Pattern Recognition, Vol. 26, No. 9, pp. 1363 1372, 1993
Printed in Great Britain

0031-3203/93 $6.00 + .00
Pergamon Press Ltd
Pattern Recognition Society

IMPLEMENTING CONTINUOUS-SCALE MORPHOLOGY VIA CURVE EVOLUTION

GUILLERMO SAPIRO,† RON KIMMEL,† DORON SHAKED,† BENJAMIN B. KIMIA‡ and ALFRED M. BRUCKSTEIN§

† Department of Electrical Engineering, Technion–I.I.T., Haifa 32000, Israel
‡ Laboratory for Engineering Man–Machine Systems, Brown University, RI 02912, U.S.A.
§ Department of Computer Science, Technion–I.I.T., Haifa 32000, Israel

(*Received* 19 *June* 1992; *in revised form* 19 *January* 1993; *received for publication* 8 *February* 1993)

Abstract—A new approach to digital implementation of continuous-scale mathematical morphology is presented. The approach is based on discretization of evolution equations associated with continuous multiscale morphological operations. Those equations, and their corresponding numerical implementation, can be derived either directly from mathematical morphology definitions or from curve evolution theory. The advantages of the proposed approach over the classical discrete morphology are demonstrated.

Mathematical morphology Scale-space Curve evolution Partial differential equations
Digital implementation Numerical algorithms

1. INTRODUCTION

A new definition of discrete mathematical morphology is presented. First, continuous mathematical morphology is given as a dynamic process, where the basic morphological operations are obtained as solutions of partial differential equations. Then, discrete mathematical morphology is defined via an efficient numerical implementation of this continuous process. The result is that this new discrete morphology approximates continuous morphology much better than the classical discrete one.

Traditionally, mathematical morphology is introduced in a set-theoretical setting.[1–3] Morphological operators are defined as operators on sets in $\mathbf{R}^N$($\mathbf{R}^2$ in case of shapes or binary images). The dilation $\delta_B: \mathbf{R}^N \to \mathbf{R}^N$ and the erosion $\varepsilon_B: \mathbf{R}^N \to \mathbf{R}^N$ of a set $X \subset \mathbf{R}^N$ by a structuring element $B \subset \mathbf{R}^N$ are defined as the sets

$$\delta_B(X) \triangleq \bigcup_{b\in B} \bigcup_{x\in X} x + b = \{x + b : x \in X, b \in B\} \tag{1}$$

$$\varepsilon_B(X) \triangleq \bigcap_{b\in B} \bigcup_{x\in X} x - b. \tag{2}$$

It is well known that erosion can be derived from dilation since[1]

$$\varepsilon_B(X) = (\delta_{\hat{B}}(X^c))^c$$

where X^c is the complement of X, and $\hat{B}$ is the "transpose" of B, $\hat{B} \triangleq \{b : -b \in B\}$. Then, dilation is obtained via vector addition of all elements of the set X and the structuring element B, and erosion is the dual operation ("dilation of the background").

The second pair of dual morphological operations is obtained via the concatenation of erosion and dilation. Opening is defined by

$$\mathcal{O} \triangleq \delta_B(\varepsilon_B(X))$$

and closing by

$$\mathcal{C} \triangleq \varepsilon_B(\delta_B(X)).$$

Figure 1 shows an example of these four operations on the plane ($\mathbf{R}^2$). Note that opening smoothes the figure, and closing smoothes the background.

From the definitions above we see that all the basic operations of mathematical morphology are derived from the dilation operator. In the sequel, we shall therefore refer to dilation only.

Function, or multi-level, morphology is usually derived from set morphology via a homeomorphism between the space of functions $f: \mathbf{R}^N \to \bar{\mathbf{R}}$ where $\bar{\mathbf{R}} = \mathbf{R} \cup \{\infty, -\infty\}$, and the subspace of umbra sets in $\mathbf{R}^{N+1}$. An umbra set S is a set for which

$$(x_1, x_2, \ldots, x_N, x_{N+1}) \in S \Rightarrow (x_1, x_2, \ldots, x_N, y) \in S, \ \forall y \leq x_{N+1}.$$

The dilation of functions can also be formulated in function terminology[1]

$$\delta_g(f)(x) = \sup_{y\in R^N} \{f(x - y) + g(y)\} \tag{3}$$

where $g: \mathbf{R}^N \to \bar{\mathbf{R}}$ is a function or multi-level structuring element. Usually the support of the structuring element of the morphological operation is finite. In set morphology, finite support simply means that the structuring element B has finite extent. In function morphology, finite support of the structuring function g means, that the support set $G = \{y : g(y) > -\infty\}$ is finite. In those

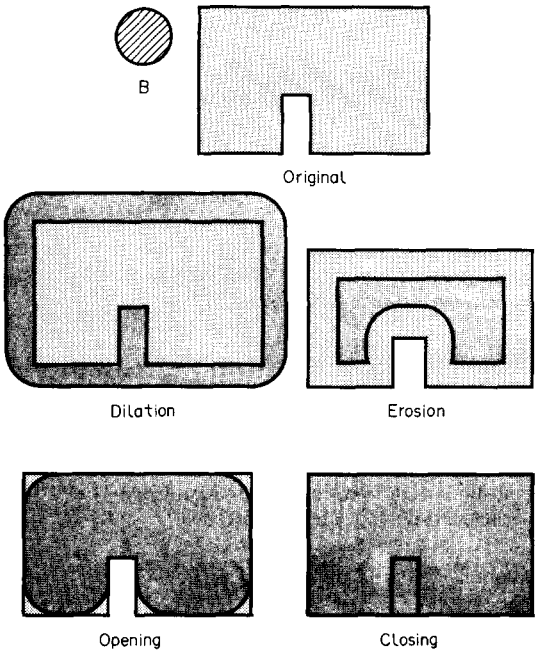

Fig. 1. Example of the four basic operations of mathematical morphology. The set X is a planar set, and the structuring element B is a disk.

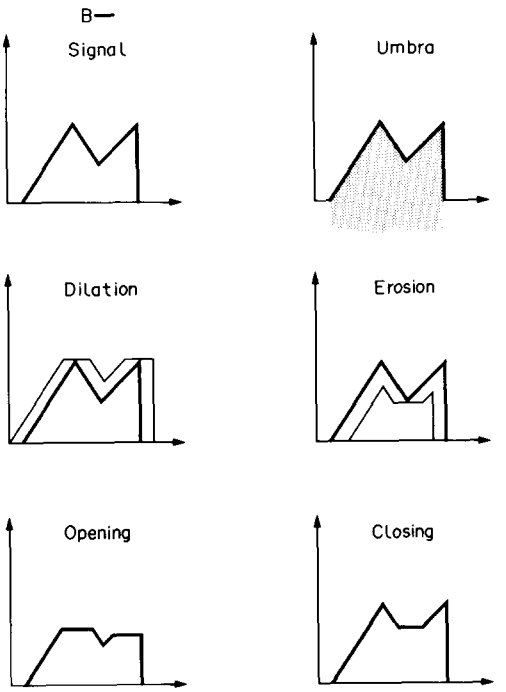

Fig. 2. Example of mixed morphology operations.

cases the dilation can be reformulated as

$$\delta_g(f)(x) = \sup_{y \in G} \{f(x-y) + g(y)\}. \tag{4}$$

Sometimes it is useful to deal with a subset of the finite support structuring functions, namely, those functions assuming a constant (usually zero) value over their support. The dilation with such a structuring function is described by the local supremum operator

$$\delta_g(f)(x) = \sup_{y \in G} \{f(x-y)\}. \tag{5}$$

The resulting operators can be considered as a special family of mixed morphology: the operators act on functions, though their structuring elements are support sets. Henceforth they will be denoted by

$$\tilde{\delta}_G(f)(x) = \sup_{y \in G} f(x-y). \tag{6}$$

Figure 2 shows examples of mixed morphology operations.

In reference (4) (see also reference (3)), Maragos describes a way to obtain the mixed morphology directly from the set morphology (sometimes called binary morphology[1]), without the need to go through function morphology. First, the function f is transformed into a family of threshold sets

$$T_a(f) \triangleq \{x: f(x) \geq a\}, \quad \forall a \in \bar{\mathbf{R}}. \tag{7}$$

Maragos proved that

$$\tilde{\delta}_G(f)(x) = \sup \{a \in \bar{\mathbf{R}}: x \in \delta_G(T_a(f))\}. \tag{8}$$

Hence, mixed morphology can be obtained via set morphology. It is also possible to obtain set morphology from mixed morphology (see Section 3).

Morphological operators can be approached as scale-space operators.[5] It can be shown that if the structuring set (or umbra of the structuring function) is convex, then[3]

$$\delta_{(t_1+t_2)B}(X) = \delta_{t_1 B}(\delta_{t_2 B}(X))$$

$$\delta_{(t_1+t_2)g}(f) = \delta_{t_1 g}(\delta_{t_2 g}(f))$$

$$\tilde{\delta}_{(t_1+t_2)G}(f) = \tilde{\delta}_{t_1 G}(\tilde{\delta}_{t_2 G}(f))$$

where for $t \in \mathbf{R}^+$, $tB = \{t \cdot x: x \in B\}$ and $tg(x) = t \cdot g(x/t)$. Scale-spaces can therefore be defined for every morphological operator $\delta_{tB}(X)$, $\delta_{tg}(f)$ or $\tilde{\delta}_{tG}(f)$, where t is the scale parameter. In addition to its theoretical importance, the scale-space approach is also important in applications such as size filtering, pattern spectrum, and smoothing.[3,6]

Figure 3 presents an example of the dilation of a continuous set with a disk of two different radii r_1 and r_2 ($r_2 > r_1$). From the property presented above, we observe that the result of dilating the set with the bigger disk, can be obtained either by dilating the original set with a disk of radius r_2, or by dilating the intermediate result (dilation with a disk of radii r_1) with a disk of radii $r_2 - r_1$.

In applications of mathematical morphology, the implementation is performed on digital computers. Indeed, the above theory is easily expressed in digital terms replacing $\mathbf{R}^N$ by $\mathbf{Z}^N$. (See Fig. 4 for examples of morphological operations on the discrete plane $\mathbf{Z}^2$.) However, in many applications where morphology is called upon because of intuitively agreeable results, discrete morphology provides results considerably

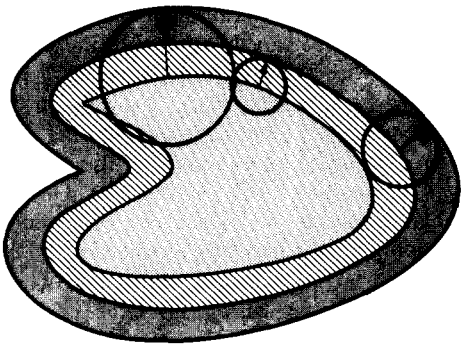

Fig. 3. Example of the dilation scale-space. The given set is dilated with two disks of different radii.

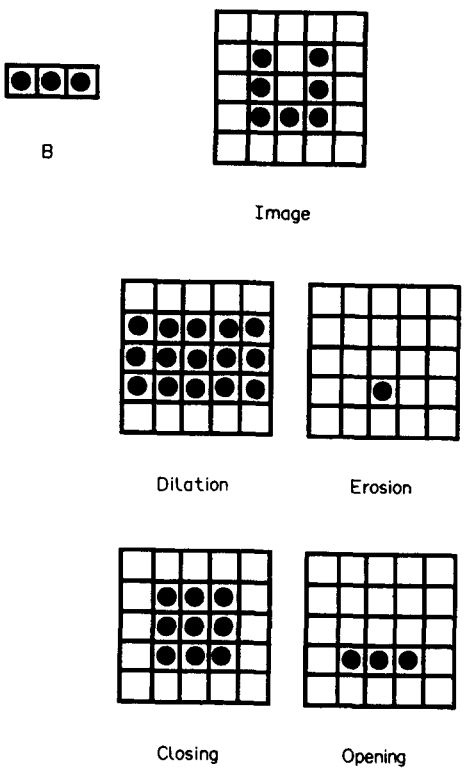

Fig. 4. Morphological operations on a discrete plane.

worse than those expected from continuous formulation. Therefore, some of the advantages of morphology as a shape analysis tool, are diminished in discrete morphology.

The problem of discrete morphology stems from the difficulty to create a convincing approximation of circles at different scales.[7,8] A square or a hexagon may be used as digital disks, in spite of the fact that those are not good approximations of continuous disks at all radii. To address such a problem is one of the goals of this work. Due to the close relation between set, function, and mixed morphology, solving this problem for one of them results in solving the problem for the others, as we shall show in this paper.

Since discrete versions of morphological techniques often fail, digital implementation of morphological operations must be based on other methods. A direct approach is based on distance transformations.[9,10] A distance transformation labels every pixel in an image with its distance to the closest boundary pixel. The dilation $\delta_{tB}(X)$, of a shape X by a circle tB of radius t, is the union of all pixels with labels less or equal to t.

In this paper we present two alternative methods for computer implementation of set and mixed morphological scale-spaces. The first results from differential geometry, and is described in Section 2. In it, instead of deforming the shape, it is the boundary that is deformed through an evolution equation. The algorithm we discuss is based on the ideas of Sethian and Osher[11,12] who developed numerical algorithms for curve and surface evolution. The other method, presented in Section 3, is derived from a recent result by Brockett and Maragos.[13] They introduced a method describing the function morphological operators as a result of a partial differential equation (PDE). For set morphology, we propose an algorithm based on the numerical implementation of this PDE, and on an "inverse definition", defining set morphology via mixed morphology. In Section 4 we discuss the two methods and show that they turn out to be identical. It is important to note that the algorithms presented here are based on discretization of continuous mathematical morphology, in contrast to the discrete morphology usually proposed to address such issues (operations on sets in $\mathbf{Z}^N$). The advantages of this method are demonstrated by examples, and are discussed in Section 4 as well.

2. MORPHOLOGICAL OPERATIONS BY CURVE EVOLUTION

A set (or shape) is well defined by its boundary (a closed planar curve in the case of planar shapes). Therefore, when performing morphological operations on sets, it is enough to investigate the influence of these operations on their boundary. In this section we will show how to perform morphological operations via curve (boundary) evolution. We first present the basic concepts of planar curve evolution.

Let $\mathscr{C}(p,t): S^1 \times [0,T) \to \mathbf{R}^2$ be a family of embedded closed curves, where t denotes time, and p parameterizes each curve. Assume that this family evolves according to the following evolution equation:†

$$\begin{cases} \dfrac{\partial \mathscr{C}}{\partial t} = \alpha \vec{T} + \beta \vec{N} \\ \mathscr{C}(p,0) = \mathscr{C}_0(p) \end{cases} \tag{9}$$

where $\vec{N}$ is the outward Euclidean normal, $\vec{T}$ the unit tangent,[15-17] and α and β are the tangent and normal components of the evolution velocity $\vec{v}$, respectively. Assuming that the normal component β of $\vec{v}$ is a geom-

† The theory can be extended to surfaces in $\mathbf{R}^n$, $n > 2$.[14]

etric function of the curve, like curvature,[17] it can be proven that the tangential component α of $\vec{v}$ does not affect the geometric behavior of the evolving curve, only its parameterization.[18] Since we are interested only in shape, we consider the evolution equation

$$\frac{\partial \mathscr{C}}{\partial t} = \beta \vec{N} \tag{10}$$

where $\beta = \vec{v} \cdot \vec{N}$, i.e. the projection of the velocity vector on the normal direction.

The evolution (10) was studied by different researchers for different functions β. If, for example, $\beta = -\kappa$ (the Euclidean curvature[17]), it can be shown that any closed, planar, simple, and smooth curve, converges to a round point.[19-21] Kimia *et al.*[22-24] recently introduced the theory of curve evolution into computer vision. They studied the evolution given by $\beta = 1 - \varepsilon\kappa$, and based on it they defined a reaction-diffusion scale-space for planar shapes. If $\beta = 1$, equation (10) simulates, under certain conditions, the grassfire flow.[22,23,25,26] The grassfire flow is the morphological scale-space created by a disk. Later in this section, we will show that morphological scale-spaces of some other structuring elements can also be simulated with different selections of β.

A number of problems must be solved when implementing curve evolution equations such as (10) in digital computers. The basic problems are:

(1) *Accuracy and stability.* The numerical algorithm must approximate the evolution equation, and it must be robust. (These are general requirements for any numerical algorithm.) Sethian[12] proved that a simple, Lagrangian, difference approximation, requires an impractically small time step in order to achieve stability. The basic problem with Lagrangian formulations is that the marker particles on the evolving curve come very close during the evolution. This can be solved by a re-distribution of the marker particles, altering the equations of motion in a non-obvious way.

(2) *Development of singularities.* If, for example, $\beta = 1$ in equation (10), even an initial smooth curve can develop singularities (see Fig. 5(a)). The question is how to continue the evolution after the singularities appear. The natural way is to choose the solution which agrees with the Hüygens principle.[26] If the front is viewed as a burning flame, this solution states that once a particle is burnt, it stays burnt.[12] The importance of this solution in shape analysis was previously pointed out and analyzed by Kimia *et al.*[22-24] Also, it can be proven that from all the weak solutions corresponding to equation (10), the one derived from the Hüygens principle, is unique, and can be obtained by a constraint denoted *entropy condition.*[27]

(3) *Topological changes.* As we see in Fig. 5(b), topological changes may occur in the curve when evolving according to equation (10). This raises the problem of handling splitting and merging curves.

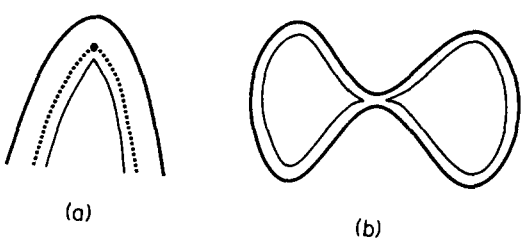

Fig. 5. Problems in curve evolution implementation. (a) Development of singularities—even a smooth initial curve can develop singularities when evolving with velocity $\vec{N}$. The first singularity occurs at the center of curvature corresponding to the maximal curvature of the initial curve (marked point). (b) Topological changes—an initial connected curve, evolving with velocity $\vec{N}$, can be disconnected when evolving in time.

Sethian and Osher[11,12,26,28] proposed an algorithm for curve (and surface) evolution that solves these problems. This algorithm consists basically of two steps. First, the curve is embedded in a higher dimensional function. Then, the equations of motion are solved using numerical techniques derived from hyperbolic conservation laws.[12,29] The basics of the algorithm are given below. For the details, see references (11, 12).

The curve $\mathscr{C}(p,t)$ is represented by the zero level set of a smooth and Lipschitz continuous function $\Phi: \mathbf{R}^2 \times [0,T] \to \mathbf{R}$. In the following we assume that Φ is negative in the interior and positive in the exterior of the zero level set. Consider the zero level set, defined by

$$\{\vec{\chi}(t) \in \mathbf{R}^2 : \Phi(\vec{\chi},t) = 0\}. \tag{11}$$

We have to find an evolution equation of Φ, such that the evolving curve $\mathscr{C}(t)$ is represented by the evolving zero level $\vec{\chi}(t)$, i.e.

$$\mathscr{C}(t) \equiv \vec{\chi}(t). \tag{12}$$

By differentiating (11) with respect to t we obtain

$$\nabla\Phi(\vec{\chi},t) \cdot \vec{\chi}_t + \Phi_t(\vec{\chi},t) = 0. \tag{13}$$

Note that for the zero level, the following relation holds:

$$\frac{\nabla\Phi}{\|\nabla\Phi\|} = \vec{N}. \tag{14}$$

In this equation, the left-hand side uses terms of the surface Φ, while the right-hand side is related to the curve $\mathscr{C}$. The combination of equations (10)–(14) gives

$$\Phi_t = -\beta\|\nabla\Phi\| \tag{15}$$

and the curve $\mathscr{C}$, evolving according to equation (10), is obtained by the zero level set of the function Φ, which evolves according to equation (15) (see Fig. 6).

The implementation of the evolution of Φ is based on a *monotone* and *conservative* numerical algorithm, derived from hyperbolic conservation laws (see Appendix and for details see reference (11)). For a large class of functions β, this numerical scheme automatically obeys the entropy condition, i.e. the condition derived from the Hüygens principle.[28,29] (In what follows, we

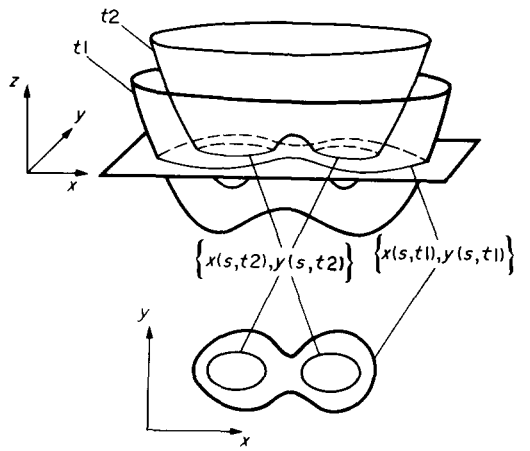

Fig. 6. Embedding the curve in a higher dimensional function solves automatically a number of topological problems. The evolving curve is obtained as a level set of the evolving surface, which remains continuous (and connected) even when topological changes as the one in Fig. 5(b) occur.

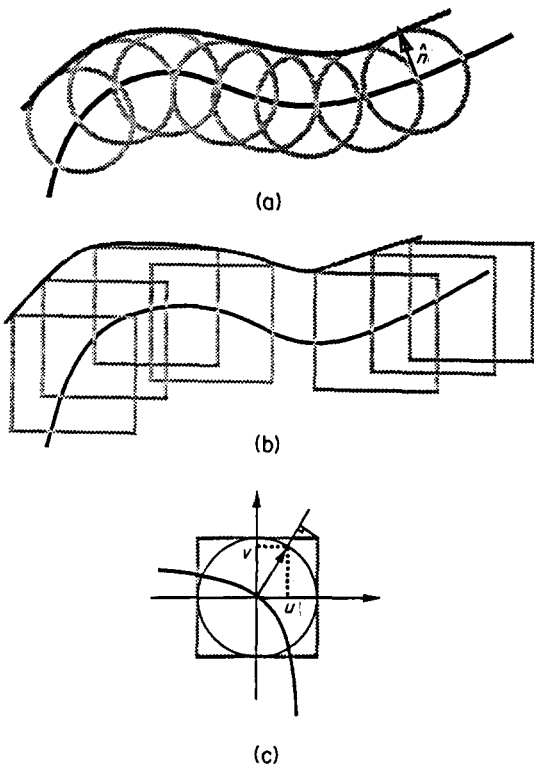

Fig. 7. The geometry of dilation by curve evolution. (a) The classical Hüygens principle. (b) Generalization of the Hüygens principle for a square—each point in the curve is the source of a "square" local wave, and the new wavefront is obtained via the envelope of the local waves. (c) Evolution velocity for a square—the velocity of the point is given by the maximal normal component of the infinite set of possible velocities.

refer only to weak solutions for which the entropy condition holds.)

As was reported elsewhere (see for example reference (22)), the weak solution $\mathscr{C}(p,t)$ of the evolution

$$\mathscr{C}_t = \vec{N}$$

simulates the grassfire flow and gives the dilated version of the curve (or shape) $\mathscr{C}_0(p)$, where the structuring element is a disk of radius t. This solution can be easily obtained by implementing the mentioned numerical algorithm. The scale-space evolution equations are presented below for set dilations with general convex structuring elements.

Let the binary, convex, structuring element B be given by a curve $\mathscr{B}$ in $\mathbf{R}^2$ ($\mathscr{B}$ is the boundary of the set B). Each point in $\mathscr{B}$ is represented by $\vec{r}(\theta)$, where $\theta \in [0, 2\pi]$, $\vec{r} \in \mathbf{R}^2$. Morphological dilation can be interpreted as a generalization of the Hüygens principle: each point in the curve is the source of a new local wave, whose shape is given by the structuring element. The new wavefront is obtained (as in the classical Hüygens principle), by the envelope of the local waves (see Figs 7(a) and (b)). The velocity of each point in the curve is given by the maximal projection of the structuring element boundary on the normal direction

$$\beta = \sup_{\theta} \{ \vec{r}(\theta) \cdot \vec{N} \}. \tag{16}$$

This can be explained as follows: each point p in the curve $\mathscr{C}(p,t)$ moves, due to the Hüygens principle, toward the boundary $\mathscr{B}$ of the structuring element B centered at the point p. This motion generates an infinite set of possible velocity vectors, one for each point in $\mathscr{B}$. Those velocity vectors can be decomposed into normal and tangential components. Since the tangential component does not affect the behavior of the evolving curve, the effective velocity is obtained from its normal component. The maximal normal component then gives the local propagation velocity (Fig. 7(c)).

From equations (10) and (16), we obtain the following evolution for a general convex structuring element:

$$\frac{\partial \mathscr{C}}{\partial t} = \sup_{\theta} \{ \vec{r} \cdot \vec{N} \} \vec{N}. \tag{17}$$

The weak solution $\mathscr{C}(p,t)$ of equation (17), holding the entropy condition, gives the dilation of the initial curve $\mathscr{C}_0(p)$ (or of the shape it defines), by the structuring element tB. Observe that from equations (15), (16), and relation (14), we obtain the surface evolution equation:

$$\Phi_t = -\sup_{\theta} \{ \vec{r} \cdot \nabla \Phi \}. \tag{18}$$

Let us now consider examples for different structuring elements. Define $u \triangleq \Phi_x$ and $v \triangleq \Phi_y$. Simple geometric computations yield the following results:

$$\begin{aligned}
&\sup_{\theta} \{ \vec{r} \cdot \nabla \Phi \} = \|\nabla \Phi\| && \text{if } B = \text{disk} \\
&\sup_{\theta} \{ \vec{r} \cdot \nabla \Phi \} = \max \{ |u|, |v| \} && \text{if } B = \text{diamond} \\
&\sup_{\theta} \{ \vec{r} \cdot \nabla \Phi \} = |u| + |v| && \text{if } B = \text{square.}
\end{aligned} \tag{19}$$

As mentioned above, the equation for a disk was previously reported elsewhere.[12,22,26]

If the structuring element B is symmetric with respect to both coordinate axes, the weak solution of equation (17) is obtained by solving equation (18) with the mentioned numerical scheme (see Appendix). For non-symmetric elements, different related numerical schemes must be implemented.[28]

We conclude in proposing the following algorithm for set morphological dilation of $X \in \mathbf{R}^2$ with the convex symmetric element B:

1. Define a smooth and Lipschitz continuous initial function $\Phi_0 : \mathbf{R}^2 \to \mathbf{R}$, such that its zero level set gives the boundary $\mathscr{C}_0$ of the initial set X.

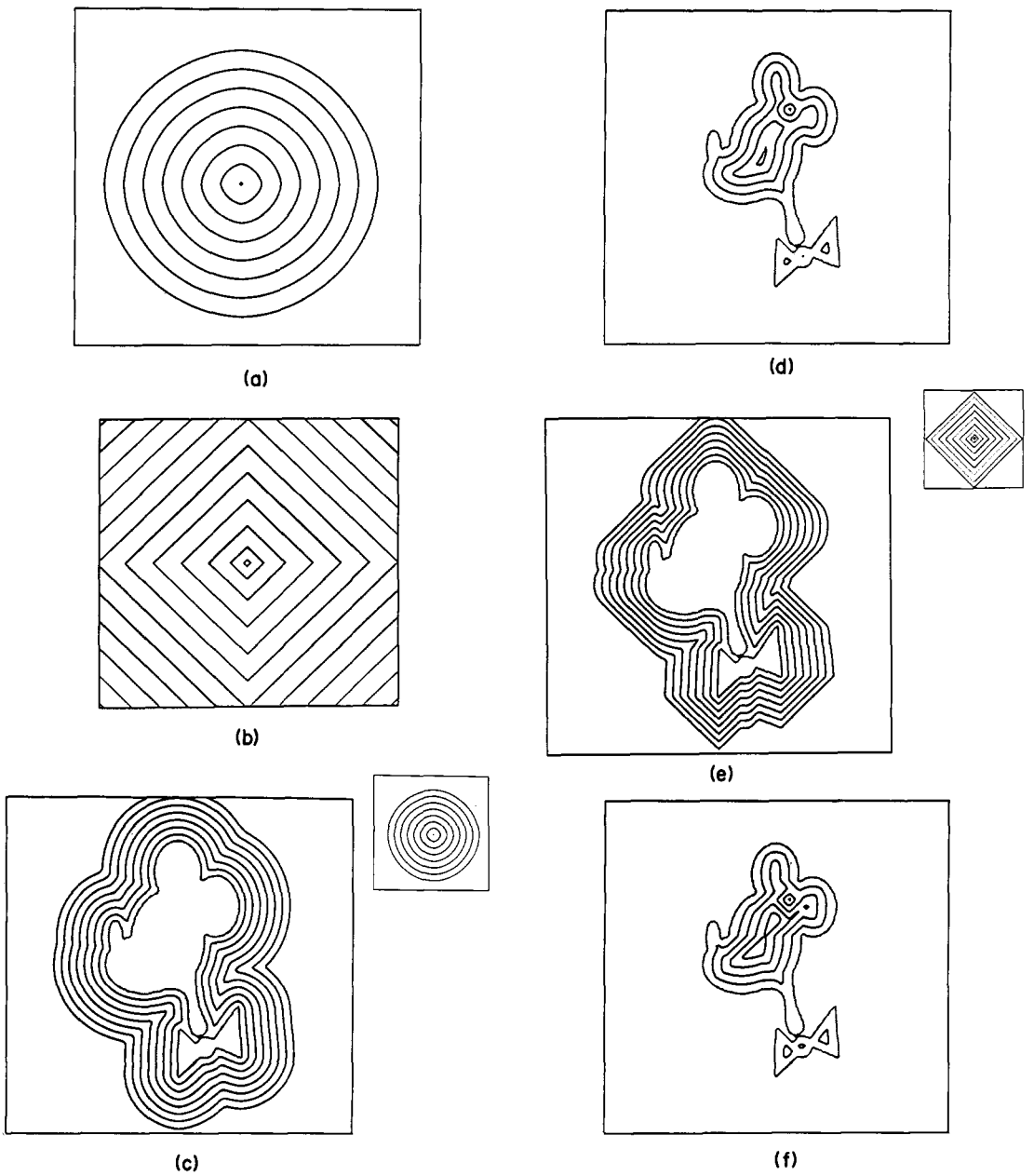

Fig. 8. Examples of morphological operations via the algorithm described in Section 2. The gray curve represents the original shape. The figure frame contains 128 × 128 pixels. The curves, which are zero level sets of the function Φ, are represented using a simple contour finding algorithm described in reference (28). (a) Dilation scale-space of a point with a circular structuring element (disk). The result is a very good approximation of a circle at all scales, in contrast with what is obtained via discrete morphology, using a discrete disk. (b) Dilation scale-space of a point with a diamond. Note that the algorithm automatically solves boundary problems. (c) Dilation scale-space of a curve (Mickey) with a disk. The initial curve is not connected, and it becomes connected after certain radius. The proposed algorithm solves this problem automatically. (d) Erosion scale-space of Mickey with a disk. (e) Dilation scale-space of Mickey with a diamond. (f) Erosion scale-space of Mickey with a diamond. (g) Erosion-dilation (opening) of Mickey with a disk. (h) Opening of a polygon with a disk.

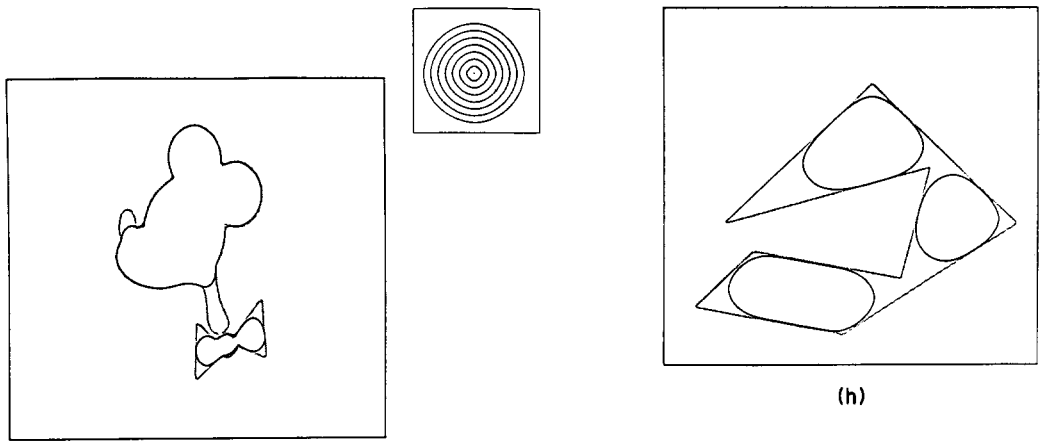

Fig. 8. (*Continued.*)

2. Compute $\sup_{\theta}\{\vec{r}\cdot\nabla\Phi\}$ for the given structuring element B.

3. Propagate the surface according to equation (18) by means of the numerical scheme described in reference (11) (see equation (A3) in the Appendix).

4. At each time t, the zero level set of Φ gives $\delta_{tB}(X)$, the boundary of the dilated version of X with the structuring element tB.

Figure 8 shows a number of morphological operations with different convex structuring elements.

Note that:

1. Erosions are obtained by taking $\vec{N}$ pointing inward, i.e. changing the sign in equation (18).

2. The results presented here for planar curves, representing sets in $\mathbf{R}^2$, can be extended to surfaces in higher dimensions, in order to operate on higher dimensional sets.

3. The reaction-diffusion scale-space studied by Kimia *et al.*[22,23] is based on dilations with disks. Using the evolution equations discussed here, this scale-space can be generalized for general convex structuring elements.

3. SET MORPHOLOGY VIA MIXED MORPHOLOGY EVOLUTION EQUATIONS

We describe now another approach for the computer implementation of set morphology. The relation between this approach, and that presented in Section 2, is discussed in Section 4.

In reference (13), Brockett and Maragos introduced a new approach to the scale-space of mixed and function continuous morphology. Their motivation is based on a feature of the image scale-space created by convolving the image with a Gaussian kernel (with variance as scale parameter). This scale-space can be described as a result of solving a partial differential equation (PDE) (diffusion in this case), with the original image as initial condition. Brockett and Maragos argue that it is possible to also describe the mixed (and function) morphological scale-space as a result of a PDE. Convolution is a linear operator, therefore the corrresponding PDE is linear. Morphological operators, on the other hand, are nonlinear. Therefore, it is not surprising that the PDEs describing the morphological scale-space are nonlinear as well. We now present the equations describing mixed morphology. For details and extensions to function morphology, see reference (13).

Brockett and Maragos derive the following differential equation for the case when the structuring element B is a sphere of radius r in $\mathbf{R}^N$:

$$\frac{\partial}{\partial t}\tilde{\delta}_{tB}(f)(x)=\lim_{r\to 0}\frac{\sup\{\nabla_x\tilde{\delta}_{tB}(f)(x)\cdot v:\|v\|\le r\}}{r}.$$

They further show that in $\mathbf{R}^2$, if the structuring element B is a disk, a diamond, or a square, this equation yields the following first-order PDEs, respectively

$$\begin{aligned}
&\frac{\partial}{\partial t}\tilde{\delta}_{tB}(f)(x)=\sqrt{\left(\left|\frac{\partial\tilde{\delta}_{tB}}{\partial x}\right|^2+\left|\frac{\partial\tilde{\delta}_{tB}}{\partial y}\right|^2\right)} && B=\text{disk}\\
&\frac{\partial}{\partial t}\tilde{\delta}_{tB}(f)(x)=\max\left\{\left|\frac{\partial\tilde{\delta}_{tB}}{\partial x}\right|,\left|\frac{\partial\tilde{\delta}_{tB}}{\partial y}\right|\right\} && B=\text{diamond}\\
&\frac{\partial}{\partial t}\tilde{\delta}_{tB}(f)(x)=\left|\frac{\partial\tilde{\delta}_{tB}}{\partial x}\right|+\left|\frac{\partial\tilde{\delta}_{tB}}{\partial y}\right| && B=\text{square.}
\end{aligned}\tag{20}$$

Those PDEs are well defined whenever $\delta_{tB}(f)$ is differentiable. When this is not the case, alternative development rules should be formulated. Those rules are formulated via the morphological derivative M

$$\frac{\partial}{\partial t}\tilde{\delta}_{tB}(f)=M(\tilde{\delta}_{tB}(f)).\tag{21}$$

The derivation of M is done through morphological considerations. In the one-dimensional case the morphological terminology can be transformed to the following expression:

$$M(f)(x)=\max\{D^+(f),-D^-(f),0\}$$

where $D^+(f)$ and $D^-(f)$ are the forward and backward derivatives of f. Note that if f is differentiable then $D^+(f) = D^-(f) = (\partial/\partial x)f$ and $M = |(\partial/\partial x)f|$. A straightforward discretization of the above continuous formulation results in an approach for implementing mixed mathematical morphology on digital computers.

Set morphology can be derived from mixed morphology. Thereby it can also be obtained from the evolution equations described above. It can be shown[3,4] that the following relation holds:

$$\delta_G(T_0(f)) = T_0(\tilde{\delta}_G(f)) \tag{22}$$

where $T_a(\cdot)$ is the threshold function defined by (7). Note that in equation (22), the dilation of the right-hand side is a mixed operator whereas the operator of the left-hand side is a set dilation.

We conclude, that if we had a good method of implementing mixed morphology, there would be a good way to implement set morphology as well. This yields the following idea: first, create any function f^*, such that its zero level set determines the initial set X

$$X = T_0(f^*). \tag{23}$$

Then, perform a mixed morphological dilation on f^*, and threshold the resulting function to obtain its zero level set. This zero level set, is the result of applying the corresponding morphological dilation on X:

$$\delta_G(X) = T_0(\tilde{\delta}_G(f^*)). \tag{24}$$

The proposed way to implement the mixed morphological operator in equation (24) is obtained by subsequent discretization of equation (21).

4. COMPARISONS, DISCUSSION, AND CONCLUSIONS

In this work, a new approach for computer implementation of mathematical morphology was discussed. The approach is based on discretization of non-linear evolution equations associated with multiscale continuous mathematical morphology, i.e. continuous mathematical morphology is presented as a dynamic process, and discrete mathematical morphology is defined as the result of the numerical implementation of this process.

The evolution equations were previously presented, based on a completely different approach, by Brockett and Maragos[13] for general convex elements, and by other authors[22,23,26] for circular structuring elements. The works in references (22, 23, 26) are based on the theory of curve evolution and numerical algorithms for surface evolution.

We first showed how to derive, from the theory of curve evolution, the corresponding evolution equations for set dilation (and any other morphological operator) with any convex structuring element tB. The evolving curve $\mathscr{C}(t)$ is the boundary of the set being dilated. This "dilation via curve evolution" approach is very intuitive, and is derived from a generalization of the Hüygens principle for any convex structuring element. In order to solve topological problems in the curve evolution implementation, $\mathscr{C}(t)$ is embedded in a higher dimensional evolving function $\Phi(t)$. This embedding is performed in such a way that $\mathscr{C}(t)$ is obtained from the zero level set of $\Phi(t)$ (see equations (11) and (18)).

We also showed how to obtain set morphology operators from the mixed morphology PDE presented in reference (13). This algorithm is also based on defining an arbitrary function f^* (see equation (23)) which has the set as its zero level set, performing mixed morphological dilation on it, and looking at the zero level set (equation (24)).

Note that the evolution equations, derived in different ways, are identical. Specifically, evolution equations (18) and (19), derived from the theory of curve evolution for set dilation, are identical to equation (20), presented in reference (13) for mixed morphology. Therefore, choosing $\Phi(x, y, 0) = f^*(x, y, 0)$, the propagated surfaces will be the same, i.e. $\Phi(x, y, t) = f^*(x, y, t)$. Then, the curve evolution approach presented here for set morphology, gives automatically mixed morphology as well: $\Phi(x, y, t)$ is the result of dilating $\Phi_0(x, y)$ with the structuring element tB (see also reference (30)).

The discretization of the evolution equations gives a new method for implementing (and defining) scale-space mathematical morphology in digital computers. The numerical approach, presented by Sethian and Osher[11,12] for curve evolution, was demonstrated to be very accurate for the implementation of the derived evolution equations. Error analysis can be found in reference (11).

The algorithm complexity is $O(N^2(r_0/\Delta t))$, where N is the picture size, r_0 the radius of the structuring element, and Δt the time step (see Appendix). Of course, for the same "price", the morphological operations for all radii $r \leq r_0$ are obtained, i.e. the whole scale-space is obtained in $O(N^2(r_0/\Delta t))$ time. The algorithm can be parallelized, reducing the complexity to $O(r_0/\Delta t)$. A different way of reducing the algorithm complexity is by performing the numerical evolution only at the neighborhood of the set boundary, i.e. just on the curve or zero level set. This makes the algorithm linear instead of quadratic. Therefore, we see that the algorithm is more time consuming than classical discrete morphology. On the other hand, in contrast with discrete morphology, the results obtained "look" like continuous morphology, making it very attractive.

The proposed method attempts to solve one of the most difficult problems in digital morphology: the digital implementation of a morphological scale-space with smooth convex structuring element. From the examples presented (see Fig. 8), we see that with this approach, we obtain a digital implementation of mathematical morphology which agrees with the intuition provided by continuous morphology. Note that in contrast with discrete morphological dilation, based on small template approximation of a circle, prolonged time dilations provided by the proposed algorithms, accurately approximate a disk. Furthermore, with the proposed algorithm, sub-pixel approximations of mor-

phological operations are obtained. Classical discrete morphology cannot achieve such accuracy.

We conclude that the discretization of continuous morphology presented here, approximates continuous morphology much better than the classical discrete morphology. This result can be applied to the areas where continuous morphology achieves good results. These include sub-pixel distance computation, shape offsetting, skeleton computation, CAD,[31] and geometric smoothing (see Fig. 8). These and other applications are currently being implemented.

Acknowledgments—It is a pleasure to thank Prof. Allen Tannenbaum for many very helpful discussions on curve evolutions.

REFERENCES

1. R. M. Haralick, S. R. Sternberg and X. Zhuang, Image analysis using mathematical morphology, *IEEE Trans. Pattern Analysis Mach. Intell.* **9**, 523–550 (1987).
2. G. Matheron, *Random Sets and Integral Geometry*. Wiley, New York (1975).
3. J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, New York (1982).
4. P. Maragos, A representation theory for morphological image and signal processing, *IEEE Trans. Pattern Analysis Mach. Intell.* **6**, 586–599 (1989).
5. M. Chen and P. Yan, A multiscaling approach based on morphological filtering, *IEEE Trans. Pattern Analysis Mach. Intell.* **11**, 694–670 (1989).
6. P. Maragos, Pattern spectrum and multiscale shape representation, *IEEE Trans. Pattern Analysis Mach. Intell.* **11**, 701–716 (1989).
7. P. A. Maragos and R. W. Schafer, Morphological skeleton representation and coding of binary images, *IEEE Trans. Acoust. Speech Signal Process.* **34**, 1228–1244 (1986).
8. R. C. Vogt, Morphological operator distributions based on monotonicity and the problem posed by digital disk-shaped structuring elements, *SPIE Digital Optical Shape Repre. Pattern Recognition* **938**, 384–392 (1988).
9. G. Borgefors, Distance transformations in digital images, *Comput. Graphics Image Process.* **34**, 344–371 (1986).
10. P. E. Danielsson, Euclidean distance mapping, *Comput. Graphics Image Process.* **14**, 227–248 (1980).
11. S. J. Osher and J. A. Sethian, Fronts propagation with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Computational Phys.* **79**, 12–49 (1988).
12. J. A. Sethian, A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed, *J. Differential Geometry* **31**, 131–161 (1989).
13. R. W. Brockett and P. Maragos, Evolution equations for continuous-scale morphology, *IEEE Int. Conf. on Acoust., Speech, Signal Processing* (1992).
14. B. White, Some recent developments in differential geometry, *Math. Intelligencer* **11**, 41–47 (1989).
15. M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, New Jersey (1976).
16. H. W. Guggenheimer, *Differential Geometry*. McGraw-Hill, New York (1963).
17. M. Spivak, *A Comprehensive Introduction to Differential Geometry*. Publish or Perish, Berkeley, California (1979).
18. C. L. Epstein and M. Gage, The curve shortening flow, *Wave Motion: Theory, Modeling, and Computation*, A. Chorin and A. Majda, eds. Springer, New York (1987).
19. M. Gage, Curve shortening makes convex curves circular, *Invent. Math.* **76**, 357–364 (1984).
20. M. Gage and R. S. Hamilton, The heat equation shrinking convex plane curves, *J. Differential Geometry* **23**, 69–96 (1986).
21. M. Grayson, The heat equation shrinks embedded plane curves to round points, *J. Differential Geometry* **26**, 285–314 (1987).
22. B. B. Kimia, Toward a computational theory of shape, Ph.D. Dissertation, Department of Electrical Engineering, McGill University, Montreal, Canada, August (1990).
23. B. B. Kimia, A. Tannenbaum and S. W. Zucker, Toward a computational theory of shape: an overview, *Lecture Notes in Computer Science*, Vol. 427, pp. 402–407. Springer, New York (1990).
24. B. B. Kimia, A. Tannenbaum and S. W. Zucker, On the evolution of curves via a function of curvature, I: the classical case, *J. Math. Analysis Applic.* **163**, 438–458 (1992).
25. H. Blum, Biological shape and visual science, *J. Theor. Biol.* **38**, 205–287 (1973).
26. J. A. Sethian, Curvature and the evolution of fronts, *Commun. Math. Phys.* **101**, 487–499 (1985).
27. J. Smoller, *Shock Waves and Reaction-diffusion Equations*. Springer, New York (1983).
28. J. A. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Computational Phys.* **98** (1992).
29. G. A. Sod, *Numerical Methods in Fluid Dynamics*. Cambridge University Press, Cambridge (1985).
30. L. Alvarez, F. Guichard, P. L. Lions et J. M. Morel, Axiomatisation et nouveaux operateurs de la morphologie mathematique, *C.R. Acad. Sci. Paris* **315**, 265–268 (1992).
31. R. Kimmel and A. M. Bruckstein, Shape offsets via level-sets, *CAD* **25**, 154–162 (1993).

APPENDIX

We now describe the numerical implementation of equation (15). For simplicity, we consider the one-dimensional case, and assume that $\beta = -1$. For more details, extensions to higher dimensions, and other functions β, see references (11, 12, 29).

The one-dimensional version of equation (15) is ($\beta = -1$)

$$\Phi_t = \|\nabla\Phi\| = \sqrt{(\Phi_x^2)}. \tag{A1}$$

Define $u \triangleq \Phi_x$ and $H(u) \triangleq -\sqrt{(u^2)}$. By differentiation of the above equation with respect to x, we obtain the following hyperbolic conservation law:[11,27,29]

$$u_t + [H(u)]_x = 0. \tag{A2}$$

To devise a numerical scheme, define $u_i^n \triangleq u(i\Delta x, n\Delta t)$. A three-point conservative and monotone, differential scheme is built for u_i^n, holding (A2).[11] It can be proven that a scheme of this kind, obeys the entropy condition.[29]

From the scheme for u, the one for Φ is obtained by integration.[11,12] If $H(u) = h(u^2)$, a simple numerical flow of (A1) is given by the following equations:

$$g(u_i^n, u_{i+1}^n) \triangleq h[(\min\{u_i^n, 0\})^2 + (\max\{u_{i+1}^n, 0\})^2]$$

$$\Phi_i^{n+1} = \Phi_i^n + \Delta t \cdot g(D^-\Phi_i^n, D^+\Phi_i^n)$$

where $D^-(\Phi_i^n) = (\Phi_i^n - \Phi_{i-1}^n)/\Delta x$ and $D^+(\Phi_i^n) = (\Phi_{i+1}^n - \Phi_i^n)/\Delta x$.

This is a so-called first-order scheme. More sophisticated higher order schemes, as well as accuracy and stability analysis, are presented in reference (11). The above scheme is easily extended to more than one dimension, and to different functions $H(u, v)$ as those in equation (19). For example, for $H(u, v) = h(u^2, v^2)$ ($u = \Phi_x, v = \Phi_y$) we obtain

$$g = h((\min(D_x^-(\Phi_{ij}^n), 0))^2 + (\max(D_x^+(\Phi_{ij}^n), 0))^2;$$
$$(\min(D_y^-(\Phi_{ij}^n), 0))^2 + (\max(D_y^+(\Phi_{ij}^n), 0))^2)$$

$$\Phi_{ij}^{n+1} = \Phi_{ij}^n + \Delta t \cdot g(D_x^-(\Phi_{ij}^n), D_x^+(\Phi_{ij}^n);$$
$$D_y^-(\Phi_{ij}^n), D_y^+(\Phi_{ij}^n)). \tag{A3}$$

Note that the discretization of the evolution equations is performed on a *rectangular grid*.[12] This rectangular grid can be associated with the *pixel* grid of digital images, making this discretization method natural for image processing. As pointed out in Section 4, this method also allows sub-pixel approximation.

About the Author—GUILLERMO SAPIRO was born in Montevideo, Uruguay, on 3 April 1966. He received the B.Sc. degree (suma cum laude) in computer engineering and the M.Sc. in electrical engineering, both from the Technion–Israel Institute of Technology, Haifa, Israel, in 1989 and 1991, respectively. He is currently working towards the D.Sc. degree in the Department of Electrical Engineering at the Technion. Since 1989 he has been a teaching assistant at the Technion and an undergraduate students project supervisor at the Signal Processing Laboratory of the Electrical Engineering Department. His research interests are in the area of curve evolution theory, image processing, and computer vision. He was awarded the Gutwirth Scholarship for Special Excellence in Graduate Studies in 1991 and is a member of SIAM.

About the Author—RON KIMMEL was born in Haifa, Israel, on 23 August 1963. He received the B.Sc. degree in computer engineerng from the Technion–Israel Institute of Technology, Haifa, Israel, in 1986, and the M.Sc. degree in electrical engineering from the same institute. From 1986 to 1991 he served in the Israeli Air Forces as a Computer Engineer. He is currently working towards the D.Sc. degree in the Department of Electrical Engineering at the Technion. Since 1991 he has been a teaching assistant and an undergraduate students project supervisor at the Center for Intelligent Systems at the Technion. His research interests are in the area of computer graphics, image processing, computer vision, and numerical analysis.

About the Author—DORON SHAKED was born in Tel-Aviv in 1962. He received the B.Sc. degree in electrical and Computer Engineering from Ben-Guryon University of the Negev, Beer-Sheva, Israel, and the M.Sc. in electrical engineering from the Technion–Israel Institute of Technology, Haifa, Israel, in 1988 and 1991, respectively. He is currently working towards the D.Sc. degree in the Department of Electrical Engineering at the Technion. His research interests are in the area of image processing, and computer vision.

About the Author—BENJAMIN B. KIMIA is a member of faculty at the Division of Engineering at Brown University. He is also a member of the Laboratory for Engineering Man/Machine Systems (LEMS), an interdisciplinary group focused on intelligent machines, signal processing, and computer engineering. Dr Kimia received the B.Eng. Honors degree from McGill University, Montréal, Canada in 1982. He subsequently pursued graduate studies there towards M.Eng. (1986) and Ph.D. (1991) in the area of computer vision and image processing. Professor Kimia's current research interests are focused on mathematical, psychophysical, and computational models for visual processing, in particular shape representation and object recognition.

About the Author—ALFRED M. BRUCKSTEIN was born in Sighet, Transylvania, Romania, on 24 January 1954. He received B.Sc. and M.Sc. degrees in electrical engineering, from the Technion, Israel Institute of Technology, in 1977 and 1980, respectively, and a Ph.D. degree in electrical engineering from Stanford University, Stanford, California, in 1984. From October 1984 until June 1989 he was with the Faculty of Electrical Engineering at the Technion, Haifa. Presently he is with the Faculty of Computer Science there. His research interest is in computer vision, image processing, estimation theory, signal processing, algorithmic aspects of inverse scattering, point processes, and mathematical models in neurophysiology. Professor Bruckstein is a member of SIAM, MAA, and AMS.