4.15

Shape Smoothing and PDEs

Frédéric Guichard
DOLabs, France

Lionel Moisan
MAP5-Université René Descartes, France

Jean-Michel Morel
CMLA-ENS Cachan, France

1 Principles for Shape Smoothing

1.1 Shape Scale Spaces

We denote by $u(\mathbf{x})$ the gray level value of an image u at point $\mathbf{x}$. We take two different definitions of a shape: for the first definition, a shape simply is a subset $\mathbf{X}$ of the image plane. Now, for sake of simplicity, we always assume that $\mathbf{X}$ is connected; otherwise we count as many shapes as connected components. A connected set of the plane can be *simply connected* or not. In intuitive terms, a set is simply connected if it has no hole. It is then entirely described as the set of points surrounded by a simple closed curve (a Jordan curve). So the analysis and recognition of a shape boils down to the analysis and recognition of a Jordan curve. In case the set $\mathbf{X}$ has holes, each hole is to be considered as a shape and is again surrounded by a Jordan curve. So we can describe a connected set $\mathbf{X}$ by its exterior Jordan curves and by the list of the Jordan curves surrounding each one of its holes. For instance, the letter "O" considered as a set of black points has a single hole and is entirely described by two Jordan curves. The same is true for the letter "A"; the exterior Jordan curve is not convex in that case. The letter "B" is entirely described by three Jordan curves (and their relative positions). So we can consider in a wide degree of generality that shape recognition must be based on Jordan curve recognition.

In many shape recognition algorithms, shapes are smoothed before the encoding and recognition tasks. The main reason why they have to be smoothed is to achieve robustness and compactness of their encoding. Local details of the shape may look different while their overall aspect is the same. So the aim of shape smoothing is to remove spurious details and keep the essential, robust features on which recognition can be based.

Now, one can smooth a shape more or less, and this is why a scale parameter $t \geq 0$ is usually associated with the smoothing task. A shape smoothing method is therefore usually called a *shape scale space*. We shall list four principles that a shape scale space (identified with a curve scale space) should satisfy. A shape scale space must, according to these principles, be a *curvature motion equation.* This term will be explained later. For some of the principles discussed here, it will be useful to consider a shape in turn as a Jordan curve and as the bounded set $\mathbf{X}$ surrounded by the Jordan curve. So we call *shape* any simply connected bounded closed set $\mathbf{X}$ whose boundary is a Jordan curve of $\mathbb{R}^2$. We denote by $T_t(\mathbf{X})$ the *shape* $\mathbf{X}$ *smoothed at scale t.* We call shape scale space any family of smoothing operators $(T_t)_{t\geq 0}$ acting on shapes and set $\mathbf{X}(t) = T_t(\mathbf{X})$.

1.2 Five Principles

We simply list the main invariance and robustness principles of the considered algorithms. All bibliographic details will be given in the last section of this chapter.

Pyramidal Principle

We suppose that the operators T_t satisfy a pyramidal assumption, that is, there exist transition operators $T_{t,s}$ such that $T_t = T_{t,s}T_s$ for any $t > s$. The reason for this assumption is a question of computational low complexity: A shape smoothed at scale $t > s$ can be directly computed from the shape smoothed at scale s. It also corresponds to the idea of a progressive pruning of the shape from s to t, where no information is added from s to t; there is only a simplification. It can be proved [3] that this principle can be reduced to a semigroup property $T_{t+s} = T_s \circ T_t$, which implies $T_{ns} = (T_s)^n$ and therefore that the smoothing is an iterative process, thus computationally simple and efficient.

Inclusion Principle

If a shape $\mathbf{X}$ in enclosed in another one, $\mathbf{X} \subset \mathbf{Y}$, we always require that this inclusion be preserved by smoothing. The reason for this is an obvious compatibility requirement of the smoothing with inclusion: $\mathbf{X}$ might be a hole of $\mathbf{Y}$ and, after smoothing, we want this inclusion to be preserved. Otherwise, no reconstruction of the shape after smoothing of its external and internal Jordan curves would be possible. So our principle states

$$\mathbf{X} \subset \mathbf{Y} \Rightarrow T_t(\mathbf{X}) \subset T_t(\mathbf{Y})$$

Local Inclusion Principle

This principle is a stronger, localized version of the inclusion principle. We denote by $B(\mathbf{x}, r)$ the open disk with center $\mathbf{x}$ and radius r, and for a shape $\mathbf{X}$, by $\partial\mathbf{X}$ the boundary of $\mathbf{X}$. Assume that $\mathbf{X}$ and $\mathbf{Y}$ are two compact shapes and that for some $\mathbf{x} \in \partial\mathbf{Y}$ and some $r > 0$, one has $\mathbf{X} \cap B(\mathbf{x}, r) \subset \mathbf{Y} \cap B(\mathbf{x}, r)$. Assume further that the inclusion is strict in the sense that $\partial\mathbf{X}$ and $\partial\mathbf{Y}$ can only meet at $\mathbf{x}$. Then we shall say that the shape $\mathbf{X}$ is included in shape $\mathbf{Y}$ around $\mathbf{x}$. We say that the scale space T_t satisfies a *shape local inclusion principle,* if for all $\mathbf{X}$ and $\mathbf{Y}$ subsets of $\mathbb{R}^2$ such that $\mathbf{X}$ is included in $\mathbf{Y}$ around $\mathbf{x}$, then for h small enough,

$$T_h(\mathbf{X}) \cap B(\mathbf{x}, r) \subset T_h(\mathbf{Y}) \cap B(\mathbf{x}, r).$$

Figure-Background Reversal Principle

We can associate with a shape $\mathbf{X}$ its complementary set $\mathbf{X}^c$, which also is a shape. Considering the shape or considering its complementary set is a perceptual decision, which depends on context: For instance a wall with windows is a shape with holes, and the windows are themselves shapes complementary to the wall. So one usually looks for smoothing operators compatible with this change of viewpoint, namely

$$T_t(\mathbf{X}^c) = (T_t(\mathbf{X}))^c$$

All smoothing operators we shall consider will satisfy this principle.

Isotropy

We say that the scale space T_t is isotropic if T_t commutes with any translation and rotation. This is an obvious requirement for shape analysis, as we wish to recognize shapes independently of their position and orientation in the image.

Affine Invariance of Shape Scale Space

We say that a shape scale space T is affine invariant if there exists a C^1 function $t'(t, A) \geq 0$ defined for any 2×2 matrix A and any real number $t \geq 0$, such that $AT_{t'(t,A)} = T_t A$. In the case where we restrict this last relation to zooms, we shall say that the scale space is scale invariant.

Let T_t be a shape scale space satisfying the four shape-analysis principles (pyramidal architecture, shape local inclusion, figure-background reversal, isotropy and regularity). Then it can be proved, under some technical additional regularity assumption, that the scale space T_t evolves any Jordan curve $\mathbf{x}_0(p)$ into a Jordan curve $\mathbf{x}(t,p)$ satisfying

$$\frac{\partial \mathbf{x}}{\partial t}(t,p) = g(|\mathrm{Curve}\,\mathbf{x}\,(t,p)|)\ \mathbf{n}(t,p) \tag{1}$$

where g is a nondecreasing function, $\mathbf{n}(t,p)$ is the signed unit normal (always pointing toward the concavity) to the curve $p \to \mathbf{x}(t,p)$ at point $\mathbf{x}(t,p)$, and $\mathrm{Curve}\,\mathbf{x}(t,p)$ denotes the curvature vector (second derivative with respect to arclength) of the curve at point $\mathbf{x}(t,p)$: If s is an arclength parameter for a curve $s \mapsto \mathbf{y}(s)$, then the curvature vector of this curve at point $\mathbf{y}(s)$ is

$$\text{Curve}\,\mathbf{y}(s) = \frac{\partial^2 \mathbf{y}}{\partial s^2}(s)$$

With obvious abbreviations, we shall simplify (1) into

$$\frac{\partial \mathbf{x}}{\partial t} = g(|\text{Curve}\,\mathbf{x}|)\,\mathbf{n}(\mathbf{x}) \tag{2}$$

The norm of the curvature vector Curve $\mathbf{x}(t,s)$ is the inverse of the radius of the osculatory circle to the curve. Thus, the more curved the curve is, the fastest the motion is. This is why a curvature motion like (2) leads to a simplification of the curve preserving its main features. If, in addition to the above-mentioned principles, the scale space has to be affine invariant, then the equation of the scale space must be

$$\frac{\partial \mathbf{x}}{\partial t} = |\text{Curve}\,\mathbf{x}|^{\frac{1}{3}}\,\mathbf{n}(\mathbf{x}) \tag{3}$$

This equation defines the so-called *affine scale space.*

Most algorithms we analyze in the next section can and will be interpreted as a curvature equation of the preceding kind and, as we shall see, the best under any aspect implements the affine scale space. These algorithms are freely available in the MegaWave2 software [1].

2 Algorithm 1: Dynamic Shape

2.1 Description

Koenderink and Van Dorn [31] define a "shape" as a closed subset X of $\mathbb{R}^N$. They associate to a shape X its characteristic function $\mathbb{1}_X$, defined by $\mathbb{1}_X(\mathbf{x}) = 1$ if $\mathbf{x} \in X$, $= 0$ otherwise.

They propose to smooth a shape X by applying directly the heat equation to its characteristic function $\mathbb{1}_X$. For that purpose, they introduce $u(t,\mathbf{x})$ a function of $t \geq 0$ (the scale of smoothing), and $\mathbf{x}$ (the spatial coordinates), such that

$$u(0,\cdot) = \mathbb{1}_X(\cdot) \quad \text{and} \quad \forall t \geq 0, \quad \frac{\partial u}{\partial t}(t,\cdot) = \Delta u(t,\cdot)$$

Of course, the solution $u(t,\mathbf{x})$ is no more a characteristic function. So, the authors define the evolved shape at scale t by thresholding $u(t,\mathbf{x})$.

$$X(t) = \left\{\mathbf{x}, u(t,\mathbf{x}) \geq \frac{1}{2}\right\} \tag{4}$$

The value $\frac{1}{2}$ is chosen by an obvious requirement, namely that the smooth version of a half plane (or space) is the half plane itself.

2.2 Implementation

The implementation of the dynamic shape is mostly the implementation of the heat equation. X is a discrete set of pixels $\mathbf{x} = (i,j)$, and image u_0 is defined over the (i,j). We recall that the convolution of u_0 by a kernel M is defined by

$$(M \star u_0)(i,j) = \sum_l \sum_m M(l,m) u_0(i-l, j-m),$$

where (l,m) goes over points where M is not zero. Implementation can be performed as follows:

1. Set $u_0(\mathbf{x}) = \mathbb{1}_X(\mathbf{x})$.
2. For a chosen scale t, construct $u(t,\mathbf{x})$ solution of the heat equation with initial condition u_0. One possibility is to construct directly $u(t,\mathbf{x})$ as the result of the convolution by a Gaussian kernel, that is

$$u(t,\mathbf{x}) = (G_t \star u_0)(\mathbf{x})$$

$$\text{with} \quad G_t(\mathbf{x}) = \frac{1}{4\pi t}\exp\left(-\frac{|\mathbf{x}^2|}{4t}\right)$$

Processing can be performed in spatial or Fourier domain.

A probably easier (approximate) solution is to convolve several times u_0 by a normalized, symmetric and separable kernel M. That is with an M such that $M(i,j) = M_1(i) \times M_1(j)$, with M_1 being a one-dimensional kernel, symmetric with respect to $(0,0)$, and whose sum is equal to 1. Writing

$$s(M) := \frac{1}{4}\sum_{i,j}(i^2 + j^2)M(i,j),$$

n convolutions by M approximate a convolution by $G_{ns(M)}$, so that choosing M and n such that $ns(M) = t$ yields the approximate solution

$$u(t,\mathbf{x}) = (M^n \star u_0)(\mathbf{x})$$

For example, one can choose $M_1(x) = \frac{1}{4}[1\ 2\ 1]$, and therefore

$$M(i,j) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Since $s(M) = 1$, n convolutions by this kernel approximates the solution of the heat equation at scale $t = n$.

3. Threshold the function $u(t, \mathbf{x})$ according to (4) to get the smoothed shape $X(t)$.

2.3 Properties

The *dynamic shape algorithm* inherits some of the geometrical properties of the heat equation. First, it is theoretically invariant by rotation. However, full invariance by rotation is difficult to obtain in practice due to the work on a discrete grid. It is also covariant to change of scale: one has, defining a "zoomed" version of X by $Z_\lambda(X) = \{\lambda\mathbf{x},\ \mathbf{x} \in X\}$,

$$T_{\lambda t}(Z_\lambda(X)) = Z_\lambda(T_t(X))$$

It is not affine invariant (in theory and in practice), since the heat equation is not.

The *dynamic shape algorithm* satisfies the inclusion principle. This is merely due to the monotony of the heat equation (or of the convolution with positive kernels) and to the monotony of the threshold operation. It is not local (and therefore does not satisfy the local inclusion principle), and indeed generates **nonlocal interactions.** Take two close disks $D(\mathbf{x}_0, 1)$ and $D(\mathbf{x}_1, 1)$, with $|\mathbf{x}_0 - \mathbf{x}_1| = 1 + \varepsilon$. Then the evolution of the union of both disks, considered as a single shape, is quite different from the evolution of each disk separately (see Fig. 1). As well, intricated shapes will not be analyzed separately and be smoothed into a single shape for large scales (see Fig. 2).

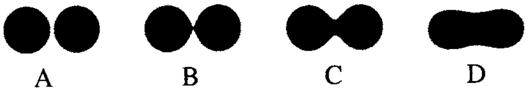

FIGURE 1 Nonlocal interactions with the dynamic shape method. Two close disks (**A**) interact as scale grows and create a qualitatively different and new shape (**C, D**). The change of topology, at the scale where both shapes merge into one (**B**), entails the appearance of a singularity (a cusp) on the shape(s) boundaries.

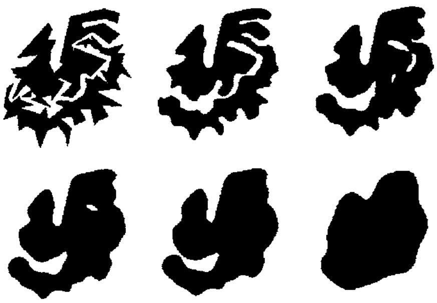

FIGURE 2 Nonlocal behavior of shapes with the dynamic shape method. **Top left:** initial image, made of two irregular shapes. **Left to right and top to bottom:** smoothing with increasing scales with the dynamic shape algorithm. Notice how, the convolution being made with Gaussians of increasing variance, the shapes merge more and more. We do not have a separate analysis of both shapes but a "joint analysis" of both. This joint analysis depends a lot on the initial distance between the two shapes.

As another consequence, **singularities** of the orientation and curvature of the boundary of the shape may appear with the evolution. Thus, the smoothing creates new salient features!

At last, the *dynamic shape algorithm* is not pyramidal, in the sense that $X(t)$ cannot be deduced from $X(s)$ for any $s > 0$. However, it is related to a pyramidal construction: for each time t, $X(t)$ is obtained by thresholding from $u(t, \cdot)$, solution of the heat equation, which is pyramidal.

3 Algorithm 2: Iterated Weighted Median Filter

3.1 Description

The preceding algorithm is suffering from nonlocality and nonpyramidality. Remarking that the smaller the scale t is, the more local the effect of the algorithm will be, one can overpass the problem by iterating the dynamic shape with a small-scale step, that is, by *alternating* a small-scale linear convolution (by a Gaussian) with the thresholding. Roughly, the iteration process generates the pyramidality, and the small-scale step ensures the locality.

Now, it is easily seen that applying to a characteristic function a linear-positive convolution followed by a thresholding is equivalent to applying a weighted median filter. Iterating weighted median filter to characteristic function has been proposed by Merriman, Bence, and Osher [42] as an approximation of the "mean curvature motion."

3.2 Implementation

At this point, the implementation will be discrete in term of scale. Let us define k the discrete version of the scale t. The scale step will be denoted by ε. The link between discrete scale k and scale t is therefore $t = k\varepsilon$. We wish to define X_k the smoothed version of X at discrete scale k.

1. Start with $X_0 = X$
2. Iterate:
   (a) Set $u_k(\mathbf{x}) = \mathbb{1}_{X_k}(\mathbf{x})$.
   (b) Set $v(k+1, \mathbf{x}) = (M \star u(k, \cdot))(\mathbf{x})$, M being a normalized symmetric kernel with $s(M) = \varepsilon$ (see Section 2.2).
   (c) Set $X_{k+1} = \{\mathbf{x}, v(k+1, \mathbf{x}) \geq \frac{1}{2}\}$.

3.3 Link with Image Partial Differential Equation

Step 2 of the implementation is in fact a weighted median filter, where weights are given by the convolution kernel M. The iteration of a median filter yields, as $\varepsilon \to 0$ and the number of iterations tends to infinity, a "motion by mean curvature." That is, letting $k\varepsilon$ tend to t and ε tend to zero, the sequence of image $u(k\varepsilon, \mathrm{x})$ tends to $u(t, \mathrm{x})$ satisfying

$$\frac{\partial u}{\partial t} = |Du| \operatorname{curv}(u), \qquad (5)$$

where $Du = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ is the gradient vector of u and the curvature curv(u) is defined by

$$\operatorname{curv}(u) = \operatorname{div}\left(\frac{Du}{|Du|}\right)$$

with $\operatorname{div} = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}$.

Considering shapes, similarly letting ε tend to zero while maintaining $k\varepsilon$ around t, the sequence of shapes $X_{k,\varepsilon}$ tends to a sequence $X(t)$. It is proved that for all t, $X(t)$ is the level set $\frac{1}{2}$ of the image $u(t, \mathrm{x})$, viscosity solution of the mean curvature motion (5).

3.4 Properties

Let us first discuss theoretical properties, that is the ones obtained asymptotically when ε tends to 0. The shape $X(t)$ evolves as a level set of the solution of the mean curvature motion. As a consequence, it inherits the properties of this equation. It is invariant by rotation, covariant by zoom (as is the preceding algorithm), but it is not affine invariant. It satisfies the local inclusion principle and has a pyramidal structure [that is, $X(t)$ can be constructed from $X(s)$ for any $s < t$].

However, these facts are slightly different when considering the discrete algorithm. Invariance by rotation is only approximate due to the work on a discrete grid (quantization effects). As well, covariance with respect to change of scale is not numerically exact, since, for example, a median filter with a disk of radius $2r$ cannot be computed from iterations of a median filter with a disk of radius r.

Concerning structural properties, the iterated weighted median filter satisfies the inclusion principle. This is merely due to the monotony of the median filter. Since the filter is local up to the size of the used kernel M, the local inclusion principle is satisfied, as illustrated in Fig. 3. At last, the algorithm is fully pyramidal on the set of discrete scales $k\varepsilon$, $k \in \mathbb{N}$.

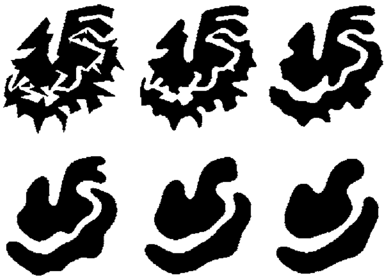

FIGURE 3 The Bence-Merriman-Osher shape smoothing method is a localized and iterated version of the dynamic shape algorithm. A convolution of the binary image with small-varance Gaussians is alternated with mid-level thresholding. It uses (**top, left**) as initial data the same shapes as in Fig. 2. **From left to right and top to bottom**: smoothing with increasing scales. Notice that the shapes keep separate. In fact, their is no interaction between their evolutions. Each one evolves as it would do alone.

4 Algorithm 3: Heat Equation on Curves

4.1 Description

To avoid quantization effects due to the image grid, the smoothing of a shape can be defined on the curve(s) corresponding to its boundary rather than on its associated binary image as in Algorithms 1 and 2.

Let us then focus on the smoothing of a single Jordan curve $\mathrm{x}_0(s)$. We assume that x_0 has finite length and is parameterized by the arclength s (that is, $|\mathrm{x}_0'(s)| = 1$ for all s), and extended as a L-periodic function from $\mathbb{R}$ to $\mathbb{R}^2$ (L is the length of the curve). As suggested by Mackworth and Mokhtarian [39], the curve x_0 may be smoothed by applying the heat equation to the vector-valued function $s \rightarrow \mathrm{x}_0(s)$, which boils down to convolve each component of $\mathrm{x}_0(s) = [x_0(s), y_0(s)]$ with a one-dimensional (1D) Gaussian kernel. This yields, for each scale t, the smoothed curve defined by

$$\mathrm{x}(t, s) = (x(t, s), y(t, s)), \qquad (6)$$

where

$$x(t, s) = (x_0 \star g_t)(s), \qquad y(t, s) = (y_0 \star g_t)(s) \qquad (7)$$

$$\text{and} \qquad g_t(s) = \frac{1}{2\sqrt{\pi t}} \exp\left(-\frac{s^2}{4t}\right)$$

4.2 Implementation

We assume that the initial curve is a closed polygon $\tilde{P}_0$ with n vertices written $\tilde{P}_0(0), \tilde{P}_0(1), \ldots \tilde{P}_0(N-1)$. The algorithm can be described as follows:

- **Initialization.** Resample uniformly (with a fixed step δ) the initial polygon $\tilde{P}_0(p), p = 0 \ldots N-1$ into a new polygon $P_0(p), p = 0..M-1$. There are several ways to define this resampling, but the simplest one consists in choosing the points $P_0(p)$ lying on the closed polygonal curve $\tilde{P}_0$ such that

$$\mathrm{dist}(\mathrm{P}_0(\mathrm{k}), \mathrm{P}_0(\mathrm{k}+1)) = \delta_0,$$

with

$$M = \lceil \mathrm{Per}(\tilde{P}_0)/\delta \rceil \quad \text{and} \quad \delta_0 = \frac{\mathrm{Per}(\tilde{P}_0)}{M} \simeq \delta$$

(dist($\cdot$) denotes the arc length distance on the curve, Per($\tilde{P}_0$) is the perimeter of $\tilde{P}_0$ and $\lceil\cdot\rceil$ means the upper integer part).

- **For k going from 1 to n,** compute for $p = 0 \ldots M-1$

$$P_k(p) = \sum_{q=-W}^{W} \tilde{g}_\tau(q)\, P_{k-1}(p+q), \tag{8}$$

with the convention that $P_{k-1}(r) = P_{k-1}(r \bmod M)$ for all r. The weights $\tilde{g}_\tau(q)$ are obtained by truncating (and normalizing) the Gaussian function, that is

$$\tilde{g}_\tau(q) = \frac{g_\tau(q\delta)\cdot \mathbb{1}_{[W,W]}(q)}{\sum_{q=-W}^{W} g_\tau(q\delta)}.$$

The integer W is chosen large enough to ensure a reasonably good approximation of the Gaussian kernel, say such that

$$g_\tau(W\delta) \le 10^{-3} g_\tau(0)$$

Notice that in (8), we use an iterative implemetation of the Gaussian convolution, numerically more efficient than the (theoretically equivalent) formula

$$P_k(p) = \sum_q g_{k\tau}(q\delta)\, P_0(p+q) \tag{9}$$

4.3 Properties

The partial differential eqaution (PDE) associated to this algorithm is, by construction, the heat equation

$$\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial^2 \mathbf{x}}{\partial s^2} \tag{10}$$

associated to the initial condition $\mathbf{x}(0,\cdot) = \mathbf{x}_0$. In the numeric scheme above, each polygon P_k corresponds to $\tilde{P}_0$ smoothed at scale $t = k\tau$.

Here, we must notice two very important facts which advocate against the method.

1. When $t > 0$, s is no more an arc length parameter of the evolved curve $\mathbf{x}(t,\cdot)$, that is, $|\frac{\partial \mathbf{x}}{\partial s}(t,s)| = 1$ is no longer ensured for $t > 0$.
2. Maximum principle holds for $x(t,s)$ and $y(t,s)$ as scalar solutions of the heat equation. In addition, $x(t,s)$ and $y(t,s)$ are C^∞ functions of (t,s), for $t > 0$. **Now, this does not imply that the curve $\mathbf{x}(t,s)$ behaves smoothly!** In fact, it can easily be seen (see Fig. 4) that a smooth curve may generate by this evolution self-crossings which by further smoothing entail the appearance of singularities.

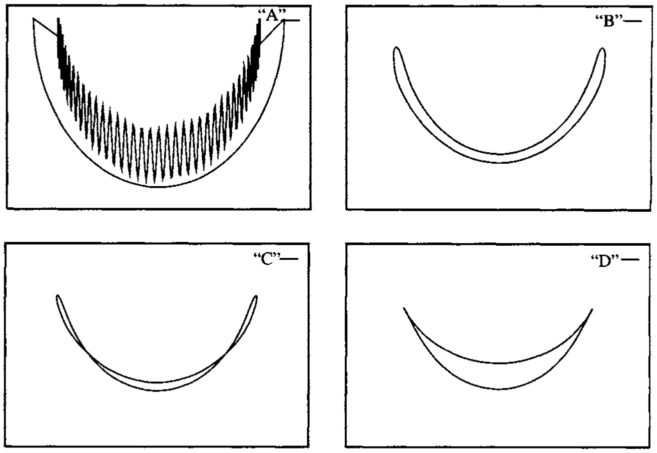

FIGURE 4 Curve evolution by the heat equation. The coordinates of the curves are parameterized by the arc length, and then smoothed as real functions of the length, by applying the heat equation to them. From **A** to **D**: The coordinates are smoothed with an increasing scale. Each coordinate function therefore is C^∞; the evolving curve can, all the same, generate self-crossings (**C**) or singularities (**D**).

The heat equation on curves satisfies the pyramidal and isotropy principles, but neither the inclusion principle nor the local inclusion principle, as shown by Fig. 4. The affine invariance principle has no particular reason to be satisfied. Concerning the numeric implementation, if we choose weights $\tilde{g}_\tau$ that are not Gaussian but simply symmetric and positive, the convergence of the numeric scheme to (10) still holds, but the isotropy principle is only asymptotically satisfied.

5 Algorithm 4: Renormalized Heat Equation on Curves

5.1 Description

The previous algorithm is pyramidal but does not satisfy the semigroup property, although the heat equation does. The reason is that the arc length parameterization is performed only at the beginning of the algorithm, and is, as we saw, no longer valid after the first iteration. To overcome the dramatic consequence of this, namely the loss of the inclusion and local inclusion principles, one can perform an arclength normalization at each iteration, as suggested, again, by Mackworth and Mokhtarian [40].

5.2 Implementation

The algorithm is very similar to the one given in Section 4.2, except that the normalization step is put inside the main loop. The initial polygon $\tilde{P}_0 = \tilde{P}_0(0)\tilde{P}_0(1)\ldots\tilde{P}_0(N_0-1)$ is smoothed according to the following procedure:

For k going from 1 to n

- **Step 1.** Resample uniformly (with a fixed step δ) the polygon $\tilde{P}_{k-1}(p), p = 0..N_{k-1}-1$ into a new polygon $P_{k-1}(p), p = 0..M_{k-1}-1$, as described in Section 4.2.
- **Step 2.** Compute for $p = 0\ldots M_k - 1$

$$\tilde{P}_k(p) = \sum_{q=-W}^{W} \tilde{g}_\tau(q)\, P_{k-1}(p+q) \qquad (11)$$

(same notations as Section 4.2).

5.3 Properties

The PDE associated with this algorithm is the mean curvature motion

$$\frac{\partial \mathbf{x}}{\partial t} = |\mathrm{Curve}\,\mathbf{x}|\;\mathbf{n}(\mathbf{x}), \qquad (12)$$

which corresponds to $g(x) = x$ in (2). It is also called the *intrinsic heat equation* since if s is the arclength parameter of $\mathbf{x}(t,\cdot)$, then

$$|\mathrm{Curve}\,\mathbf{x}|\;\mathbf{n}(\mathbf{x}) = \frac{\partial^2 \mathbf{x}}{\partial s^2}$$

This PDE satisfies not only the pyramidal and isotropy principles, but also the inclusion and the local inclusion principles (only the affine invariance principle has no particular reason to be satisfied). So does the algorithm above, asymptotically when $\delta \to 0$. This can be checked on Fig. 5: Even a very intricated curve has to evolve smoothly and cannot intersect itself or develop singularities.

6 Algorithm 5: Iterated Affine Erosion

6.1 Description

The renormalized heat equation is a reasonable way to smooth a shape, but it does not achieve the ultimate geometrical invariance allowed by shape smoothing PDEs, namely the affine invariance. We shall now describe a curve smoothing algorithm that, surprisingly enough, achieves affine invariance at a lower computational cost than all previous algorithms.

This algorithm is based on the iteration of an operator called *affine erosion* [44]. Given a real parameter $\sigma > 0$, the σ-affine erosion of a convex shape X is the shape that remains when all σ-chord sets of X have been removed from X. A σ-chord set of X is a domain with area σ which is limited by a chord of X (that is, a segment whose endpoints lie on the boundary of X) and the boundary of X (see Fig. 6). The definition is a little more complicated (though similar) in the case of nonconvex shapes, but we shall not need it here.

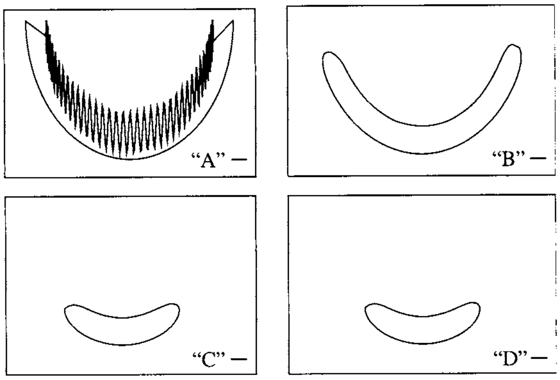

FIGURE 5 Curve evolution by the "renormalized heat equation" (Mackworth-Mokhtarian). At each smoothing step, the curve is reparameterized by arc length before a small-scale Gaussian convolution is applied. From **A** to **D**, the curve is smoothed with an increasing scale. Note that, in contrast with the linear heat equation (Fig. 4), the evolving curve shows no singularities and does not cross itself.

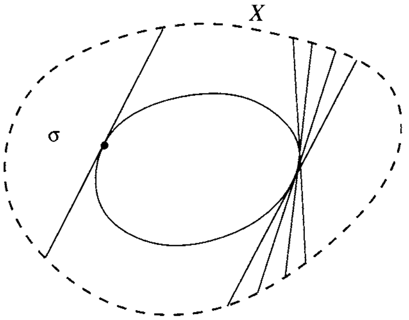

FIGURE 6 Affine erosion of a convex shape.

6.2 Implementation

The affine erosion of a convex shape X can be computed *exactly* in linear time (its boundary is made of straight segments and pieces of hyperbolae). However, this is of little interest for shape smoothing since a polygon is required at each iteration. Hence, from the numeric point of view it is enough to approximate the affine erosion at each iteration by a new polygon. Besides, the case of a nonconvex shape X can be dealt with by computing the affine erosion of each *convex component* of the boundary of X, provided that the erosion area σ is small enough.

The algorithm starts with a polygon P_0, and builds iteratively successive polygons $P_1, P_2, \ldots P_N$ (until the desired smoothing scale is attained) according to the following process:

1. **Break the curve P_k into convex components.** This operation permits to apply the affine erosion to convex pieces of curves, which is much faster (the complexity

is linear) and can be done simply in a discrete way. The main point is to take into account the finite precision of the computer to avoid spurious (small and almost straight) convex components only due to numeric errors.

2. **Sample each component.** At this stage, points are removed or added to guarantee a good sampling of the curve.
3. **Apply discrete affine erosion to each component.**
4. **Concatenate the pieces of curves obtained at step 3 to obtain the new polygonal curve P_{k+1}.** This way, we obtain a new closed curve on which the whole process can be applied again.

- **Break the curve into convex components**

  Let $P = P(0)P(1)\ldots P(n-1)$ be the polygonal curve to be broken into convex components. P has to be broken at points where the sign of the determinant

  $$d_i = [P(i-1)P(i), P(i)P(i+1)]$$

  changes. Numerically, we set $P(i) = (x_i, y_i)$ and use the formula

  $$d_i = (x_i - x_{i-1})(y_{i+1} - y_i) - (y_i - y_{i-1})(x_{i+1} - x_i). \qquad (13)$$

  Since we are interested in the sign of d_i, we must be careful because the finite numeric precision of the computer can make this sign wrong. Let us introduce the relative precision of the computer

  $$\varepsilon_0 = \max\{x > 0,\ (1.0 \oplus x) \ominus 1.0 = 0.0\}. \qquad (14)$$

  In this definition, $\oplus$ (resp. $\ominus$) represent the computer addition (resp. substraction), which is not associative. When computing d_i using (13), the computer gives a result $\tilde{d}_i$ such that $|d_i - \tilde{d}_i| \leq e_i$, with

  $$\begin{aligned} e_i = \varepsilon_0(&|x_i - x_{i-1}|(|y_{i+1}| + |y_i|) \\ &+ (|x_i| + |x_{i-1}|)|y_{i+1} - y_i| \\ &+ |y_i - y_{i-1}|(|x_{i+1}| + |x_i|) \\ &+ (|y_i| + |y_{i-1}|)|x_{i+1} - x_i|). \end{aligned}$$

  In practice, we take ε_0 a little bit larger than its theoretical value to overcome other possible errors (in particular, errors in the computation of e_i). For 4-bytes C *float* numbers, we use $\varepsilon_0 = 10^{-7}$, whereas the theoretical value (that can be checked experimentally using (14) is $\varepsilon_0 = 2^{-24} \simeq 5.96\ 10^{-8}$. For 8 byte C *double* numbers, the correct value would be $\varepsilon_0 = 2^{-53} \simeq 1.11\ 10^{-16}$

The algorithm that breaks the polygonal curve into convex components consists in the iteration of the following decision rule:

1. If $|\tilde{d}_i| \leq e_i$, then remove $P(i)$ (which means that the new polygon to be considered from this point is $P(0)P(1)\ldots P(i-1)P(i+1)\ldots P(n-1)$)
2. If $|\tilde{d}_{i+1}| \leq e_{i+1}$, then remove $P(i+1)$
3. If $\tilde{d}_i$ and $\tilde{d}_{i+1}$ have opposite signs, then the middle of $P(i), P(i+1)$ is an inflexion point where the curve must be broken
4. If $\tilde{d}_i$ and $\tilde{d}_{i+1}$ have the same sign, then increment i

This operation is performed until the whole curve has been visited. The result is a chained list of convex pieces of curves (with connecting endpoints).

- **Sampling**

  At this stage, we add or remove points from each polygonal curve in order to ensure that the Euclidean distance between two successive points lies between ε and 2ε (ε being the absolute space precision parameter of the algorithm).

- **Discrete affine erosion**

  This is the main step of the algorithm: Compute quickly an approximation of the affine erosion of scale σ of the whole curve. The first stage consists in the calculus of the "area" A_j of each convex component

  $$C^j = P^j(0)P^j(1)\ldots P^j(n_j - 1),$$

  given by

  $$A_j = \frac{1}{2}\sum_{i=1}^{n_j-2} [P^j(0)P^j(i), P^j(0)P^j(i+1)]$$

  Then, the effective area used to compute the affine erosion is

  $$\sigma_e = \max\left\{\frac{\sigma}{8}, \min_j A_j\right\}$$

  We restrict the erosion area to σ_e (which is less than σ in general) because the simplified algorithm for affine erosion (based on the breaking of the initial curve into convex components) may give a bad estimation of the continuous affine erosion + dilation when the area of one component is less than the erosion parameter. The term "$\sigma/8$" is rather arbitrary and guarantees an upper bound to the number of iterations required to achieve the final scale.

  Once σ_e is computed, the discrete erosion of each component is defined as the succession of each middle point of each segment $[AB]$ such that

  1. A and B lie on the polygonal curve
  2. A or B is a vertex of the polygonal curve

3. The area enclosed by $[AB]$ and the polygonal curve is equal to σ_e

These points are easily computed by keeping in memory and updating the points A and B of the curve plus the associated chord area.

Notice that if the convex component is not closed (which is the case if the initial curve is not convex), its endpoints are kept. Hence, step 4 (concatenation of pieces of curves) consists only in storing curves with connecting endpoints into a single curve. Note that new inflexion points (that is, curve breaking points found in step 1 of the next iteration) will not be exactly these endpoints in general, hence inflexion points are, as it should be, allowed to move in the whole iterative process.

6.3 Properties

The PDE associated with this algorithm is the affine scale space (3). An iteration of the previous process with erosion area σ corresponds to a scale increase of $\Delta t = \omega\sigma^{2/3}$ in (3), with $\omega = \frac{1}{2}(\frac{3}{2})^{2/3}$. Hence, to achieve a given scale t the previous process has to be applied with successive effective erosion areas $\sigma_e(1), \sigma_e(2), \ldots \sigma_e(N)$ such that

$$t = \omega \sum_{k=1}^{N} \sigma_e(k)^{2/3}$$

The scale space (3) satisfies all principles enumerated in Section 1. The numeric scheme above also satisfies these principles, up to the scale discretization σ and the computer precision ε. Notice that the inclusion principle is a property of the affine erosion itself, so that this property is not only satisfied asymptotically but with any number of iterations.

Not only the *iterated affine erosion* achieves the highest degree of invariance possible (in particular affine invariance), but among the five algorithms presented in this chapter, it is the fastest. Indeed, it has linear complexity in time and memory, and its stability is ensured by the fact that each new curve is obtained as the set of the middle points of some particular chords of the initial curve, defined themselves by an integration process (an area computation). Hence, no derivation or curvature computation appears in the algorithm. An example of curve smoothing with the iterated affine erosion is shown on Fig. 7.

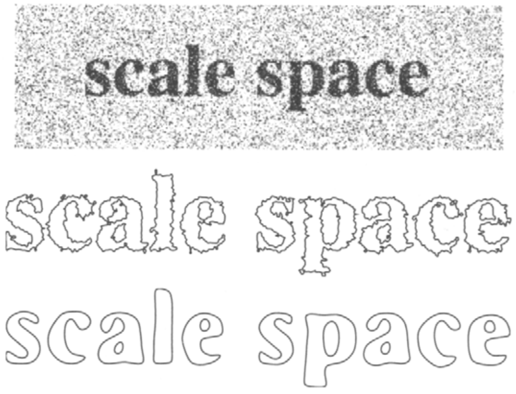

FIGURE 7 Smoothing curves with the affine scale space. **Top:** A text image corrupted by noise. **Middle:** Thresholding the image reveals characters as irregular level lines. **Bottom:** The same level lines, smoothed with the affine scale space. The smoothing process produces curves almost independent of the noise, which is a requirement for robust pattern recognition.

7 Bibliographic Notes

"Shape" can have different meanings. Most authors refer to shape as a common denominator between several identical or similar three-dimensional (3D) objects seen from different points of views. The problem of "recognizing a three-dimensional (3D) object from a single view" has been extensively studied [25, 29, 38, 47]. When several uncalibrated views of the same 3D object are available, its 3D model can be reconstructed [22, 27].

The common use of planar shape always involves a set of possible deformations. The usual ones are translation, rotation and zoom. Now, authors have explored in recent years more general deformations, particularly elastic ones. In [19, 58], the matching of images or shapes is made up to a flow of diffeomorphisms deforming one of the images (or shapes) onto the other one. These papers, and many references therein, give a well-developed mathematic treatment of a problem raised as early as 1983 [8]: the matching of deformed radiographic images to idealized atlas images.

We therefore have two meanings for shape: one is the 3D bulk of an object and the other one denotes an equivalence class of planar shapes, or images, under a deformation group.

Shape recognition must be performed in spite of *occlusion.* The phenomenology of occlusion was thoroughly studied by Kanizsa [30] and his school. Figure 8 illustrates how invariant our vision is to occlusion. Most viewers describe it as made of circles and rectangles; they do not even notice that such a description is inferential, rectangles and disks being simply not there. The robustness to occlusion requirement explains why we need a **local smoothing of the shape.** If the smoothing were global, it would depend on hidden parts of the shapes, which we cannot infer.

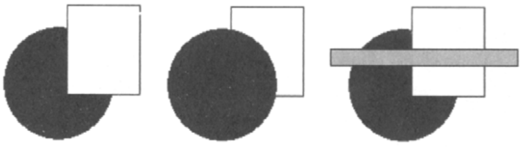

FIGURE 8 According to the theory of G. Kanizsa [30] and his school, shapes can be recognized even when they undergo several occlusions. Our perception is trained to recognize shapes which are only seeable in part.

The *figure-background problem,* leading to the **figure-background reversal principle** was studied by Rubin [52]. It is the other face of the occlusion problem: A shape is superimposed to a background, which can be made of various objects: how to extract, to single out, the shape from that clutter? This can also be viewed as a dilemma: Do we first extract the shape and then recognize it or, conversely, do we extract it *because* we had it recognized?

There are other perturbations affecting the identity of shapes. Shapes are easily recognized in images in spite of a change in the color and luminance scale; shapes are also easily recognized when noise and blur are present. Thus, one can list no less than five kinds of perturbations for planar shapes, which do not affect shape recognition, the elastic or projective deformations, the classic noise and blur, inherent to any perception and any image by Shannon's theory, the changes of contrast, of the occlusions and the background. These invariance requirements are classic and adopted in Rothwell [51] and many references therein.

7.1 The Local Contrast Invariance Argument

According to the founding Attneave paper [7], 1954,

> *The concentration of information in contours is illustrated by the remarkable similar appearance of objects alike in contour and different otherwise. The "same" triangle, for example, may be either white on black or green on white. Even more impressive is the familiar fact that an artist's sketch, in which lines are substituted for sharp color gradients, may constitute a readily identifiable representation of a person or thing.*

The contrast invariance of shape recognition (Fig. 9) was still earlier noticed by the gestaltist Wertheimer [56] in 1923. So shape perception is independent of the gray level scale or of the measured colors. Let us formalize this and define a digital image as a function $u : \mathbb{R}^2 \to \mathbb{R}^+$, where $u(\mathrm{x})$ represents the gray level or luminance at $\mathbf{x}$.[1] According to the contrast invariance principle, a first task of shape recognition is to extract from the image a topological information fairly independent from the varying and unknown contrast change function of the optical and/or biologic apparatus. One can model such a contrast change function as any continuous increasing function g from $\mathbb{R}^+$ to $\mathbb{R}^+$. The real datum when we observe u could be as well any image $g(u)$. This simple argument leads to select the set of level sets of the image, or its set of level lines, as a complete contrast invariant image description [11]. Mathematic morphology [54] considers the family of the connected components of the level sets of u,

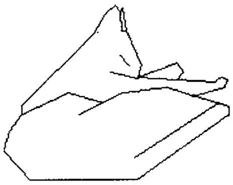

FIGURE 9 Sleeping cat? According to the perception psychologist Attneave [21], the main information in shapes is contained in the high curvature points: "Common objects may be represented with great economy, and fairly striking fidelity, by copying the points at which their contours change direction maximally, and then connecting these points appropriately with a straight-edge. The shape smoothing methods we have considered are all curvature motions, which filter out local spurious peaks of curvature, and keep the main ones."

$$\chi_\lambda(u) = \{\mathrm{x}, u(\mathrm{x}) \geq \lambda\}, \quad \lambda \in \mathbb{R}$$

An image can be reconstructed from its upper level sets by the obvious formula

$$u(x) = \sup\{\lambda, \mathrm{x} \in \chi_\lambda(u)\} \tag{15}$$

We define the level lines as the boundaries of the level sets. There are several frameworks to define the level lines: if u is considered to be a function with bounded variation, the level lines can be defined as a set of nested Jordan curves [4]. The set of all level lines is called the *topographic map* of the image.

7.2 The Concentration of Information Argument

The set of the image level lines is a complete contrast invariant information. Somewhat in contradiction with this contrast invariance principle, many authors assert, like Attneave, that information is concentrated along contours (i.e., regions where color changes abruptly). Thus, it makes sense to prune the tree of level lines by only keeping a selection of the most contrasted level lines. A simplification of the tree of level lines can be performed by using the method proposed in Desolneux et al. [16], which retains roughly all "meaningful" level lines. This method usually reduces by a factor 10 the number of level lines. Figure 10 shows an example of the set of meaningful level lines for a given image.

7.3 The Smoothing Argument

To justify the necessity of a multiscale smoothing for shapes, one can again rely on Attneave:

> *It appears, then, that when some portion of the visual field contains a quantity of information grossly in excess of the observer's perceptual capacity, he treats those components of information which do not have redundant representation*

[1] Shape recognition algorithms do not use much color. The geometric information brought by color usually is redundant with respect to luminance [12].

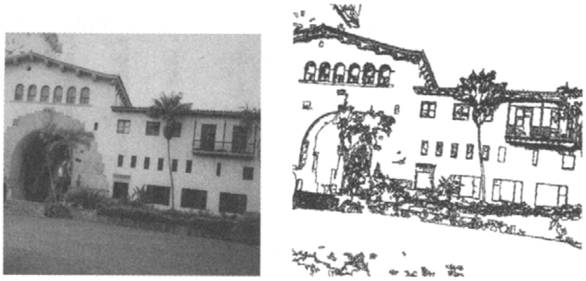

FIGURE 10 **Left:** Original image. **Right:** Meaningful level lines. These lines make a sparse nested set of Jordan curves. Each one of the Jordan curves can be used directly for shape-based image comparison.

> *somewhat as a statistician treats "error variance," averaging out particulars and abstracting certain statistical homogeneities. Such an averaging process was involved in drawing the cat for Fig. 9. It was said earlier that the points of the drawing corresponded to places of maximum curvature on the contour of the cat, but this was not strictly correct; if the principle had been followed rigidly, it would have been necessary to represent the ends of individual hairs by points. In observing a cat, however, one does not ordinarily perceive its hairs as individual entities; instead one perceives that the cat is furry.... The perceived contour of a cat (e.g., the contour from which the points of Fig. 9 were taken) is the resultant of an orthogonal averaging process in which texture is eliminated or smoothed out almost entirely, somewhat as if* ***a photograph of the object were blurred and then printed on high-contrast paper.***

A correct shape encoding, which does not get lost in textural details therefore asks for a previous blurring (see Fig. 11). The process suggested (in bold in the citation) is to convolve the silhouette of the object ("blurred") and then to enhance the result, which amounts to threshold the image ("high-contrast paper"). This algorithm, studied in Section 3, is by now known as the Bence, Merriman, Osher algorithm [42] (an iterated version of the Koenderink and Van Dorn [32] algorithm, studied in Section 2). It was proved to be equivalent to the curvature motion [9, 21] and to the iterated median filter [26].

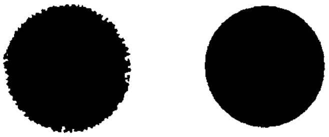

FIGURE 11 One can immediately see that those two objects are disks, with approximately the same size. The second one is obtained from the first by the affine curvature equation [2]. Such a curvature equation was anticipated by Attneave, who proposed to smooth the silhouette of a cat by (**A**) blurring the cat image and (**B**) enhancing the resulting image to get a smooth silhouette: "somewhat as if the photograph of the object were blurred and then printed on high-contrast paper."

7.4 Affine Invariant Mathematic Morphology and Affine Scale Space

The general process by which an image or a shape is smoothed at several scales to eliminate spurious or textural details and extract its main features is called "scale space." The main developments of scale space theory in the past 10 years involve invariance arguments: Indeed, a scale space will be useful to compute invariant information only if it is itself invariant. Let us summarize a series of arguments given (e.g.) in Alvarez et al. [3]: A scale space computing contrast invariant information must in fact deal directly with the image level lines; to be local (not dependent on occlusions), it must be a PDE. To be a smoothing, this PDE must be of a parabolic kind. Then, the further affine invariance requirement and the invariance with respect to reverse contrast (if we want a self-dual operator, in the mathematic morphology terminology [54]) lead to a single PDE,

$$\frac{\partial u}{\partial t} = |Du\,|\mathrm{curv}(u)^{\frac{1}{3}}, \tag{16}$$

where the power $\frac{1}{3}$ is signed, i.e., $s^{\frac{1}{3}} = \mathrm{sign}(s)\,|s|^{\frac{1}{3}}$. This equation is equivalent to the "affine curve shortening" (Eq. 3) [53] of all of the level lines of the image.

No further invariance requirement is possible: (16) is the only affine invariant local contrast invariant smoothing. The use of curvature-based smoothing for shape analysis is well established. Founding papers are [6], [45], and [18]. These authors define a multiscale curvature which is similarity invariant, but not affine invariant. Abbasi et al. [2] used the curvature motion (5) and an affine length parameterization of the boundary of the shapes to get an affine shape encoding. Of course, such a coding cannot be fully affine invariant, since the curvature motion is not affine invariant. It is much more natural to use directly (16), or, more interestingly, the affine curve evolution (3).

7.5 Curvature Equations

We insisted on curve smoothing algorithms, but the aim of this small section is to give some links to the major mathematic contributions to the mathematic aspect, namely the invention and analysis of curvature equations. Probably the first statement of a curvature equation as a smoothing process is due to Firey [23], who modelled the shape evolution of worn stones on a beach. The first proof that the two-dimensional (2D) curvature equation corresponding to the algorithms of Section 3 indeed smoothes curves is due to Grayson [24] in his famous paper "The heat equation shrinks embedded plane curves to round points." He shows that the smoothing of a curve ends up in a circular shape and that the number of inflexion points decreases along the smoothing process. This gives the mathematic foundation to the observed shape simplification. Sapiro and Tannenbaum [53] and, independently, Alvarez et al. [3] introduced the affine scale space equation in two different contexts. For the last authors, existence of solutions to the image equation was performed via viscosity solutions, following the theory of Evans-Spruck [20] and Chen et al. [13]. Angenent et al. established mathematic existence and regularity results for the curve affine scale space.

7.6 Affine Erosions and Dilations

The *affine erosion* (and dilation) of a shape was introduced in [44] as a geometric operator yielding a numeric scheme for the curve affine scale space. In general, the affine erosion of X is not simple to compute, but if X is convex, it can be exactly computed in linear time. This property is used, through convex decomposition, in the fast algorithm presented in Section 6.2, which was first described in [33].

Convergence of the iterated affine erosion to the affine scale space of images is established [43]. In [37], the authors go back to the mathematic morphology formalism ([41, 54]), in particular the Matheron theorem (theorem 6.2 in [26]) to define an affine distance and then affine invariant set erosions and dilations. Starting from a very simple (probably the simplest possible) definition of affine invariant distance of a point to a set, they show that alternate affine erosions and dilations are consistent with (16). The relevance of this result can be explained as follows. Applying the alternate scheme to the image u is equivalent to apply alternate affine erosions and dilations to each level curve of u, i.e., the iterated affine erosion of Section 6. Thus, this algorithm moves these curves according to Eq. 3. It is equivalent to move all level curves of u by this equation or to apply to u the evolution (16). This presentation follows the general line of the book in preparation [26].

7.7 From Global to Local Recognition Methods

In global recognition methods, the shape is considered as a whole and is described by a sequence of characteristics of the shapes [17, 50], such as perimeter, algebraic moments [28, 46, 55] or Fourier coefficients [34, 36, 49]. A comparison of two shapes is led back to the comparison of the vectors of computed characteristics. Most of the methods we just mentioned are Euclidean but not affine invariant. It is possible, however, to compute affine invariant moments. In that case, we face another difficulty: How can we define the relative weights of each moment in a shape comparison distance? This difficulty is overcome by the more clever *normalization methods* [14]. Normalization methods allow the transformation of any element of an equivalence class of shapes under a group of geometric transforms into a specific one, fixed once for all in each class. A good account of global normalization methods can be found in Reiss [50]. Affine normalization [28] can be used as a tool to match final invariance requirements on shape recognition, namely the locality (robustness to occlusion) and the affine invariance (invariance with respect to orthoprojections). Indeed, the affine invariance entails an affine normalization of the level lines. In addition, the robustness under partial occlusions implies that the normalization of the level lines must be done with respect to several local reference systems. Thus, several pieces of the same level line can be normalized by using different reference systems, providing a local and redundant description of the level line. Affine invariant robust semilocal descriptors are given by the lines which are bi-tangent to the curve. These descriptors can be used as starting points for an invariant sampling of the set of the tangent lines to the curve. A bi-tangent to a given curve is any straight line tangent to the curve at two different points. From each bi-tangent, one can define a local system of coordinates. These coordinate systems are then invariant under affine transforms and they can be used to normalize a portion of the curve. The just-sketched method is systematically used in Lamdan et al. [35] and in the recent shape recognition prototype described in Rothwell [51] and in Lisani et al. [37]. As pointed out in this last reference, an affine normalization of a curve needs three affine invariant reference points. As examples of finalized algorithms using shape smoothing, let us mention the one described in the Rothwell book [51]. The very interesting method of Abbasi and Mokhtarian [2] involves curvature scale space and similarity invariance. Lisani et al. [37] described an algorithm implementing the shape affine scale space described here to get compact codes of shapes and perform image comparison tasks, based on a systematic comparison of all level lines of two images. Several generic shape recognition algorithms stem from the founding paper by Wolfson [57] under the generic name of geometric hashing. Let us mention [10] as a very recent extension of this technique. In geometric hashing, a shape is described by local descriptors, usually points with an orientation (*minutiae* in the fingerprint recognition terminology), or corners [15]. The shape matching problem is then reduced to the problem of finding a given configuration of *minutiae*, describing a target shape, in a clutter of *minutiae* computed in another image. In the affine invariant case, the complexity of such a method grows as N^3, where N is the number of *minutiae* in the explored image. The reason is that no spatial order can be given between the *minutiae* and any group or subgroup can belong or not to a given shape. Lisani et al. [37] proved that one can single out preformed chains of *minutiae* in an image, namely the encoded pieces of level lines. The practical existence of these chains reduces considerably the complexity of a shape search, since a single chain may be enough to characterize a target shape.

Acknowledgments

The second and third authors gratefully acknowledge partial support by C.N.E.S (Centre National d'Etudes Spatiales), Centre National de la Recherche Scientifique, and Ministère de la Recherche et de la Technologie. This work was supported by Office of Naval Research under grant N00014-97-1-0839.

References

[1] The MegaWave2 software. *Available at* http://www.cmla.ens-cachan.fr/Utilisateurs/megawave.

[2] S. Abbasi and F. Mokhtarian, "Retrieval of similar shapes under affine transformation," in *Proc. International Conference on Visual Information Systems* (Amsterdam, The Netherlands, 1999), 566–574.

[3] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing: Multiscale analysis and P.D.E.," *Arch. Rat. Mech. Anal.* 16, 200–257 (1993).

[4] L. Ambrosio, V. Caselles, S. Masnou, and J. M. Morel, "The connected components of Caccioppoli sets and applications to image processing," *J. Eur. Soc. Math.* 3, 213–266 (2001).

[5] S. Angenent, G. Sapiro, and A. Tannenbaum, "On the affine heat flow for nonconvex curves," *J. Amer. Math. Soc.* (1998).

[6] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans PAMI* 8, 2–14 (1986).

[7] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.* 61, 183–193 (1954).

[8] R. Bajcsy, R. Lieberson, and M. Reivich, "A computerized system for the elastic matching of deformed radiographic images to idealized atlas images," *J. Comput. Assist. Tomog.* 7, 618–625 (1983).

[9] G. Barles and C. Georgelin, "A simple proof of convergence for an approximation scheme for computing motions by mean curvature," *SIAM J. of Numer. Anal.* 32, 484–500 (1995).

[10] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," in *Proc. of ICCV2001* (2001), 454–461.

[11] V. Caselles, B. Coll, and J. M. Morel, "Topographic maps and local contrast changes in natural images," *Int. J. Comput. Vis.*, 33, 5–27 (1999).

[12] V. Caselles, B. Coll, and J. M. Morel, "Geometry and color in natural images," *J. of Math. Imag. Vis.* 16, 89–105 (2002).

[13] Y.-G. Chen, Y. Giga, and S. Goto, "Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations," *Proc. Japan Acad. Ser. A* 65 (1989).

[14] T. Cohignac, C. Lopez, and J. M. Morel, "Integral and local affine invariant parameter and application to shape recognition," in *ICPR94* (1994), 164–168.

[15] R. Deriche and G. Giraudon, "On corner and vertex detection," in *CVPR91* (1991), 650–655.

[16] A. Desolneux, L. Moisan, and J. M. Morel, "Edge detection by Helmholtz principle," *J. Math. Imag. Vis.* 14, 271–284 (2001).

[17] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* (Wiley, New York, 1973).

[18] G. Dudek and J. K. Tsotsos, "Shape representation and recognition from multiscale curvature," *CVIU* 2, 170–189 (1997).

[19] P. Dupuis, U. Grenander, and M. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Qua. Appl. Math.* 56, 587–600 (1998).

[20] L. C. Evans and J. Spruck, "Motion of level sets by mean curvature, I," *J. Differ. Geom.* 33, 635–681 (1991).

[21] L. C. Evans, "Convergence of an algorithm for mean curvature motion," *Indiana University Math. J.* 42, 553–557 (1993).

[22] O. Faugeras, Q. T. Luong, and T. Papadopoulo, *The Geometry of Multiple Images* (MIT Press, Cambridge, MA, 2001).

[23] W. J. Firey, "Shapes of worn stones," *Mathematika* 21, 1–11 (1974).

[24] M. A. Grayson, "The heat equation shrinks embedded plane curves to round points," *J. Differ. Geom.* 26, 285–314 (1987).

[25] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints* (MIT Press, Cambridge, MA, 1990).

[26] F. Guichard and J. M. Morel, Image iterative smoothing and P.D.E.'s. Book in preparation, 2000.

[27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision* (Cambridge University Press, United Kingdom, 2000).

[28] R. K. Hu, "Visual pattern recognition by moments invariants. *IEEE Trans. Inf. Theory* 179–187 (1962).

[29] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int. J. Comput. Vis.* 5, 195–212 (1990).

[30] G. Kanizsa, "Organization in vision: Essays on Gestalt perception," in *Praeger*, 1979.

[31] J. J. Koenderink and A. J. van Doorn, "Dynamic shape," *Biol. Cyber.* 53, 383–396 (1986).

[32] J. J. Koenderink and A. J. van Doorn, "Dynamic shape," *Biol. Cyber.* 53, 383–396 (1986).

[33] G. Koepfler and L. Moisan, "Geometric multiscale representation of numerical images," in *ScaleSpace99* (1999), 339–350.

[34] A. Krzyzak, S. Y. Leung, and C. Y. Suen, "Reconstruction of two-dimensional patterns from Fourier descriptors," *MVA* 2, 123–140 (1989).

[35] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Object recognition by affine invariant matching," *Proc. CVPR* (1988) 335–344.

[36] C. C. Lin and R. Chellappa, "Classification of partial 2-d shapes using Fourier descriptors," in *CVPR86* (1986) 344–350.

[37] J.-L. Lisani, P. Monasse, L. Moisan, and J.-M. Morel, "On the theory of planar shape," *SIAM J. on Multiscale Modeling and Simulation* 1, 1–24 (2003).

[38] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Art. Intell.* 31 355–395 (1987).

[39] A. Mackworth and F. Mokhtarian, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Trans. Pattern Anal. and Mach. Intell.* 8 (1986).

[40] A. Mackworth and F. Mokhtarian, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Anal. and Mach. Intell.* 14, 789–805 (1992).

[41] G. Matheron, *Random Sets and Integral Geometry* (John Wiley, New York, 1975).

[42] J. Merriman, B. Bence, and S. Osher, "Diffusion generated motion by mean curvature," in *Computational Crystal Growers Workshop* (American Mathematical Society, 1992), 73–83.

[43] L. Moisan, Traitement numerique d'images et de films: équations aux dérivées partielles preservant forme et relief. PhD thesis, UNIV. Paris Dauphine, 1997.

[44] L. Moisan, "Affine plane curve evolution: A fully consistent scheme," *IEEE Trans. Image Process.* 7, 411–420 (1998).

[45] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *PAMI* 14, 789–805 (1992).

[46] P. Monasse, "Contrast invariant image registration," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 6, 3221–3224 (Phoenix, Arizona, 1999).

[47] H. Murase and S. K. Nayar, "Learning and recognition of 3d objects from appearance," in *Proc. IEEE Workshop on Qualitative Vision* (1993) 39–50.

[48] S. Osher and J. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation," *J. Comput. Phys.* 79, 12–49 (1988).

[49] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *SMC* 7, 170–179 (1977).

[50] T. H. Reiss, *Recognizing Planar Objects Using Invariant Image Features, volume 676 of Lecture Notes in Computer Science* (Springer Verlag, New York, 1993).

[51] C. A. Rothwell, *Object Recognition Through Invariant Indexing* (Oxford Science Publications, Cambridge, UK, 1995).

[52] E. Rubin, *Visuell Wahrgenommene Figuren* (Copenhagen, Gyldendals, 1921).

[53] G. Sapiro and A. Tannenbaum, "Affine invariant scale-space," *Int. J. Comput. Vis.* 11, 25–44 (1993).

[54] J. Serra, *Image Analysis and Mathematical Morphology* (Academic Press, New York, 1982).

[55] C. H. Teh and R. Chin, "On image analysis by the method of moments," *IEEE Trans. on PAMI* 10, 496–513 (1988).

[56] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt ii. *Psychologische Forschung*, 4:301–350, 1923. Translation published as Laws of Organization in Perceptual Forms, in Ellis, W. (1938). A source book of Gestalt psychology (71–88). London: Routledge & Kegan Paul.

[57] H. J. Wolfson, "Model-based object recognition by geometric hashing," in *ECCV90* (1990) 526–536.

[58] L. Younes, "Computable elastic distances between shapes," *SIAM J. Appl. Math.* 58, 565–586 (1998).