Scale-spaces and affine curvature

Olivier Faugeras[1] and Renaud Keriven[2]

[1] I.N.R.I.A. Sophia-Antipolis, 06561 Valbonne, France, faugeras@sophia.inria.fr
[2] E.N.P.C. , 93167 Noisy Le Grand, France, keriven@cermics.enpc.fr

Abstract

We present a new way to compute the affine curvature of plane curves. We explain how an affine scale-space can be used to gain one order of derivation in the numerical approximation of affine curvature. We outline our implementation and compare our results with previous ones. This paper ends by showing a simple application in pattern recognition using affine curvature.

Keywords: Geometry, Invariants, Pattern Recognition.

Introduction

A great amount of work in object recognition deals with semi-differential affine or projective invariants [7]. Local invariants seem much more difficult to obtain because of the large orders of derivatives required; for example, the affine curvature is a fourth order quantity, the projective curvature a seventh order one, which makes their actual computation quite a challenge, to say the least.

Among the numerous applications of scale-spaces, this paper presents a surprising one: one can use the affine scale-space to compute the affine curvature of a plane curve with third order derivatives only.

As a result, we present a way to obtain affine curvature with enough accuracy to perform tasks like recognition.

Affine curvature and the affine scale-space

In this section, we present some basic concepts in affine differential geometry and introduce the property of the affine scale-space we will use to compute the affine curvature.

1. Affine differential geometry

Let $\mathcal{L}$ be a Lie group operating on some objects. A quantity q depending on these objects is called an *invariant* of $\mathcal{L}$ if, whenever a transformation $L \in \mathcal{L}$ changes q into q', we have $q' = \alpha(L)q$, where α is a function of L alone, i.e. does not depend on the object which is transformed. If $\alpha \equiv 1$, then q is called an *absolute invariant*.

Differential invariants are special invariants based on local transformations (see [6]).

Let $\mathcal{C} : \mathbf{R} \rightarrow \mathbf{R}^2$ be a plane curve of parameter p. The first and the second differential invariants for the Euclidean group $\{m \mapsto Rm + T \mid R \text{ rotation}, T \text{ translation}\}$ are the well known Euclidean arc-length parameter v and curvature κ defined by:

$$\left\{ \begin{array}{lcl} \frac{\partial v}{\partial p} & = & \|\frac{\partial \mathcal{C}}{\partial p}\| \\ \kappa & = & \|\frac{\partial^2 \mathcal{C}}{\partial v^2}\| \end{array} \right.$$

which are preserved by rotations and translations.

$$\left\{ \begin{array}{lcl} \frac{\partial s}{\partial p} & = & [\frac{\partial \mathcal{C}}{\partial p}, \frac{\partial^2 \mathcal{C}}{\partial p^2}]^{1/3} \\ \mu & = & [\frac{\partial^2 \mathcal{C}}{\partial s^2}, \frac{\partial^3 \mathcal{C}}{\partial s^3}] \end{array} \right. \quad (1)$$

which are invariants for affine proper motions, and absolute invariants for special affine motions ($\{m \mapsto Am + B \mid [A] = 1, B \in \mathbf{R}^2\}$) .

Circles (and straight lines) are the only curves with constant Euclidean curvature. In the affine case, constant affine curvature is obtained for the conics ($\mu = 0$ for a parabola, $\mu > 0$ for an ellipse and $\mu < 0$ for an hyperbola).

While κ is a second order derivative quantity, μ is a fourth order one. This is the reason why it has not been used so much in image processing so far. These two quantities are linked by the handsome relation:

$$\mu = -\kappa^{4/3} + \frac{1}{2}\frac{\partial^2 \kappa^{-2/3}}{\partial v^2} \quad (2)$$

2. An interesting property of the affine scale-space

We now consider a closed planar curve $\mathcal{C}_0(p) : S^1 \rightarrow \mathbf{R}^2$ and the associated affine scale-space , i.e. the family of closed curves $\mathcal{C}(p,t) : S^1 \times [0, \tau) \rightarrow \mathbf{R}^2$ evolving according to the following law (see [12]) :

$$\left\{ \begin{array}{lcl} \frac{\partial \mathcal{C}(p,t)}{\partial t} & = & \frac{\partial^2 \mathcal{C}(p,t)}{\partial s^2} \\ \mathcal{C}(p,0) & = & \mathcal{C}_0(p) \end{array} \right. \quad (3)$$

This scale-space is known to be affine-invariant (absolute for special affine motions). Any smooth embedded curve first becomes convex and then converges to an elliptical point [12].

What we will focus on is the evolution law of the Euclidean curvature, given in [12], for the affine scale-space (rewritten to give μ from κ):

$$\mu = -\frac{1}{\kappa}\frac{\partial \kappa(p,t)}{\partial t} \quad (4)$$

Given the affine scale-space , we can see that the affine curvature depends on the Euclidean curvature and its first derivative with respect to time, thus a gain of one order of derivation (compare with (2)). We assume here that the computation of the affine scale-space can be made accurate enough not to be considered as effectively introducing through the back door one more derivation. This assumption will be confirmed by experimentation:

- Equation (4) leads to much more accuracy than methods directly using fourth order derivatives (see section 2.).
- The error on $\frac{\partial \kappa}{\partial t}$ has proven to be of the same order as the one on $\frac{\partial \kappa}{\partial v}$ which is another third order derivative (see next sections and equation (12) to understand why this derivative must be used).

Computing affine curvature

In this section, we show how (4) can be used to compute μ.

1. The problem

Given the curve $\mathcal{C}_0(p)$, the first thing to do is to compute the associated affine scale-space . This is achieved with the numerical method proposed by Osher and Sethian [8] and used by Sapiro and Tannenbaum [11]. It is based on Hamilton-Jacobi theory [5]. The curves $\mathcal{C}(p,t)$ are represented by the zero level sets of a family of surfaces $\Phi : \mathbf{R}^2 \times [0, \tau) \rightarrow \mathbf{R}$ governed by some evolution law depending on the desired scale-space . $\Phi(x, y, t)$ is discretized, its evolution is computed and $\mathcal{C}(., t)$ is obtained from $\Phi(., ., t)$ using a zero-crossing algorithm. Except its reliability, accuracy and stability, this method has the following properties:

1. *Scale preservation.* The evolution law used is not exactly (3). Only the normal component of the velocity is considered and (3) becomes:

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{C}(p,t)}{\partial t} = \beta \mathbf{n} = \kappa^{1/3}\mathbf{n} \\ \mathcal{C}(p,0) = \mathcal{C}_0(p) \end{array} \right. \quad (5)$$

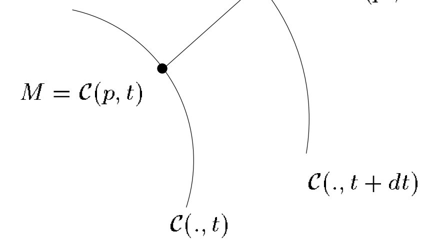

Figure 1: The normal velocity is $\beta\mathbf{n}$ with $\beta = \kappa^{1/3}$

where $\mathbf{n}$ is the Euclidean normal vector.

Yet, the geometric curves $\mathcal{C}(.,t)$ obtained are the solutions of (3). Furthermore, the family of curves is not only the same, but the scale parameter t is the same. No time rescaling is needed: it is essential in order to use (4).

2. *Loss of the curve parameter.* Because the curves are obtained from the zero-crossing of some surfaces (and anyway because (5) is used instead of (3)), the parameter p is lost. This seems to make impossible the use of (4) because $\frac{\partial \kappa(p,t)}{\partial t}$ is a derivative *at constant* p for the solutions of (3).

We now explain how to use (4) in spite of the loss of the curve parameter.

2. Solution

The clue is to derive κ *at constant Euclidean arc-length parameter* v. Denoting the derivative of f with respect to x at constant y by $\left.\frac{\partial f}{\partial x}\right|_y$, any function $f(p,t)$ verifies: $\frac{\partial f}{\partial t} = \left.\frac{\partial f}{\partial t}\right|_v + \frac{\partial f}{\partial v}\frac{\partial v}{\partial t}$. We will expand the term $\frac{\partial \kappa}{\partial t}$ in (4) this way:

$$\frac{\partial \kappa}{\partial t} = \left.\frac{\partial \kappa}{\partial t}\right|_v + \frac{\partial \kappa}{\partial v}\frac{\partial v}{\partial t} \tag{6}$$

We now show how to compute the terms in the right hand side of (6). Projecting $\frac{\partial \mathcal{C}}{\partial t} = \left.\frac{\partial \mathcal{C}}{\partial t}\right|_v + \frac{\partial \mathcal{C}}{\partial v}\frac{\partial v}{\partial t}$ onto the Euclidean tangent vector $\mathbf{t}$ and using (3), we obtain:

$$\frac{\partial v}{\partial t} = \langle \frac{\partial^2 \mathcal{C}}{\partial s^2}, \mathbf{t}\rangle - \langle \left.\frac{\partial \mathcal{C}}{\partial t}\right|_v, \mathbf{t}\rangle \tag{7}$$

Using (1) with $p \equiv v$, it is quite straightforward to show that:

$$\frac{\partial^2 \mathcal{C}}{\partial s^2} = \kappa^{1/3}\mathbf{n} - \frac{1}{3\kappa^{5/3}}\frac{\partial \kappa}{\partial v}\mathbf{t} \tag{8}$$

which gives the first term of the right hand side of (7)

Let $M = \mathcal{C}(p,t)$ and M' be the intersection of $\mathcal{C}(.,t+dt)$ with the normal $[M,\mathbf{n})$ at M (figure 1). We know from [12] that: $M' = M + \beta\mathbf{n} = M + \kappa^{1/3}\mathbf{n}$. Let us now fix p and consider equation (4) at the point $\mathcal{C}(p_0,0)$ where μ is to be computed. At each time t, the Euclidean arc-length parameter $v(p,t)$ if defined up to the choice of an origin $Or(t) \in \mathcal{C}(.,t)$. Supposing $\beta\mathbf{n}$ smooth enough, we choose for $Or(t)$ the unique solution in a neighborhood of $\mathcal{C}(p_0,0)$ of the following differential equation:

$$\frac{dOr}{dt} = \beta\mathbf{n}, \quad Or(0) = \mathcal{C}(p_0,0)$$

On the trajectory $(\mathcal{T})$ of Or (figure 2), in particular at point $\mathcal{C}(p_0,0)$, derivatives at constant v become Lie derivatives in the direction $\mathbf{n}_\beta = [\beta\mathbf{n}^T, 1]^T$ of the tangent plane of the spatio-scale surface $[\mathcal{C}(p,t)^T, t]^T$ (see [4]):

$$\text{on } (\mathcal{T}),\ \left.\frac{\partial \kappa}{\partial t}\right|_v = \lim_{dt\to 0}\frac{\kappa(M') - \kappa(M)}{dt} = L_{\mathbf{n}_\beta}\kappa \tag{9}$$

$$\text{on } (\mathcal{T}),\ \left.\frac{\partial \mathcal{C}}{\partial t}\right|_v = \lim_{dt\to 0}\frac{M' - M}{dt} = L_{\mathbf{n}_\beta}\mathcal{C} \tag{10}$$

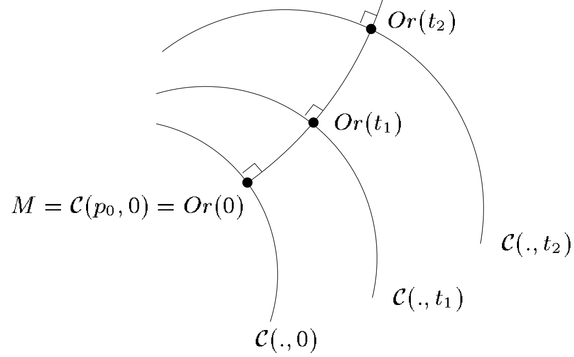

Figure 2: Euclidean arc-length parameter is choosen constant along $(\mathcal{T})$

As a result, $\frac{\partial \mathcal{C}}{\partial t}\Big|_v$ is normal to the curve and:

$$\langle \frac{\partial \mathcal{C}}{\partial t}\Big|_v, \mathbf{t}\rangle = 0 \tag{11}$$

This completes the evaluation of $\frac{\partial v}{\partial t}$ in (7).

By the way, we also got the first term of the right hand side of (6) in (9). Equation (4) becomes the final formula:

$$\mu = \frac{1}{3\kappa^{8/3}}(\frac{\partial \kappa}{\partial v})^2 - \frac{L_{\mathbf{n}_\beta}\kappa}{\kappa} \tag{12}$$

Note that, in spite of the apparition of new terms due to the loss of the parameter p in the process of computing the affine scale-space , μ still requires third order derivatives only. The term $\frac{\partial \kappa}{\partial v}$ is directly computed on the curve. The Lie derivative is obtained by considering locally $\kappa(p,t)$ as a scalar field $K(x,y)$ defined on a neighborhood of $\mathcal{C}(p_0, 0)$:

$$L_{\mathbf{n}_\beta}\kappa = \beta\langle\nabla K, \mathbf{n}\rangle = \kappa^{1/3}\langle\nabla K, \mathbf{n}\rangle \tag{13}$$

Experimental results

This section describes the way we used (12) to compute the affine curvature of closed planar curves. We present some results and an example of application in object recognition.

1. Practical scheme

Our implementation performs the following steps:

1. Get the curve from the image with a subpixelic edge detector ([13], [2]).
2. Compute an accurate Euclidean curvature with a method based on Chebyshev polynomials (see [9]). Use the same method to get the Euclidean and affine arc-length parameters and $\frac{\partial \kappa}{\partial v}$. (The affine arc-length parameter , though unused in (12), may be used to normalize the affine curvature – see section 3.).
3. Compute the affine scale-space with the numerical method of [8].
4. For some time steps, extract $\mathcal{C}(.,t)$ from the scale-space using a subpixelic zero-crossing as in 1. Compute the Euclidean curvature along these curves as in 2.
5. For each point of the curve, construct the scalar field $K(x,y)$ from the results of steps 2 and 4. Fit it with a smooth surface using a least squares method. Compute ∇K.
6. Compute the affine curvature according to (12) and (13).

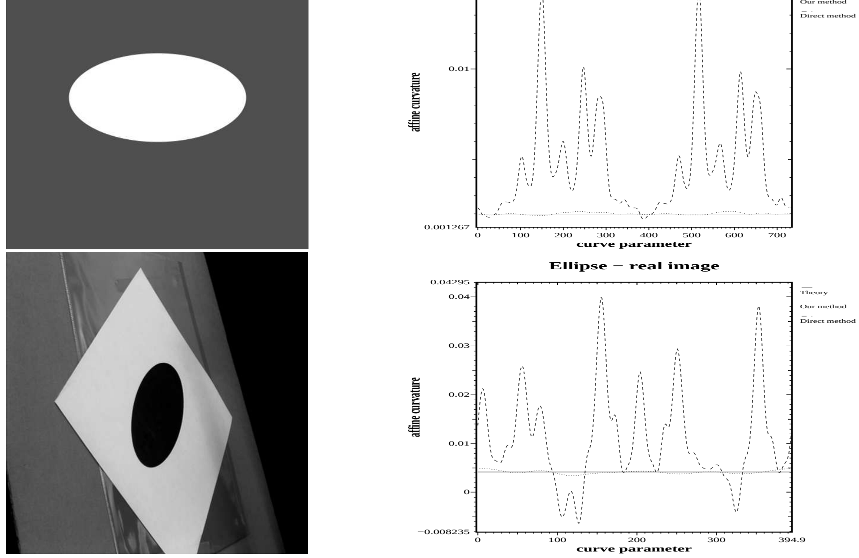

Figure 3: A synthetic (up-left) and a real ellipse (down-left). Their respective affine curvatures in (top-right) and (bottom-right): the theoretical curvature, the one obtained directly with fourth order derivatives and the one obtained with our method.

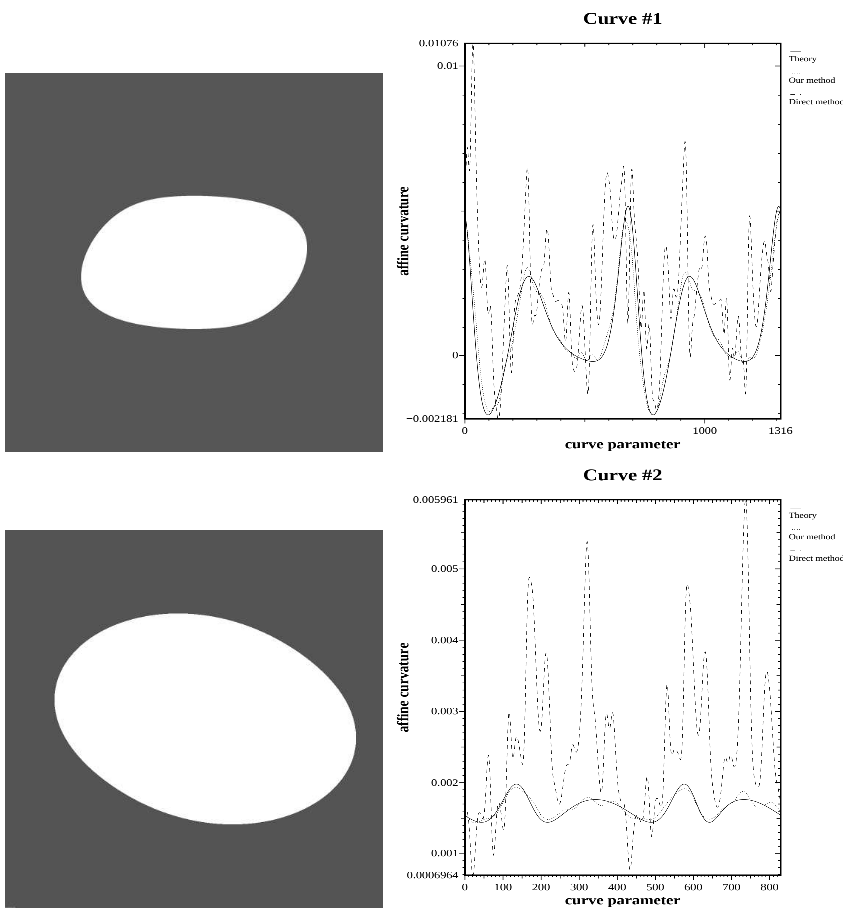

Figure 4: Two synthetic curves: (top-left) $\mathcal{C} = [a\cos(p)(1+\epsilon\sin(2p)), b\sin(p)(1+\epsilon\cos(2p))]$ and (bottom-left) $\mathcal{C} = [a\cos(p)(1+\epsilon\sin^2(2p)), b\sin(p)(1+\epsilon\sin^2(2p))]$. Their respective affine curvatures in (top-right) and (bottom-right): the theoretical curvature, the one obtained directly with fourth order derivatives and the one obtained with our method.

method. Each time, the computed affine curvature was compared to the theoretical one, and to the affine curvature obtained from a method using fourth order derivatives (several methods have been tried, the less inaccurate one has proven to be the one using Chebyshev polynomials like in [9]).

The gain of one order of derivation is obvious: see figures 3 and 4.

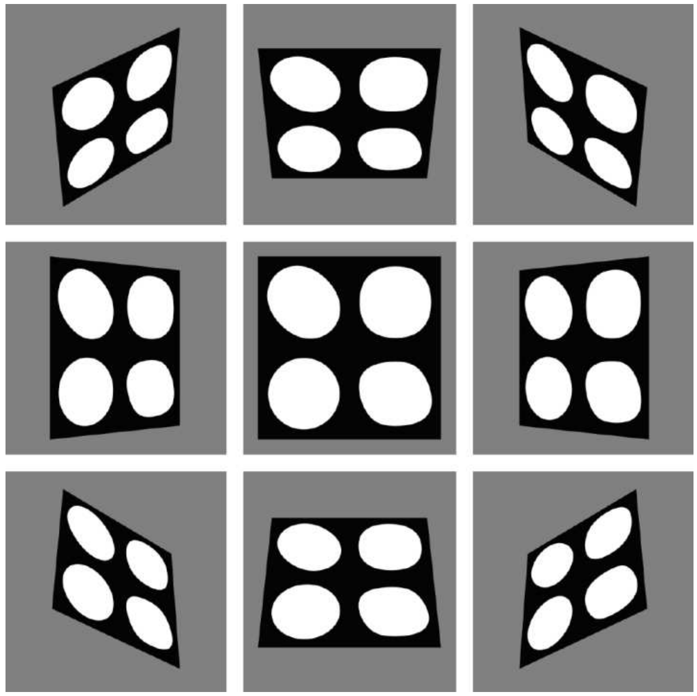

Figure 5: Four closed curves on a plane. Some of the different views used in the recognition experiment; the central image is the front view.

3. Application

In this section, we investigate the use of the computed affine curvature for performing recognition tasks.

Two different views of a three-dimensional plane curve are related by a projective transformation (homography). Thus, their projective curvatures (see [1]) are equal. Yet, if the view points are far from the curve and close to each other, the homography can be approximated by an affine transformation. As a result, the affine curvatures are equal. Actually, they are proportional because the affine curvature is not an absolute invariant for proper affine motions. We can obtain such an absolute invariant by normalizing the curvature by multiplying it by the square of the affine perimeter:

$$\tilde{\mu} = P_{aff}^2 \mu \quad \text{where } P_{aff} = \oint ds$$

We tested this idea on the following problem: Given four closed convex plane curves in the same plane viewed from a wide variety of viewpoints, can we use the normalized affine curvature to recognize those curves independently of viewpoint? Several synthetic views of the four curves were generated (see figure 5: note that the objects are not so far and the view points not so close, thereby pushing to the limit the assumption that the homographies are well approximated by affine transformations). For a curve $\mathcal{C}_i^v$ $(1 \leq i \leq 4)$ on a view v, the normalized affine curvature $\tilde{\mu}_i^v$ is computed as a function of a normalized arc-length parameter $\tilde{s} = s/P_{aff}$ $(0 \leq \tilde{s} \leq 1)$ (figure 6). We then consider the distance between $\tilde{\mu}_i^v$ and the curvature of each curve j of the front view v^f:

$$d_{ij}^v = \min_{0 \leq \tilde{s}_0 \leq 1} [\oint (\tilde{\mu}_i^v(\tilde{s} - \tilde{s}_0) - \tilde{\mu}_j^{v^f}(\tilde{s}))^2 d\tilde{s}]^{1/2}$$

Let four points $\{M_j\}$ of $\mathbf{R}^2$ represent the four curves (e.g. the corners of a square). The bary-center of $\{M_j\}$ with respective weights $1/d_{ij}^v$ shows which curve of the front view is the nearest to $\mathcal{C}_i^v$ and how reliable

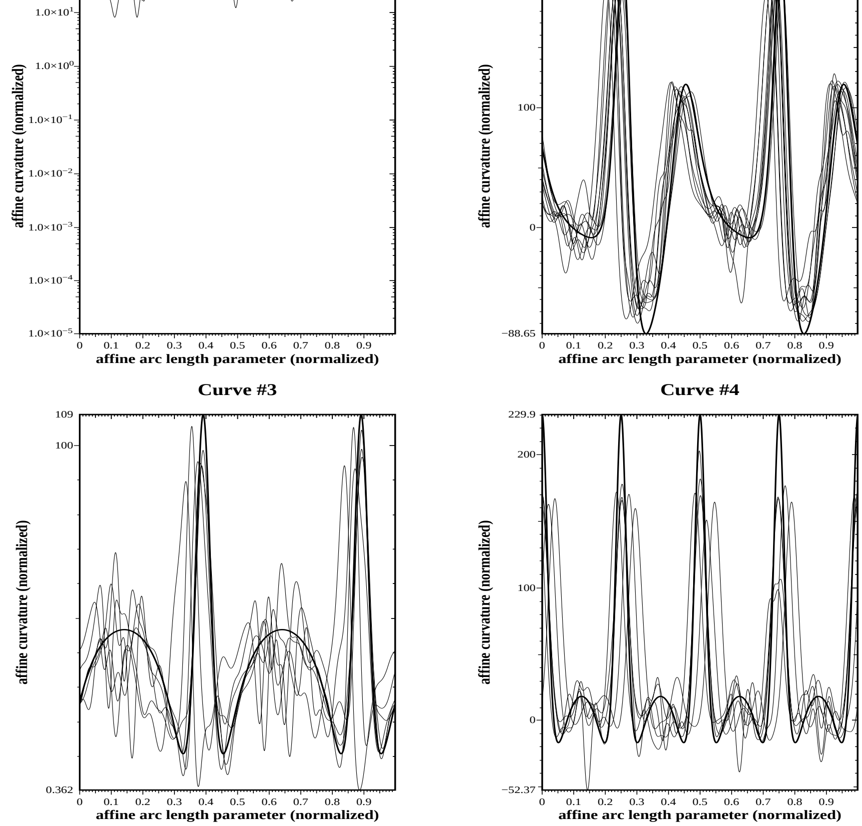

Figure 6: The affine curvatures obtained for the four curves viewed from different points. In bold, the curvature in the front view.

the result is. This scheme always recognized which curve of the front view was being seen from another view point (figure 7).

Concluding remarks

We showed how the gain of one order of derivation through the use of affine scale-space leads to an affine curvature accurate enough to perform useful tasks such as pattern recognition. It could also be used to help stereo matching where Euclidean curvature is still used as a (poor) approximation to the projective curvature (see [10]).

Work in progress includes the extension of our method to non convex or open curves. This does not seem to pose hard problems. The main difficulty still remains: the computation of projective differential invariants such as projective curvature. With the affine curvature we compute, the projective curvature could be obtained with fifth order derivatives instead of seventh order, using the result of [3]. This is still quite a challenge.

References

[1] Elie Cartan. *La Théorie des Groupes Finis et Continus et la Géométrie Différentielle traitée par la Méthode du Repère Mobile.* Jacques Gabay, 1992. Original edition, Gauthiers-Villars, 1937.

[2] Frédéric Devernay. A fast and efficient subpixelic edge detector. unsubmitted, October 1993.

[3] Olivier Faugeras. On the evolution of simple curves of the real projective plane. *Comptes rendus de l'Académie des Sciences de Paris, Tome 317, Série I*, 0(6):565–570, September 1993. Also INRIA Technical report number 1998.

[4] Olivier D. Faugeras and Théo Papadopoulo. A theory of the motion fields of curves. *The International Journal of Computer Vision*, 10(2):125–156, 1993.

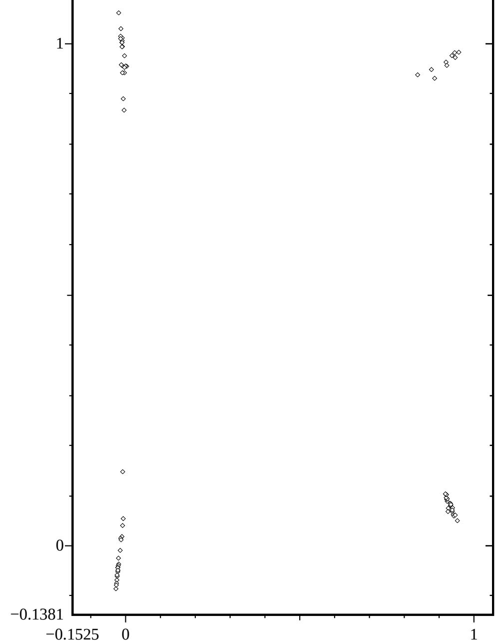

Figure 7: The result of the classification. The points $[0,0], [1,0], [0,1]$ and $[1,1]$ represent the four curves of the front view. All the curves are recognized.

[5] M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. *J. of Differential Geometry*, 23:69–96, 1986.

[6] W. Guggenheimer, Heinrich. *Differential Geometry*. Dover Publications, New York, 1977.

[7] Joseph L. Mundy and Andrew Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[8] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[9] Théo Papadopoulo and Olivier Faugeras. Estimation of the Second Order Spatio-temporal Derivatives of Deforming Image Curves. In *Proc. International Conference on Pattern Recognition*, volume 1, pages 179–184, Jerusalem, October 1994. IEEE Computer Society Press.

[10] Luc Robert and Olivier Faugeras. Curve-Based Stereo: Figural Continuity And Curvature. In *CVPR91*, pages 57–62, Maui, Hawai, June 1991. IEEE.

[11] Guillermo Sapiro and Allen Tannenbaum. Affine shortening of Non-Convex Plane Curve. Technical Report EE PUB 845, Technion Israel Institute of Technology-Haifa, August 1992.

[12] Guillermo Sapiro and Allen Tannenbaum. Affine Invariant Scale Space. *The International Journal of Computer Vision*, 11(1):25–44, August 1993.

[13] Ali J. Tababai and O. Robert Mitchell. Edge location to subpixel values in digital imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):188–201, March 1984.