Correspondence

Noise-Resistant Invariants of Curves

Isaac Weiss

***Abstract*— Projective invariants are shape descriptors that are independent of the point of view from which the shape is seen, and therefore, they are of major importance in object recognition. They make it possible to match an image of an object to one stored in a database without the need to search for the correct viewpoint. In this paper, we obtain an invariant representation ("signature") of a general curve. The calculation is local and does not suffer from the occlusion problem of global descriptors. To make the method robust, we have developed differentiation techniques that give much more reliable results than previous ones. These differentiation methods are useful in many other applications as well.**

***Index Terms*— Invariance, matching, object recognition, perspective, pose, projective, robust estimation.**

I. Introduction

A major problem in object recognition is the fact that the same shape can be seen from different points of view, resulting in different images. In order to compare a given image with one that is stored in a library of images, existing methods had to search in a multidimensional parameter space to find the appropriate "pose," or point of view. Invariants of shapes, because they are independent of the point of view, free us from this search and allow direct matching between the observed and the stored images.

Projective invariants were a very active mathematical subject in the latter half of the 19th century. However, in vision, only one projective invariant (the cross ratio of four points on a line [6]) was used until recently.

Projective invariants of curves and surfaces were first introduced in vision by this author [13]. Both algebraic and differential invariants were described in that paper, which pointed out their usefulness for object recognition. The algebraic invariants were then successfully applied to industrial objects in [7]. Recognition of occluded surfaces using invariants was treated in [4], and semi-differential invariants were used by [12] and [1]. Many other contributions have been made since. A recent comprehensive review is presented by Weiss [18].

Algebraic invariants are well suited to be used with algebraic shapes, namely, shapes that can be expressed as an implicit polynomial $f(x, y) = 0$, e.g., conics. The polynomial coefficients are obtained by fitting the appropriate polynomial to the whole visible shape and calculating the invariants from the polynomial coefficients. These invariants have several problems: first, most shapes are not algebraic, making it hard to fit simple polynomials to them. Second, the algebraic method is global, requiring dealing with whole shapes. Like any global descriptors, they are vulnerable to occlusion problems.

Manuscript received March 25, 1991; revised June 16, 1992. This work was supported by the Air Force Office of Scientific Research under grant F49620-92-J-0332, the Office of Naval Research under grant N0004-91-J-1222, the Advanced Research Projects Agency (ARPA Order No. 8459), and the U.S. Army Topographic Engineering Center under Contract DACA76-92-C-0009. Recommended for acceptance by Associate Editor J. Eklundh.

The author is with the Center for Automation Research, University of Maryland, College Park, MD 20742.

IEEE Log Number 9208439.

Differential invariants overcome these problems because they are local, i.e., invariants are found for each point on a general curve. An important property of these invariants is *completeness*, namely, that a small set of invariants contains all the essential information about the curve. In the plane, the two lowest order invariants determine all others (see p. 144 of [8]). Furthermore, these two are sufficient to identify the curve up to a relevant transformation. In the Euclidean case, for example, the curvature and arclength determine a curve up to a Euclidean transformation.

In this paper, we use this completeness property for object recognition under projective and affine transformations. At each point of the given curve, we calculate two projective invariants I_1, I_2. We plot these numbers as a point in an "invariant plane" whose coordinates represent invariants. In effect, we plot one invariant against the other. In this way, the given curve is represented by an invariant "signature" curve in the invariant plane. Because of the completeness, the signature identifies the curve up to the transformation to which I_1, I_2 are invariant.

It is useful to know the amount of information that needs to be obtained from the image. To find invariants, we have to eliminate the information in the image that is specific to the coordinate system. For example, given a pencil that can move or rotate on a table, the position of the pencil and its orientation are not invariant, but its length is a Euclidean invariant. Given the coordinates, say, of the ends of the pencil, we can eliminate the position and orientation and calculate the distance. Thus, from the four measured coordinates, we have eliminated the three Euclidean transformation coefficients and found one invariant.

Similar arguments apply for other transformations. In the projective case, we want to eliminate eight coefficients of the transformation; therefore, the number of data quantities to be obtained from the image should exceed eight. It is easy to apply this to algebraic shapes. For example, one conic does not contain enough coefficients, but two conics do, with ten independent ones, and so does a cubic with nine coefficients.

In the differential case, similar arguments apply. However, in traditional methods of obtaining invariants, a complication arises because the curve is represented parametrically as $x(t), y(t)$ with an arbitrary curve parameter t. Wilczynski's method [19] requires the eighth derivative (of both $x(t)$ and $y(t)$) to find two invariants to both the projection and the change of parametrization, for a total of 18 quantities. If we disregard the parameter problem, we need only the fourth derivatives of x, y, namely, only ten numbers. Thus, if we could overcome the parameter problem, the number of data quantities needed would be no more than that required by an algebraic method.

The parameter problem can be avoided from the outset because the parameter is not part of the geometry of the curve; the coordinates x, y of each point are sufficient to determine the curve. The parameter is an arbitrary function introduced for convenience; therefore, it is clear that one can improve on these methods. In this paper, we present an approach that avoids the parametrization problem and enables us to find two local invariants with only the number of quantities that are needed from a purely geometrical point of view. In this way, we combine the advantage of the differential method (its locality) with that of the algebraic method, which does not need a curve parameter.

Of course, the fourth derivative, or similar quantities needed in our new invariant method, are still high by computer vision standards, and common methods completely fail to obtain them. We have found a smoothing and differentiation method that gives good results for high derivatives. In our experiments, the error in the derivatives and in the invariants was no more than the error in the data. Obviously, the method has applications beyond invariants.

In the following sections, we review the previous methods for deriving differential invariants, and then, we describe our new methods for finding differential invariants and for obtaining derivatives. Finally, experiments are presented.

II. Wilczynski's Method

In this section, we describe the method developed by Wilczynski [18], who obtained closed-form formulas for a complete set of differential projective invariants of curves. (This is also in [10] and [13].)

A projective transformation in the plane can be written in Cartesian coordinates as

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{pmatrix} = \frac{1}{xT_{31} + yT_{32} + T_{33}} T \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

where T is a constant matrix. The factor multiplying T in this equation contains the coordinates; therefore, the transformation is nonlinear and leads to infinities when it vanishes. To avoid these problems, it is common to generalize the treatment by working in homogeneous coordinates $\mathbf{x} = (x_1, x_2, x_3)^t$ and writing the transformation as

$$\tilde{\mathbf{x}} = \lambda_{\mathbf{x}} T \mathbf{x}$$

where $\lambda_{\mathbf{x}}$ is an arbitrary factor that can be different at each point $\mathbf{x}$. This can be generalized to curves in n-D homogeneous space.

A curve in n-D can be represented parametrically as $\mathbf{x}(t)$. We want to find quantities at each point of the curve that are independent of both the coordinate system and the parameter t. To find invariants, one can proceed in stages:

i) Find invariants to the linear part T of the transformation above.
ii) From these, form invariants to multiplication by λ.
iii) From the latter, construct invariants to the curve parameter t.

We will proceed to describe each stage.

The basic method of obtaining invariants in this approach is by using derivatives. The advantage of differentiation is that it eliminates constants, which are often associated with the coordinate system. For instance, a straight line can be written as $y = ax + b$, with the constants a, b both depending on the coordinates. However, the equation $y'' = 0$ represents straight lines (and only them) and is invariant of any coordinate system. The more general case is more involved, but the same principle applies.

Invariance to T can be obtained by taking the derivatives of the curve $\mathbf{x}(t)^{(k)}$ and forming the set of linear algebraic equations

$$\mathbf{x}^{(n)} + \binom{n}{1} p_1 \mathbf{x}^{(n-1)} + \binom{n}{2} p_2 \mathbf{x}^{(n-2)} + \ldots + p_n \mathbf{x} = 0 \quad (1)$$

for the unknowns $p_1 \ldots p_n$ at each point t. For a plane curve ($n = 3$), we have

$$\mathbf{x}''' + 3p_1\mathbf{x}'' + 3p_2\mathbf{x}' + p_3\mathbf{x} = 0$$

with the three unknowns p_1, p_2, p_3. It is easy to see that multiplying $\mathbf{x}$ by a constant matrix T has no effect on the above equation since the matrix factors out. Thus, the p_n are invariant to the linear transformation T.

Next, we have to deal with the transformation of multiplying by λ, namely, $\tilde{\mathbf{x}} = \lambda_{\mathbf{x}}\mathbf{x}$. We find quantities P_i, which are functions of p_i that are invariant to this transformation. Since P_i are not invariant to transforming the parameter t, they are called *semi-invariants*. For a planar curve, we have (see p. 58 of [18])

$$\begin{aligned} P_2 &= p_2 - p_1^2 - p_1' \\ P_3 &= p_3 - 3p_1p_2 + 2p_1^3 - p_1''. \end{aligned} \quad (2)$$

The final stage is finding invariants to the parameter t. The transformation in question is $t \to \tilde{t}(t)$, to which we can find *relative invariants of weight* w, namely, quantities that transform as

$$\tilde{\Theta}_w = \frac{1}{(\tilde{t}')^w}\Theta_w.$$

These invariants are

$$\begin{aligned} \Theta_3 &= P_3 - \frac{3}{2}P_2' \\ \Theta_8 &= 6\Theta_3\Theta_3'' - 7(\Theta_3')^2 - 27P_2\Theta_3^2. \end{aligned}$$

These are relative invariants of weights 3 and 8, respectively.

The question arises of how many independent invariants exist for a curve[1] and to what extent one can reconstruct the curve given the invariants. We have the completeness theorem (see p. 53 of [18]).

Theorem: The invariants Θ_3, Θ_8 completely determine a plane curve except for a projective transformation.

This is a special case of the more general property mentioned in the introduction. From these, other invariants can easily be derived. In particular

$$\Theta_{12} = 3\Theta_3\Theta_8' - 8\Theta_8\Theta_3'.$$

Although the invariant set Θ_3, Θ_8 is complete, it cannot be used as is to identify the curve because they are relative invariants, i.e., the transformation contains the unknown $(\tilde{t}(t)')^{-w}$. However, we can derive from them absolute invariants ($w = 0$). We can choose

$$I_1 = \frac{\Theta_3^8}{\Theta_8^3}, \qquad I_2 = \frac{\Theta_3^4}{\Theta_{12}}.$$

We can now define an "invariant plane," with coordinates I_1, I_2. For each point on the given curve, we can calculate the two invariants and draw a point I_1, I_2 in the invariant plane. In this way, we obtain an invariant curve, or signature, of the original curve.

Calculating the above invariants requires the eight derivative with respect to the parameter, and this poses a hard problem from a practical point of view. The invariants Θ_3, Θ_8 were implemented numerically, using simple finite difference methods, by Brown [3]. He concluded that in this simple implementation, the above invariants are quite unreliable and hard to use in practice.

The method of invariants can be made reliable in several ways, two of which are treated here:

1) Calculate derivatives using more sophisticated numerical analysis methods. We have succeeded in reliably obtaining a fourth derivative from numerical data (Section V).
2) Develop methods that need fewer quantities such as derivatives that need to be obtained from the image. We have developed a method that does not require a parametrization of the curve and thus requires fewer data quantities (Section IV).

[1]There are some semantic fine points regarding the definition of an independent invariant.

TABLE I
CANONICAL COORDINATE SYSTEM

	f	f'	f''	f'''	f''''
$\bar{x}$	0	1	0	1	0
$\bar{y}$	0	0	1	I_1	I_2

A. Our Modified Semi-Invariants

We will now modify the semi-invariants written above to reduce the number of derivatives needed from five to four. From (2), we see that the semi-invariant P_2 contains only the first derivative p_1'. Since the p_i themselves depend on the third derivatives of the curve, P_2 depends on the fourth derivatives. However, P_3 depends on the fifth derivative of the curve. This can be eliminated by subtracting P_2':

$$P_3^* = P_3 - P_2' = p_2' - 2p_1p_1' + 3p_1p_2 - 2p_1^3.$$

Our curve can now be described by P_2, P_3^*, which involve only fourth derivatives. They are invariant to projectivities (T plus λ) but not to the change of the parameter t.

III. Our Canonical Methods

We develop new approaches to obtaining local invariants based on different principles than previously described. The methods reduce the amount of data needed from the image and offer more intuitive insight into the geometrical nature of the invariants. In this section, only semi-invariants are obtained because we still depend on a curve parameter. In the next section, we describe a different treatment that avoids the parameter altogether and yields full invariants.

The basic idea is to transform the given coordinate system to a "canonical," or standard system, which is determined by the shape itself. Since this canonical system (or frame) is independent of the original system, it is invariant. All quantities defined in it are, thus, invariant. The concept can be illustrated by examples of simpler transformations. If a 1-D function $x(t)$ is subject to scale transformation in x, we can obtain scale invariance by transforming to a new coordinate $\bar{x}$ in which the derivative at the origin is fixed, say, $\bar{x}'(0) = 1$. We achieve this by a simple normalization $\bar{x} = x/x'(0)$. This also fixes other scale-dependent quantities such as the second derivative $\bar{x}''(0)$; therefore, they are now scale invariant.

An important 2-D example is the Euclidean invariants. To find an invariant at a given point on a curve, we can change the x, y axes so that the new $\bar{x}$ axis is tangent to the curve at that point. We thus have set $\bar{y}' = 0$, whereas the second derivative $\bar{y}''$ at this point is now the curvature. It is invariant since we can obtain the same canonical system regardless of the system with which we started. We see that by determining some of the properties of the system, the others are also determined and become invariant. We generalize this process to the affine and projective cases.

Canonical Projective Semi-Invariants

We define a projective canonical system in Table I (with nonhomogeneous coordinates).

In Table I, $\bar{x}, \bar{y}$ are the canonical coordinates, f in the first line is either $\bar{x}$ or $\bar{y}$, and the primes denote derivatives with respect to t. The I_1, I_2 mean that there are no conditions on these derivatives $\bar{y}''', \bar{y}''''$; they are our desired invariants. In all, there are eight conditions here, and they can be satisfied by employing the eight parameters of a projective transformation to arrive at this canonical coordinate system. The following theorem holds.

Theorem: Given a curve and its first four derivatives with respect to some parameter t, one can always find a coordinate system that is unique for each point and that satisfies these canonical conditions. Conversely, given a curve in the canonical system of Table I, its parameterization around the origin is determined uniquely up to and including the fourth derivative.

The proof is constructive and gives the appropriate transformation from the given coordinate system to the canonical one. The details are given elsewhere [17].

These invariants have been calculated explicitly and tested numerically [2]. It was confirmed that they are invariant to the projectivity but not to a change of the parameter. Full invariance requires either higher derivatives, as in Wilczynski's method, or getting rid of the parameter altogether, as is described next.

IV. Differential Invariants without Derivatives

We summarize here a new method that combines the locality of the differential invariants with the advantages of the algebraic method, such as avoiding the need for curve parametrization. A detailed description is given by Weiss in [15].

As discussed before, in order to eliminate the projective transformation, having eight coefficients, we have to extract from the image a number of data quantities greater than eight. To obtain two absolute invariants at each point, we thus need ten quantities. However, Wilczynski's method needs the eighth derivatives of $x(t), y(t)$, namely, 18 quantities in all (counting the zeroth derivatives). The semi-invariants only need the fourth derivatives, but they are not invariant to the parameter.

A problem with the differential method, as mentioned before, is that the arbitrary parameter is a source of ambiguity whose elimination requires higher derivatives then the projectivity itself needs. Although the projectivity has only eight parameters that have to be eliminated, requiring no more than fourth derivatives, the unknown parameter forces us to obtain more information from the image (or higher derivatives) to eliminate it. On the other hand, the parameter is not, in fact, part of the geometry of the curve. The coordinates x, y of each point are sufficient to characterize the curve, and the parameter is introduced artificially for convenience. Thus, it is desirable to use a method that does not require a curve parameter.

A way to approach the problem is to deal with an implicit representation of the curve, i.e., one of the form $f(x, y) = 0$, without an explicit parameter. The given curve itself is quite arbitrary, and it is hard to find an f that will represent it in this way. However, at each point of it, one can find a simpler, osculating curve that can be represented implicitly and whose invariants are relatively easy to find.

An osculating curve is a generalization of the tangent. A tangent is a line having at least two points in common with the curve in an infinitesimal neighborhood, i.e., two "points of contact." This can be expressed as a condition on the first derivative. Similarly, a higher order osculating curve has more (independent) points of contact, and the condition on the derivatives can be written as

$$\frac{d^k}{dt^k}(f^*(x, y) - f(x, y)) = 0, \qquad k = 0 \ldots n \tag{3}$$

where f^* is the osculating curve, f is the given curve, and n is the order of the contact. Since the derivatives vanish, this condition is invariant to the parameter t. Since it has a geometric interpretation with points of contact, the condition is also projectively invariant.

In practice, we will need neither the parameter nor the derivatives. The data quantities needed here are the coefficients of the given curve f, which can be obtained by fitting f to the data points. We need no more of them than in the algebraic method. Thus, the robustness is increased relative to the explicit differential method.

To find invariants at a point $\mathbf{x}_0$ of a given curve, we proceed in the following steps:

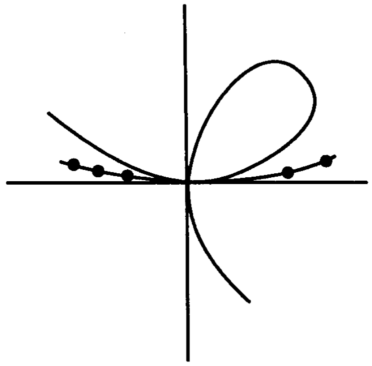

Fig. 1. Osculating folium of Descartes.

1. Fit any convenient curve f to the data pixels around the given point x_0.
2. Find an algebraic (implicit) curve f^* that osculates f at that point. The coefficients of this osculating curve are independent of any parametrization.
3. Eliminate the factors of the projectivity by moving to a canonical coordinate system in which the osculating curve has a simple, predetermined form.

The osculating curve f^* is chosen as the simplest one that enables us to eliminate the eight projectivity factors. Three factors are eliminated by moving the origin to the given point $\mathbf{x}_0$ and rotating so that the x axis is tangent to the given curve f there. We now need a five-coefficient osculating curve f^* that passes through the origin. A suitable choice is the "nodal cubic" [9]

$$f^* = c_0x^3 + c_1y^3 + c_2xy^2 + c_3x^2y + c_4y^2 + xy = 0. \quad (4)$$

This curve intersects itself at the origin; therefore, it has two tangents there: one lying along the x axis. The other tangent is called the "projective normal" [10].

Our goal is now to transform the coordinates so that this nodal cubic takes on the simple coefficient-free form

$$x^3 + y^3 + xy = 0. \quad (5)$$

This is known as a *folium of Descartes*; see Fig. 1.

In a nutshell, we obtain it as follows. We skew the coordinates so that the projective normal becomes perpendicular to the x axis, thus providing a canonical y axis. This eliminates c_4. We scale the axes to eliminate c_0, c_1, obtaining an affine canonical system with new $\bar{c}_2, \bar{c}_3$. These are now *affine invariants*. We tilt and slant to eliminate them as well, obtaining the projective canonical system. The full development is in [15].

Semi-Differential Invariants

One can reduce the number of data quantities needed at a curve point if some other information about the shape is known, such as the appearance of straight lines or known feature points. For example, a silhouette of an airplane can contain both curved parts and straight lines. Invariants involving both derivatives and reference points were found in [1] and [12]. However, their method still uses an unknown parameter that has to be eliminated so that robustness is reduced.

The "derivativeless" method described above is perfectly suited for this situation and again leads to both saving in the number of data quantities needed from the image and increased reliability. In [15], we describe various configurations involving one or two feature points or lines. In all these cases, the osculating curve is a conic; therefore, it is easier to obtain. In the simplest case, given "feature" point $\mathbf{x}_1$, we can connect it with a line to the curve point $\mathbf{x}_0$ and then skew the coordinates to make this line perpendicular to the x axis. We thus obtain a canonical y axis. Further transformations elminate all conic coefficients and yield a canonical system having a unit parabola and a unit distance between $\mathbf{x}_0$ and $\mathbf{x}_1$ (Fig. 2). Of course, the correspondence between the appropriate parts of the shapes has to be known.

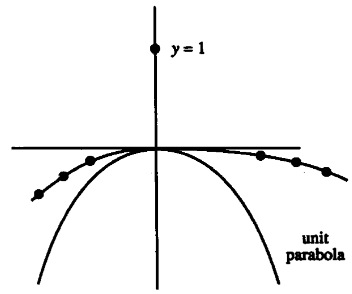

Fig. 2. Canonical conic and point.

V. Noise-Resistant Derivatives

Investigating differential invariants has led us to develop reliable methods for obtaining high derivatives. Our results have significant uses beside invariants. We analyze the requirements that a differentiation method has to meet, show that the Gaussian is not good enough for our purposes, and present the method that was successful in our experiments. More details can be found in [14] and [11].

Two basic, and conflicting, requirements are involved:

1. *Accuracy:* At least in the noiseless case and for low orders, one would like to obtain the correct analytic derivatives.
2. *Smoothing (to alleviate the effect of noise and discretization):* Generally, smoothing reduces the accuracy, even in the analytic case, as the more rapid changes in the function are smoothed out. The goal in designing a derivative filter is then to strike the correct balance between accuracy and smoothing. In this section, we describe how this balance can be achieved.

As an illustration, we first show that the Gaussian gives the wrong result even for the simplest functions. Smoothing over x^2 gives

$$g(x, \sigma) \otimes x^2 = x^2 + \sigma^2$$

where $g(x, \sigma)$ is the Gaussian, and $\otimes$ means convolution. We can see that an error is introduced that increases as we increase the smoothing, i.e., higher σ. Similar results are obtained for higher powers x^n and for taking derivatives. For effective smoothing, σ should not be too small; therefore, this error can be substantial. This is a *systematic, analytic error*, as opposed the random noise that we want to smooth. For our purpose of obtaining accurate derivatives, this is "oversmoothing," and we want to eliminate it. As noted before, we do not expect a smoothing operator to give accurate results, but we do want to improve the balance between accuracy and smoothing so that at least the simple, smooth functions will remain accurate.

We will now discuss this noise versus signal problem in terms of the ratio between the smoothing parameter σ and a "natural" scale of the shape s_0. For a smooth shape $f(x)$, we estimate this s_0 as the scale over which the relative change in the shape, $\Delta f/f$ is of the order of magnitude of 1. We then rescale the x axis so that $\bar{x} = x/s_0$ and rewrite the shape as $\bar{f}(\bar{x})$. For example, if $f(x) = \sin(x/s_0)$, then $\bar{f}(\bar{x}) = \sin \bar{x}$. This makes the derivatives of $\bar{f}$ with respect to

$\bar{x}$ of the order of magnitude of 1. In this way, we "normalize" the signal in the x direction and deal with its scale separately.

A natural tool in dealing with the derivative in some neighborhood of a smooth shape is the Taylor expansion around $x = 0$

$$\bar{f}(\bar{x}) = \bar{f}(\frac{x}{s_0}) = \sum_{\nu} \frac{\bar{f}^{(\nu)}}{\nu!}(\frac{x}{s_0})^{\nu}$$

with the derivatives $\bar{f}^{(\nu)} = \frac{d^{\nu}\bar{f}}{d(x/s_0)^{\nu}} \approx O(1)$. We will look at the result of Gaussian filtering at $x = 0$. It is easy to show by smoothing each term above that the error introduced by the Gaussian is

$$g \otimes f - f \approx (\frac{\sigma}{s_0})^2 \bar{f}'' + \frac{1}{8}(\frac{\sigma}{s_0})^4 \bar{f}^{(4)} + \dots \tag{6}$$

If we want accurate results, we have to keep this error small. Looking at the leading term, we see that the error is proportional to σ^2/s_0^2; therefore, we have to keep σ small. Unfortunately, this limits the ability of the filter to smooth out the noise. A similar result is obtained for derivatives.

A way to improve the situation is to eliminate the leading terms in the expansions of the errors above. If the first term is eliminated, for instance, then the error will be reduced to $\approx \frac{1}{8}(\frac{\sigma}{s_0})^4$. This way, we can obtain a much better accuracy for the same smoothing parameter σ as before (keeping $\sigma \leq s_0$). Alternatively, we can increase the smoothing without compromising accuracy.

The first error terms in (6) are the errors of smoothing the corresponding powers in the Taylor expansion of f. Thus, we need a smoothing filter F_l that will preserve the first l powers x^n:

$$F_l \otimes x^n = x^n, \qquad n = 0 \dots l$$

(For the Gaussian, $l = 1$.) It can be easily shown that the powers will be preserved if the first l moments of the filter vanish (and the 0th moment is normalized to 1). Defining the "normalized moments" as

$$m_n = \int (\frac{x}{\sigma})^n F_l(x) dx$$

where σ is now a measure of the filter size, e.g., the variance, we can write the conditions on the filter as

$$m_0 = 1; \qquad m_n = 0 \quad \text{for} \quad n = 1 \dots l.$$

Using such a filter, the first l terms of the smoothing error (6) vanish, and we are left with a term proportional to $(\frac{\sigma}{s_0})^{l+1}$:

$$F_l \otimes f^{(k)} - f^{(k)} \approx \frac{m_{l+1}}{(l+1)!}(\frac{\sigma}{s_0})^{l+1} \bar{f}^{(l+k+1)} + \dots$$

This is the analytic error in smooting the derivative $f^{(k)}$.

Accuracy Criterion

For good accuracy, the above error has to be small. Since $\bar{f}^{(\nu)} \approx O(1)$, we obtain from above the condition on the analytic error

$$\epsilon = \frac{m_{l+1}}{(l+1)!}(\frac{\sigma}{s_0})^{l+1} < \epsilon_{max}. \tag{7}$$

This can be called an "accuracy criterion," namely, a filter design criterion that links σ, s_0, l to some acceptable accuracy tolerance ϵ_{max}. This criterion can be used to estimate the parameters in several ways. Given the meaningful scale of change of the signal s_0 and the smoothing parameter σ, we can calculate the order l of the filter needed to lower the analytic oversmoothing error to the acceptable tolerance. σ is chosen to smooth the estimated noise. Conversely, for a given order l, we can calculate the largest smoothing parameter σ that will still keep the error below the tolerance. Generally speaking, at low orders l, we need $\sigma < s_0$, but at high orders, we can afford to have the smoothing σ bigger than s_0 because the factor $m_{l+1}/(l+1)!$ in the accuraccy criterion (7) is small.

An example of such zero-moment filters F_l can be obtained by multiplying a Gaussian with appropriate polynomials:

$$F_l = \sum_{i=0}^{l}(a_i P_i(x))g(x).$$

The $P_i(x)$ are Hermite polynomials that are orthogonal with respect to the Gaussian weight function. The coefficients a_i are chosen so that the first l moments of the filter vanish, in accordance with our conditions.

Oversmoothing is only one problem with the Gaussian. The truncation of the infinite filter is also a serious problem because the derivative $g^{(n)}$ becomes meaningless at the ends of the window and contributes a large error [14]. Discrete versions of our method on a finite window are described in detail in [11]. Closed-form filters are derived there that yield differentiation filters of the desired orders. In the following experiments, we use one of these methods, which was based on the Krawtchouk polynomials. They are defined by the condition of orthogonality with respect to the binomial weight function.

In summary, we have been able to obtain good estimation of high derivatives by increasing the filter size σ and its order l to meet the above accuracy criterion.

VI. Experiments

We describe here some experiments done on synthetic shapes. Despite the limitations of such experiments, we were able to obtain a good idea of the reliability of the high derivatives by perturbing the data and looking at the change in the invariants. The results, with the differentiation method that we employed, were very good.

We tested known differential affine invariants. These are invariant to affinities with unit determinants and to changing the curve parameter except for a starting point while involving the fourth derivative. We have the *affine length* and *affine curvature* [8], [4], [16], [17]:

$$\begin{aligned} \tau &= \int_{t_0}^{t} \text{abs}|\mathbf{x}_t, \mathbf{x}_{tt}|^{1/3} dt \\ \kappa_a(t) &= -\frac{5}{9}|\mathbf{x}_t, \mathbf{x}_{tt}|^{-8/3}|\mathbf{x}_t, \mathbf{x}_{ttt}|^2 \\ &\quad + \frac{4}{3}|\mathbf{x}_t, \mathbf{x}_{tt}|^{-5/3}|\mathbf{x}_{tt}, \mathbf{x}_{ttt}| \\ &\quad + \frac{1}{3}|\mathbf{x}_t, \mathbf{x}_{tt}|^{-5/3}|\mathbf{x}_t, \mathbf{x}_{tttt}| \end{aligned}$$

where $\mathbf{x}_t$ denotes a derivative of $\mathbf{x}$, and $|\dots|$ denote a determinant.

Fig. 3 shows a "peanut" shape created as

$$\begin{aligned} x(t) &= 2\cos t \\ y(t) &= \sin t + \frac{1}{2}\sin 3t \end{aligned}$$

and Fig. 4 is a projection of it. Each projection was discretized independently, and the derivatives were calculated from the discrete images. We then plotted the signatures, i.e., κ_a versus τ. The signatures of the two projections are superimposed in Fig. 5. We can see that the match is very good.

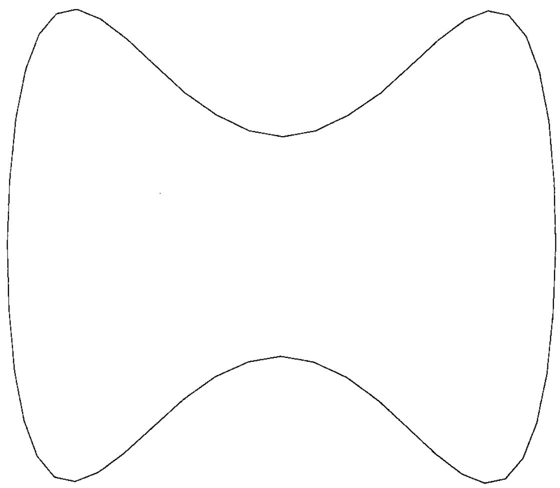

Fig. 3. Projection 1.

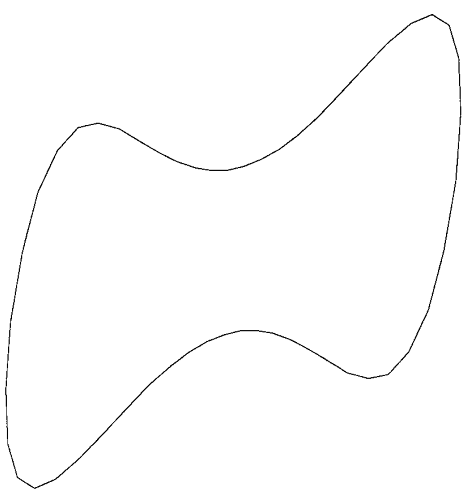

Fig. 4. Projection 2.

To suppress noise in the derivatives, we would like the smoothing parameter factor σ of the filter to be big. However, according to the analysis of the last section, σ cannot be much bigger than the scale of variation of the shape, which here is $s_0 = 1/3$. We chose the order of the filter as $l = 10$ and found that very good accuracy is obtained with $\sigma = 0.4$, which is in line with our accuracy criterion (7). With these parameters, we obtained very reliable and noise-resistant invariants. Perturbing the data by 10% yielded perturbations in the invariants of less than that amount without sacrificing the distinciveness of the curve. Our particular filter with this σ needed a very wide window. A smaller σ would require a smaller window but would result in more sensitivity to noise. This shows that there is no serious problem in using high derivatives as long as the differentiation method and its parameters are chosen properly.

VII. Conclusions

We have developed new methods of obtaining differential invariants and implemented some of them experimentally. We have obtained an invariant signature that can be used to recognize a plane curve regardless of the point of view from which the curve is

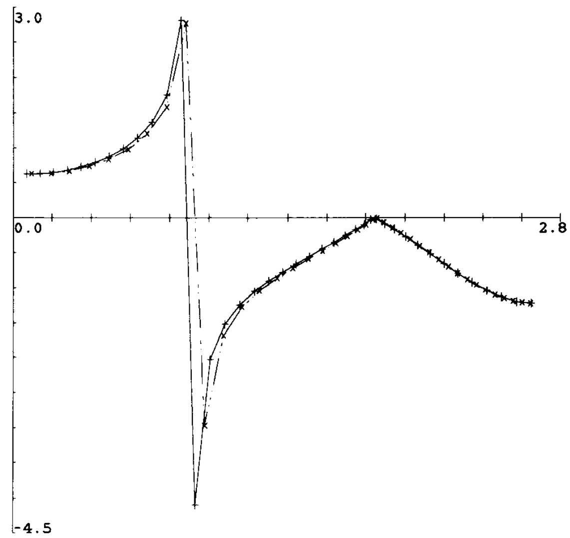

Fig. 5. Invariant signature.

seen. We have found a new method of smoothing and differentiation and showed that it is robust to noise, which makes the signature reasonably reliable. Our differentiation method, and the criterion for its accuracy, have many other significant applications as well.

References

[1] E. Barrett., P. Payton, N. Haag, and M. Brill, "General methods for determining projective invariants in imagery," *CVGIP:IU*, vol. 53, pp. 45–65, 1991.

[2] M. Brill, private communication, 1991.

[3] C. M. Brown, "Numerical evaluation of differentialand semi-differential invariants," Comput. Sci. Dept., Univ. of Rochester, TR-393, Sept. 1991.

[4] A. Bruckstein and A. Netravali, "On differential invariants of planar curves and recognizing partially occluded planar objects," AT&T TR, July 1990.

[5] J. B. Burns, R. Weiss, and E. M. Riseman, "View variation of point set and line segment features," in *Proc. DARPA IU Workshop*, 1990.

[6] R. O. Duda and P. E. Hart, *Pattern Recognition and Scene Analysis.* New York: Wiley, 1973.

[7] D. Forsyth *et al.*, "Invariant descriptors for 3-D object recognition and pose," *IEEE Patt. Anal. Machine Intell.*, vol. 13, pp. 971–991, 1991.

[8] H. Guggenheimer, *Differential Geometry.* New York: Dover, 1963.

[9] Halphen, "Sur les invariants différentiels des courbes gauches," *J. Ec. Polyt.*, vol. 38, no. 1, 1880.

[10] E. P. Lane, *A Treatise on Projective Differential Geometry.* Chicago: Univ. of Chicago Press, 1942.

[11] P. Meer and I. Weiss, "Smoothed differentiation filters for images," *J. Visual Commun. Image Represent.*, vol. 3, pp. 58–72, 1992.

[12] L. Van Gool, J. Wagemans, J. Vandeneede, and A. Oosterlinck, "Similarity extraction and modeling," in *Proc. 3rd Int. Conf. Comput. Vision*, 1990, pp. 530–534.

[13] I. Weiss, "Projective invariants of shapes," in *Proc. DARPA Image Understanding Workshop*, 1988, pp. 1125–1134.

[14] ——, "High order differentiation filters that work," Univ. of Maryland, Cent. for Automat. Res., TR CAR 545, 1991.

[15] ——, "Local projective and affine invariants," Univ. of Maryland, CAR-TR 612, 1992.

[16] ——, "Noise resistant projective and affine invariants," in *Proc. DARPA Image Understanding Workshop*, 1992, pp. 683–692.

[17] ——, "Noise resistant invariants of curves," in *Geometric Invariants in Machine Vision* (A. Zisserman and J. Mundy, Eds.). Cambridge, MA: MIT Press, 1992.

[18] ——, "Geometric invariants and object recognition," *Int. J. Comput. Vision*, vol. 10, no. 3, pp. 201–231, 1993.

[19] E. J. Wilczynski, *Projective Differential Geometry of Curves and Ruled Surfaces.* Leipzig: Teubner, 1906.