CAR-TR-868
CS-TR-3825

N00014-95-1-0521
August 1997

# Discrete, Nonlinear Curvature-Dependent Contour Evolution

Scott Thompson and Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

COMPUTER VISION LABORATORY

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742-3275

CAR-TR-868
CS-TR-3825

N00014-95-1-0521
August 1997

# Discrete, Nonlinear Curvature-Dependent Contour Evolution

Scott Thompson and Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

## Abstract

There has been much recent interest in curvature-dependent contour evolution, particularly when the resultant family of contours satisfies the heat (diffusion) equation. Modeling the evolution of a shape's boundary as a real-valued solution to the reaction-diffusion equation has been shown to be useful for shape decomposition [3]. This approach to contour evolution involves solving a partial differential equation (PDE), is computationally demanding, and must deal with the problem of singularities. In this paper, we describe a low-precision discrete method of contour evolution, based on the 8-connected chain code of the contour, that performs analogously to PDE-based methods and avoids the singularity problem. (Preliminary work along these lines was described in [12].) Our discrete method is not limited to linear functions of curvature; we give several examples of contour evolution processes that depend nonlinearly on curvature, including examples studied in [6], and illustrate their possible uses.

**Keywords:** Contour evolution, Curvature, Shape

The support of the Office of Naval Research under Grant N00014-95-1-0521 is gratefully acknowledged. The first author acknowledges the support of the Henry M. Jackson Foundation.

## 1 Preliminary Definitions

Digital **shapes** are non-empty, finite sets of grid points $S = \{P : P \in \mathbb{Z}^2\}$. The grid points can be regarded as the centers of unit squares ("cells"). The **background** (complement) of a shape (or set of shapes) $S$ is the set of grid points $\bar{S} = \{P : P \notin S\}$. The **digitization** of a shape in the real plane is the set of cells that intersect the shape (or the set of grid points at the centers of these cells).

For two grid points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, $P$ and $Q$ are called 4-**adjacent** (or 4-**neighbors**) if

$$|x_1 - x_2| + |y_1 - y_2| = 1.$$

Similarly, they are called 8-**adjacent** (or 8-**neighbors**) if

$$\max(|x_1 - x_2|, |y_1 - y_2|) = 1.$$

The same terminology is used for the cells that have the points at their centers. The reflexive, transitive closure of $i$-adjacency ($i = 4$ or 8) is called ***i*-connectedness**; in other words, $P$ and $Q$ are called ***i*-connected** if there exists a sequence of grid points $P = P_0, P_1, \ldots, P_n = Q$, such that $P_k$ and $P_{k+1}$ are $i$-adjacent, $0 \leq k < n$.

The ***i*-boundary** $S_i'$ of a digital shape $S$ is the set of cells of $S$ that are $i$-adjacent to cells of the background:

$$S_i' = \{P : P \in S, P \text{ } i\text{-adjacent to } Q \in \bar{S}\}.$$

From now on, "boundary" and "adjacency" will refer to the 4-boundary and 8-adjacency, respectively, and we will drop the subscript $i$.

It can be shown [8] that if a digital shape $S$ is 8-connected and its complement is 4-connected, the cells of its boundary can be cyclically ordered, such that successive cells in the boundary are 8-adjacent. (More generally, this can be done for the boundary associated with each 4-connected component of the background of $S$, e.g., if $S$ is a shape with holes.) The contour $C = \{P_k = (x_k, y_k), k = 1, \ldots, n\}$ is the 8-connected path obtained by following the boundary of $S$ in a clockwise fashion. The **chain code** of $C$ consists of the $n$ vectors $\vec{v_k} = \overline{P_{k-1}P_k}$, each of which can be represented by an integer $j \in [0, 7]$, where $j$ denotes the vector whose slope is $45j^o$. The chain code can be regarded as representing a **digital curve.**

The remainder of this paper is organized as follows. Section 2 describes a method for discrete approximation of the curvature of a boundary, which takes into account the nonisotropicness of the grid. Section 3 describes continuous contour evolution guided by solutions to the reaction-diffusion equation, and also discusses discrete contour evolution processes. Section 4 illustrates discrete contour evolution with examples using both linear and nonlinear functions of boundary curvature.

## 2 Curvature Estimation

In contour evolution processes, a contour is progressively deformed, with the amount of deformation at a point depending on the curvature of the contour at that point. In this paper we will define discrete contour evolution processes that are applied to the digital curve defined by the chain code of the boundary of a digital shape.

The curvature of a twice-differentiable curve at a point can be expressed in terms of the first and second derivatives of the curve at that point. No such simple definition of curvature exists for a digital curve. One might try to define it by substituting differences (derived from the chain code) for derivatives in a formula for the curvature, but this approach is not satisfactory

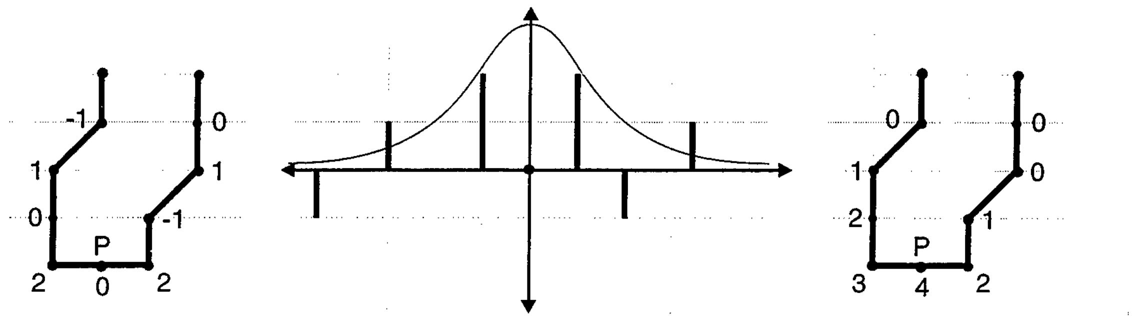

Figure 1: (a) Raw signed differences between consecutive chain codes. (b) Differences as a function of distance from the cell P. (c) The curvature estimates when these differences are convolved with the Gaussian shown in (b).

since the vectors in a chain code have slopes that differ by multiples of $45^o$; thus, small changes in slope cannot be locally described.

The $45^o$ limitation is a result of the standard use of 8-neighbor chain codes for curve representation. Generalized chain codes[9] provide finer angular resolution, but it can be shown that they offer no advantage over the 8-neighbor chain code from the point of view of representational precision. The average quantization error between a curve and its discrete approximation is directly proportional to the grid size (scale) regardless of the particular code chosen. In general, better approximations of a curve can be obtained by reducing the grid size; but small changes in slope still go undetected. Smoothing the signed differences between adjacent chain codes usually improves slope and curvature estimation, but the amount of smoothing needed to properly handle a shape that has features of multiple sizes is difficult to determine.

Several methods of estimating the curvature of the boundary of a real shape from the chain code of the boundary of its digitization are described in [13]. It is shown there that better results are achieved when the nonisotropicness of the square grid is taken into account. In this paper we use a variant of the "resampling method" discussed in [13]; it consists of two steps:

1. Compute the signed differences between consecutive chain codes (Figure 1a).
2. Smooth these differences with a Gaussian, taking into account the nonisotropicness of the grid (Figure 1b).

Specification of the Gaussian used for smoothing requires, in addition to the scale parameter $\sigma$, a truncation parameter $m$ which specifies the size of the smoothing neighborhood:

$$G_\sigma(i) = (\frac{1}{\sigma\sqrt{2\pi}}e^{-i^2/2\sigma^2}), i \in [-m, m]$$

In order to minimize truncation error, $m$ is set to the smallest integer larger than $3\sigma$ [11]. We discuss the choice of $\sigma$ in the next section, where we also discuss the precision used in the digital computations.

## 3 Contour Evolution

### 3.1 Continuous Contour Evolution

Contour evolution refers to a process of progressively deforming a contour or curve. An important class of deformations of a curve can be described as a linear sum of two local displacements along the normal $\vec{N}(P)$ at a given point $P$ of the curve:

$$(\beta_0 + \beta_1 \kappa(P))\vec{N}(P),$$

where $\kappa(P)$ is the curvature at $P$. The repeated displacement results in motion (i.e., deformation) of the contour; the $\beta_0$ term gives rise to a constant motion and the $\beta_1\kappa$ term gives rise to motion with magnitude proportional to $\kappa(P)$. Constant motion is equivalent to the morphological operations of erosion or dilation. It can be shown[3] that curvature-dependent motion corresponds to the effects of smoothing the curve with a Gaussian. These two processes have different properties and can be used to capture various properties of the shape bounded by the curve. It can be shown [2] that the combined process satisfies a viscous conservation law—a reaction-diffusion equation. It has also been shown that the processes described by this equation can be used to decompose a shape into parts and protrusions by applying the process for a sufficient amount of time. When run in reverse, the process can simulate a natural evolution of the shape from an initial oval [4].[1]

### 3.2 Discrete Contour Evolution

#### 3.2.1 Iterative Erosion and Dilation

In this paper we will study discrete iterative contour deformation processes in which cells on the boundary of a digital shape are either removed, or produce additional cells. A boundary cell that is removed from the shape is said to be "eroded," and a cell that produces an additional cell is said to be "dilated". To make the erosion or dilation of the boundary at each cell dependent on the estimated curvature, we control the number of iterations before a boundary cell $P$ erodes or dilates; specifically, we let this number be

$$\begin{array}{cc} [2^b/|f(P)|] & f(P) \neq 0 \\ \infty & f(P) = 0 \end{array}$$

where $f(P)$ is a function of the digital curvature of the boundary at $P$, and $b$ is the number of bits of precision used in the computations. The sign of $f$ determines whether $P$ erodes or dilates: by convention, $P$ erodes when $f$ is negative; it dilates when $f$ is positive; and it neither erodes nor dilates when $f$ is zero.

As previously stated, a cell that erodes simply disappears from the shape. A cell that dilates produces a cell in the direction of the outward pointing normal to the contour of the shape at that cell. The normal direction is computed by combining the directions from the cell to the two adjacent cells on the boundary. Specifically, if the difference between these directions is a

[1]It is observed in [3] that when used separately, morphological operators and Gaussian smoothing cannot produce intuitive shape decompositions. The former class of operators (closing, for example) can be used to decompose an object into blob-like and ribbon-line parts (see, e.g., [1]; other methods of decomposing a shape are described in [7, 10]). The latter type of operator smooths the object into a circle, and suggests a protrusion-based view of shape representation (see also [5]). The reaction-diffusion equation provides a mathematical framework in which these contrasting views of shape representation are combined, reflecting the idea that objects are naturally perceived as compositions of both blob- and ribbon-like parts, and of protrusions.

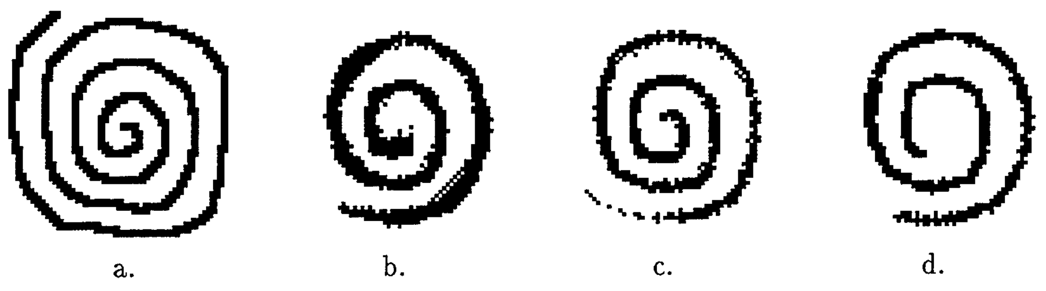

Figure 2: (a) A two-pixel-thick spiral. (b) Dilation occurs faster in the diagonal directions. (c) Scaling dilation along the diagonals. (d) Redefining dilation along a diagonal to be dilation in the two adjacent axial directions.

multiple of $90^o$, the normal direction is midway between them; if it is an odd multiple of $45^o$, it is the axial direction nearest to midway between them.

Ideally, the rate of erosion or dilation should depend solely on the value of $f$. In fact, this is difficult to achieve, since dilation in a diagonal direction is $\sqrt{2}$ times faster than dilation along an axis. This is illustrated in Figure 2, where $f(P) = \kappa(P)$ is applied to a two-pixel-thick spiral. Figure 2b shows a thickening (and a "splitting") in the diagonal directions. Two methods were used to reduce this effect of the nonisotropy of the grid. In the first method, dilation in the diagonal directions was scaled by $1/\sqrt{2}$ (Figure 2c). In the second, diagonal dilation was replaced by dilations in the two adjacent axial directions (Figure 2d). As the figures illustrate, the second method gave better results. This method was therefore used in the remainder of our experiments.

### 3.2.2 Parameter Selection

In Section 2, we described how to measure the curvature of a digital curve by using Gaussian smoothing of the chain code. To determine a reasonable value for the scale parameter $\sigma$ of the Gaussian, we experimented with the spiral shape shown in Figure 2a. When $f(P) = \kappa(P)$, we expect the following:

1. The spiral should shrink to a single cell.
2. The spiral should never intersect itself.
3. The spiral should never break.
4. The thickness of the spiral should remain constant.

The results shown in Figure 3 indicate that $\sigma = 3$ is adequate. As we recall from Section 2, the truncation parameter, $m$, that defines the size of the neighborhood over which Gaussian smoothing is done, is then set to 10, which is the smallest integer less than $3\sigma$. Evidently, higher values for $\sigma$ would have produced similar results, but at a higher computational cost.

We also used the spiral shape to test the effect of varying the number of bits, $b$, of precision used in the computations. The results shown in Figure 4 indicate that $b = 6$ is adequate. Evidently, higher values for $b$ would have produced similar results.

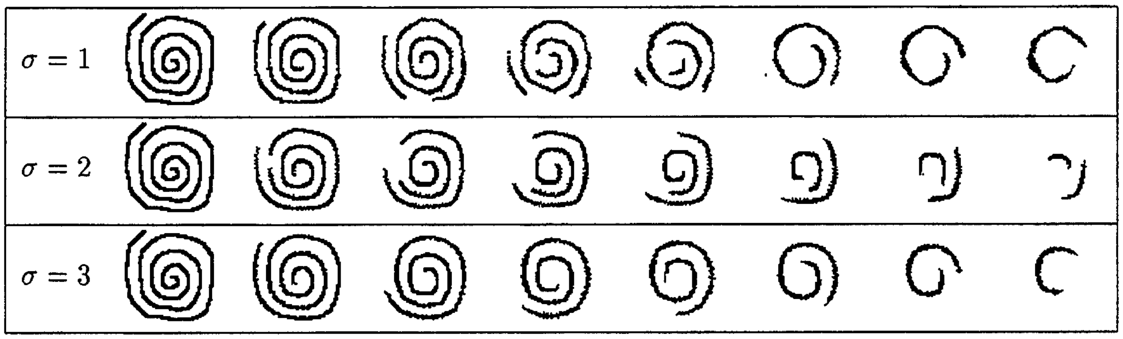

Figure 3: The effect of varying the Gaussian scale parameter, $\sigma$, on discrete contour evolution.

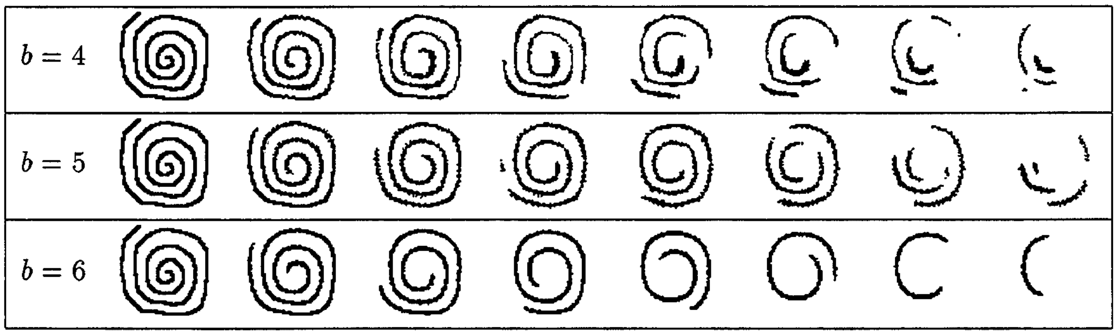

Figure 4: The effect of varying the number of bits, $b$, used in the computations.

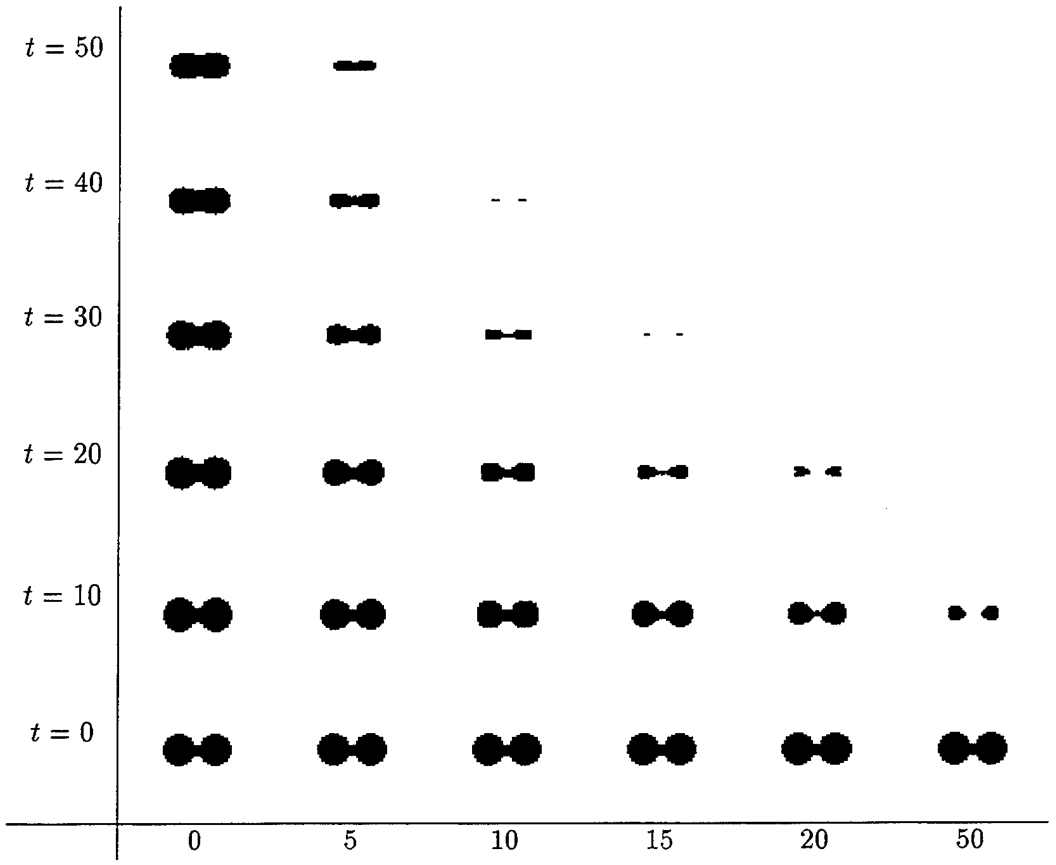

Figure 5: Discrete, linear curvature-based contour evolution of a dumbbell. The bottom row corresponds to $t = 0$, and the $x$-axis shows values of $\beta_0$ (with $\beta_1 = 1$).

## 4 Experiments

### 4.1 Linear Functions of Curvature

Linear functions of curvature,

$$f(P) = \beta_0 + \beta_1 \kappa(P),$$

were used in our initial experiments. Such functions were extensively studied in [2], and represent reaction-diffusion processes. Recall that $\beta_0$ corresponds to constant motion of the boundary curve, and $\beta_1 k$ corresponds to curvature-dependent motion. When $\beta_1 = 0$, the rate of motion is constant; negative $\beta_0$ produces morphological dilation and positive $\beta_0$ produces morphological erosion. When $\beta_0 = 0$, the rate of motion at a cell is proportional to the curvature of the boundary at that cell.

The results of applying this process to various test shapes for various values of $\beta_0$ (using $\beta_1 = 1$) are shown in Figures 5-9. Each test shape was scaled to 64 by 64 pixels, which matched the dimensions of the spiral test shape used in Figures 2-4. In the first example (Figure 5), the "dumbbell" splits into two pieces as constant motion is added to the process. Similarly, the fingers split from the "hand" as $\beta_0$ increases (Figure 6). In the spiral example (Figure 7), small values of $\beta_0$ cause the spiral to disintegrate. In the "bird" example (Figure 8), when the constant term exceeds the curvature of the negatively curved circular hole, the hole dilates. In

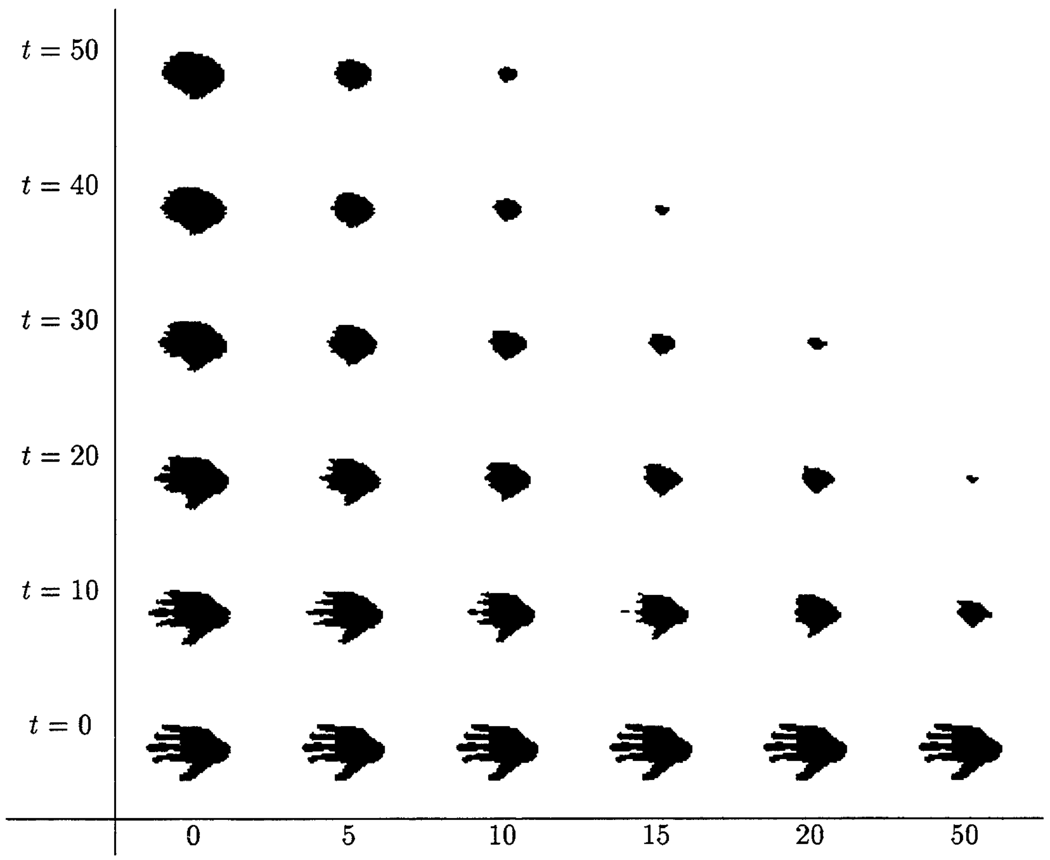

Figure 6: Discrete, linear curvature-based contour evolution of a hand. The bottom row corresponds to $t = 0$, and the $x$-axis shows values of $\beta_0$ (with $\beta_1 = 1$).

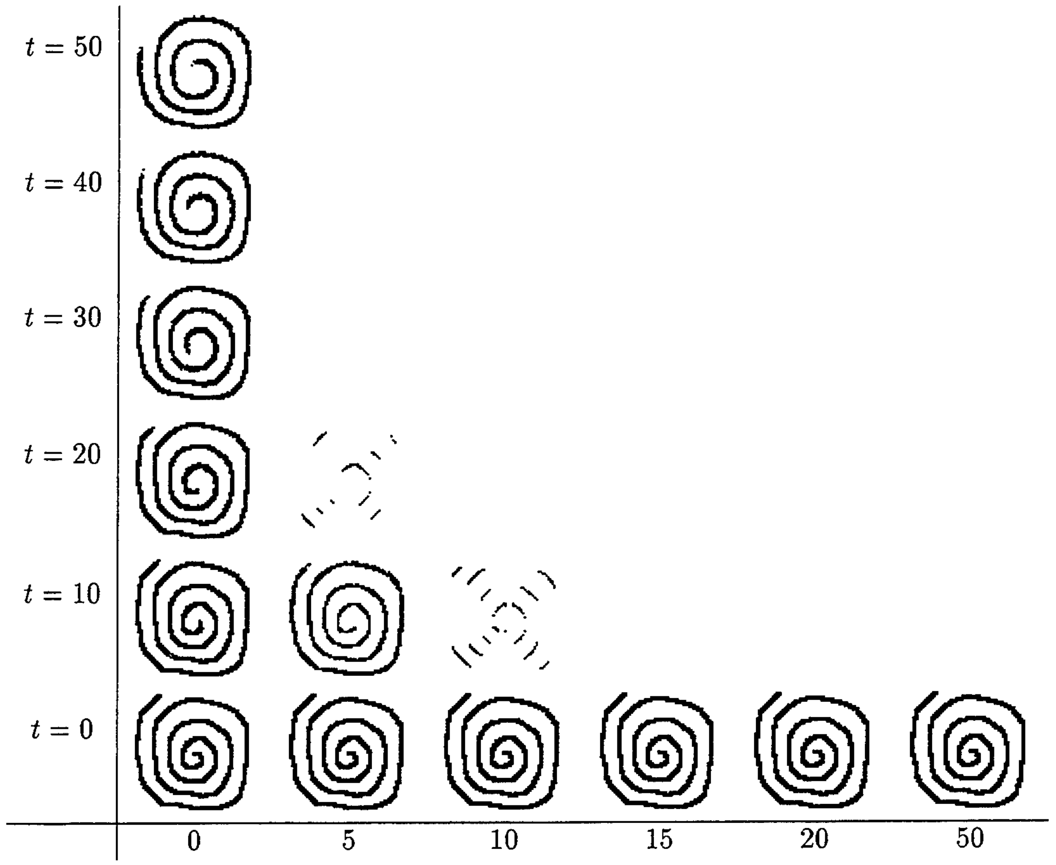

Figure 7: Discrete, linear curvature-based contour evolution of a spiral. The bottom row corresponds to $t = 0$, and the $x$-axis shows values of $\beta_0$ (with $\beta_1 = 1$).

the final "coffee bean" example (Figure 9), adding a constant term causes the coffee beans to separate. These results are very similar to those achieved by continuous methods [2].

## 4.2 Nonlinear Functions of Curvature

Most of the past work on curvature-dependent contour evolution has assumed linear dependency on curvature, but one recent paper [6] also considered nonlinear functions of curvature based on using only non-negative, or only non-positive curvature values (i.e., the contour does not change at points where its curvature has the wrong sign). As we shall see in Section 4.2.1, our digital approach yields results closely similar to those obtained in [6].

Our digital contour evolution process is also not restricted to linear functions of the contour's curvature. In this section we define three simple types of nonlinear functions of curvature, and show the results of applying these functions to various test shapes. The first type consists of "rectified" functions, where either negative or positive curvatures are suppressed. The second type consists of "thresholded" functions, where either low or high curvatures are suppressed. The last group of nonlinear functions we consider are based on the absolute value of the curvature.

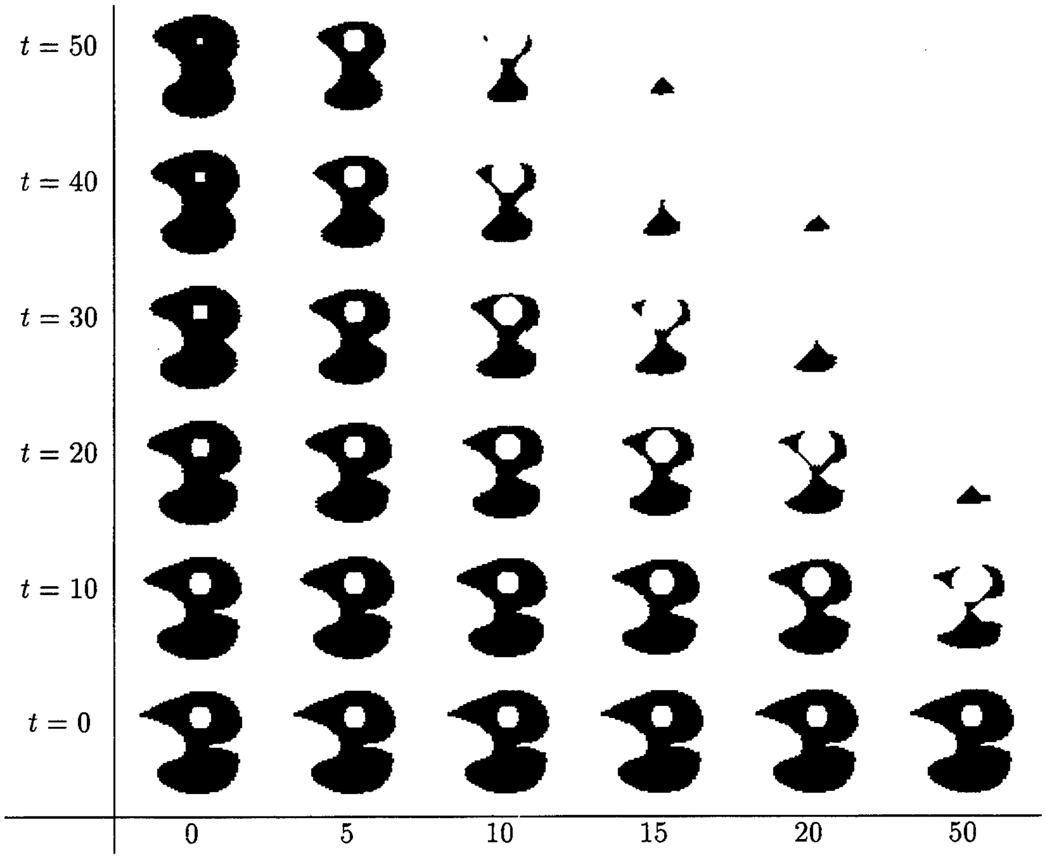

Figure 8: Discrete, linear curvature-based contour evolution of a bird. The bottom row corresponds to $t = 0$, and the $x$-axis shows values of $\beta_0$ (with $\beta_1 = 1$).

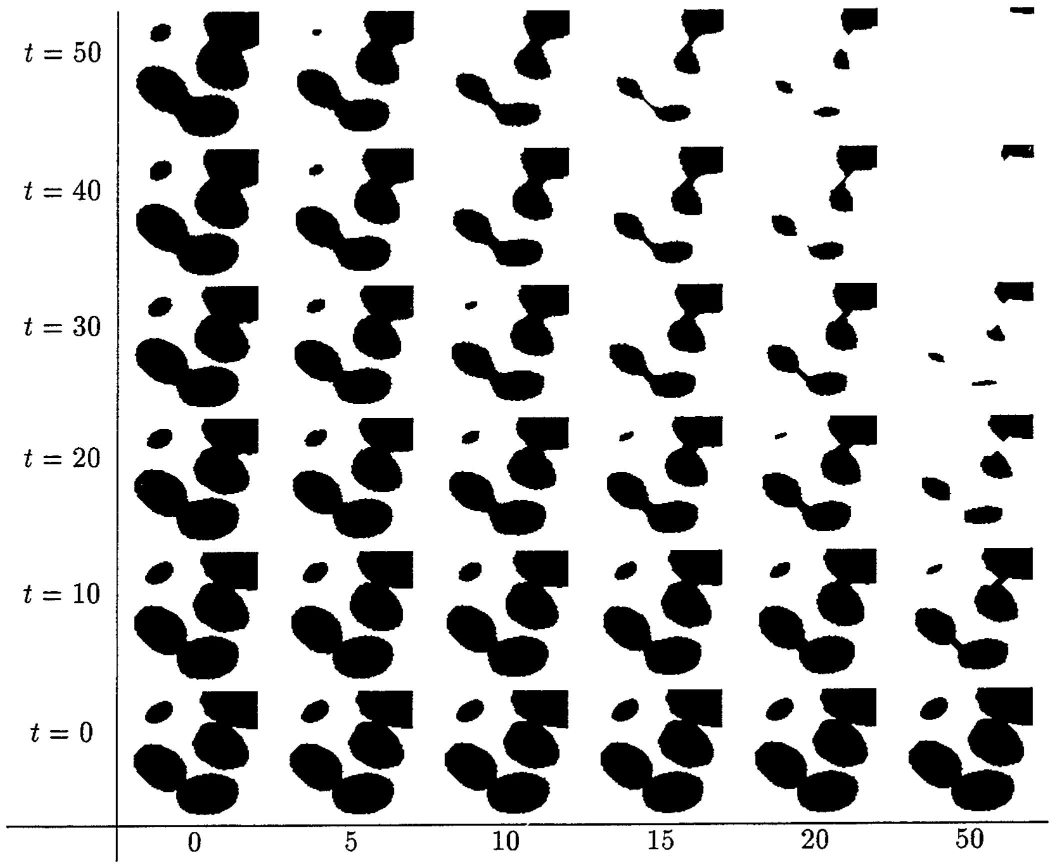

Figure 9: Discrete, linear curvature-based contour evolution of a windowed image of coffee beans. The bottom row corresponds to $t = 0$, and the $x$-axis shows values of $\beta_0$ (with $\beta_1 = 1$).

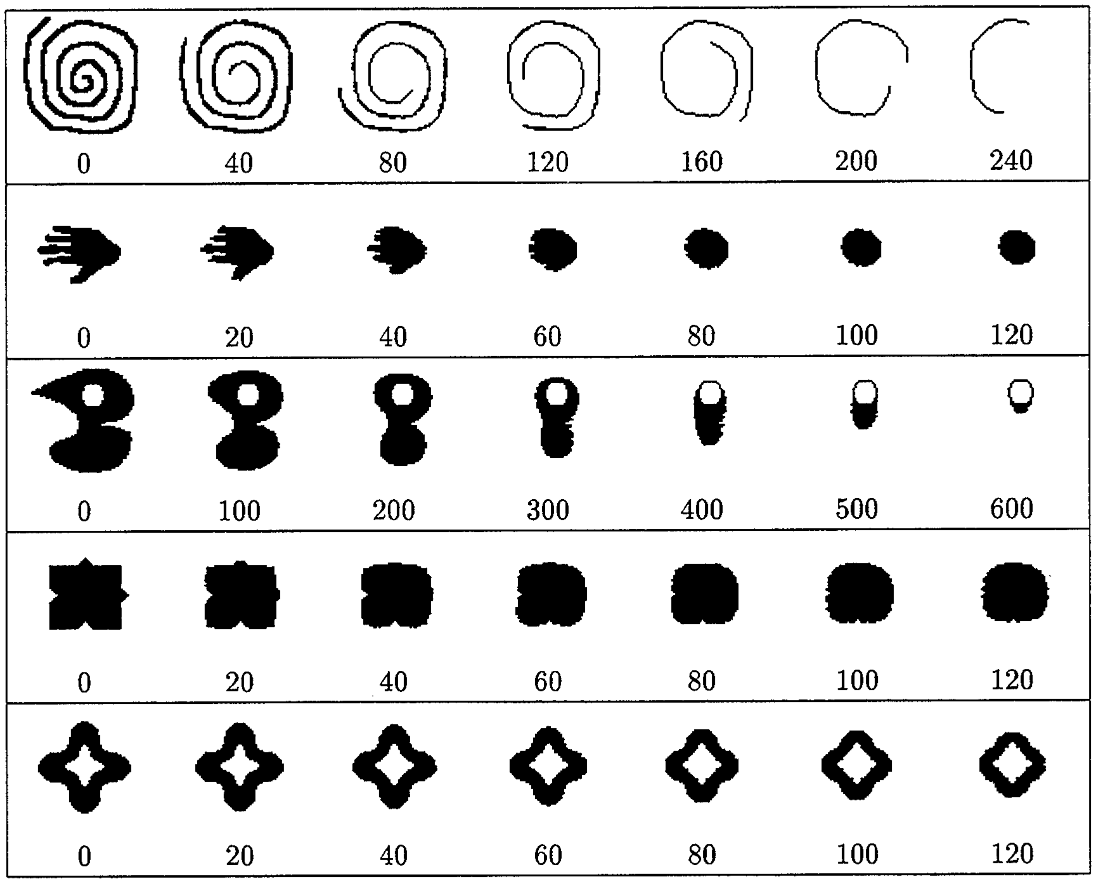

Figure 10: Erosion proportional to positive curvature.

### 4.2.1 Rectified Curvature

1. Positive Curvature:

$$f(P) = \begin{cases} 0 & \kappa(P) < 0 \\ \kappa(P) & \kappa(P) \geq 0 \end{cases}$$

The results of applying this function to three of the test shapes, as well as two shapes used in [6], are shown in Figure 10. In these examples, the rate of erosion is determined by positive curvature; there is no dilation. The examples are shown over different time periods to illustrate the long-term behavior of the process.

In the first example, the spiral shortens due to its high positive curvature at both ends. In addition, the spiral gradually "thins" to one pixel thick. The thinning begins at the center of the spiral (which is more strongly curved) and continues outward. At one pixel thick, there are two curvatures at each non-end cell since it is on two boundaries. The average curvature at these cells is zero, which stops the thinning and prevents the spiral from breaking.

In the second ("hand") example, the fingers of the hand erode and eventually disappear; the remaining disc then slowly erodes.

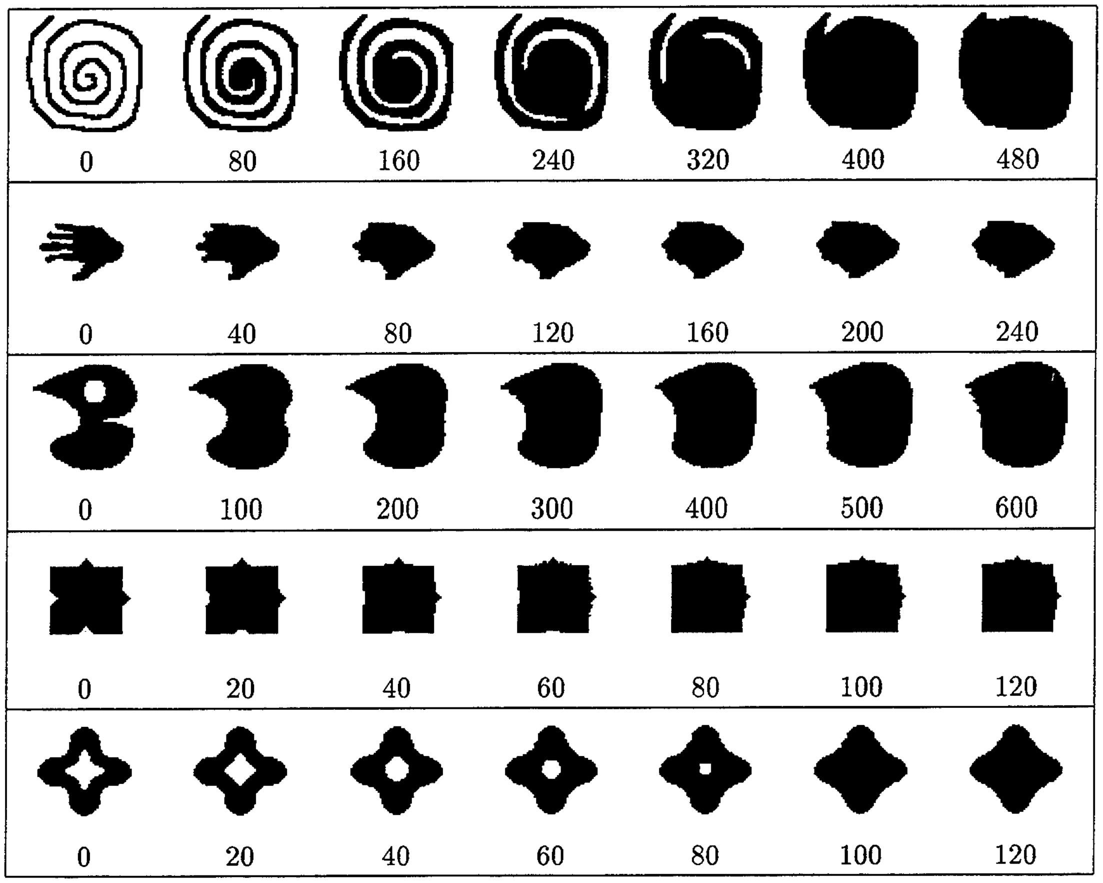

Figure 11: Dilation proportional to negative curvature.

In the third ("bird") example, the bird's beak quickly erodes relative to the other convexities. Erosion stops at the border of the bird's eye, leaving a one-pixel-thick circle.

In the fourth and fifth examples, which should be compared to Figures 3 and 4 of [6], the process removes the positive notches in the square, and erodes the "rosette" into a hollow square.

2. Negative Curvature:

$$f(P) = \begin{cases} \kappa(P) & \kappa(P) \leq 0 \\ 0 & \kappa(P) > 0 \end{cases}$$

The results of applying this function to the same five test shapes are shown in Figure 11. In these examples, the rate of dilation is determined by negative curvature; there is no erosion.

In the spiral example, dilation begins near the center of the spiral. The process eventually reaches a "steady state" configuration, essentially the spiral's (digital) convex hull.

In the hand example, the concavities between the fingers fill in, and eventually a convex hull is produced. The bird example shows that holes are filled in by this process.

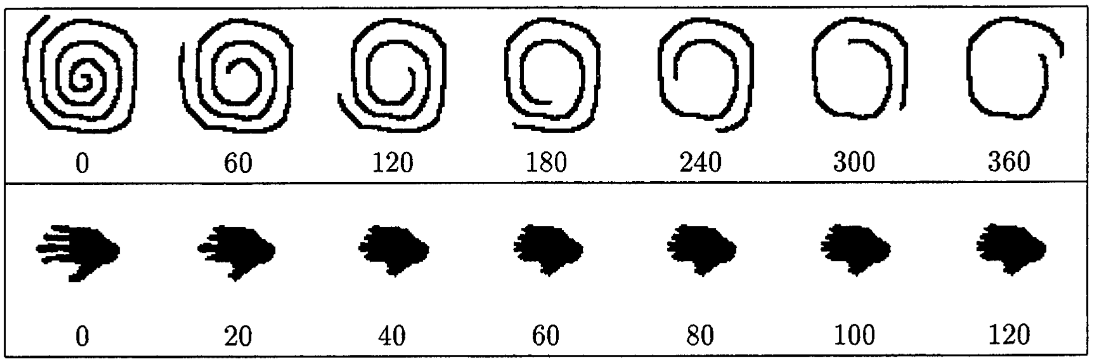

Figure 12: Using only high absolute curvature.

Similarly, in the fourth and fifth examples, the process removes the negative notches in the square, and gradually fills in the concavities between the notches; it also fills the hole in the rosette.

#### 4.2.2 Thresholded Curvature

1. High Curvature:

$$f(P) = \begin{cases} 0 & |\kappa(P)| < \text{MAX}/2 \\ \kappa(P) & |\kappa(P)| \geq \text{MAX}/2 \end{cases}$$

   where MAX is the initial maximum curvature. The results of applying this function to the spiral and hand shapes are shown in Figure 12. Note that this function involves both dilation and erosion.

   In the spiral example, the spiral shrinks at both ends, where its curvature is high. The dilation that occurs along the inner edge of the spiral is quickly re-eroded, and the thickness of the spiral does not change.

   In the hand example, the process removes the highly curved features of the hand; the fingertips are removed and the deep concavities between the fingers are filled in.

2. Low Curvature:

$$f(P) = \begin{cases} \kappa(P) & |\kappa(P)| \leq \text{MAX}/2 \\ 0 & |\kappa(P)| > \text{MAX}/2 \end{cases}$$

   where MAX is the initial maximum curvature. The results of applying this function to the same two test shapes are shown in Figure 13. Since high positive or negative curvatures are ignored, this process requires a relatively large number of iterations until steady state.

   In the spiral example, the combination of erosion along the spiral's outer edge and dilation along its inner edge produces a "loosening" and "straightening". Since the ends of the spiral remain fixed, the dilations eventually result in the spiral intersecting itself at the center; this creates tiny "holes." Since the holes are highly negatively curved, they are not filled in. Thus the process leads to a pattern of tiny holes near the center. As the spiral

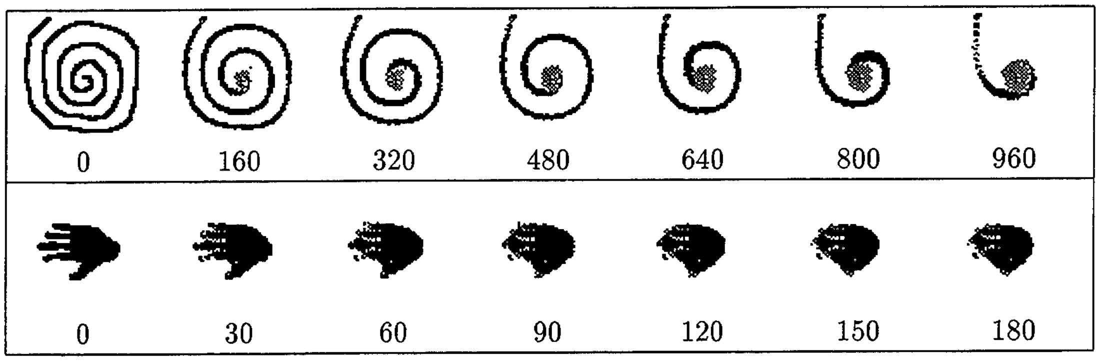

Figure 13: Using only low absolute curvature.

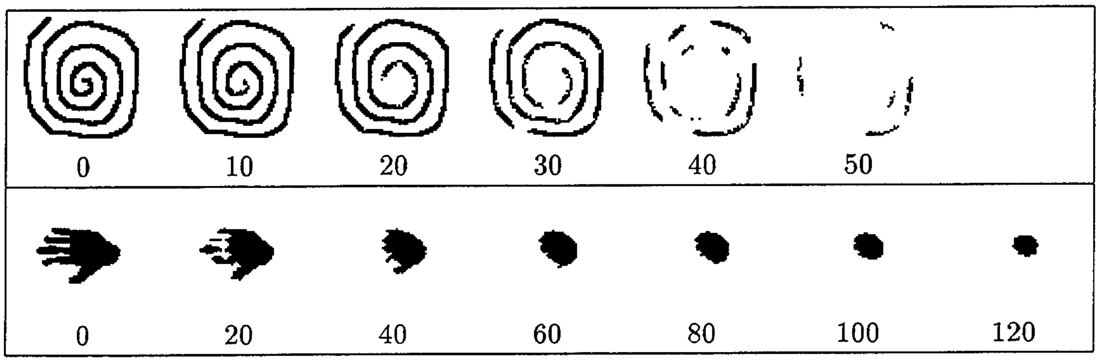

Figure 14: Erosion proportional to absolute curvature.

straightens, erosion begins to occur on both sides near the outer end, which causes small pieces of the spiral to eventually break off. These pieces are highly curved and therefore do not erode.

In the hand example, the positively curved fingers erode on both sides near their tips, which eventually separate from the hand. In addition, the negatively curved areas near the bases of the fingers merge together. This process does not appear to produce a meaningful decomposition.

### 4.2.3 Absolute Curvature

1. Absolute Value of Curvature:

$$f(P) = |\kappa(P)|$$

The results of applying this function to the two test shapes are shown in Figure 14. In these examples, the rate of erosion is determined by the absolute value of the curvature; there is no dilation.

In the spiral example (note the time scale), the spiral rapidly disintegrates along both its exterior and interior sides. It breaks first where its negative curvature is highest.

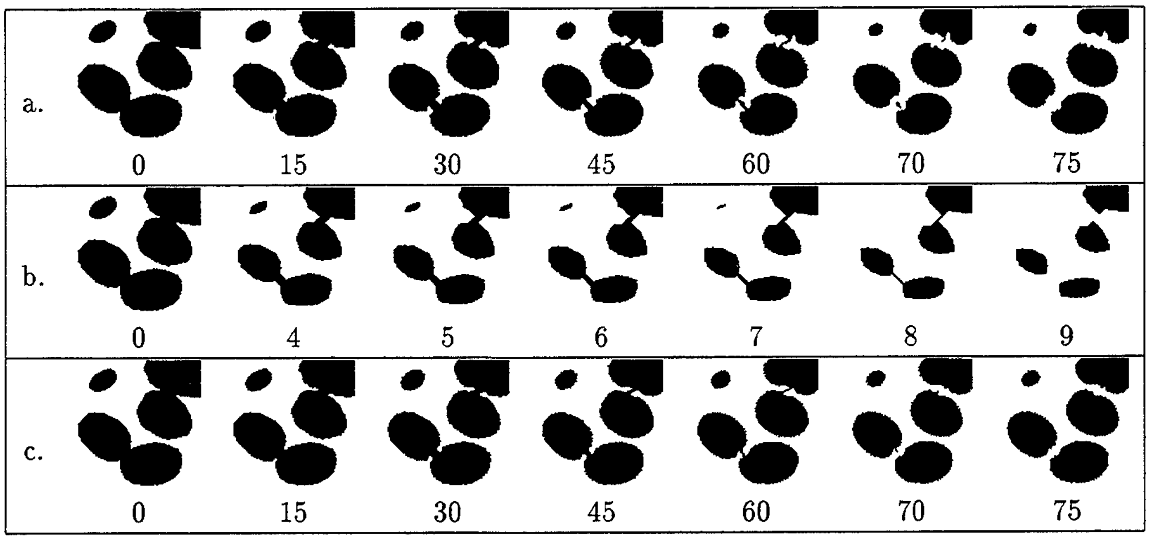

Figure 15: Separating "coffee beans". (a) Absolute value of curvature. (b) Morphological erosion. (c) A variation on absolute value.

In the hand example, the concavities between the fingers deepen as the fingertips erode. The three middle fingers break off from the hand; this is apparently caused by the smoothing, since the near-zero curvatures on the sides of the fingers near the base are offset by the nearby concavity. The disconnected fingers then rapidly erode.

To illustrate a possible use for the absolute value function, we also applied it to the "coffee bean" example shown in Figure 15 (see also Figure 9). Here the deep concavities are flanked by relatively low positively curved sides. Using absolute value thus produces a "cutting" process that separates the beans: the concavities deepen while the positively curved perimeter remains relatively unchanged, as we see in Figure 15a.

Separation of the beans could also be achieved by morphological erosion (Figure 15b):

$$f(P) = \beta_0,$$

but the use of absolute value has two advantages. First, the absolute value process removes 60% fewer pixels before the beans split. Second, the absolute value process preserves the small bean, while morphological erosion removes it.

2. A variation:

The "cutting" process can be improved further by using a variation on absolute value which erodes at a faster rate at local minima of the curvature, e.g.,

$$f(P) = \begin{cases} |\kappa(P)| & \text{if } \kappa(P) \text{ is a local minimum} \\ \frac{2}{3}|\kappa(P)| & \text{otherwise} \end{cases}$$

The result of applying this function to the beans is shown in Figure 15c. This process removes 76% fewer pixels than morphological erosion.

## 5 Discussion

This paper has described a discrete method of curvature-dependent contour evolution. The method is based on the 8-connected chain code of the contour, and on discrete curvature measurement. Discrete estimation of the curvature of the boundary of a real shape from its digitization is difficult; the more advanced methods take into account the multiple feature sizes of the shape. Our method of curvature measurement takes differences between neighbors, uses truncated Gaussian smoothing at a computational precision of only six bits, and takes into account the nonisotropicness of the grid. Using linear functions of the discrete curvature for contour evolution gives results similar to those produced by continuous solutions to the reaction-diffusion equation—in other words, our method generates similar evolutionary sequences. We have also investigated several simple non-linear functions of the digital curvature; contour evolution processes based on these functions can behave in qualitatively different ways. For example, using absolute curvature (guided by negative curvature extrema) results in a "cutting" process.

## References

[1] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, 1984.

[2] B. Kimia, A. Tannenbaum, and S.W. Zucker. Entropy scale space. In C. Arcelli, L.P. Cordella, and G.S. di Baja, editors, *Visual Form*, pages 333–344. Plenum Press, New York, 1992.

[3] B. Kimia, A. Tannenbaum, and S.W. Zucker. Shapes, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

[4] J.J. Koenderink. *Solid Shape*. MIT Press, Cambridge, MA, 1990.

[5] M. Leyton. A process grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[6] R. Malladi and J.A. Sethian. Image processing: Flows under min/max curvature and mean curvature. *Graphical Models and Image Procesing*, 58:127–141, 1996.

[7] W. Richards and D.D. Hoffman. Codon constraints on closed 2D shapes. *Computer Vision, Graphics and Image Processing*, 31:265–281, 1985.

[8] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. Academic Press, New York, 1982.

[9] J.A. Saghri and H. Freeman. Generalized chain codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:533–539, 1981.

[10] E. Saund. *The Role of Knowledge in Visual Shape Representation.* PhD thesis, MIT, Cambridge, MA, 1988.

[11] B. Shahraray and D.J. Anderson. Uniform resampling of digitized contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:674–681, 1985.

[12] S. Thompson, A. Rosenfeld, and S. Iwata. Discrete curvature-based contour evolution. Technical Report CS-TR-3593, Unversity of Maryland, College Park, MD, 1995.

[13] M. Worring and A.W.M. Smeulders. Digital curvature estimation. *CVGIP: Image Understanding*, 58:366–382, 1993.

| REPORT DOCUMENTATION PAGE | *Form Approved* *OMB No. 0704-0188* |
|---|---|

| | | |
|---|---|---|
| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** August 1997 | **3. REPORT TYPE AND DATES COVERED** Technical Report |

| | |
|---|---|
| **4. TITLE AND SUBTITLE** Discrete, Nonlinear Curvature-Dependent Contour Evolution | **5. FUNDING NUMBERS** N00014-95-1-0521 |
| **6. AUTHOR(S)** Scott Thompson and Azriel Rosenfeld | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Center for Automation Research<br>University of Maryland<br>College Park, MD 20742-3275 | **8. PERFORMING ORGANIZATION REPORT NUMBER** CAR-TR-868<br>CS-TR-3825 |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** Office of Naval Research<br>800 North Quincy Street, Arlington, VA 22217-5660<br><br>Advanced Research Projects Agency<br>3701 North Fairfax Drive, Arlington, VA 22203-1714 | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** | |
| **12a. DISTRIBUTION/AVAILABILITY STATEMENT** Approved for public release.<br>Distribution unlimited. | **12b. DISTRIBUTION CODE** |

**13. ABSTRACT** *(Maximum 200 words)*

There has been much recent interest in curvature-dependent contour evolution, particularly when the resultant family of contours satisfies the heat (diffusion) equation. Modeling the evolution of a shape's boundary as a real-valued solution to the reaction-diffusion equation has been shown to be useful for shape decomposition [3]. This approach to contour evolution involves solving a partial differential equation (PDE), is computationally demanding, and must deal with the problem of singularities. In this paper, we describe a low-precision discrete method of contour evolution, based on the 8-connected chain code of the contour, that performs analogously to PDE-based methods and avoids the singularity problem. (Preliminary work along these lines was described in [12].) Our discrete method is not limited to linear functions of curvature; we give several examples of contour evolution processes that depend nonlinearly on curvature, including examples studied in [6], and illustrate their possible uses.

| | | | |
|---|---|---|---|
| **14. SUBJECT TERMS** Contour evolution, Curvature, Shape | | | **15. NUMBER OF PAGES** 19 |
| | | | **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT** UNCLASSIFIED | **18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED | **19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED | **20. LIMITATION OF ABSTRACT** UL |