Journal of Computational Physics **169,** 678–707 (2001)
doi:10.1006/jcph.2000.6580, available online at http://www.idealibrary.com on IDEAL®

Convolution–Thresholding Methods for Interface Motion

Steven J. Ruuth*,[1] and Barry Merriman†,[2]

*Department of Mathematics and Statistics, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, Canada V5A 1S6; and †Department of Mathematics, University of California at Los Angeles, Los Angeles, California 90095-1555
E-mail: sruuth@cs.sfu.ca, barry@math.ucla.edu

Received November 29, 1999; revised June 29, 2000

Convolution–thresholding is a new approach to describing interface motion that unifies and generalizes Huygens' principle, threshold growth cellular automata, and reaction–diffusion equations. Convolution methods have many desirable properties, including automatic capture of topological change, production of curvature motion without explicit computation of curvature, natural extension to the motion of triple-point junctions, and fast, accurate implementation. In this paper, we summarize the relation of convolution–thresholding schemes to previous methods, and we review the theoretical and algorithmic development of this approach. We also review recent applications to computer vision, developmental biology, excitable media, and material science.

Key Words: Huygens' principle; diffusion-generated motion; convolution; threshold dynamics; cellular automata; curvature motion; spectral method.

1. INTRODUCTION

There are many phenomena in which sharp interfaces form, persist, and propagate. Notable examples include optical or acoustic wavefronts moving through materials, the growth of crystalline materials, the deblurring of photographic images, the evolution of detonation fronts in explosive materials, and the propagation of excitation waves in heart and neural tissue.

Modeling these processes often leads to equations of motion for a surface moving with a normal speed that depends on the surface geometry. However, these models—and their numerical solution procedures—are complicated by the fact that the interfaces can merge

[1] This research was partially supported by the President's Research Fund at SFU.
[2] The work of this author was partially supported by NSF DMS94-04942, ONR NOOO14-92-J-1890.

0021-9991/01 $35.00

or break up, or form junctions and more complicated networks. It is challenging to devise models and associated numerical algorithms that are simple, yet robust enough to capture such topological changes. Our focus here is on a variety of novel ways of describing interface motion that meet this challenge, and which can be unified through the idea of convolution–thresholding.

While investigating the problem of evolving surfaces with junctions, Merriman *et al.* [22, 23] developed the diffusion-generated motion by mean curvature algorithm, which is the focus of Section 3. This simple algorithm automatically evolves surfaces with a normal speed equal to mean curvature without ever directly computing the mean curvature. Topological changes such as merger and breakage are automatically handled without any special algorithmic procedures. Accurate, efficient discretizations are possible using adaptive resolution and fast Fourier transform techniques [32]. Finally, and perhaps most remarkably, the algorithm extends directly to the motion of triple-point junctions and arbitrary networks of surfaces [21–23, 31].

Independent of the work on diffusion-generated motion, a variety of interesting related methods have arisen in cellular automata modeling. (See Section 5.) Of these models, the "threshold growth dynamics" of Gravner and Griffeath was among the first to be rigorously analyzed [12]. In threshold growth dynamics, an unoccupied site of the lattice becomes occupied if a certain proportion of its neighbors are occupied, while occupied sites are never vacated. Although simple, these automata rules generalize in a natural way to a variety of systems arising in developmental biology and excitable media.

Another well-known class of models for evolving interfaces is the Ginzburg–Landau (or more generally the reaction–diffusion or phase field) partial differential equation (PDE) models. These models typically represent the interface as a rapid transition layer in some state—or "phase"—parameter which evolves by a (strong) reaction (weak) diffusion equation. In such a system, the state is driven to the nearest equilibrium value of the reaction, except in the thin transition layer in which the diffusion dominates. In the asymptotic limit of infinitely strong reactions, these fronts shrink to ideal surfaces that move by mean curvature or other various geometric motion laws.

Underlying each of these models are the two processes of local averaging (or diffusion) and thresholding (or projecting to a discrete set of values) applied to a representing function for the surface of interest. This can be abstracted and generalized to the convolution–thresholding motion described in Section 4. In essence, a set (and thus its bounding surface) is evolved by convolving its characteristic function with an averaging kernel and then thresholding to recover an updated characteristic function. Generalizations for multiple kernel functions or more complicated thresholding schemes are clearly possible. These general *convolution–thresholding methods* provide a natural model intermediate between cellular automata and PDEs—it turns out they can simultaneously achieve the long length scale limit of automata and the short length scale limit of reaction–diffusion PDEs, both of which are difficult limits to analyze theoretically or investigate numerically.

The outline of the paper follows. In Section 2 we review the standard Huygens' principle and its generalizations. Section 3 describes the diffusion-generated motion by mean curvature algorithm and discusses its discretization. In Section 4, we show how these methods can be generalized to give convolution–thresholding motion. This section also provides an overview of the class of obtainable motion laws, discusses fast discretization methods, and reviews some interesting related methods. Section 5 describes threshold dynamic models

and related automata arising in developmental biology and excitable media applications. It is also shown that convolution–thresholding methods arise naturally as the fine grid limit of these automata and that the fast discretizations developed for convolution–thresholding motion once again apply. In Section 6, we compare convolution–thresholding motion and phase-field methods and review a recent convolution–thresholding method for evolving filaments which can be motivated as a formal splitting for the complex Ginzburg–Landau equation. A closely related method for the multiscale treatment of images is also reviewed here. Finally, Section 7 concludes with a short summary and a description of some of the interesting open problems related to convolution–thresholding methods for interface motion.

2. HUYGENS' PRINCIPLE

In this section, we review Huygens' principle constructions for both curvatureindependent and curvature-dependent motions of interfaces. Later sections will show how these intuitive, geometrical methods are precisely a special case of convolution–thresholding methods.

2.1. Huygens' Principle for Constant Normal Velocity

The classical Huygens' principle is a geometric construction for moving a curve (in two-dimensions, or in general a codimension 1 surface) with a constant normal velocity, c. The principle states that the evolved curve at a time Δt can be obtained from the initial curve by drawing discs of radius $r = c\Delta t$ which are centered on the initial curve. The forward envelope of these discs is the curve at time $t = \Delta t$.

For our purposes, it is more convenient to draw discs of radius $r = c\Delta t$, centered so they are entirely on one side of the curve and tangent to it. The locus of the disc centers forms the new curve position after a time $t = \Delta t$. See Fig. 1.

2.2. Huygens' Principle for Motion by Mean Curvature

The classical construction can be modified to produce a motion where the surface normal velocity is proportional to the local mean curvature. Instead of the usual procedure, position

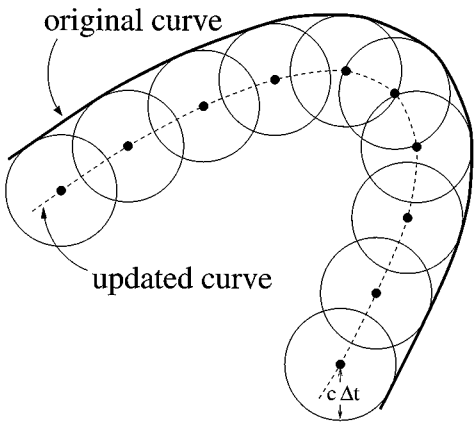

FIG. 1. Huygens' principle.

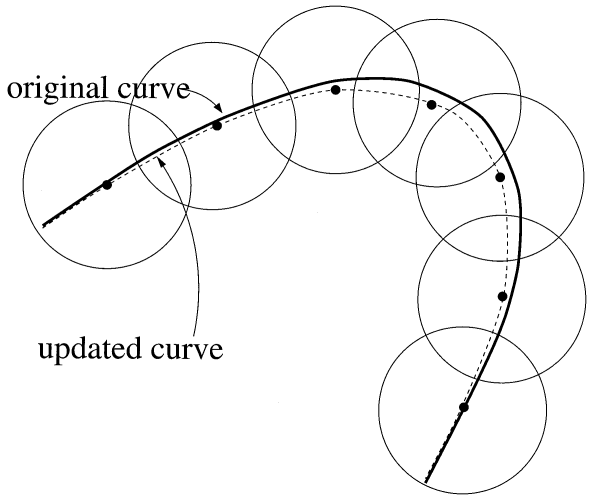

FIG. 2. Huygens' principle for a curvature-dependent motion.

each disc so that exactly half its *area* lies inside the curve to be evolved, and then take the locus of all disc centers as the new curve, as illustrated in Fig. 2. Although this is just a slight modification of the standard Huygens' principle shown in Fig. 1, it yields a qualitatively different type of motion.

Clearly the most curved portions of the interface are displaced the most by this modified process, so that it induces some form of curvature-dependent motion. A simple geometric analysis (see Fig. 3) shows that if the local radius of curvature of the curve is R, and we position a disc of radius $r \ll R$ so that it is cut exactly in half (by area) by the curve, then the disc center is displaced normal to the curve by a distance $d \sim r^2/R$. We would like this displacement to represent one time step of motion by mean curvature, so we want $d = v_n \Delta t$, with $v_n = \kappa = \frac{1}{R}$. This will indeed be the case as long as $r \sim \sqrt{\Delta t}$. Note that this also explains why this geometric procedure (Huygens' principle for mean curvature) uses discs of radius $r \sim \sqrt{\Delta t}$, while that for constant motion uses discs of radius $r \sim \Delta t$. (This distinction has practical importance for the convolution-based numerical implementations discussed later, since more spatial resolution is required to treat the smaller discs of the constant motion.)

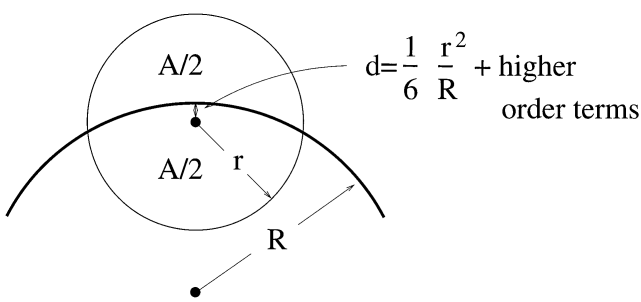

FIG. 3. The geometry of Huygens' principle for motion by mean curvature.

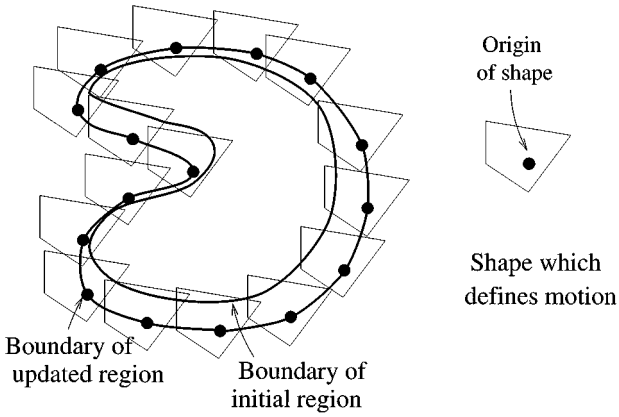

FIG. 4. Huygens' principle with a general shape and a nonzero fraction λ.

2.3. Generalized Huygens' Principles

Variations on Huygens' principle can be obtained by using shapes other than discs, taking the locus of designated points other than the disc centers, and positioning the shapes fractionally outside the curve, rather than entirely outside or half in and half out. Combining these observations, we can obtain a generalization of the geometric Huygens' principle (see, e.g., Fig. 4):

> Select an arbitrary shape (generalizing the disc of the standard principle) and an "origin point" for the shape (generalizing the disc center), which can be any point inside or outside the shape. Allow the shape and its associated origin to be moved in the plane only by rigid translation (not rotations). Given an initial curve, everywhere possible position the shape so that a fraction λ of its total area is enclosed by the curve. Then the updated curve is the locus of all the corresponding origin points.

Clearly, by using nonsymmetric shapes anisotropic motions can be derived and by varying the fraction λ the relative importance of the curvature component can be changed. This approach was fully developed, including explicit formulas for the limiting surface evolution, in the recent work of Ishii *et al.* [18]. These and other theoretical results will be summarized in Section 4 after another closely related method is discussed—diffusion-generated motion by mean curvature.

3. DIFFUSION-GENERATED MOTION BY MEAN CURVATURE

The diffusion-generated motion algorithm introduced in [22, 23] is a surprisingly simple procedure for approximating motion by mean curvature of a surface without computing curvature. In this section we give the basic algorithm and its extension to surfaces with multiple junctions. Later sections will show how this procedure is another special case of convolution–thresholding. We also present efficient discretization techniques for diffusion-generated motion which have direct extensions to the general convolution–thresholding schemes.

3.1. The Basic Method

It is intuitively clear that if we allow a set of points to "diffuse," sharp corners rapidly smooth out (e.g., Fig. 5). Based on this observation, we might expect that diffusion can be used to evolve the boundary of a set in a curvature-dependent way.

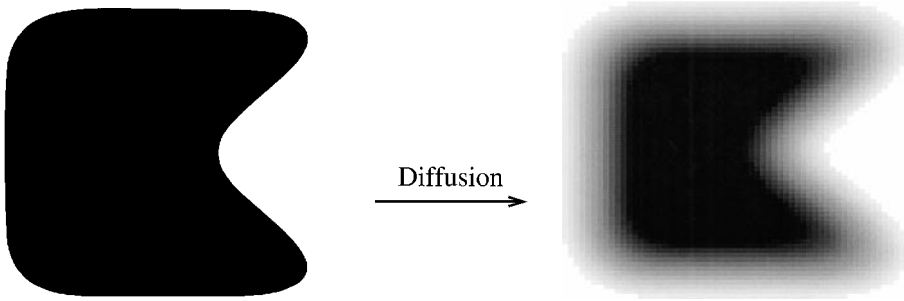

FIG. 5. Sharp corners are rapidly smoothed out by diffusion.

Consider, for example, setting χ equal to the characteristic function for some initial region. We then apply diffusion to χ,

$$\frac{\partial \chi}{\partial t} = \nabla^2 \chi$$

and consider the evolution of the $\frac{1}{2}$-level contour. Rewriting using local polar coordinates with origin at the center of curvature (see Fig. 6),

$$\frac{\partial \chi}{\partial t} = \frac{1}{r}\frac{\partial \chi}{\partial r} + \frac{\partial^2 \chi}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2 \chi}{\partial \theta^2},$$

we find that the motion is dominated by the radial advection and diffusion terms. (The remaining term involving θ-derivatives vanishes to highest order, since χ is locally a function of r only, independent of θ.)

Taking a radial view of the evolution (Fig. 7), we find that the first term simply advects the initial profile with a speed $\frac{1}{r} = \kappa$, while the second diffusion term smears out the profile. Because the smearing is symmetric, however, it does not affect the $\frac{1}{2}$-level contour. Thus we are left with an advection–diffusion equation which advances the level set $\frac{1}{2}$ with a speed equal to the local curvature, κ.

Using this simple intuition, an algorithm for moving an interface by mean curvature can be constructed [22, 23]:

ALGORITHM **DGM** (Two Regions).

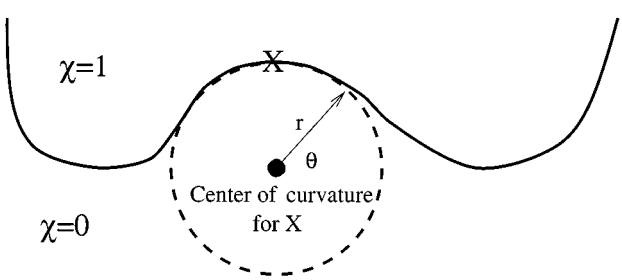

FIG. 6. Local polar coordinates with origin at the center of curvature for X.

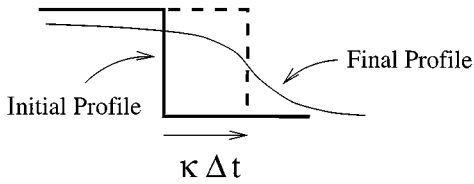

FIG. 7. Radial view of the time evolution. The radial advection term advects the initial profile with a speed equal to the local curvature (dashed), while the radial diffusion term merely contributes a smearing to the final result and does not affect the $\frac{1}{2}$-level contour.

GIVEN: An initial region R.

BEGIN

(1) "Initialize": Set $\bar{\chi}$ equal to the characteristic function for the region R.

(2) Repeat for all steps:

(a) "Diffuse": Starting from $\bar{\chi}$, evolve χ for a time Δt according to $\chi_t = \nabla^2 \chi$.

(b) "Threshold": $\bar{\chi} = \begin{cases} 1 & \text{if } \chi > 1/2 \\ 0 & \text{otherwise.} \end{cases}$

END

The location of the interface is given by the boundary of the set defined by the characteristic function $\bar{\chi}$, or by the $\frac{1}{2}$ level of the smooth χ.

Notice that this procedure can be described informally as diffusing the set for a short time; and then thresholding at the $\frac{1}{2}$ level to obtain a new set. As we have seen, such a diffusion will cause a curvature-dependent blurring of the set boundary, and a formal analysis of the diffusion equation [21–23] shows that this should result precisely in motion by mean curvature. Indeed, an interesting variety of rigorous proofs have been given to show that this simple algorithm converges to motion by mean curvature as the time step goes to zero [1, 6, 18]. Note that the rate of convergence for smooth interfaces is first order and that more rapid convergence is often possible using extrapolation. See [32].

This algorithm has several remarkable properties: Motion by mean curvature is obtained in any number of dimensions without ever directly computing the mean curvature. Topological mergers such as pinch off, which occur in higher dimensions, are captured with no special algorithmic procedures. Note also that motion by mean curvature is a nonlinear evolution, and yet the diffusive evolution is entirely linear, with the only nonlinear part of the algorithm being the final, trivial, thresholding step.

Perhaps most remarkable, this procedure has a direct extension to the motion of multiple junctions. This extension will be the focus of the next section.

3.2. Extension to Multiple Junctions

Diffusion-generated motion has a direct extension to surfaces with multiple junctions. Let the intersecting surfaces partition the domain into regions with characteristic functions $\chi_1, \chi_2, \ldots, \chi_N$. Note that $\sum \chi_i = 1$ everywhere, reflecting the partition. We independently diffuse each region—i.e., convolve χ_i with the Gaussian—to obtain smoothed-out characteristic functions $\chi_i(\Delta t)$. Note that these still sum to one, by the linearity of the convolution:

$\sum \chi_i(\Delta t) = \sum K * \chi_i = K * \sum \chi_i = K * 1 = 1$. Thus the smoothed-out characteristic functions still partition the domain into "fuzzy" sets. To obtain a partition into geometric sets, we simply define set i to be the set on which $\chi_i(\Delta t)$ is greater than all the other smoothed-out characteristic functions.

This approach leads to the following algorithm for the motion by mean curvature of multiple regions:

ALGORITHM **DGM** (Multiple (r) Regions).

GIVEN: Several regions R_j, $1 \le j \le r$, which divide the domain $\Omega = \bigcup_{1 \le j \le r} R_j$.

BEGIN

(1) "Initialize": Set each $\bar{\chi}_j$ equal to the characteristic function for the region R_j.

(2) Repeat for all steps:

(a) "Diffuse": Starting from $\bar{\chi}_j$, evolve each χ_j for a time Δt according to $\frac{\partial \chi_j}{\partial t} = \nabla^2 \chi_j$.

(b) "Threshold": $\bar{\chi}_j = \begin{cases} 1 & \text{if } \chi_j = \max_{1 \le k \le r}\{\chi_k\} \\ 0 & \text{otherwise.} \end{cases}$

END

For any time t, the interfaces are given naturally as the boundaries of the characteristic sets. Note that in the case of two regions, i.e., a set and its complement, this reduces to the original algorithm for χ_1 alone because $\chi_2 = 1 - \chi_1$ is not an independent quantity. It remains an open problem to prove that this algorithm converges to motion by mean curvature for the interesting case of three or more regions.

Because the original diffusion-generated-motion algorithm uses a symmetrical χ comparison, it produces symmetrical triple-point junctions. To obtain arbitrary desired junction angles a nonsymmetrical comparison can be used, as described in [21, 23, 31]. This approach has been further generalized by Ruuth [31] to produce a normal velocity which depends on a positive multiple of the curvature of the interface plus the difference in bulk energy densities for prescribed junction angles (see Fig. 8). Similarly to the basic algorithm, these generalizations naturally treat topological merging and breaking and produce no overlapping regions or vacuums. Also similarly to the basic algorithm, there are no rigorous results

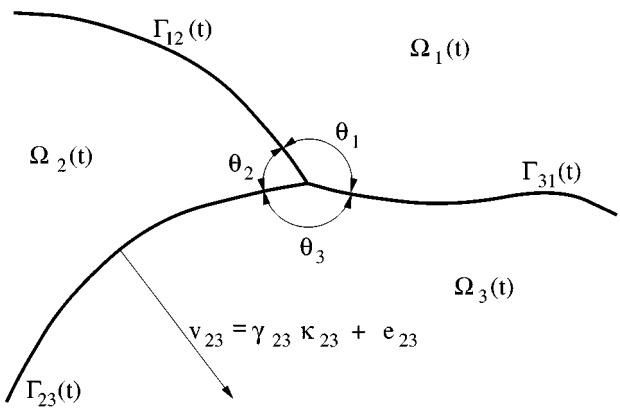

FIG. 8. The interfaces, Γ_{ij}, move with a velocity $v_{ij} = \gamma_{ij}\kappa_{ij} + e_{ij}$ and are subject to angles $\theta_1, \theta_2, \theta_3$.

concerning these methods when multiple junctions occur, but the numerical experiments in [30–32] demonstrate their convergence.

3.3. Efficient Discretizations

As outlined, diffusion-generated motion is only discrete in time. We now discuss possible spatial discretizations for the method. Most important is that the efficient spectral methods described here have direct extension to the general convolution–thresholding methods described in later sections.

3.3.1. *Finite Difference Methods*

A very simple way to spatially discretize diffusion-generated motion is to use a finite-difference method on a fixed grid [22, 23]. Unfortunately, this simple approach leads to several problems [23, 32].

In particular, the time step Δt must be large enough so that the motion of the interface over each step can be resolved by the spatial discretization. For the case of a finite-difference discretization, *the level set* $\frac{1}{2}$ *must move at least one grid point; otherwise the front will remain stationary* (see Fig. 9). This produces the restriction that

$$\begin{aligned}(\text{speed of motion of the interface}) \times \Delta t &\gg \text{grid spacing}\\ \kappa \Delta t &\gg h, \end{aligned} \tag{1}$$

which is prohibitively expensive whenever κ is small.

Furthermore, even when this restriction (1) is satisfied throughout space, an extremely fine grid may be needed to achieve the desired accuracy. Consider, for example, the evolution of a smooth surface (i.e., no junctions or self-intersections) according to diffusion-generated motion. Here, a simple Taylor-series expansion can be used to demonstrate that an $O((\Delta t)^2)$ error in the position of the front is generated at each step [30]. If the function χ is represented using a fixed grid then each thresholding produces an error which is comparable to the mesh spacing; i.e., each thresholding step produces an $O(h)$ error in the position of the front. To preserve the overall accuracy of the method we must take $h = O((\Delta t)^2)$. Using a uniform mesh, this leads to $O((\frac{1}{\Delta t})^{2d})$ grid points and $O((\frac{1}{\Delta t})^{2d})$ operations per step in d dimensions, which is often impractical even for simple two-dimensional applications.

As we shall see in the next section, these inefficiencies are easily overcome using adaptive resolution and fast Fourier transform techniques.

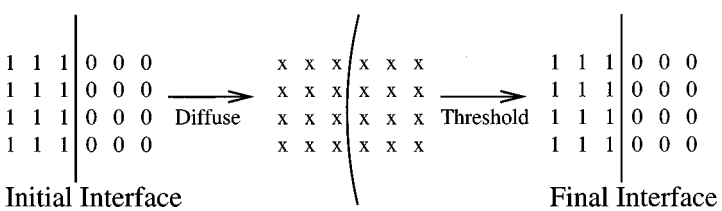

FIG. 9. If the level set $\frac{1}{2}$ moves less than one grid point, the front remains stationary.

3.3.2. *A Spectral Discretization*

The standard discretization of diffusion-generated motion can be expensive, even for simple two-dimensional applications. Fortunately, much faster results can be obtained using a simple spectral method on adaptive grids [32].

To begin, a method is needed to solve the heat equation,

$$\chi_t = \nabla^2 \chi,$$

repeatedly over intervals of length Δt. This is accomplished using a Fourier series. Notice that χ is initially discontinuous so it will contain a high-frequency error from truncating the Fourier series. However, we only require χ after a time Δt. After a time Δt, high-frequency error modes have been damped out. Since the problem is linear, the various modes do not interact—thus *there is never a need to approximate the high-frequency components of* χ. Thus a Fourier series is an excellent choice, because far fewer basis functions are required than might otherwise be expected.

The thresholding step is also straightforward. Using the usual orthogonality conditions, it is easy to show that the Fourier coefficients of the characteristic function after thresholding are

$$c_{jk} = \iint_{R_t} \exp(-2\pi i j x) \exp(-2\pi i k y)\, dA, \tag{2}$$

where

$$R_t = \left\{ \mathbf{x}\colon \chi(\mathbf{x}, t) > \frac{1}{2} \right\}$$

is the approximation to the phase we are following.

To complete the discretization, the integrals (2) must be evaluated. These are accurately and efficiently treated using the quadrature methods described in [32]. Briefly,

- If $R(t)$ is a square, the integration step is carried out exactly. More general regions are treated by dividing the domain into small squares (see, e.g., Fig. 10) and summing the contributions from each. At the finest level, the contributions to the Fourier coefficients are approximated using a quadrature over triangles.
- During mesh refinement, a large number of unequally spaced function evaluations are required (see, e.g., Fig. 10). Because the fast Fourier transform requires an equally spaced grid, fast implementations use a recent unequally spaced fast Fourier transform method [2]. This method is also used for the rapid evaluation of the Fourier sums that arise in the quadrature steps of the algorithm.

We now consider how this discretization compares with the usual finite-difference approach.

3.3.3. *Comparison*

This spectral discretization is preferred over finite-difference discretizations of diffusion-generated motion for several reasons [32]. These reasons are outlined below.

1. A lattice-based method must satisfy (1) globally, or part of the front may erroneously remain stationary. By recursively refining near the interface and interpolating at the finest cell level, the spectral discretization eliminates this restriction.

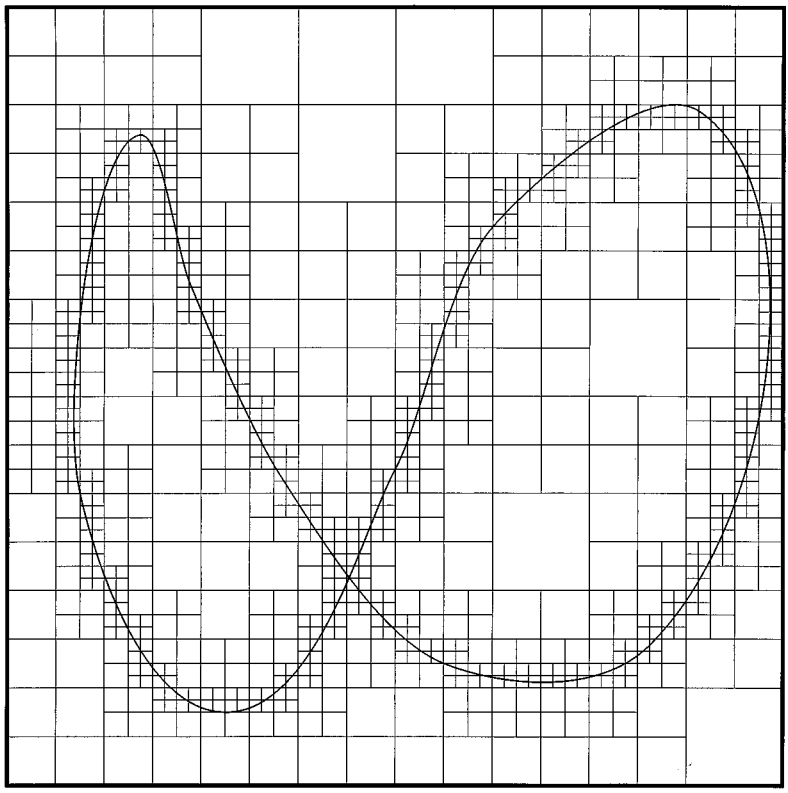

FIG. 10. Integration is carried out by dividing the domain into squares. Contributions from all but the finest regions can be evaluated exactly.

2. An unwanted anisotropic component to the motion is generated whenever a regular lattice is used since the front must travel an integer number of cells per time step. No such restriction occurs with the spectral approach since interpolation is used to locate the front at the finest cell level.

3. A lattice-based method produces an irregular error which makes the construction of higher order accurate, extrapolated results impractical. Because the spectral discretization uses interpolation to locate the front at the finest cell level, the error arising from the thresholding step is relatively small. In many instances, this makes an accelerated convergence to the limiting motion law possible using Richardson extrapolation in the time-step size. See [32] for further details.

4. Far fewer operations are required to obtain an accurate representation of the front using the spectral discretization. Here, the proposed method requires only

$$O\left(\frac{1}{\Delta t}\log^2(\Delta t)\right)$$

operations per step to preserve the overall accuracy of the method [32]. This compares very favorably to the result for smooth curves $O(1/(\Delta t)^4)$, which was derived in Section 3.3.1.

In practice, finite-difference or pseudo-spectral methods on a uniform grid are often adequate for obtaining crude but illustrative results (e.g., [21–23, 35]). However, when *accurate*

solutions are sought ($\leq$3% relative error) the spectral discretization is preferred. Indeed, a finite-difference discretization on a uniform grid often requires hours of computation to obtain the same relative error that a spectral discretization obtains in a few seconds. See [32, 34] for some sample calculations illustrating this property.

4. CONVOLUTION–THRESHOLDING MOTION

We now show how the general framework of convolution–thresholding motion of surfaces unifies the geometric Huygens' principles with the analytic method of diffusion-generated motion by mean curvature. This section also gives recent generalizations of the basic method and discusses their discretization. We also review related methods that have appeared in the literature.

4.1. Huygens' Principle as Convolution–Thresholding

The Huygens' principle described in Section 2 is a geometric technique for moving a curve or surface. As the first step toward generalization, this geometric construction can be translated into an analytic form. We represent curves as the boundaries of regions, and in turn represent regions by their characteristic functions, i.e., functions that are 1 on the region and 0 off the region. We represent the discs (or any other shape) used to advance the front by their characteristic functions as well. Suppose the original region has characteristic function χ, and let K be the characteristic function for the motion-generating shape, scaled so that it has unit mass. Let $*$ denote the convolution,

$$\chi * K(\mathbf{x}) = \int_{R^2} \chi(\mathbf{y}) K(\mathbf{x}-\mathbf{y})\, d\mathbf{y}. \tag{3}$$

Then for constant normal motion, the updated region in Huygens' construction can be defined as

$$\{\mathbf{x}\colon \chi * K(\mathbf{x}) > 0\}, \tag{4}$$

and the updated curve is the boundary of this region. Similarly, for motion by mean curvature the updated region in Huygens' construction can be defined as

$$\left\{\mathbf{x}\colon \chi * K(\mathbf{x}) > \frac{1}{2}\right\}. \tag{5}$$

For example, for the Huygens' principles using discs in 2-D, the kernel is the (normalized) characteristic function for a disc of radius r, centered at the origin,

$$K(\mathbf{x}) = \begin{cases} \frac{1}{\pi r^2} & \text{if } |\mathbf{x}| < r \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $r \sim \sqrt{\Delta t}$ for motion by mean curvature, or $r \sim \Delta t$ for constant normal motion.

Thus, the geometric Huygens' principle is equivalent to the analytic procedure of convolving the characteristic function for the original region with an appropriate kernel function and obtaining a new characteristic function from this via thresholding.

4.2. Diffusion-Generated Motion as Convolution–Thresholding

The diffusion-generated motion [22, 23] can also be viewed as a convolution–thresholding algorithm. If the initial surface bounds a region with characteristic function χ, then the solution to the linear diffusion equation at a time Δt later is $\chi * K$, where K is a Gaussian of width $\sqrt{\Delta t}$,

$$K(\mathbf{x}) = K_{\mathrm{G}}^{\Delta t}(\mathbf{x}) \equiv \frac{1}{4\pi\,\Delta t} \exp\left(-\frac{1}{4\Delta t}|\mathbf{x}|^2\right),$$

and the updated surface is the boundary of the region

$$\left\{\mathbf{x}\colon \chi * K(\mathbf{x}) > \frac{1}{2}\right\}. \tag{7}$$

Indeed, any positive, radially symmetric kernel may be used in place of the Gaussian to obtain a convolution-generated mean curvature motion, as was pointed out by Merriman *et al.* [22] and proven rigorously by Ishii [17]. Thus diffusion plays no deep special role in generating the motion by mean curvature and probably obscures the greater significance of the convolution. The main value of the diffusion PDE description of the convolution process is that it allows a convenient formal analysis, as was indicated in Section 3, and it highlights the connection with phase-field models, as described in Section 6.

4.3. Convolution–Thresholding Motion

Based on the update rule (4), it is clear that we are interested in more general forms of convolution-generated motion. In particular, it is natural to consider the following generalizations of (7):

1. Allow different convolution kernel functions, K. The method formally allows arbitrary kernel functions, and asymmetrical kernels can be used to produce anisotropic motion laws, as originally suggested in [22]. Without loss of generality, we shall assume that the kernel has been normalized to satisfy $\int K(\mathbf{x})\,d\mathbf{x} = 1$.

2. Allow a general threshold, λ, in $\{\mathbf{x}\colon \chi * K(\mathbf{x}) > \lambda\}$. This provides a continuum of convolution–thresholding methods parameterized by $\lambda \in [0, 1)$ with $\lambda = 0$ corresponding to the standard Huygens' principle for constant motion (see [33]), and $\lambda = \frac{1}{2}$ corresponding to motion by mean curvature. In general, λ can also be allowed to depend on other quantities. For example, a variety of $v_n = a + b\kappa$ diffusion-generated motions can be obtained with $\lambda = \frac{1}{2} + c\sqrt{\Delta t}$ [18, 21, 31], so $\lambda = \lambda(\Delta t)$ is a useful form. More generally, λ may be selected locally as a function of the normal direction defined by the level sets of $K * \chi$ to achieve an interesting variety of anisotropic motions [33].

These generalizations produce semidiscrete methods—i.e., continuous in space but discrete in "time". To determine the corresponding continuous dynamics, we must somehow introduce a time step and clarify what it means to take the small-time-step limit (assuming such a limit exists). Intuitively, the time step is determined by the effective size of the support of K, since the larger the effective support of K, the further its convolution will move the set boundary. Thus the small-time-step limit is obtained by scaling down K in a suitable fashion. More precisely, let us scale the fixed kernel $K(\mathbf{x})$ by the mass-preserving form $K(\mathbf{x}/r)/r^d$, so that the effective radius of its support scales like $r \ll 1$. By convolving

this scaled kernel with χ and thresholding the result at λ, the set boundary is displaced by an amount that is some function of r, $s(r)$. If we demand that in the limit of small r this displacement be one time step of some limiting motion law, $s(r) = v_n \Delta t$, this fixes the relation between the size of the kernel, r, and the time step, Δt. Note, in particular, that if K is a Gaussian kernel with effective support of size r, this general procedure yields $\Delta t \sim r^2$. This is precisely the scaling relation between kernel size and time step used in the diffusion-generated case discussed above, although there it can also be motivated by the simple fact that diffusion for a time Δt will smear (and thus move) the set boundary over a distance $r \sim \sqrt{\Delta t}$.

4.4. Obtainable Motion Laws

It is natural to ask what motion laws arise from convolution-generated motion and how the radius of the kernel scales with Δt.

In the case where K is nonnegative and $\lambda = 0$, convolution-generated motion reduces to Huygens' principle for the curvature-independent motion described in Section 2. Notice in particular that if we assume that each update corresponds to one time step of length Δt, then $\mathcal{N}$ and hence K have radii which scale like Δt.

Another interesting case occurs in two dimensions when $\lambda = \frac{1}{2}$ and K is the scaled characteristic function of a symmetric region $\mathcal{N}$ (i.e., $\mathcal{N} = -\mathcal{N}$). If we define $r(\theta)$ to be the polar representation of the boundary of $\mathcal{N}$, then it is easy to show that a leading-order approximation of the displacement of a smooth initial boundary is $r^2(\theta)\kappa/6$ [33]. Thus general "anisotropic curvature motions" of the form

$$v_n = b(\theta)\kappa \tag{8}$$

are obtained simply by taking

$$r(\theta) = \sqrt{6b(\theta)\Delta t}.$$

Similar to the case of constant motion, this algorithm also has a simple geometric version [33]:

> Using only translations, place copies of $\mathcal{N}$ so that exactly half of their area lies inside the original region. The locus of shape centers forms the boundary of the updated set.

A combination of these two types of motion can be obtained by varying the threshold, λ. This class of methods has been studied in the recent and comprehensive work of Ishii *et al.* [18] for the case where λ is a constant or $\lambda = \lambda(\Delta t)$. They give explicit formulas for the limiting surface normal velocity v_n in terms of various moments of the kernel function, in any number of dimensions. Moreover, they also give rigorous proof that the convolution-generated motions converge to their stated v_n motion laws in the limit as $\Delta t \to 0$.

One notable implication of their results is that it is impossible to obtain many interesting curvature-dependent motions in three dimensions with this class of generalizations.[3] For

[3] Note that the converse is also true. It is not possible to approximate many interesting convolution–thresholding combinations using finite-motion laws. See Section 5.4 for an example.

example, consider motion by *weighted mean curvature* (e.g., [16, 38]),

$$v_n = \left(\gamma + \frac{\partial^2 \gamma}{\partial \theta_1^2}\right)\kappa_1 + \left(\gamma + \frac{\partial^2 \gamma}{\partial \theta_2^2}\right)\kappa_2, \tag{9}$$

where γ represents the anisotropic surface energy, $\{\kappa_i\}$ are the principal curvatures of the surface, and $\{\theta_i\}$ are the associated local angles made by the normal vectors along the principle circles of curvature. Such motions can only be obtained if γ is constant—in which case the original diffusion-generated motion algorithm applies. The origin of this limitation in more than two dimensions can be understood by a straightforward geometric analysis [33]. Briefly, when a nonspherical Huygens' shape is positioned to be some fraction inside the surface, the principal curvatures of the surface have independent, and generally different, influences. Thus the motion cannot depend only on the symmetrical combination $\kappa = (\kappa_1 + \kappa_2)$, and motion laws of the form $b(\hat{n})\kappa$ with nonconstant b are not possible. The same is true for other forms that require constrained combinations of principal curvatures, such as the surface-tension-weighted mean curvature.

4.5. Extensions

To produce more general motions, λ may be allowed to depend on other quantities. For example, in [33] λ is defined locally as a function of the normal direction to obtain motions in two dimensions of the form

$$v_n = a(\theta) + b(\theta)\kappa,$$

where b is nonnegative and continuous.

Even more generally, multiple-kernel algorithms may be desired since these provide a convenient way to generate interface velocities that are unobtainable with single kernels (e.g., anisotropic mean curvature motion in more than two dimensions). In this approach, the characteristic function is convolved with multiple kernels, $\chi * K_1, \chi * K_2, \ldots, \chi * K_N$, and these are combined in some convex combination or differencing combination prior to the thresholding stage. For example, suppose that we wish to evolve a surface according to a normal velocity

$$v_n = b(\hat{n})\kappa, \tag{10}$$

where $0 < b_{\min} \le b(\hat{n}) \le b_{\max}$. Set $K_{\min}$ equal to the heat kernel for

$$\chi_t = b_{\min}\nabla^2 \chi$$

and $K_{\max}$ equal to the heat kernel for

$$\chi_t = b_{\max}\nabla^2 \chi$$

for a time-step size of Δt. Thresholding the convex combination

$$(1-c)\chi * K_{\min} + c\chi * K_{\max}$$

at the level $\frac{1}{2}$ then produces a velocity proportional to mean curvature *for any constant*

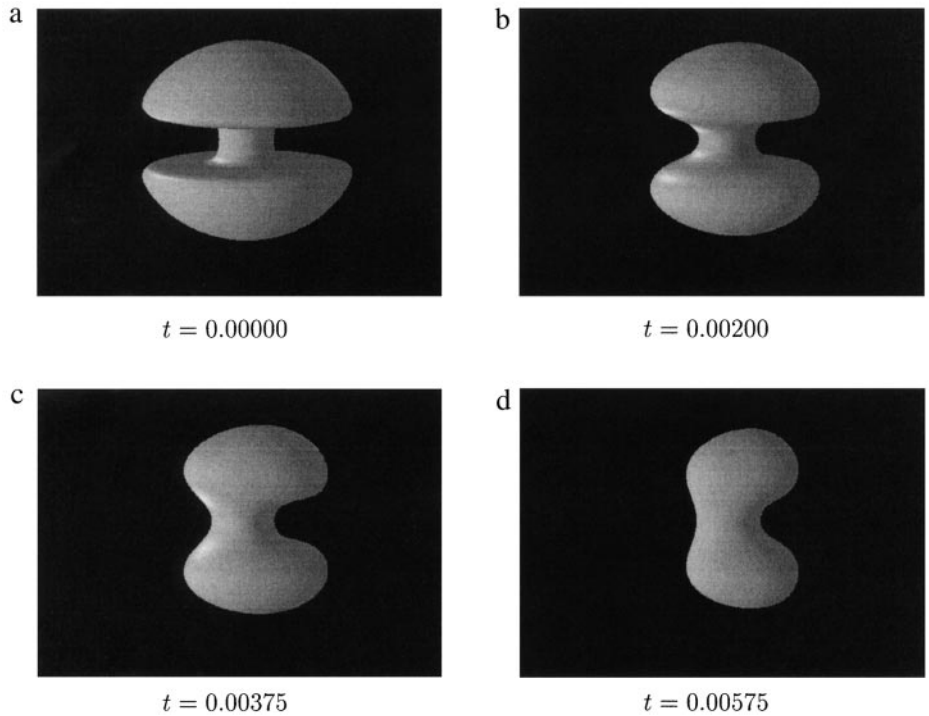

FIG. 11. A normal velocity $v_n = (1 + \sqrt{n_1^2 + n_2^2} + \sin(\pi n_1))\kappa$.

$0 \le c \le 1$. (This observation follows from Ishii [17], since the effective kernel is positive, symmetric, and decreases exponentially quickly away from the origin.) In fact, it is easily shown (cf. [21]) that a normal velocity $b\kappa$ is obtained if

$$c = \frac{b - b_{\min}}{b - b_{\min} + \sqrt{b_{\min} b_{\max}} + b\sqrt{b_{\min}/b_{\max}}}.$$

To produce motions of the desired form (10), we simply select $b = b(\hat{n})$ locally as a function of the normal direction of the level sets of $\chi * K$. See Fig. 11 for an example.

As a final observation, nonlocal choices for λ also produce interesting flows. For example, volume-preserving motion by mean curvature [3, 29], i.e. $v_n = \kappa - \bar{\kappa}$ where $\bar{\kappa}$ is the surface average of the mean curvature, is realized by selecting the level surface of $\chi * K$ that encloses the same volume as the original set in diffusion-generated motion, instead of the $\frac{1}{2}$ level [30]. Convergence of the three procedures discussed in this section has been demonstrated numerically, but not proven analytically. Given its simplicity, it would be especially interesting to obtain a proof for the volume-preserving diffusion-generated motion by mean curvature.

4.6. Numerical Approximation

Perhaps the most obvious method for approximating convolution generated motion is pseudospectrally. Using this approach, functions are represented by their values on a

regular lattice of grid points. This makes the thresholding step trivial since it can be carried out pointwise. The convolution step is also straightforward since it reduces to a multiplication in Fourier space using fast Fourier transform (FFT) methods. Unfortunately, however, this simple approach is rarely adequate because of the strong grid effects discussed in Section 3.3.3. See [34] for an example.

For fast, accurate results, the discretization given in Section 3.3.2 may be used. Briefly, the characteristic function for the initial region and the kernel are approximated by Fourier tensor products. Multiplying in Fourier space then gives a simple estimate for the convolution product. The Fourier representation of the characteristic function for the updated region is then determined using an adaptive quadrature method rather than a pseudospectral method. Note that the convolution step acts as a filter, removing high-frequency components. Since this convolution step is linear, the different Fourier modes do not interact and there is never a need to treat the highest frequency components. Thus, an excellent approximation is obtained using fewer Fourier modes than might otherwise be expected. See [33, 34] for some examples and full details.

4.7. Related Methods

The "spatially continuous automata" of MacLennan [19] are another independent development similar to diffusion-generated motion. They arise from cellular automata, again as a method intended to capture the smoother, long-wavelength aspects of automata patterns. MacLennan achieves this simply by taking continuous versions of the spatially discrete aspects of cellular automata evolution. The resulting method consists of taking a continuous initial data function, evolving for a discrete time step by convolving it with a continuous convolution kernel, and then applying a continuous pointwise sharpening step that tends to undo some of the blurring of the convolution step. This procedure is quite similar to diffusion-generated motion (and the general convolution–thresholding motion we present in this section), except for one minor but crucial distinction. The simple asymptotics that yield motion by mean curvature in diffusion-generated motion arise precisely because the initial data are the discontinuous characteristic function, and because the sharpening step is discontinuous, replacing the blurred-out characteristic function by a new discontinuous characteristic function. Replacing these by continuous analogues destroys simple sharp interface motions in the first few time steps. Thus, these spatially continuous automata do not tend to yield well-behaved limiting interface motions amenable to asymptotic and rigorous analysis, although they do produce an interesting and varied class of evolutions.

5. CONNECTION TO CELLULAR AUTOMATA MODELS

Cellular automata are discrete dynamical systems. They consist of a lattice of sites, each of which may take on a finite number of "states," or values. The site values evolve in synchronous, discrete time steps according to an evolution rule that specifies the updated value in terms of the current values at neighboring sites [44].

In this section we review a particularly fundamental class of automata models—the threshold dynamics—and discuss some of their mathematical properties. As we shall see, convolution–thresholding motion arises naturally as the fine grid limit of these automata, giving a numerically and analytically tractable link between cellular automata models and the smooth features of pattern dynamics. We conclude this section with extensions to models for pattern dynamics in developmental biology and excitable media.

5.1. Connection to Threshold Dynamics

An important class of automata can be obtained by imagining each neighbor's contribution to be a simple "vote" for or against a certain state value of the site in question; any number of affirmative votes above a certain threshold will yield that outcome. For example, consider a simple voting automaton where there are two states, 1 and 0. A sum of the cell's own vote and that of its eight nearest neighbors is formed. Where this sum is greater than or equal to the threshold value λ the cell is assigned state 1; elsewhere it is assigned state 0. By denoting the state of cell (j, k) at time step n by C^n_{jk}, we obtain a simple analytic representation for the automata model,

$$C^{n+1}_{jk} = \begin{cases} 1 & \text{if } \sum_{-1\le j',k'\le 1} C^n_{j-j',k-k'} \ge \lambda \\ 0 & \text{otherwise,} \end{cases}$$

where λ represents the threshold value.

More generally, each vote can be assigned some weight. Letting $N \subset Z^2$ be the neighborhood of interest and W be the matrix of weights, we obtain the update rule for threshold dynamics,

$$C^{n+1}_{jk} = \begin{cases} 1 & \text{if } \sum_{j',k'\in N} W_{j',k'} C^n_{j-j',k-k'} \ge \lambda \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Note that the function C is precisely the characteristic function of a set on the lattice, and the combination appearing above is precisely the discrete convolution $C * W$ with the discrete kernel function W. Thus these threshold automata can be viewed as discrete versions of the convolution–threshold method. Moreover, they can also be viewed as discrete approximations to continuum convolution–threshold models, and this is a convenient framework for understanding the long-wavelength aspects of automata pattern formation, as described in the next sections.

5.2. Connection to Limiting Shapes

A natural and very interesting problem is to find the limiting shapes for threshold dynamics and related automata models. In an early paper, Packard and Wolfram [25] found that

> Most two dimensional patterns generated by cellular automaton growth have a polytropic boundary that reflects the structure of the neighborhood in the cellular automaton rule. Some rules, however, yield slowly growing patterns that tend to a circular shape independent of the underlying cellular automaton lattice.

To derive a more detailed, rigorous theory, Gravner and Griffeath [12] developed and studied a class of set-evolution algorithms of a form somewhat similar to that of the threshold dynamics. In these "threshold growth dynamics" an unoccupied site becomes occupied if a certain proportion of its neighbors are occupied, while occupied sites are never vacated. The main goal of this work was to prove that such discrete evolution rules lead to a certain asymptotic limiting shape for the evolving set as time (the number of iterations) goes to infinity. Gravner and Griffeath accomplish this in full rigor and generality, both on a continuum and a lattice.

In continuum terms, and from the viewpoint of convolution–thresholding motion as developed in Section 4, we would say they were analyzing a discrete approximation to a certain continuum convolution-generated motion. If the continuum convolution kernel were

known, then an anisotropic surface normal velocity law $v_n = a(\hat{n})$ could be determined from the general kernel-velocity relations obtained in [18]. From this normal velocity, the limiting shape would follow by the classical geometric Wulff construction [24]. As an aside, note that at the continuum level it is a classical observation about crystal growth (dating back to Gross in 1908) that such anisotropic velocity laws result in well-described limiting shapes as t goes to infinity and that the geometric Wulff construction on the function $v(n)$ yields the corresponding shape. However, rigorous proofs of this did not appear until recently. A simple direct proof for the standard continuum formulation, as well as more detailed discussion and references, are contained in a work of Osher and Merriman [24]. See also Ishii *et al.* [18] for recent results about the asymptotic shape of fronts propagating by threshold dynamics and Gravner and Griffeath [13] for some simple growth rules with more complex iterates which can nevertheless be determined by a combination of computer experiment and exact recursion.

5.3. Relation to Finite-Grid Effects in Automata

Notice that threshold dynamics can be viewed as a method for evolving interfaces since the boundary between state 0 and state 1 represents a crude interface that is evolved by each update step of the algorithm (see, e.g., Fig. 12). An interesting question is how to select an appropriate neighborhood and weight values to model a desired front motion.

In early cellular automata, a neighborhood of nearest neighbors on a uniform lattice was selected. This choice has the advantages of speed and simplicity but is inadequate for modeling many interesting natural phenomena. In particular, rules which use these small neighborhoods are unable to model the effects of curvature on the speed of propagation [15, 42]. The reason for this can be understood by referring back to the section on the Huygens' principle for mean curvature motion, where we showed that the size of the neighborhood must scale like $O(\sqrt{\Delta t})$, which is much larger than the $O(\Delta t)$ scale neighborhoods required for constant normal motion. These simple automata also add grid-based anisotropy to the front motion [36]. See Fig. 13 for an excitable automaton with a strong grid-based anisotropy.

In an attempt to reduce grid effects, several modifications of cellular automata such as random grids, stochastic local functions, and asynchronous evaluations have been designed. Based on extensive numerical experiments, Schönfisch [36] found that random grids are the most useful of these, but that even these randomized methods have deficiencies from a theoretical or a practical point of view. In particular, fluctuations in the front occur for random grids and these "fluctuations become more prominent for higher values of threshold" [36]. Thus, while randomized methods do produce a marked improvement in the isotropy of automata, they are still not adequate for many problems of practical interest.

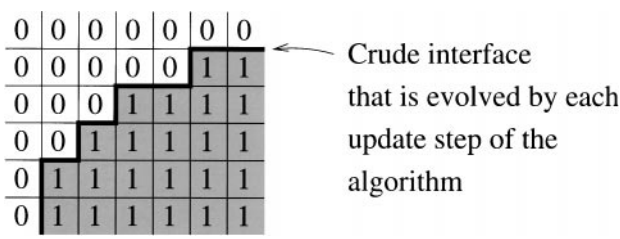

FIG. 12. Threshold dynamics can be viewed as a method for interface motion.

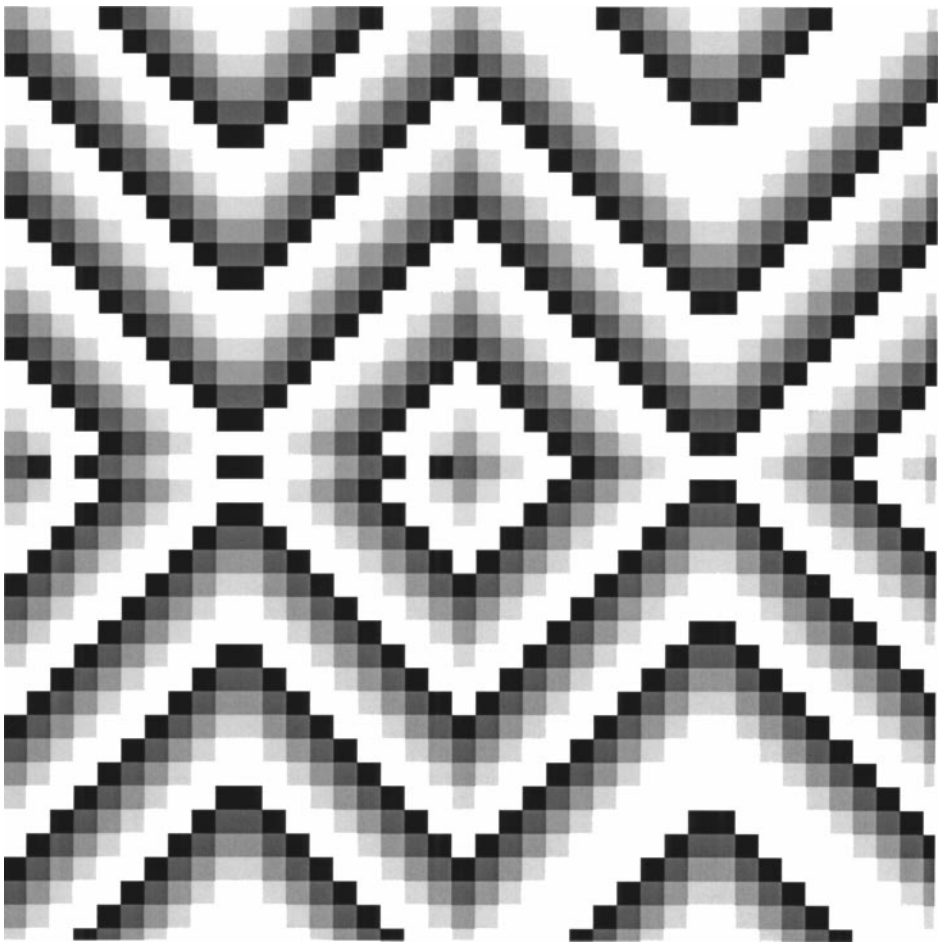

FIG. 13. Spiral wave in the excitable automata model [5]. There are six states, 0 through 5. If any cell is at state 5 it is set to state 0 (resting). Any other excited or refractory (i.e., nonzero) state is incremented by 1. If a cell is resting and one of its four neighbors is excited (state 1), then the cell becomes excited; otherwise it remains at rest. White corresponds to state 0 and black to state 5.

Alternatively, reduced grid effects and an improved curvature contribution can be obtained by refining the lattice and taking larger neighborhoods [8–12, 15, 20, 39, 40]. In the limit as the lattice is refined and larger and larger neighborhoods are used the summation step leads to a convolution

$$C * W(\mathbf{x}) = \int_{R^d} C(\mathbf{y}) W(\mathbf{x} - \mathbf{y})\, d\mathbf{y},$$

where C is the characteristic function for the initial region in the fine grid limit and W: $R^d \to R$ is the fine-grid large-neighborhood limit of the discrete convolution function. Thus in the limit relevant for eliminating lattice effects from the automata, threshold dynamics becomes convolution–thresholding (4).

Note that this means that the limiting motion can be accurately and efficiently treated using the methods outlined in Section 4.6. Alternatively, it is possible to approximate the sum pseudospectrally [34] or with a number of one-dimensional convolutions [9–11, 15,

39, 40]. However, these methods use a pointwise thresholding so each step displaces the front a distance which is comparable to the mesh spacing. This leads to strong grid effects that are often unacceptable in practical applications.

5.4. Application to Developmental Biology

Threshold dynamics have arisen in a variety of disciplines in developmental biology [5]. For example, Young devised an interesting model of vertebrate skin patterns which is based on local activation and inhibition [45]. This model assumes that cells are in one of two states—differentiated (colored) and undifferentiated. Each differentiated cell produces two diffusive chemicals: a short-range "activator" and a longer range "inhibitor." The activator stimulates the differentiation of nearby undifferentiated cells and the inhibitor stimulates nearby differentiated cells to become undifferentiated. The combined effect of these two chemicals is modeled as the weighted difference of concentrations.

To discretize this continuum model, Young uses an automaton. The convolutional form of Young's automaton is easily derived [34]. Simply set

$$\chi\colon R^d \to R$$

equal to the characteristic function for the differentiated region, Ω, and define the updated region, Ω^{new}, to be the set

$$\Omega^{\text{new}} = \{\mathbf{x}\colon \chi * K(\mathbf{x}) > 0\}$$

for the kernel function, K. The kernel K is not refined with Δt since the time evolution of the model is naturally discrete.

A variety of patterns are possible by varying the threshold, the size and symmetry of the neighborhood, and the relative weights of the activator and inhibitor [45]. For example, the steady patterns given in Fig. 14 arise from a kernel that represents the difference

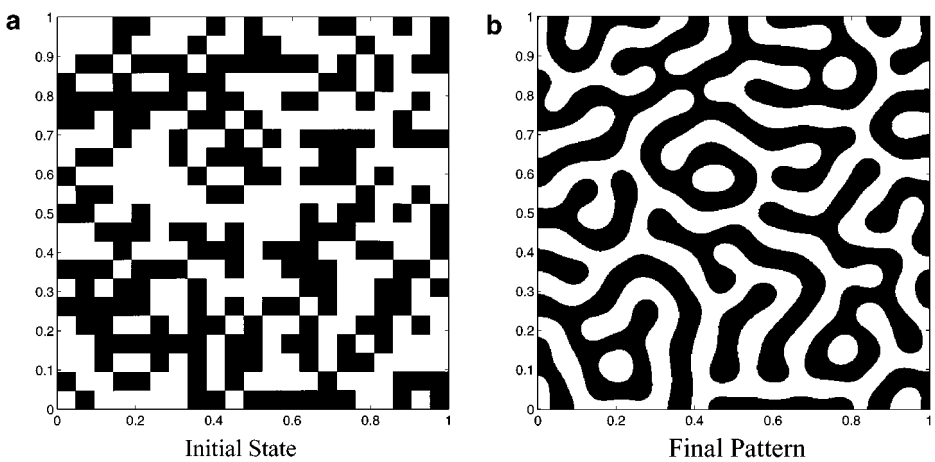

FIG. 14. Isotropic pattern formation after 100 steps starting from a random checkerboard pattern. In this case, the kernel consists of the difference of two Gaussian distributions: $K(x, y) = \frac{2500}{\pi}\exp(-2500|\mathbf{x}|^2) - \frac{1250}{3\pi}\exp(-\frac{1250}{3}|\mathbf{x}|^2)$.

of two symmetrical Gaussian distributions. Stronger contributions of either activator or inhibitor tend to generate spotty patterns, while nonsymmetrical kernels can produce interesting striped patterns [45]. Note that solutions to this convolution-based model are efficiently obtained using the discretization methods of Ruuth [32]: Only 128×128 basis functions were required to obtain the steady patterns in Fig. 14 whereas a lattice of 2048×2048 grid points is required using an automaton-based discretization. See [34] for further details.

It is interesting to note that similar convolution–thresholding schemes have also arisen in neural models for the visual cortex. For example, Ringach *et al.* develop a convolution–sharpening model for the study of simple cells in the primary visual cortex of cats and macaque monkeys [27]. Note, however, that these authors seek solutions to the interesting *inverse* problem of determining a kernel function, based on experimental input image sequences and output spike trains. In particular, they propose a new subspace reverse correlation technique which has several advantages over standard white-noise techniques, including an improved signal-to-noise ratio, increased spatial resolution, and the possibility of restricting the study to particular subspaces of interest. See also Swindale for a related model for generating patterns of ocular dominance in the visual cortex [37].

5.5. Application to Excitable Media

In the biological and physiological literature, the best-known examples of cellular automata are the excitable media [5]. In an excitable system, a sufficient stimulus (i.e., above some threshold value) leads to a large response followed by a period of recovery to a stable rest state. An *excitable medium* is a spatially distributed excitable system coupled in such a way that excitation can provoke excitation in neighboring regions. Note that these systems often experience a recovery or *refractory* period during which the medium is unable to be reexcited regardless of the size of stimulus. Examples of excitable media arise in diverse physical, chemical, and biological systems including models for nerve cells, muscle cells, cardiac function, developmental biology, chemical reactions, and star formation. See [5, 39, 41, 43] for further details and references.

In early cellular automata models for excitable media, update rules were based on the values in a neighborhood of nearest neighbors. Because this choice produces waves which propagate at a speed of one cell per time step, several serious shortcomings occur [9, 42]. The most serious of these are [15, 40]:

1. The speed of propagation does not depend on the extent of recovery of the medium.
2. The speed of propagation does not depend on the wavefront curvature.
3. Unwanted anisotropy is added to the front motion. See, e.g., Fig. 13.

To treat the first shortcoming, more recent automata select threshold values according to the recovery of the medium [8–11, 15, 20, 39, 40]. Averages over large neighborhoods are used in an attempt to reduce unwanted anisotropy and to obtain an approximation to the curvature component of the motion [8–11, 15, 20, 39, 40]. Typically, this averaging step is carried out either directly [8, 20] (which is slow but general), using a number of one-dimensional convolutions [9–11, 15, 39, 40] (which is efficient but specialized), or pseudospectrally (which is efficient *and* general—see [34]).

Consider, for example, the excitable automata introduced in Gerhardt *et al.* [9–11], Weimar *et al.* [39, 40], and Henze and Tyson [15]. In these automata, update rules are

chosen to mimic the dynamics of a two-variable system of reaction–diffusion equations,

$$\frac{\partial u}{\partial t} = \frac{1}{\epsilon} f(u, v) + D_u \nabla^2 u$$
$$\frac{\partial v}{\partial t} = g(u, v) + D_v \nabla^2 v,$$

where ϵ is a small parameter and $f(u, v)$ and $g(u, v)$ specify the local kinetics of the system. Note that the scalar u (the excitation variable) changes on a time scale which is much faster than the scalar v (the recovery variable). To derive the corresponding automaton model, the reaction–diffusion system is split in a nonconvergent way into four steps which are carried out sequentially (see [34, 40] for details):

1. The excitation variable is diffused.
2. The diffused excitation variable is thresholded to 0 (resting) or 1 (excited) according to the value of the recovery variable, i.e., $\lambda = \lambda(v)$.
3. The recovery variable is evolved according to the local kinetics.
4. The result from Step 3 is diffused to give the updated recovery variable.

Finally, the discretization is completed by representing u and v by their pointwise values on a regular lattice.

The advantages of this automaton over earlier models are clear. Since large neighborhoods are used, the motion of the wavefront will exhibit fewer grid effects (i.e., reduced anisotropy) and will have an improved dependence on curvature. Furthermore, the wave speed will depend on the extent of recovery of the medium since thresholding is carried out according to the value of v. Indeed, simulation results reported for FitzHugh–Nagumo kinetics [15] and the Oregonator model [40] agree well with PDE simulations for the period, wavelength, and motion of the tip of the spiral wave for a wide range of parameters. When compared to PDE simulations, the automata model has the practical advantage that it ignores the details of the fast kinetics so that "the time step in the cellular automaton can exceed that in PDE simulations by 1 or 2 orders of magnitude" [15].

Note that discretizations which use a pointwise thresholding should be avoided because this type of thresholding displaces the front a distance which is comparable to the mesh spacing. Fortunately, an improved discretization is easily obtained [34]. Steps 1 and 2 above are trivially treated using the discretization methods of diffusion-generated motion (see Section 3.3.2). The evolution of the recovery variable is similar, except that we must use Gaussian quadrature rather than exact integration to evaluate the Fourier coefficients. Similar to automata-based discretizations, this approach allows very large time steps (relative to PDE simulations) since it ignores the details of the fast dynamics. Relative to automata-based discretizations, however, it is clear that this spectral discretization gives a much more accurate treatment of the front since it recursively refines near the interface and interpolates at the finest cell level. This allows for accurate estimates of quantities defined on the interface and is particularly valuable for computing curvature-dependent motions. Furthermore, discontinuities and unwanted anisotropy in the front motion are eliminated since interpolation is used to locate the front at the finest cell level. Finally, this discretization has the benefit that Δt can be selected independently of other parameters, unlike the methods proposed in [9–11, 15, 39, 40]. See Fig. 15 for an evolving spiral wave which was computed using these fast methods.

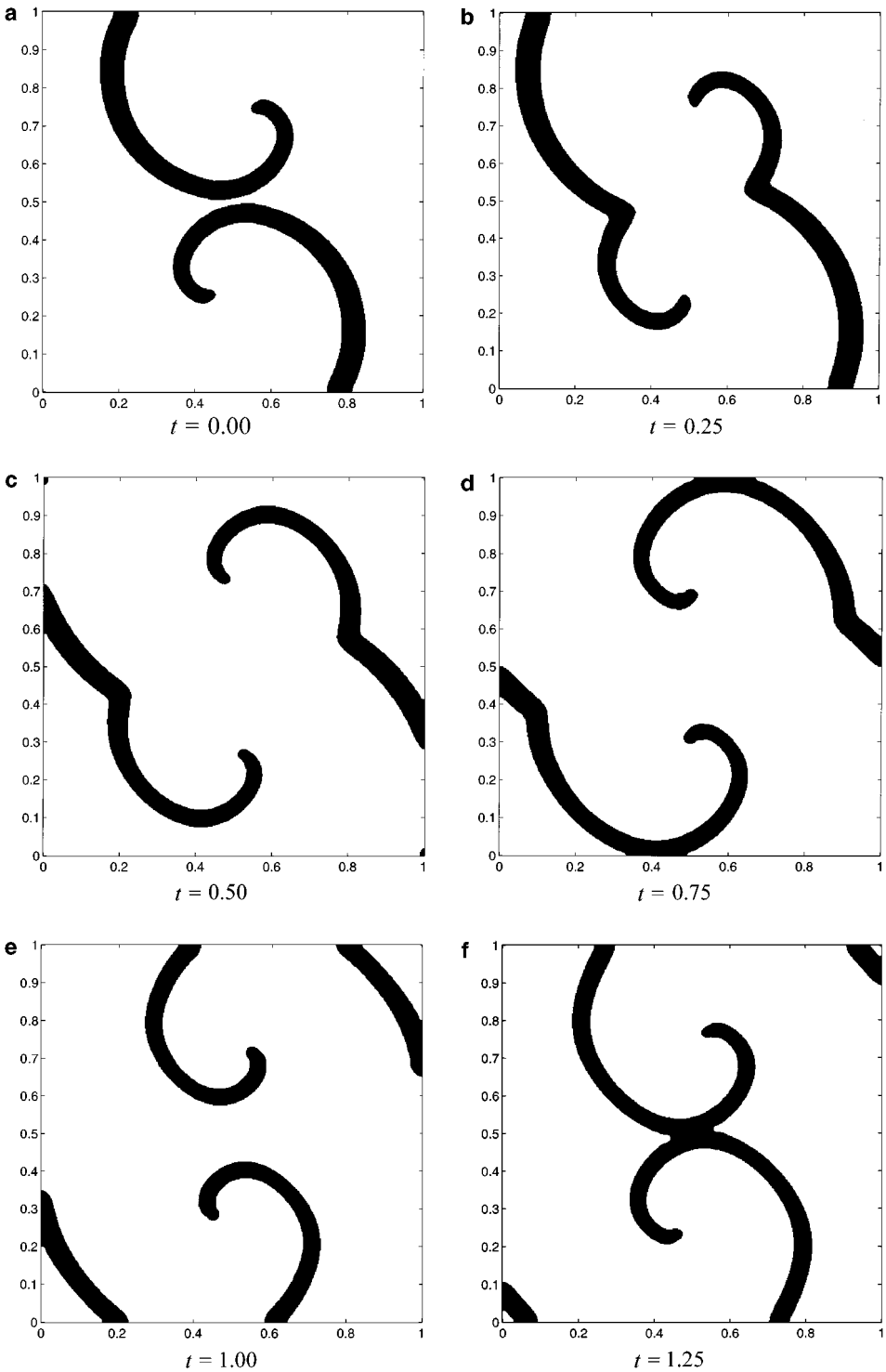

FIG. 15. Excitation variable for two interacting spiral waves. See [34] for details.

6. CONNECTION TO PHASE-FIELD PDES

As we have seen in Section 4.2, diffusion-generated motion by mean curvature is a special case of convolution–thresholding. We will show in turn how the diffusion-generated motion methods are related to phase-field models and thereby establish the general connection between convolution–thresholding and phase-field PDE models. We also show how this connection can be used to motivate a recent approach for evolving filaments with a normal speed equal to curvature. A closely related method for evolving orientation vector fields is also reviewed.

6.1. Phase-Field and Diffusion-Generated Motion

The diffusion-generated motion procedure of alternately diffusing and thresholding is reminiscent of an operator-splitting approximation of the (real-valued) Ginzburg–Landau equation,

$$u_t = \nabla^2 u - \frac{1}{\epsilon^2} u(u^2 - 1).$$

In this PDE model, a reaction front of width ϵ develops, separating large regions of constant equilibrium states for the reaction, i.e., where $u \approx 1$ or $u \approx -1$. In the asymptotic limit $\epsilon \to 0$ of a strong reaction and weak diffusion, the reaction front moves by mean curvature [7]. At a formal level, by splitting the process into separate time steps of diffusion and reaction, and driving the reaction step to equilibrium (i.e., set $u(\mathbf{x})$ to be the closer of the equilibrium states -1 and 1) we arrive at the diffusion-generated motion by mean-curvature algorithm. Thus (as $\Delta t \to 0$) in this formal time splitting we actually achieve the asymptotic mean-curvature motion of the phase-field model. Moreover, the split process is considerably simpler than the nonlinear PDE dynamics, as it consists of just linear diffusion and thresholding, and since there is no development of artificial small $O(\epsilon)$ spatial length scales.

From a theoretical and computational standpoint, this phase-field model has the benefit that topological shape changes such as merger and pinch off are treated automatically. Unfortunately, if phase-field PDEs are used in computation it is necessary to resolve the thin $O(\epsilon)$ wide reaction zone to obtain numerical accuracy [23]. In contrast, diffusion-generated motion does not have this artificial small scale. Thus diffusion-generated motion has in effect passed to the asymptotic limit of the phase-field class of models, a simplified and more accurate evolution scheme being obtained in the process.

More generally, this suggests the formal "meta-principle" that we could replace certain phase-field PDE models whose asymptotic limit produces an interface motion by a diffusion-generated motion procedure that achieves the limiting motion (in the $\Delta t \to 0$ limit) without any artificial small spatial scales. The process would be to simply do the linear diffusion evolution on a suitable representing function whose values are all equilibrium states, and which is singular at the ideal interface, and then threshold by projecting smoothed-out values back to the equilibrium states of the reaction. It is an open question as to how generally valid this meta-principle is, but as is illustrated below for filaments, it seems to have some general validity.

6.2. The Diffusion-Generated Motion of Filaments

Interestingly, the idea of treating phase-field equations in a formal split-step manner can also be used to produce a diffusion-generated method for the curvature motion of *filaments* in three dimensions (or, more generally, 1-D filaments in any number of dimensions, or even k-D "filaments" moving in a higher dimensional space, for example by general vector mean-curvature flow) [35].

Consider the complex Ginzburg–Landau equation

$$u_t = \nabla^2 u - \frac{1}{\epsilon^2} u(|u|^2 - 1),$$

where $u(\mathbf{x}, t)$, $\mathbf{x} \in R^3$, is a complex scalar and ϵ is a small positive parameter. In this PDE model, the filament is given by the curve where $|u|$ vanishes, which is in a tube of width ϵ outside of which $|u| \approx 1$. Notice that the Laplacian term dominates in a neighborhood of width ϵ of the filament, while farther away the reaction term dominates (see Fig. 16). In the asymptotic limit $\epsilon \to 0$ of a strong reaction and weak diffusion, the filament moves in the principal normal direction with a speed equal to its curvature [28].

Similar to the case of diffusion-generated motion, a formal splitting method can be applied to the complex Ginzburg–Landau equation to obtain an algorithm for evolving filaments [35]. This "diffusion-generated motion by mean curvature for filaments" alternates one step of normalizing u (i.e., replacing u by $u/|u|$, which is a natural generalization of the usual $(\chi - \lambda)/|\chi - \lambda|$ thresholding step) with a step of diffusion over a time Δt. The corresponding algorithm evolves the filament in the Frenet normal direction to the curve with a speed equal to curvature, and it naturally captures topological merging and breaking of filaments without fattening curves. It also gives an improved computational efficiency over direct numerical simulation of the Ginzburg–Landau equations because it obtains the $\epsilon \to 0$ limit of the phase-field model without this artificial, small scale.

See [35] for a variety of interesting numerical experiments and generalizations as well as an asymptotic analysis justifying the convergence of the algorithm.

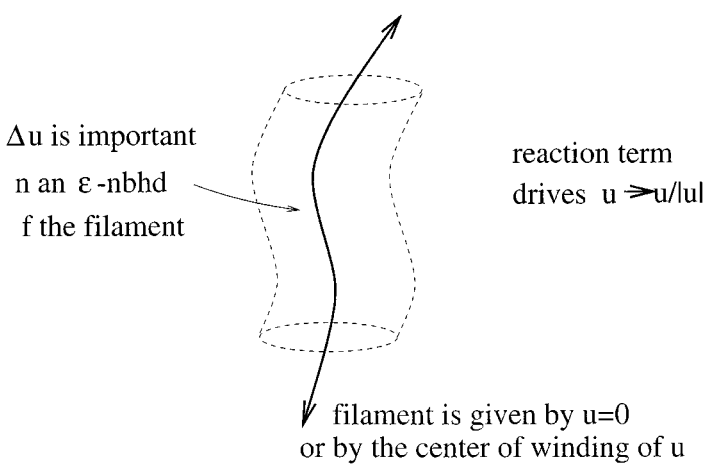

FIG. 16. The filament is given by $u = 0$, or the center of winding of u. In an ϵ-neighborhood of the filament, the Laplacian term dominates. Further away, the reaction term drives u to $u/|u|$.

6.3. Orientation Diffusions

Independent of the work on diffusion-generated motion, Perona [26] developed and studied a diffusion-based algorithm for evolving orientation-like quantities. His motivation was to develop methods appropriate for smoothing noisy data, analyzing images at multiple scales, and enhancing discontinuities for applications in image processing and computer vision. In particular, he was interested in problems where the important information is contained in the orientation of lines, rather than the brightness values. To accomplish these goals, Perona embeds the orientation θ in the plane via the map

$$w = [\cos(\theta), \sin(\theta)].$$

The orientation vector w is then alternately diffused for a short time and projected onto the unit circle to give an algorithm that is remarkably similar to diffusion-generated motion for filaments.

Perona also provides a simple discretization for his method and gives a variety of interesting examples that demonstrate that his approach eliminates noise and gives useful image information at multiple scales. See [26] for full details.

7. SUMMARY AND DIRECTIONS FOR FUTURE WORK

Convolution–thresholding is a flexible, general framework for defining interface motions. In this approach, an interface is represented as the singular set of a suitable representing function, and the function is updated in time by alternatively convolving with a smoothing kernel and thresholding (or, more generally, projecting back onto the restricted set of values) to obtain an updated valid representing function. This approach is intrinsically discrete in time and is amenable to fast, accurate spatial discretization via Fourier transform techniques. The resulting schemes tend to be simple, and yet they can describe complex, curvature-dependent flows that include topological changes and triple-point motions. The approach also generalizes to describe the curvature motion of filaments or arbitrary dimension subsurfaces within higher dimensional spaces. The method has illuminating relations to Huygens' principle, cellular automata, and reaction–diffusion/phase-field PDE models of interface motion and can provide a valuable alternative formulation in various applications or theoretical investigations.

A key area of future work is the *inverse* problem (cf. [14]): given a surface-motion law, find a kernel (or kernels) and some thresholding technique that achieves that law. As an example, we are currently seeking methods for the anisotropic curvature-dependent motion of junctions such as those arising in materials science applications. Other areas of interest include the development of new methods for more general (or possibly nonlocal) motion laws or methods for constrained curvature-dependent flows (cf. [4]). More generally, statistical methods offer great promise in modeling a variety of interesting experimental processes. See Ringach *et al.* for an example [27].

Another interesting theoretical problem is to establish the range of validity of the meta-principle from Section 6, i.e., to determine when a phase-field or reaction–diffusion type of PDE model has the same limiting behavior as its diffusion-generated motion analogue [35]. When applicable, this principle allows interface or filament motion to be immediately translated into the simpler convolution–thresholding schemes.

It is also of great interest to couple convolution–thresholding schemes to physical or biological processes occurring off the interface. An example of such a coupling was given in Section 5.5 where the thresholding level was set according to a second recovery variable to simulate an excitable medium, but it remains to fully develop such coupling strategies for general classes of coupled interface–external equation models.

Computationally, the algorithms of Ruuth [32] give a simple and efficient means for treating most curvature-dependent motions. However, the method can be inefficient for motions which are independent of curvature since the corresponding kernels have smaller supports and so require more spatial resolution than those for curvature motion [33]. Thus, an interesting research project would be to develop fast algorithms for these small kernels.

Theoretically, rigorous treatments of two-phase motions have been developed in codimension one (e.g., curves in two dimensions or surfaces in three dimensions). These proofs assume positive, symmetric kernels and a fixed threshold [1, 6, 17, 18]. For extensions to arbitrary codimension, multiple junctions, and variable thresholds, a variety of heuristic arguments, asymptotics, and experimental evidence supporting convergence have been developed [21–23, 31, 33, 35], but a rigorous theory has proven elusive. Indeed, many interesting kernels have not yet been the subject of systematic numerical investigation. These include nonsymmetric kernels and kernels involving both positive and negative components. Also, the extremely simple volume-preserving motion by mean-curvature algorithm described in Section 4.5 would be an excellent target for a convergence proof.

As can be seen from this brief review, convolution–thresholding methods for interface motion have attracted considerable theoretical and computational interest and have interesting relations and contrasts with other methods for surface evolution. They have arisen independently in varied fields of research, and they provide an interesting bridge connecting geometric, PDE, and cellular automata models that produce moving interfaces. We anticipate a great amount of future development as these connections and applications are explored more thoroughly.

REFERENCES

1. G. Barles and C. Georgelin, A simple proof of convergence for an approximation scheme for computing motions by mean curvature, *SIAM J. Numer. Anal.* **32**(2), 484 (1995).
2. G. Beylkin, On the fast Fourier transform of functions with singularities, *Appl. Comput. Harmonic Anal.* **2**, 363 (1995).
3. L. Bronsard and B. Stoth, *Volume Preserving Mean Curvature Flow as a Limit of a Nonlocal Ginzburg–Landau Equation*, Technical Report 94-NA-008, Centre for Nonlinear Analysis, Carnegie Mellon University, 1994.
4. D. L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* **106**(1), 77 (1993).
5. G. B. Ermentrout and L. Edelstein-Keshet, Cellular automata approaches to biological modeling, *J. Theor. Biol.* **160**, 97 (1993).
6. L. C. Evans, Convergence of an algorithm for mean curvature motion, *Indiana Univ. Math. J.* **42**, 553 (1993).
7. L. C. Evans, H. M. Soner, and P. E. Souganidis, Phase transitions and generalized motion by mean curvature, *Commun. Pure Appl. Math.* **45**(9), 1097 (1992).
8. V. G. Fast and I. G. Efimov, Stability of vortex rotation in an excitable cellular medium, *Physica D* **49**, 75 (1991).
9. M. Gerhardt, H. Schuster, and J. J. Tyson, A cellular automata model of excitable media. II. Curvature, dispersion, rotating waves and meandering waves, *Physica D* **46**, 392 (1990).
10. M. Gerhardt, H. Schuster, and J. J. Tyson, A cellular automata model of excitable media. III. Fitting the Belousov–Zhabotinsky reaction, *Physica D* **46**, 416 (1990).

11. M. Gerhardt, H. Schuster and J. J. Tyson, A cellular automata model of excitable media including curvature and dispersion, *Science* **247**, 416 (1990).
12. J. Gravner and D. Griffeath, Threshold growth dynamics, *Trans. Am. Math. Soc.* **340**(2), 837 (1993).
13. J. Gravner and D. Griffeath, Cellular automata growth on z^2, *Adv. Appl. Math.* **21**, 241 (1998).
14. H. A. Gutowitz, Introduction, in *Cellular Automata Theory and Experiment* (MIT/North-Holland, Amsterdam, 1991), p. vii.
15. C. Henze and J. Tyson, Cellular automaton model of three-dimensional excitable media, *J. Chem. Soc. Faraday Trans.* **92**(16), 2883 (1996).
16. C. Herring, The use of classical macroscopic concepts in surface energy problems, in *Structure and Properities of Solid Surfaces*, edited by R. Gomer and C. S. Smith (University of Chicago, Chicago, 1952), p. 5.
17. H. Ishii, A generalization of the Bence, Merriman and Osher algorithm for motion by mean curvature, in *Curvature Flows and Related Topics*, edited by A. Damlamian, J. Spruck, and A. Visintin (Gakkôtosho, Tokyo, 1995), p. 111.
18. H. Ishii, G. E. Pires, and P. E. Souganidis, Threshold dynamics type schemes for propagating fronts, *TMU Math. Preprint Ser.* **4**, (1996).
19. B. J. MacLennan, *Continuous Spatial Automata*, Technical Report CS-90-121, University of Tennessee, Dept. of Computer Science, Knoxville, 1990, available at http://www.cs.utk.edu/∼mclennan/fieldcomp-biblio.html.
20. M. Markus and B. Hess, Isotropic cellular automaton for modeling excitable media, *Nature* **347**, 56 (1990).
21. P. Mascarenhas, *Diffusion Generated Motion by Mean Curvature*, CAM Report 92-33, University of California, Dept. of Math., Los Angeles, 1992.
22. B. Merriman, J. Bence, and S. Osher, Diffusion generated motion by mean curvature, in *Computational Crystal Growers Workshop*, edited by J. E. Taylor (American Mathematical Society, Providence, Rhode Island, 1992), p. 73. Also available as UCLA CAM Report 92-18, April 1992.
23. B. Merriman, J. Bence, and S. Osher, Motion of multiple junctions: A level set approach, *J. Comput. Phys.* **112**(2), 334 (1994).
24. S. Osher and B. Merriman, The Wulff shape as the asymptotic limit of a growing crystalline interface, *Asian J. Math.* **1**(3), 560 (1997).
25. N. Packard and S. Wolfram, Two-dimensional cellular automata, *J. Stat. Phys.* **38**, 901 (1985).
26. P. Perona, Orientation Diffusions, *IEEE Trans. Image Process.* **7**(3), 457 (1998).
27. D. Ringach, G. Sapiro, and R. Shapley, A subspace reverse-correlation technique for the study of visual neurons, *Vision Res.* **37**(17), 2455 (1997).
28. J. Rubinstein, Self-induced motion of line defects, *Q. Appl. Math.* **49**(1), 1 (1991).
29. J. Rubinstein and P. Sternberg, Nonlocal reaction–diffusion equations and nucleation, *IMA J. Appl. Math.* **48**, 248 (1992).
30. S. J. Ruuth, *Efficient Algorithms for Diffusion-Generated Motion by Mean Curvature*, Ph.D. thesis, University of British Columbia, Vancouver, Canada, 1996.
31. S. J. Ruuth, A diffusion-generated approach to multiphase motion, *J. Comput. Phys.* **145**, 166 (1998).
32. S. J. Ruuth, Efficient algorithms for diffusion-generated motion by mean curvature, *J. Comput. Phys.* **144**, 603 (1998).
33. S. J. Ruuth and B. Merriman, Convolution generated motion and generalized Huygens' principles for interface motion, *SIAM J. Appl. Math.* **60**(3), 868 (2000).
34. S. J. Ruuth, B. Merriman, and S. Osher, Convolution generated motion as a link between cellular automata and continuum pattern dynamics, *J. Comput. Phys.* **151**, 836 (1999).
35. S. J. Ruuth, B. Merriman, J. Xin, and S. Osher, *Diffusion-Generated Motion by Mean Curvature for Filaments*, CAM Report 98-47, University of California, Los Angeles, 1998.
36. B. Schönfisch, Anisotropy in cellular automata, *Biosystems* **41**, 29 (1997).
37. N. V. Swindale, A model for the formation of ocular dominance stripes, *Proc. R. Soc. London B* **208**, 243 (1980).
38. J. E. Taylor, II, Mean curvature and weighted mean curvature, *Acta Metall. Meter.* **40**(7), 1475 (1992).

39. J. R. Weimar, J. J. Tyson, and L. T. Watson, Diffusion and wave propagation in cellular automata models of excitable media, *Physica D* **55**, 309 (1992).
40. J. R. Weimar, J. J. Tyson, and L. T. Watson, Third generation cellular automata for modeling excitable media, *Physica D* **55**, 328 (1992).
41. N. Wiener and A. Rosenblueth, The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specifically in cardiac muscle, *Arch. Inst. Cardiol. Mexico* **16**, 205 (1946).
42. A. T. Winfree, *When Time Breaks Down* (Princeton Univ. Press, Princeton, NJ, 1987).
43. A. T. Winfree, Stable particle-like solutions to the nonlinear wave equations of three-dimensional excitable media, *SIAM Rev.* **32**(1), 1 (1990).
44. S. Wolfram, Universality and complexity in cellular automata, *Physica D* **10**, 1 (1984).
45. D. A. Young, A local activator-inhibitor model of vertebrate skin patterns, *Math. Biosci.* **72**, 51 (1984).