that should ultimately be merged at a $C_A$ point:

1. The snaxels are initially on an arc of circle that can be identified, for example, from curvature morphology analysis [51], [53]. We can thereby create a snake segment or fire front that we denote $S_A$.
2. During the grassfire propagation, snaxels of $S_A$ should get closer as the iterations progress.

Note that any method capable of extracting curvature extrema at many scales could be used (e.g., see [76]).
joined by straight lines. $C_A$ and $CA$ are the centers of the arcs. In the neighborhood of $CA$, the ridge is made of increasing absolute distance as shown by the potential surface. If the snake is attached at $CA$, the true skeleton is obtained as shown in (b). Otherwise, the snake shrinks as shown in (c).

Point 2 arises from the fact that pannormals to the boundary are all merge at one point: the center of curvature of $S_A$. Since along any pannormal, the slope of the potential surface is maximal, snaxels follow such directions and ultimately converge to one point if such a point exists in the interior of the object. This point of convergence is $C_A$. Therefore, we propose the following extraction procedure for $C_A$ points that are also end points of skeleton branches:

- If the snaxels of segment $S_A$ converge to one single point $C_A$ of the object interior $O$, then
  a) $C_A$ is a ridge point (may be verified directly on $H$).
  b) $C_A$ is the end of a skeleton branch if the snaxels that converged at its location are all constituents of a single fire front, namely of $S_A$. In other words, $C_A$ becomes a point at which the fire front $S_A$ collapses and gives birth to a symmetric axis.
  c) We consider $C_A$ as a critical point like the C's (Section V). Therefore, the snake should be attached at $C_A$, and optionally, the number of snaxels forming the fire front $S_A$ can be reduced to one single snaxel.

We now have all of the elements required to obtain a skeleton made of merged fire fronts where the fire propagation is modeled and simulated by an active contour model. The next stage in the skeletonization process consists of creating a graph representation of the skeleton, which is a step not often considered in the literature (but see [19], [52], [63], [68]). This is the subject of the following section where we propose a method for building the skeletal graph, which takes full advantage of the ensured connectivity of our active contour model.

VI. GRAPH REPRESENTATION

The connectivity of our skeleton obtained by fire front propagation is ensured due to the internal constraints of the active contour model. Thus, we can easily create a graph representation of the skeleton in terms of branches, end nodes, branching nodes, as well as relations between these elements (i.e., links).

The graph is generated at the termination of the propagation process, which is modeled by the snake segments (Fig. 9). These are completely connected and have no gaps. The actual curve approximation to the snake can be easily improved by increasing the number of snaxels once a “stable” skeleton has been obtained. Otherwise, interpolation by straight lines can be used if one is not looking for high spatial accuracy. To obtain the graph representation, we first label each individual fire front, that is, each individual snake segment $S$ defined between two consecutive critical points (Fig. 9). Critical points may be of two kinds: positive curvature extrema on the boundary $C$ or the center of curvature of circular boundary arcs $C_A$ that map to ends of skeleton branches.

The traversal of the snake segments from an initial curve extremum (labeled $C_1$ in Fig. 9) is done by associating pairs of snaxels of different snake segments that correspond to the same skeletal locus. For example, in Fig. 9, snaxels of both the first segment (label $S_1$) and the last segment (label $S_4$) should be found at each skeletal locus on the first branch (e.g., $b_1$).

This will be so until a point is reached at the branch where

---

19 As mentioned in Section II-A, methods for skeletonization based on ridge following on a Euclidean or quasi-Euclidean distance map do not, in general, directly produce a connected skeleton. This is because distances are evaluated only at integer coordinates.

20 In general, snaxels of different snake segments will overlap. However, because we are using a digital grid to simulate the grassfire transform, ridges may be of thickness two (pixels) rather than one. Therefore, snaxels of different snake segments may not always overlap but rather be (connected) neighbors. A simple matching procedure then needs to be defined by taking into account the cases of overlapping and connected neighbors and using the spatial connectivity of the snake segments.

---
more than two labels are found. This point corresponds to a branch node (e.g., \( n_i \)). The traversal can then be pursued to identify other branches, for example, the branch consisting of pairs of snaxels labeled \( S_1 \) and \( S_3 \) in Fig. 9. The traversal terminates when the initial snaxel is reached. Four simple rules can be used to generate the graph:

1) Each critical point (C or \( C_A \)) corresponds to an end node.
2) Each branch point consists of two snaxels from two different fire fronts or snake segments (\( S \) or \( S_A \)).
3) Each branch node consists of three or more snaxels from three or more different fire fronts.
4) Links, that is, the order relationship between branches, are defined at branch nodes during the traversal of the snake.

Branches are processed twice by traversing the complete snake. However, this can be altered so that the pairs of snaxels forming branches are visited only once.

A. Graph Pruning and Branch Significance

Not all branches of the skeleton are actually significant. Thus far, the branches terminating at an end node have been generated based on curvature criteria (see Section V) or the extraction procedure of center of curvature \( C_A \) (see Section V-A). In the case of the branches generated from positive curvature extrema \( C \), if the thresholds on relative curvature amplitude and region of support were relaxed, we might obtain spurious branches. However, it is possible to evaluate the skeletal branches on another basis, that is, by considering the ridge support along the branch.

We examine ridge support computed at each skeletal pixel by looking at the local deformation introduced by the ridge on the distance surface at that skeletal pixel position [52]. If the deformation is limited in its spatial extent after a certain point along the axis branch, then it is not “significant.” The use of ridge support as a branch significance criterion is justified by the fact that noise or small deformations on the object contour generate small deformations of the distance surface along the corresponding axis branch. The relative number of skeletal pixels that lie on that part of the ridge that is changing significantly can be used in combination with the curvature significance measures to decide whether or not a branch should be pruned from the graph (Fig. 10).

The notion of ridge support based on the local deformation introduced by a ridge on the distance surface is an intuitive or qualitative formulation; we need an easily computable measure to apply such a notion for graph pruning. For this, let us first return to the work of Blum, who first proposed the use of the velocity of skeleton branch formation, that is, the speed at which fire fronts merge, as a measure of the “smoothness of [fire] fronts” as they collapse at symmetry points [17], [18]. We will refer to this velocity of formation of a skeleton branch or symmetric axis by the symbol \( V_A \). Smoothness, as Blum refers to it, corresponds to our idea of local deformation of the distance surface. The distance surface is, in fact, a kind of photographic memory of the complete grassfire propagation.

Where fire fronts progress “smoothly,” the distance surface is locally smooth. This corresponds to points of maximal slope of one. Where fire fronts collapse or merge with each other, the distance surface deforms, and a ridge is created. The relative lack of smoothness of the deformation introduced by merging fronts depends on the angle at which they meet. Blum called such an angle the generating angle [17], [18]. If the generating angle is sharper, the velocity of formation of the symmetric axis \( V_A \) and the local deformation of the distance surface is greater. The limiting cases are the parallel fire fronts that merge at an infinite \( V \), and create a maximal deformation of the distance surface (Fig. 11).

In the continuous domain, \( V_A \) can be defined as the time (directional) derivative of length along a symmetric axis \( l_A \):

\[
V_A = \frac{\partial l_A}{\partial t}.
\]
Fig. 11. Velocity of formation of a symmetric axis and ridge support. In (a), fire fronts generated by two isolated points \(P_1\) and \(P_2\) for an unlimited field of grass are shown. The fire fronts are illustrated as an alternating wave pattern. The vertical centered line between the excitation points (SA) is first created at a midpoint \(R_0\) on a straight line segment joining the two points with an infinite speed \(V(A(R_0)) = \infty\). As the symmetric axis \(SA\) grows, its velocity decreases asymptotically to the space velocity \(V(A) \rightarrow 1\). For example, \(V(A(R_0)) = \sqrt{2}\). Note that as one travels along the symmetric axis, away from the initial symmetry point, the "generating angle" (see text) increases asymptotically from a completely closed angle of \(0^\circ\) at \(R_0\), to a completely open angle of \(180^\circ\). For example, at \(R_1\), the generating angle equals \(90^\circ\) (adapted from Fig. 1 of [17]). In (b), the potential surface representation of this grassfire transform is shown. The only ridge on this surface corresponds to the perpendicular SA between \(P_1\) and \(P_2\). Note how the deformation produced by merging fire fronts (in (a)) varies along the SA. In correspondence to the symmetry point \(R_0\), the deformation is maximal; at this point, the surface folds with an angle of \(90^\circ\). If you are further away from \(R_0\) along the SA, the deformation is smaller.

This derivative is taken in the direction of increasing time.

Note that \(V_A\) is not, in general, well defined at its extremities (e.g., at branch nodes). However, in practice, for the purpose of graph pruning, we will evaluate \(V_A\) only along branch points while not considering extremities such as branch nodes. However, even along a single open branch (without extremities), \(V_A\) values may not vary smoothly, that is, the associated velocity or quench function may have singular points, where the above derivative is undefined. Nevertheless, in the discrete domain, for our purpose of graph pruning, these limit problems will be of little concern, especially when approximating derivatives by finite differences.

Therefore, in the discrete domain, we can approximate \(V_A\) at branch points by finite differences

\[
\bar{V}_A = \frac{\Delta \bar{I}_A}{\Delta \bar{t}}
\]

where \(\bar{I}_A\) is the Euclidean distance of the discrete path along the symmetric axis from one symmetry point to another and \(\Delta \bar{t} \geq 0\), but since we are using a distance surface to "memorize" the complete grassfire, we can replace time by the height of the distance surface obtained from the EDT:

\[
\bar{V}_A = \frac{\bar{I}_A}{|\Delta AlternativeEDT(\bar{I}_A)|}.
\]

As we have seen, in the case of parallel merging fire fronts, \(\bar{V}_A\) may reach infinite values. Furthermore, \(\bar{V}_A\) is always greater than one, that is, it is always greater than the space propagation velocity of nonmerging fire fronts [17], [62]. Therefore, rather than using \(\bar{V}_A\) directly as a measure of ridge support, we propose to use its inverse, that is, \(\bar{V}_A^{-1} (0 \leq \bar{V}_A^{-1} < 1)\).

In fact, \(\bar{V}_A^{-1}\) can be interpreted as the discrete version of the amplitude of the directional slope at a symmetry point in the direction of the tangent (if it exists) to the symmetric axis.\(^{22}\)

We refer to this slope amplitude by using the symbol \(M_A\):

\[
M_A = \frac{1}{\bar{V}_A} = \frac{|\Delta AlternativeEDT(\bar{I}_A)|}{\bar{I}_A}.
\]

\(M_A\) varies from null values (for parallel merging fronts) to unitary values at which no symmetry point can exist (i.e., \(M_A < 1\); see Section B of Appendix B). Fixing a threshold \(M_{A_{max}}\), on the highest acceptable values for \(M_A\) (\(M_A \leq M_{A_{max}} < 1\)) provides us with a simple quantitative criterion for graph pruning on the basis of ridge support (Fig. 10).

Finally, we note that the slope amplitude at a symmetry point \(M_A\) provides a good first approximation to the boundary-axis weight of Blum and Nagel [19]. The boundary-axis weight is another kind of branch significance measure assessing the "importance of [the symmetry points in] representing the boundary" [19]. Essentially, the boundary-axis weight is defined as a "limiting ratio of boundary length to symmetric axis length." Because this measure is defined in the continuous domain within the framework of differential geometry, it is, in general, difficult to evaluate it accurately and use it for practical applications. In contrast, \(M_A\) is easily evaluated in the discrete domain and provides for accurate results as a consequence of our use of a Euclidean metric in the computation of the distance surface. It can also be shown that \(M_A\) is closely related to the boundary-axis weight and is even equivalent to it for straight portions of a symmetric axis and for symmetry points with infinite \(\bar{V}_A\).

VII. NEW METHOD FOR SHAPE SKELETONIZATION: CHARACTERISTICS

In this section, we first summarize the main strengths of our method for skeletonization of planar shapes and then present extensions that show the flexibility of our approach.

A. Strengths of Our Method

In addition to the benefits of using a Euclidean distance mapping, our algorithm for shape skeletonization possesses some new features and integrates the strengths of most previously reported algorithms, as described below.

• The integration of boundary information permits us to reduce noise sensitivity and to obtain an explicit multiscale description in terms of boundary convexity significance.

• Our skeletons are always connected due to the nature of the active contour model we have used. We are actually fitting

\(^{22}\) Montanari has previously used \(\bar{V}_A^{-1}\) as a criterion for branch significance [61].

\(^{23}\) Note that the directional slope of a symmetry point is not defined (geometrically) on the same basis as the directional slope of the distance surface (see Appendix B). In fact, at symmetry point locations, the directional slope with respect to the distance surface is undefined (i.e., the surface is locally nonregular). Rather, the directional slope of a symmetry point is defined with respect to a 3-D curve, the trace of which is given by the symmetric axis.
elastic strings to a potential surface, where these strings are implicitly connected due to their internal constraints.

- Combining convexity significance with ridge support along a skeleton branch permits us to further prune the skeleton graph, if necessary, thereby rendering skeletons with no spurious branches.
- Due to the connectivity of our skeleton model, a graph representation can be easily obtained. A hierarchical model of the object could be built by integrating the convexity significance and ridge support of a branch with the graph representation.
- The active contour model permits "friendly" user interaction when required. For example, in certain cases, it might be useful to observe the effect of different curvature criteria on the multiscale representation. In addition, we may wish to compare different kinds of skeletons that can be obtained: SAT-like, PISA-like [57], or augmented skeletons possessing branches terminating at concavities (see Section VII-B). New branches can be added simply by using a spring model to attach the snake to particular points on the distance surface. Conversely, branches can be deleted by simply removing the corresponding spring forces.
- The active contour model also provides us with a natural way of defining dynamic skeletons. "Dynamic" refers to skeletons that evolve with time. This should be important when processing nonrigid objects such as cells [32], [52], [66]. With an appropriate sampling rate, generally small deformations will occur from frame to frame, and we can use the previously computed skeleton (at time \( t - \Delta t \)) to initialize the snake on the new distance surface at time \( t \). A stable skeleton is then rapidly obtained since the snake is already close to the optimal solution. If larger deformations occur, that is, new curvature extrema appear or old ones disappear, we can use the previously computed skeleton as an initial guess but must then add or delete spring forces to generate or drop branches, respectively (Fig. 12).

- In comparison with the fastest known serial algorithms for skeletonization of 2-D shapes based on distance mapping, our algorithm has comparable numerical complexity. Our implementation of the Euclidean distance mapping is as fast as the chessboard distance mapping [49], [50]. The next step (grassfire propagation using the active contour model) necessitates less computation than most other methods. These require a "visit" to each pixel in the distance map to detect ridges, followed by an additional pass over the distance map to fill-in gaps between detected ridges (e.g., [44]) or to thin down overly "fat" ridges (e.g., [74]). In general, a "visit" to a pixel of the distance map implies access to the eight neighbors of this pixel. In our case, by using snake segments bounded by critical points \( C \)'s and \( C_A \)'s with a low spatial sampling rate, that is, using few snaxels, and due to the implicit connectivity of our active contour model, a solution is found in fewer steps (see also Section VII-B and Fig. 13). On the other hand, our method requires a supplementary step when compared with most methods, namely, the extraction of curvature extrema. This step can be accelerated by using techniques such as the hierarchical discrete correlation method [26] to filter orientation data along the boundary [51], [53] as well as by using the notion of ridge support to obtain a simple curvature significance criterion. Parallel implementation of our algorithm could also be achieved since both the EDT and active contour model permit parallel computation. Furthermore, in addition to being fast, our method is accurate and provides a simple way of building a graph from the skeleton.

One of the most interesting advantages of our method is the flexibility that the active contour model provides in order to compute skeletons in different or more efficient ways. This idea will become clearer in the following paragraphs.

\( \text{Fig. 12. Example of the use of dynamic skeletons for the shape representation of a living cell in two successive frames: } f_0 \text{ and } f_1. \text{ Springs used to fix the snake at curvature extrema along the boundary are shown by dots. In (a), the cell shape at frame } f_0 \text{ with its skeleton (stable state reached after 35 iterations) is shown. The new cell shape is shown at frame } f_1 \text{ in (b) to (f). In (b), the snake is initialized at the previously computed skeleton position. The effect of the removal of all springs corresponding to the curvature extrema found at frame } f_0 \text{ is shown in (c). In (d), two new springs are used to attach the snake to the two most significant curvature extrema. An intermediate stage between (d) and the final result is shown in (e). Finally, in (f), we show the new stable state of the snake reached after only 15 iterations.} \)
creasing number of snaxels. In (a), a first coarse solution is obtained after 5 iterations. In (b), a more refined solution is then rapidly obtained by increasing the number of snaxels. In (c), the final result for a continuous snake is shown.

Fig. 13. Example of a grassfire propagation in three stages, with an increasing number of snaxels. In (a), a first coarse solution is obtained after 5 iterations. In (b), a more refined solution is then rapidly obtained by increasing the number of snaxels. In (c), the final result for a continuous snake is shown.

snake segments merge. If snaxels of different snake segments overlap or are connected neighbors, they create a new symmetry point, thereby implying that no further processing is required for these snaxels. This further implies that as fire fronts merge, the grassfire propagation will move faster in terms of computer time since fewer snaxels will have to be updated. The best way to implement such a grassfire simulation is by “shortening” the snake as snaxels from different snake segments merge. The removal of snaxels can be performed in two ways: introducing position discontinuities by breaking the snake into parts [49] or reinitializing shorter snakes (with fixed extremities) (one for each shortened snake segment or fire front).

- Different kinds of skeletons can be obtained by using boundary information. For example, if one extracts the center of curvature of arcs of circles osculating the boundary, SAT-like skeletons are obtained [19]. If, instead, end nodes are placed at the middle of these arcs of circles (i.e., on the boundary), PISA-like skeletons [56] are generated.

- Our method seems to be naturally extendible to 3-D problems. In this case, the active contour model becomes an active surface model [79]. The potential surface becomes a potential volume and is computed using the same EDT algorithm but this time for 3-D objects (for examples of DT’s in 3-D, see [21], [67], and [80]). Curvature information can also be used, this time by looking at the extrema of principle curvature of the bounding surface defining the shape of the object [85]. The physical analogy becomes one of wave propagation in space instead of fire propagation on a planar surface. Loci where the waves merge define the skeleton in terms of three types of geometrical entities: points, curves, and surfaces.

- Finally, we note that an alternate sequential approach to skeleton generation based on the same main features we have used to implement the grassfire propagation algorithm is possible. This is based on a ridge following algorithm. The idea would be to allow snakes to “grow” from curvature extrema toward local maxima of the distance surface. Thus, snaxels would descend along ridges of the distance surface as they are generated. Branches would therefore be generated one by one, and the graph representation would be directly obtained. However, problems might occur with the connectivity of the generated graph. Such “growing” snakes can also be used after the grassfire transform has been performed to segment a region into subparts. For example, branches can be generated in correspondence with concavities by initially fixing snakes at negative curvature extrema and then making them grow by descending the distance surface in the steepest gradient direction.

This provides a new way of segmenting a planar shape (Fig. 14). Note that since concavities generate panormals that are radials orthogonal to the boundary [18], there exists more than one direction (of maximal slope amplitude equal to one) for the snake to crawl down the distance surface when starting at a concavity. In Fig. 14, we use the direction of the radial orthogonal to the average orientation at the negative curvature extremum. The “average orientation” may be obtained from the smoothed or averaged chain code representation of the boundary [49].

VIII. CONCLUSIONS

In this paper, we have presented a new method for shape description of amorphous planar objects on the basis of an active contour model. Such a model has permitted us to combine a class of contour-based shape descriptors (i.e., curvature features) with their counterparts: a class of region-based shape descriptors (i.e., symmetry points). We have been careful in optimizing numerous implementation aspects of our shape description method. We have used a Euclidean metric for optimal accuracy, and the active contour model has permitted us to bypass some of the discretization limitations inherent in using a digital grid. We have performed noise filtering on the basis of both contour feature measures and region measures, that is, curvature extremum significance and ridge support, respectively, to obtain “robust” shape descriptors. We have also proposed other improvements and variations of the algorithmic implementation. Our motivations for the design

24. Extracting curvature information from a 3-D surface becomes the most difficult problem to solve in comparison with the 3-D EDT computation and the simulation of wave propagation. This might reduce the applicability of our method for 3-D objects.

25. This idea of using growing snakes has been previously reported and applied to the contour extraction problem in noisy images [14], [29], [86].
The usual skeleton generated growing snake segments from significant concavities. In (c), the segmentation significant convexities address these needs.

In the following subsection, we describe the form of a cell or nonrigid body in both the static and the dynamic cases [66]. In the following paragraphs, we summarize how our shape deformation in time and the evolution of pseudopods can also be favorably described by a skeleton-based approach. The “history” of the skeleton is used for this purpose. For example, branches corresponding to convex deformations that persist over a long period of time are kept or labeled as significant historical events that have the potential to lead to the formation of pseudopods. On the other hand, branches that do not survive can be removed from the process history of a cell and are considered to be noisy events. This suggests another meaning of noise filtering, which should be compared with the ridge support and curvature extremum significance criteria. Pseudopod activity can also be described using the skeleton representation. As the symmetric axis varies in its attributes, such as length and associated width function, or in comparison with other axes (e.g., pseudopod dominance with respect to the cell motion [66]), the process history of the cell subpart can be recovered a posteriori and predictions made about future deformations and motion direction. From the point of view of computational efficiency, our particular implementation on the basis of an active contour model permits us to simplify the tracking of deformations and the evolution of protrusions (Fig. 12).

Following the path pioneered by Blum, we have attained our initial goals, within the scope of this research, of designing and implementing a powerful shape description method particularly well suited to tracking amorphous forms such as cells. We are now in a position to perform experiments with cell tracking in order to refine our understanding of the “social behavior” of groups of cells and to achieve even more insight into biological shape problems.

APPENDIX A
DYNAMICS OF THE SNAKE

Consider a deformable curve \( v(s, t) \) with parameters \( s \) (spatial index) and \( t \) (time), defined on given open intervals \( \Omega \) and \( T \), respectively. Let us consider this deformable curve to be a function of two variables \( x \) and \( y \) (e.g., spatial coordinates) having the same parametrization as \( v \):

\[
v(s, t) = (x(s, t), y(s, t)) : s \in \Omega, t \in T.
\] (A.1)
The potential energy function of the snake $E_{\text{snake}}(v)$ is defined as [43]:

$$E_{\text{snake}}(v) = \frac{1}{2} \int_{\Omega} \left[ E_{\text{int}}(v) + E_{\text{ext}}(v) + E_{\text{field}}(v) \right] ds.$$  \hspace{1cm} (A.2)

$E_{\text{int}}$ represents the internal potential energy of the snake. It is a function of both bending and stretching forces applied to the snake. $E_{\text{ext}}$ gives rise to external constraint forces. $E_{\text{field}}$ gives rise to gravitational field forces. The internal potential energy of the snake $E_{\text{int}}$ is defined as follows:

$$E_{\text{int}}(v(s)) = \omega_1(s)|v_x|^2 + \omega_2(s)|v_y|^2$$  \hspace{1cm} (A.3)

where $v_x \equiv \frac{\partial v}{\partial x}$ and $v_y \equiv \frac{\partial v}{\partial y}$. The first order term $\omega_1(s)|v_x|^2$ makes the snake behave like a string (i.e., resists stretching), whereas the second order term $\omega_2(s)|v_y|^2$ makes the snake behave like a rod (i.e., resists bending). The weight $\omega_1(s)$ regulates the tension of the snake, whereas $\omega_2(s)$ regulates its rigidity. Position or tangent discontinuities may be introduced if a discontinuity can occur at this snake point. Furthermore, if we set $\omega_1(s) = \omega_2(s) = 0$, no internal constraint exists between $s_0$ and its immediate successor $s_0 + 1$, forcing a position discontinuity between these two snakes.

In the distance transform domain $DT$, the external potential energy $E_{\text{ext}}$ arises from spring like forces. A spring force $f_{\text{spring}} = -k_{\text{spring}}(P_1 - P_2)$ may be created between a fixed point $P_1(x_1, y_1)$ on the potential surface and a snake element, or snaxel, with coordinates $P_2(x_2, y_2)$ by adding the energy term $E_{\text{spring}} = -\frac{1}{2}k_{\text{spring}}(P_1 - P_2)^2$ to $E_{\text{ext}}$ [43]. Springs are used to force the snake to attach to desirable features, such as curvature extrema of a contour (S). The potential field energy $E_{\text{field}}$ can be derived from the classical gravitational potential energy equation (see, for example, [36]) in a point-by-point fashion as follows:

$$E_{\text{field}}(v(s, t)) = \mu g z(v(s, t))$$  \hspace{1cm} (A.4)

where $\mu$ is the constant mass density of the snake, $g$ is the magnitude of the gravitational acceleration, and $z(v(s, t))$ is the height or potential value at a snake point or element and at time $t$ on the potential surface $H$.

Given the potential energy function $E_{\text{snake}}$ (see (A.2)) for a specific initial position, a minimization procedure can be applied to reach a more stable energy state by converting potential energy to kinetic energy and then dissipating this kinetic energy through an energy dissipation function. In such a case, the Euler-Lagrange equations of motion are the following (see [49], [77] for details):

$$\mu x_{tt} + \gamma_x + \frac{\partial}{\partial s}(\omega_1(s)x_x) + \frac{\partial^2}{\partial s^2}(\omega_2(s)x_{xx}) = -\frac{1}{2}(E_{\text{ext}}(v) + E_{\text{field}}(v))$$  \hspace{1cm} (A.5a)

$$\mu y_{tt} + \gamma_y + \frac{\partial}{\partial s}(\omega_1(s)y_y) + \frac{\partial^2}{\partial s^2}(\omega_2(s)y_{yy}) = -\frac{1}{2}(E_{\text{ext}}(v) + E_{\text{field}}(v))$$  \hspace{1cm} (A.5b)

where $x_{tt} = \frac{\partial x}{\partial t^2}$, $y_{tt} = \frac{\partial y}{\partial t^2}$, $E_{\text{ext}} = \frac{\partial}{\partial s}(E_{\text{ext}})$, $E_{\text{ext}} = \frac{\partial}{\partial s}(E_{\text{ext}})$, $E_{\text{field}} = \frac{\partial}{\partial s}(E_{\text{field}})$, $E_{\text{field}} = \frac{\partial}{\partial s}(E_{\text{field}})$, and where $\gamma$ is the constant damping density or viscosity factor. Note that associated with such differential equations are appropriate initial and boundary conditions at $t = 0$ and at the extremities of the interval $\Omega$, respectively. This issue needs only be considered when discretizing the equations of motion [49]. Let us summarize the way these equations model the dynamics of the snake. On the left-hand-sides (LHS’s), the first two terms represent inertial $(\mu(x_0))$ and damping $(\gamma(s))$ influences; the former is proportional to acceleration, and the second is proportional to speed, as one would expect. The next two terms represent the elasticity forces between snaxels in terms of tension $(\omega_1(s))$ and rigidity $(\omega_2(s))$. The former is proportional to the spacing between snaxels, whereas the latter is proportional to curvature [49]. On the right-hand-sides (RHS’s), the first term represents the influence of the external constraints as a function of the first spatial derivatives of the energy $E_{\text{ext}}$ (i.e., giving rise to external forces). The second term represents the influence of the potential surface as a function of its negative slope in the $x$ and $y$ directions. The latter will force the snake to follow the potential surface topography in the direction of highest negative slope in a fashion similar to a steepest descent technique. Therefore, the LHS’s represent the intrinsic forces acting on or within the snake, whereas the RHS’s represent the extrinsic forces which are independent of the snake’s nature.

A. Discretization of the Equations of Motion

For the snake model, the space and time domains, the external forces, the snake itself, and the equations of motion must be discretized so that the model is compatible with the digital nature of the computer [43], [78]. First, consider the discretization of a region $\Omega$ of the space $\mathbb{R}^3$, in which the potential surface $H$ exists. This region $\Omega$ can be discretized by defining three discrete sets $X$ and $Y$ for the spatial coordinates and $Z$ for the field values or valid potential surface heights. This permits the confinement of the potential values in $\mathbb{R}^3$ to a restricted set of values $\{x, y, z\} \in \Omega$ to a restricted set of values $\{x, y, z\} \in \Omega$ that represents the restricted discrete domain. The sets $X$ and $Y$ consist of integer values varying from 0 to some given maximum indexing values $M_X$ and $M_Y$, respectively: $\{x \in X : x = \{0, \ldots, M_X\}\}$ and $\{y \in Y : y = \{0, \ldots, M_Y\}\}$. These two sets define indices on the potential surface. A third set is then defined for the restricted surface values. It consists of real numbers varying from a given minimum $Z_{\text{min}}$ to a given maximum $Z_{\text{max}}$ such that $\{z \in Z : z = \{Z_{\text{min}}, \ldots, Z_{\text{max}}\}\}$. Examples of such discrete domains $\Omega$ are the image domain $\Omega$ containing digital intensity images $(I(x, y))$ and their filtered versions and the distance transform domain $\Omega^D$, containing digital distance transforms of 2-D regions or objects [49].

Like the region $\Omega$, the external forces must also be spatially discretized. This is done by using the same discrete spatial coordinates $(x, y) \in (X, Y)$ when evaluating the spring forces.

---

26Barred symbols refer to discretized variables (e.g., $(x, y) \rightarrow (\bar{x}, \bar{y})$).
Unlike the previous cases, the snake must be discretized in not one, but two, domains: space and time. The spatial discretization is done by regularly tessellating the interval $\Omega$ into $M_z$ nodes (i.e., $M_z$ snaxels), leading to a discrete set $\Omega = \{ s \in \bar{\Omega} : \bar{\Omega} = \{ 0, \ldots, M_z - 1 \} \}$. We use the symbol $\Delta s$ to represent the distance or spatial step between successive nodes. The time discretization is achieved by considering a discrete time interval $T$, defined to start at time 0. It consists of values $t$ regularly separated by a constant time step $A_t$. Theoretically, we can activate a snake for an infinite period of time; therefore $T$ is the open set: $T = \{ t \in T : T = \{ 0, At, 2At, \ldots \} \}$.

Having discretized the two parameters $s$ and $t$ that describe the snake (see (A.1)), its discrete version can now be stated as follows:

$$v(s, t) = v(s + 1, t - 1)$$

This spatially relates the discrete snake to the same grid $(\bar{x}, \bar{y})$ used to discretize the domain $\Omega$ and the external forces.

The next step involves the discretization of the equations of motion of the snake (see (A.5)). As stated previously, a discretization in both space and time is required. For such differential equations, we use the finite difference method as the discretization technique to approximate the first- and second-order derivatives in both space and time [43], [78].

The discrete version of the equations of motion can then be derived. This results in a set of two systems of $M_z$ linear equations:

$$AX = Bx$$

where on the LHS’s we use the following identities:

$$A = \frac{\gamma}{2} I + K$$

with $I$ denoting the identity matrix. Matrix $K$, called the stiffness matrix, represents all internal elasticity relations of the snake. It is defined on the top of this page, using the following identities:

$$a_x = \omega_2 (s + 1)$$

$$b_x = -2\omega_2 (s) - 2\omega_2 (s + 1) - \omega_2 (s + 1)$$

$$c_x = \omega_2 (s - 1) + 4\omega_2 (s) + \omega_2 (s + 1) + \omega_2 (s + 1)$$

Both $A$ and $K$ possess the properties of being symmetric and pentadiagonal. Furthermore, $A$ can be shown to be positive definite. On the RHS’s of (A.7) the following identities are used:

$$B_x(s, t - 1, t - 2) = -\frac{1}{2} \left( E_{extx} (v(s, t - 1)) + E_{fieldx} (v(s, t - 1)) \right) + [2\mu] v(s, t - 1) + \left[ \frac{\gamma}{2} - \mu \right] v(s, t - 2)$$

$$B_y(s, t - 1, t - 2) = -\frac{1}{2} \left( E_{exty} (v(s, t - 1)) + E_{fieldy} (v(s, t - 1)) \right) + [2\mu] v(s, t - 1) + \left[ \frac{\gamma}{2} - \mu \right] v(s, t - 2)$$

with $s = 0, \ldots, M_z - 1$. Note that the RHS’s depend on two prior snake positions at $t - 1$ and $t - 2$. Then, the solution of the system of (A.7) can be obtained by retrieving the new snake position at time $t$:

$$X = A^{-1} B_x$$

$$Y = A^{-1} B_y.$$
diagonal matrix, \( L \) is a lower triangular matrix, and \( U = L^T \) is an upper triangular matrix.

Such a factorization scheme proves to be most efficient for iterative processes where the matrix \( A \) does not vary at each iteration [12]. This will be the case for the snake model in most situations, where the mass density \( \mu \) and the damping density \( \gamma \) are considered to be constant and where the elasticity constraints or links \( \omega_1(s) \) and \( \omega_2(s) \) will be set to be constant most of the time. In the case where the matrix \( A \) changes with each iteration, for example, by varying the elasticity properties of the snake for every \( \Delta t \), the Choleski factorization\(^{28}\) will be more efficient [12].

To insert discontinuities in the snake, some entries of matrix \( K \) need to be modified, thereby requiring a new factorization step. A tangent discontinuity will impose the modification of six different entries in \( K \), whereas nine entries need to be modified in the case of a position discontinuity (for details, see [49]).

**APPENDIX B**

**THE DISTANCE SURFACE**

In this Appendix, we first briefly give definitions and properties related to the notion of a distance surface. (See [37] for proofs and more extensive details.) We then characterize the slope magnitude at ridge points of the distance surface (Section A). Finally, in Section B, we show how the snake model can take advantage of the distance surface as seen as a potential surface.

We call a **distance surface** a surface \( H = \phi(x, y) \) such that \( \phi \) is a solution of the following differential equation\(^{29}\):

\[
(\nabla \phi)^2 = \left( \frac{\partial \phi}{\partial x} \right)^2 + \left( \frac{\partial \phi}{\partial y} \right)^2 = 1. \tag{B.1}
\]

Let \( C \) be the base set (or object contour) on which \( \phi(x, y) = 0 \). An associated set at \( \phi(x, y) = q \), where \( q \) is a constant, consists of planar curves \( C_q \) that are parallel to \( C \) at a signed distance \( q \) from \( C \). The orthogonal trajectories (i.e., the directions of maximal gradient of \( \phi \)) of these level curves are straight lines. These straight lines emerge from the base set \( C \). Therefore, \( H \) is a **ruled surface** generated by such a set of lines.\(^{30}\) These lines make an angle \( \pi/4 \) with the \( x-y \) plane (i.e., the slope of a tangent to these lines is always of magnitude one). Furthermore, \( H \) is also a developable surface (i.e., it can be rolled out flat onto a plane), which implies that any of its regular points has a null Gaussian curvature [25]. This is easily understood if we note that at any regular point of \( H \), one of the principal directions is always along an orthogonal trajectory of a level curve \( C_q \) (i.e., a straight line). Therefore, the corresponding **principal curvature** is null.

From (B.1), we directly have that the gradient magnitude of \( \phi \) is constant and equal to one at all regular points (i.e., \( \nabla \phi = 1 \)). Furthermore, the directions in which this gradient magnitude of \( \phi \) is maximal and equal to one are, by definition, the directions of the orthogonal trajectories of the level curves \( C_q \). Therefore, \( \phi \) varies with the Euclidean distance along these orthogonal trajectories (i.e., straight lines of unit speed). In other words, the maximal slope amplitude of \( H \) is always along these orthogonal trajectories (direction of maximal gradient). At singular points (i.e., nonregular points where \( \frac{\partial \phi}{\partial x} \) and \( \frac{\partial \phi}{\partial y} \) are undefined), \( \phi \) is the radius of curvature of \( C \) (i.e., the center of a maximal inscribed circle). At these points, the orthogonal trajectories emerging from \( C \) intersect. We call these points of intersection **ridge points** of the distance surface \( H \). In the following subsection, we demonstrate that the slope along successive ridge points is always smaller in magnitude than the maximal slope at regular points of \( H \); this is a consequence of the fact that ridge points do not belong to the minimal path, that is, a path of minimum distance, of any other point [47].

### A. Ridge Points of the Distance Surface

Let us define a ridge segment to be a finite length 3-D curve (open, nonplanar, and simply connected) whose trace consists of a connected set of ridge points and a **symmetric axis segment** to be the projection onto the \( X-Y \) plane of this ridge segment. The trace of the ridge segment is constrained to be on the distance surface \( H = \phi(x, y) \). Since the orthogonal trajectories of \( \phi \) are of unit speed, the slope magnitude along a tangent to a ridge segment is necessarily less than one.\(^{31}\) This follows from the distance constraint imposed on ridge points (i.e., being part of \( H \)). For example, consider two ridge points \( R_1 \) and \( R_2 \) separated by a distance \( \Delta L \) along the symmetric axis segment (i.e., when projected onto the \( X-Y \) plane; Fig. 15). By definition, \( R_1 \) and \( R_2 \) are at a minimal distance \( d_1 \) and \( d_2 \), respectively, from the base set \( C \). Let us assume that \( R_2 \) is farther away from \( C \) than \( R_1 \); that is

\[
d_2 > d_1 \quad \text{or} \quad d_2 = d_1 + \Delta d. \tag{B.2}
\]

At \( R_2 \), because of the minimal distance constraint, we have the following:

\[
d_2 < d_1 + \Delta L. \tag{B.3}
\]

Therefore, combining (B.2) and (B.3), we get

\[
\Delta d < \Delta L. \tag{B.4}
\]

However, the average slope magnitude \( M_A \) along the ridge from \( R_1 \) to \( R_2 \) is simply

\[
M_A = \frac{\phi(R_2) - \phi(R_1)}{\Delta L} = \frac{\Delta d}{\Delta L}. \tag{B.5}
\]

Therefore, (B.4) and (B.5) imply that \( M_A \) is necessarily less than one. The same result holds in the limit as \( R_2 \) is chosen

---

\(^{28}\) The complexity of the Choleski factorization is also linear in time \((O(M^2))\). See, for example, Jacobs [42] for a complete discussion of this technique.

\(^{29}\) This equation is known as the **eikonal equation** in geometrical optics [36].

\(^{30}\) The projection of these lines onto the \( x-y \) plane correspond to the "pannormals" of Blum (Section V-A).

\(^{31}\) Note that our analysis is restricted to those ridge segments for which \( V_A \) varies smoothly (see Section VI-A).
B. The Snake Model and the Distance Surface

LEYMARIE AND LEVINE: SIMULATING THE GRASSFIRE TRANSFORM 73

between to converge at symmetry points (or ridge points) from the base set \( C \).

Two lines of minimal distance from the base set \( C \) are shown. Two lines of minimal distance from \( C \) are shown. Two lines of minimal distance from \( C \) are shown. Two lines of minimal distance from \( C \) are shown.

\[
\phi_x = \frac{L_x}{\sqrt{L_x^2 + L_y^2}} \quad \phi_y = \frac{L_y}{\sqrt{L_x^2 + L_y^2}}.
\]

Obviously, these equations are not valid at ridge points (where \( \phi_x \) and \( \phi_y \) are undefined). Nevertheless, we can still evaluate \( \phi_x \) and \( \phi_y \) and observe variations from expected behavior if these points were regular points. For example, we can observe how the computed orientation of the slope \( \theta = \) angle given by \((L_x, L_y)\) varies over the iterations (when simulating the grassfire propagation). If we are in a region consisting of regular points only, the orientation will change smoothly (ideally along an orthogonal trajectory or pannormal). However, in the case of a neighborhood centered at a ridge point, the orientation \( \theta \) will change abruptly (corresponding to the intersection of pannormals). When such a case is detected, we can backtrack (i.e., reset a snaxel to the previous position) and evaluate the directional slopes on the basis of finite differences.

The spatial sampling of snaxels (i.e., the number of snaxels per arc length unit) along a snake segment can be efficiently updated by observing how the convergence of pannormals influences the snaxels' interdistance \( \Delta \). Snaxels should follow the directions imposed by pannormals when evaluating \( \phi_x \) and \( \phi_y \). Therefore, for each snake segment, we propose the following two heuristics to reset snaxel sampling. If the average \( \Delta \) (averaged over all snaxels of a snake segment) becomes lower than a desired limit, then the number of snaxels is reduced. This may be the case of a snake segment initialized on a convex contour segment (e.g., circular arc). Otherwise, if the average \( \Delta \) becomes larger than a desired limit, the number of snaxels is increased. This may be the case of a snake segment initialized on a concave contour segment.

Furthermore, this resampling procedure permits us to keep the stiffness parameter \( \omega_1(\theta) \) constant, thereby simplifying the snake computations [49].

REFERENCES


Frédéric Leymarie received the B.Eng. degree (honors in aeronautics) from the the Electrical Engineering Department of McGill University, Montréal, Canada, in 1986. He received the M.Eng. degree from the Electrical Engineering Department of McGill University, Montréal, Canada, in 1986. He is currently a Professor in the Department of Electrical Engineering, McGill University, and a CIAR/PRCARN Associate. He is also Director of the McGill Research Center for Intelligent Machines (McGRCM). During 1972–1973, he was a member of the Technical Staff at the Image Processing Laboratory of the Jet Propulsion Laboratory, Pasadena, CA. During the 1979–1980 academic year, he was a Visiting Professor in the Department of Computer Science, Hebrew University, Jerusalem, Israel. His research interests encompass computer vision, biomedical image processing, and robotics, and he has consulted for various government agencies and industrial organizations in these areas. He has authored Vision in Man and Machine and has coauthored Computer Assisted Analyses of Cell Locomotion and Chemotaxis.

Martin D. Levine (S‘59–M‘66–SM‘74–F‘88) received the B.Eng. and M.Eng. degrees in electrical engineering from McGill University, Montréal, Canada, in 1960 and 1963, respectively, and the Ph.D. degree in electrical engineering from the Imperial College of Science and Technology, University of London, London, England, in 1965. He is currently a Professor in the Department of Electrical Engineering, McGill University, and is the Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Dr. Levine is on the Editorial Board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, an Associate Editor of Computer Vision, Graphics and Image Processing: Image Understanding, and is the Editor of the Plenum Series on Advances in Computer Vision and Machine Intelligence. He was the General Chairman of the Seventh International Conference on Pattern Recognition held in Montreal during the summer of 1984 and served as President of the International Association of Pattern Recognition during 1988–1990.