Local Invariants For Recognition

Ehud Rivlin and Isaac Weiss

***Abstract*—Geometric invariants are shape descriptors that remain unchanged under geometric transformations such as projection or changing the viewpoint. A new method of obtaining local projective and affine invariants is developed and implemented for real images. Being local, the invariants are much less sensitive to occlusion than global invariants. The invariants' computation is based on a canonical method. This consists of defining a canonical coordinate system by the intrinsic properties of the shape, independently of the given coordinate system. Since this canonical system is independent of the original one, it is invariant and all quantities defined in it are invariant. The method was applied without the use of a curve parameter. This was achieved by fitting an implicit polynomial to an arbitrary curve in a vicinity of each curve point. Several configurations are treated: a general curve without any correspondence and curves with known correspondences of one or two feature points or lines. Experimental results for different 2D objects in 3D space are presented.**

***Index Terms*—Object recognition, invariants, image matching, geometry.**

I. INTRODUCTION

GEOMETRIC invariants are shape descriptors which remain invariant under geometrical transformations such as projection or viewpoint change. They are important in object recognition because they enable us to obtain a signature of an object which is independent of external factors such as the viewpoint. In this paper we treat projective (viewpoint) and affine invariants in various geometrical configurations.

The subject of invariants has recently gained in importance and recognition in the vision community. 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 [6], was used until recently.

Projective invariants of curves and surfaces were first introduced in vision by Weiss [1988]. In that paper we described both algebraic and differential methods for obtaining invariants and pointed out their usefulness for object recognition. A comprehensive review of recent developments in the field is given by Weiss [17].

One can distinguish between two kinds of invariants: global and local. Global invariants describe a shape as a whole so they require knowledge about the whole shape. Examples of global invariants of Euclidean and affine transformations are moment invariants [12] and Fourier descriptors [4]. Global projective invariants were described by Weiss [15]. They have been applied successfully in [7] to industrial objects. Affine invariants of implicit polynomials were used in [13], [10]. Like any global descriptors, these quantities are quite susceptible to occlusion. Although attempts to overcome the problem were made [5], they involved an exhaustive search.

Local invariants are more immune to occlusion. They have been treated by Weiss [15], [16], [18]. So-called mixed (hybrid) invariants were developed in [14], [1], and [3]. In this paper we develop both local and mixed invariants using a new approach that is simpler and more robust to noise than previous methods.

Local invariants are defined at each point of a shape separately, which makes them less sensitive to occlusion. Recognition can be done through an invariant "signature" of the shape. For instance, in the Euclidean case, it is common to plot the curvature against the arc length, both of which are local Euclidean invariants. Such plots or "signatures" of curves can then be matched even if part of a curve is missing due to occlusion. No search is involved. We obtain such signatures in the projective and affine cases.

One can build an object recognition system that uses invariant signatures of curves, rather than the curves themselves, for storage in a visual database and matching. Because of the invariance, the matching does not require a search for the correct point of view. This is possible because of a general property that determines the *uniqueness* and *completeness* of the invariants.

The uniqueness and completeness property of differential invariants can be described as follows. Given a plane curve and a transformation group, there are two independent invariants $I_1(t)$, $I_2(t)$ of the transformations at each point t of the curve. These two invariant functions contain all the information about the curve, except for the transformation to which they are invariant. Accordingly, given two invariants for each curve point, we can reconstruct the original curve up to a transformation belonging to the group.

More accurately, the following theorem holds [8, p. 144]: *All differential invariants of a (transitive) transformation in the plane are functions of two invariants of the lowest order and their derivatives.*

This uniqueness property can be used in a method of building an invariant signature that describes a given curve uniquely, up to the relevant transformation. The method applies to all kinds of local invariants. At each point of the given curve we calculate two invariants, I_1, I_2. We plot these numbers as a point in an "invariant plane" whose coordinates rep-

Manuscript received Nov. 24, 1992; revised June 21, 1994.
I. Weiss was supported in part by Air Force Office of Scientific Research Grant F49620–92–J–0332. E. Rivlin was supported in part by the Defense Advanced Research Projects Agency (ARPA Order No. 6989).
E. Rivlin is with the Computer Science Department at the Israel Institute of Technology; e-mail ehudr@cs.technion.ac.il.
I. Weiss is with the Center for Automation Research, University of Maryland, College Park, MD 20742 USA; e-mail weiss@cfar.umd.edu.
IEEECS Log Number P95024.

resent the two independent invariants. In effect we plot one invariant against the other. In this way the given curve maps into an invariant signature curve in the invariant plane. The signature uniquely identifies the curve regardless of the modifying transformation such as a different viewpoint. Various methods can be used to find I_1, I_2. Examples for the projective and affine cases are described by Weiss [18], [19].

Global invariants are often associated with algebraic methods, and no differentiation is needed (although integration may be used for finding moments). Local invariants involve some form of differentiation. Larger transformation groups need higher orders of differentiation; projective invariants need a higher order of differentiation than affine, which in turn need a higher order than Euclidean invariants.

We deal here mainly with curves. General curves can be treated in several ways. Two main approaches exist in geometry for curve representation: the explicit and the implicit one. In the explicit method a curve is represented as functions of some parameter along the curve, e.g., $x(t)$, $y(t)$. In the implicit approach a curve is represented by a relation $f(x, y) = 0$, without a parameter. The advantage of the implicit approach is that it does not require introduction of a parameter, which is not in fact part of the geometry of the curve itself. The relation between x, y is sufficient to completely characterize the curve. The explicit method makes it easier to obtain closed form formulas for general curves.

In finding invariants, the parameter is undesirable for the following reasons. The essence of finding invariants is the elimination of unknowns from the system, such as the unknown quantities describing the point of view. The parameter is also in general unknown since it can be chosen in an arbitrary way. It has to be eliminated so that the invariants will not depend on it. The more unknowns we have to eliminate, the more information we have to extract from the image, which translates in the explicit method to higher, and less reliable, derivatives. From the viewpoint of fitting the curve to a set of data points, the implicit method minimizes the fitting errors in perpendicular to the curve, while the explicit method tries to also minimize errors in fitting the parameter along the curve, which is geometrically unnecessary and only adds to error accumulation. We will return to this subject.

Most previous work on local invariants [21] was done using the explicit approach. An implicit approach was used in [9] but it did not provide all the invariants and was cumbersome to implement. We present here a simple way of deriving local invariants in the implicit approach, without a curve parameter. The approach is based on transforming the shape to a canonical (intrinsic) system of coordinates, rather than obtaining closed form formulas for the invariants. The canonical method is very general and was first introduced for vision invariants by Weiss [16]. [18] in an explicit representation.

Several kinds of situations will be treated here. The first involves general plane curves without any correspondence information. These require the highest number of derivatives so their signatures are the hardest to obtain. Next, shapes consisting of a curve and one known feature point will be treated. For the feature (or reference) point it is assumed that a correspondence can be established between two images. This enables us to eliminate some of the transformation parameters and reduce the amount of information needed from the curve itself, i.e., the order of the derivatives. Similarly, a curve and two reference points reduce this amount even further. The references mentioned earlier treated these situations with the explicit approach, using derivatives with respect to a curve parameter. We will treat them here without a parameter. We will also treat curves with reference *lines*, which have not been previously treated, to our knowledge.

In the remainder of this section we briefly summarize the notions of projective and affine transformations. Further details are in [17] and [11].

The formation of an image of a plane curve on a planar film, using a "pinhole" camera, can be described as a 2D projective transformation:

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

where T is a nonsingular 3×3 constant matrix, with eight significant parameters.

Affine transformations are a subgroup of the projective ones. If the object is far away from the camera, the affine transformation can be used as a good approximation to the projective one. The affine transformations are linear, and they preserve parallelism in the plane. The affine group contains the smaller subgroup of Euclidean transformations (rotation and translation), along with scaling (in the x and y directions) and shear.

The elements of the matrix T can be identified as

$$T = \begin{pmatrix} aff_1 & aff_2 & trans_x \\ aff_3 & aff_4 & trans_y \\ proj_1 & proj_2 & 1 \end{pmatrix}.$$

The elements marked aff_i represent rotation, scaling, and shear. Together with the translation elements $trans_x$, $trans_y$ they represent the affine group. The $proj_1$, $proj_2$ elements represent tilt and slant, which are nonlinear transformations.

In an affine transformation the $proj_i$ elements above vanish so the transformation is linear:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} aff_1 & aff_2 \\ aff_3 & aff_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} trans_x \\ trans_y \end{pmatrix}.$$

The terms defined above should be distinguished from similar, commonly used terms such as perspective projection or perspective camera. These latter refer to a projection from a 3D object to a 2D image, while the traditional terms used here refer to transformations from nD to nD. In this paper we deal only with transformations in the plane.

II. Finding Local Invariants—A Canonical Approach

Our goal here is to find two local invariants I_1, I_2 at each curve point, so we can plot one against the other. There are several ways to derive such invariants, but the canonical method is probably the simplest and most intuitive. It is also the most general since it can be applied to any other forms of invariants. It was first used for vision invariants by Weiss [16], [18].

The basic idea is to transform our coordinate system to a canonical one, i.e., a standard system which is defined by the intrinsic characteristics of the shape itself. Since this system is intrinsic, all quantities measured in it are independent of the initial system and are therefore invariants. One can give a simple example as follows: Given an image of a rod, we can calculate its length, which is a Euclidean invariant, by applying the formula for Euclidean distance. An alternative way is to transform the coordinate system to a canonical one, in which the rod lies along the x axis and the origin is at one end of the rod. Then the x coordinate of the other end is the rod's length. We see that by moving to a canonical coordinate system we have obtained the invariant length without an explicit formula. This canonical system was determined by the properties of the shape rather than by some external factors.

An important differential example is finding Euclidean invariants of curves. We can move the coordinate system so that the x axis is tangent to the curve $y(x)$ at some point that we choose on it, i.e., $y = y' = 0$ there. The second derivative (y'') at this point is now equal to the curvature and is invariant since we obtain the same canonical system regardless of which system we started with. We see that by determining some of the properties of the system, the others are also determined and become invariant.

We generalize this approach to larger transformation groups. In general, the unknown factors in a transformation can be eliminated by using the same kind of transformation, with the same number of factors, to go over to the canonical coordinates. The Euclidean invariants can be obtained by using a Euclidean transformation to obtain a Euclidean canonical system, etc.

The general projective transformation can be decomposed into simpler transformations: translation, rotation, skewing, scaling (making up the affine group), tilt, and slant. We will use these to move to canonical coordinates step by step. At each step some of the viewpoint parameters will be eliminated until we are left with a coordinate system independent of the original viewpoint and defined by the shape itself.

There are two basic requirements that the canonization process has to meet: it has to be *invariant,* i.e., produce a result that is independent of the original system, and it has to be *local,* i.e., it should be based on curve properties that can be extracted from a small neighborhood around each point.

The Euclidean example above meets these requirements. The requirement of tangency is an invariant one, because the tangency property is unchanged under a projective transformation. The locality requirement is also met, because the tangency means that the first derivative y' vanishes. A derivative is a local property and can be obtained independently at each curve point. We will later deal with the problem of obtaining the derivatives.

For the Euclidean case we used the tangent to obtain a canonization process that met our requirements of invariance and locality. We can generalize the method by using an osculating curve, which 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 can be defined as having more (independent) contact points with the original curve, infinitesimally close to each other. The condition on the derivatives can be written as

$$\frac{d^k}{dt^k}\left(f^*(x,y)-f(x,y)\right)=0, \qquad k=0\ldots n \tag{1}$$

with f^* being the osculating curve, f the given curve, and n the order of contact. Since the derivatives vanish, this condition is invariant to the parameter t. (We will derive the osculating curve without this parameter.) Since it has a geometric interpretation with points of contact, the condition is also projectively invariant. And since it is defined infinitesimally, it is also local. Thus all the independence requirements set forth are met.

In the following we will use an osculating implicit curve f^* satisfying the above condition. This curve will be chosen as the simplest one that meets our needs; its shape is thus known. Thus it will be easier to handle than the original f which can be any function extracted from the image. According to our needs we find either a cubic or a conic which osculates the original curve. We then transform the coordinates so that this cubic or conic takes on a particularly simple, predetermined form, i.e., we eliminate all its coefficients. In this new (canonical) system all quantities are invariants and we pick the ones that best suit our needs.

We will describe the correspondenceless case in full and summarize the other cases.

III. Local Projective Invariants Without Correspondence

We use the osculating curve method to eliminate all the projective unknowns and obtain two local invariants at any curve point. The outline of our method is as follows:

- Repeat the following steps for each pixel that belongs to the curve to obtain two independent invariants at that point of the curve:
  - Define a window around the pixel and fit a suitable curve f to it. f should fit the data as accurately as possible. All the following stages are performed analytically.
  - Find an osculating curve f^* which osculates the above fitted curve f with a known order of contact, at the point at which we want the local invariants. (This is an invariant operation.)
  - Derive a canonical coordinate system by demanding that in it the osculating curve has a simple, predetermined form (e.g., vanishing first derivatives). This ensures that the canonical system is based invariantly on the intrinsic properties of the shape itself and is not dependent on the given co-

ordinate system. Thus we will obtain a unique coordinate system regardless of the original system. By doing so we eliminate all the unknown quantities of the original system (e.g., the viewpoint).

– Transform the fitted curve f to this new system. Since the system is canonical, all shape descriptors defined in it are independent of the original coordinate system and are therefore invariants. Pick two independent local quantities in this system, e.g., certain derivatives, as our invariants I_1,I_2.

- Plot one invariant against the other to obtain an invariant signature curve. This is based on the completeness and uniqueness property discussed above.

In the following sections we will describe the above steps in more detail.

A. Curve Fitting

The first step of the algorithm above involves fitting a curve to the given data points. Since we are dealing with imperfect data, some modeling approximation needs to be made.

The basic assumption we make is that the data represent a smooth differentiable function. Therefore it can be expanded locally in a Taylor series whose higher terms are negligible. It is thus natural to fit a polynomial curve, representing the meaningful terms of the Taylor expansion. However, there is room for exploring other kinds of fitted curves. The fit is not necessarily invariant, but we assume that the fitting error is smaller than the noise, which is also not invariant, so there is harm. This kind of issue is further discussed in [13].

Two main parameters of the fit are the order of the polynomial fit and the size of the fitting window around the point of interest. These should be chosen so as to guarantee a good fit, but the final result should not depend on them. The final result is a function of the first derivatives of the fitted curve, and these are independent of the curve order and of the window size as long as the fit is good.

To determine the order of the polynomial curve, we need to know the minimum number of coefficients needed or the amount of information that needs to be obtained from the image. To find invariants, we have to eliminate the information in the image which is specific to the coordinate system. In the example given earlier of the rod that can only move or rotate in the plane, we can measure the coordinates of the rod's two ends. From these four quantities we can eliminate three, related to the three unknown parameters of the transformation (translations and rotation). This leaves one Euclidean invariant, the length.

Similar enumeration applies to other transformations. In the projective case, we want to eliminate eight parameters of the transformation, so the number of coefficients to be obtained from the image should exceed eight. Since we need two independent invariants at each pixel, we need 10 independent quantities. A cubic has nine coefficients, but we also have the position of the point on the cubic for a total of 10 quantities. Thus it is sufficient from purely geometrical considerations to fit a cubic to our data. However, other considerations push us towards a higher order curve.

We can see here the advantage over the explicit method that requires differentiation of $x(t)$, $y(t)$ with respect to the curve parameter t. The elimination argument above applies to this unknown parameter, i.e., this parameter has to be eliminated along with the coordinates, so that the invariants will be independent of it. This increases the amount of data that needs to be extracted from the image, e.g., the orders of the derivatives. In Wilczynski's method, the eighth derivatives of both x and y were needed, a total of 18 quantities. This reduced the reliability of the invariants. Thus avoiding the parameter from the outset reduces the number of quantities we need to obtain from the image and improves reliability.

We now expand in more detail on the merits of the implicit relative to the explicit representation.

From a purely mathematical viewpoint, namely, for a *noiseless* curve, the two methods are quite similar. They require the same minimal number of data points to obtain the same order of derivative, and it is not hard to transform from one representation to the other (locally). For instance, given nine points, we can pass through them either an implicit cubic or two explicit eighth-order polynomials. Both will provide an eighth derivative. However, in real life we have hundreds of noisy pixels, and fitting a shape to them is only an approximation. Fitting thus implies making some assumptions about the noise or modeling the noise. The explicit and implicit methods differ greatly in their suitability for dealing with different noise models.

In the explicit method, we have the two separate functions $x(t)$, $y(t)$. This is most suitable to a noise model that assumes independent errors in the x and y directions. That is, the noise error can be decomposed into two independent components along the axes. With the same noise model, the error can also be decomposed into a component perpendicular to the curve and one tangent to it, namely, going along the parameter t. This kind of noise model is not particularly useful for our purpose because we are not interested in errors of fitting t along the curve. These errors have no geometric meaning since the parameter t is not part of the geometry of the curve; it was introduced only for convenience. This parameter, along with the fitting error along it, will later have to be eliminated when finding the invariants.

The implicit method, on the other hand, does not have a parameter in the first place so it does not have fitting errors along the parameter. It is suited to a noise model which considers an error only perpendicular to the curve. In theory, there may be a way to use the explicit representation too with this noise model, but that would be rather complicated and unnatural. In any case, the implicit method has half as many variables to fit as the explicit one does because it does not have to describe the dependence on the parameter. In other words, it has fewer noisy variables with which to deal.

The practical importance of this can be seen if we look into the way that errors propagate in the calculation of invariants. The invariant equations are nonlinear, and errors in variables do not cancel out even if the variables themselves cancel out. In the explicit method, the parameter is eliminated in the course of finding invariants, but the errors associated with it do not cancel out. In many situations these errors can be quite large. The implicit method does not have to deal with a parameter or any errors along it to begin with, so there are fewer errors. In a sense, the implicit method eliminates the parameter at an earlier stage than does the explicit method, namely, at the curve fitting stage rather than at the invariant finding stage. This way the nonlinear accumulation of errors in t at the later stage is eliminated.

Another important difference is the "conditionality" of the problem. In the fitting process, one has to solve a system of equations for the coefficients, and the condition number of the system is a measure of the stability of the solution to small errors in the data. This condition number is related to the ratio of the biggest eigenvalue to the smallest one. Generally speaking, big disparities in the magnitudes of the quantities in the equation lead to high condition number and poor stability. A polynomial with powers of eight has much more disparity of magnitudes among its values than does a cubic, whose highest power is three. Therefore fitting the cubic has a much lower conditionality and is thus much more robust. Again, in an ideal world without round-off and noise errors, this would not make a difference, but in the real world it can be a crucial difference.

It may be argued that t can be avoided by using x as a parameter, but this does not address the other problems above: it is the wrong noise model (no error allowed along x) and has high conditionality.

Similar arguments apply to the case of the semidifferential invariants, to be discussed later.

The other important fitting parameter is the size of the window. It was found by Weiss [20] that the wider the window, the more reliable the fitting becomes. This is because a larger window averages out random noise better. Thus we need to find a function that fits the data over a large window. This means using a higher order Taylor approximation of the data curve, namely, a higher order implicit polynomial. As mentioned before, the order and the window size have no influence on the invariants as long as the fit is good. This is due to the fact that the invariants are functions of local derivatives. The higher window size and curve order are needed to ensure a good fit, in the sense of good noise suppression.

In practice we have found it convenient to use a quartic implicit polynomial, although a cubic would be enough in the noiseless case. In the sequel we will deal with the fitted quartic

$$\begin{aligned} f(x, y) = a_0 + a_1x + a_2y + a_3x^2 \\ + a_4xy + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 \\ + a_9y^3 + a_{10}x^4 + a_{11}x^3y + a_{12}x^2y^2 + a_{13}xy^3 + a_{14}y^4 = 0 \end{aligned} \tag{2}$$

with the cubic being the special case in which the coefficients $a_{10}, \ldots, a_{14}$ vanish.

Once the curve order and window size have been chosen, the fitting itself can be done by standard methods. Simple least square fitting is quite ill-conditioned because of the relatively large number of unknowns. The Singular Value Decomposition method is very successful in overcoming this problem, and we obtain a quite reliable fit.

We have thus obtained a local algebraic (parameterless) representation for the data around some curve point. We will now find its invariants (analytically).

B. Deriving a Canonical Coordinate System

We will find the canonical system in stages, eliminating more unknowns at each stage. As a convention, we denote the new coordinates after each canonization step by $\bar{x}, \bar{y}$ and drop the bars before going to the next step, and similarly for other quantities.

Eliminating the Euclidean Unknowns. First we will detail the Euclidean canonization stage, in which the unknown translation and rotation are eliminated. These transformations are the smallest subgroup of interest here, therefore this stage is common to all our canonizing schemes.

The first step is eliminating translation, done by moving the origin to our curve point. Our pixel x_0, y_0 does not necessarily lie on the fitted curve but it is close to it. Thus, we find a point $x_0, \overset{*}{y}_0$ which does lie on the curve, i.e., we solve (2) for $\overset{*}{y}_0$, given x_0. This is easy to do with Newton's method because y_0 is a close initial guess. We now translate the origin to $x_0, \overset{*}{y}_0$. (We could simplify the solution by first translating so that $x_0 = 0$ and then solving for $\overset{*}{y}_0$.

We now transform the curve coefficients to the new system and obtain new $\bar{a}_i$. This is done by expressing the old coordinates in terms of the new, $\bar{x} = x - x_0, \bar{y} = y - \overset{*}{y}_0$, substituting in (2) and rearranging. In this new system we have $\bar{a}_0 = 0$ which can be seen by simply substituting the point (0, 0) in equation (2). We now drop the bars from $\bar{y}_0$, $\bar{a}_i$.

The next step is to rotate the coordinates so that the x axis will be tangent to the curve. It is easy to see that in the rotated system we must have $\bar{a}_i = 0$ (because $df(x, y)/dx = 0$). To satisfy this condition we again express the old coordinates in terms of the new, with the rotation factor u_r

$$x = (\bar{x} + u_r\bar{y})/v \qquad y = (\bar{y} + u_r\bar{x})/v \tag{3}$$

Here v is a normalization factor $\left(1 + u_r^2\right)^{1/2}$ that makes the transformation orthogonal. Now a_1 is transformed to

$$\bar{a}_1 = a_1 - u_r a_2.$$

To make this vanish we thus have to rotate by the amount

$$u_r = a_1/a_2.$$

Since translation and rotation make up the Euclidean transformations, we have reached a Euclidean canonical system. All quantities defined in it are Euclidean invariants. The curvature at x_0 is now simply the second derivative, d^2y/dx^2. The arc length is $|dx|$ since $dy = 0$.

Eliminating the projective unknowns. Of the eight parameters of the general projectivity we have already eliminated three by translation and rotation, so our osculating curve should have five coefficients, while passing through the origin and being tangent to the x axis. Following [9] we choose the "nodal cubic" (Fig. 1)

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

This curve intersects itself at the origin so it has two tangents there, one lying along the x axis. The other tangent is called the "projective normal." Our treatment of the nodal cubic differs from Halphen's and yields the full range of invariants. (We also had the advantage of a symbolic manipulation program.)

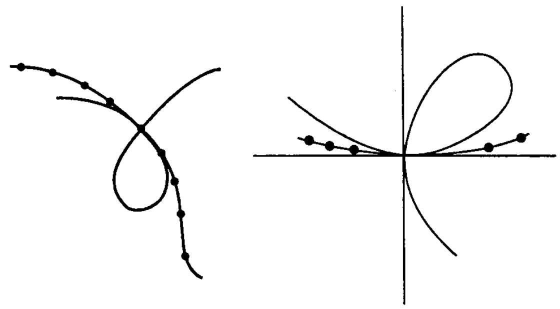


Fig. 1. Osculating nodal cubic (left), folium of Descartes (right).

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

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

It is known [2] as a *folium (leaf) of Descartes*. 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 too, obtaining the projective canonical system.

We will now find the nodal cubic f^* using the osculation condition, i.e., the equality of the first n derivatives of f and f^* (1). The first derivative (and the 0th) vanish because of the tangency to the x axis. To determine the five coefficients c_i we need five more derivatives to be equal, i.e., up to the sixth one. The condition of equal derivatives ensures the locality of the treatment and also its invariance, as discussed earlier.

To go further, we need to calculate the derivatives d^ny/dx^n of the fitted curve. This is done analytically from $f(x, y)$. To do it we use the fact that all the total derivatives of f vanish, since f vanishes identically, equation (2). The first total derivative, for example, is

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = 0.$$

This is a linear equation for dy/dx. It is superfluous because we have already demanded its vanishing (tangency). However, each successive differentiation gives one linear equation for one higher $y^{(n)}$, in terms of lower derivatives. The calculation is tedious, and we used a symbolic manipulation program to calculate up to $y^{(8)}$ in terms of a_i.

Setting $a_2 = 1$ and denoting

$$d_n = \frac{1}{n!}\frac{d^ny}{dx^n}(0)$$

we have

$$d_2 = -a_3 \qquad (6)$$

$$d_3 = -a_6 - d_2a_4 \qquad (7)$$

$$d_4 = -a_{10} - d_2a_7 - d_2^2a_5 - d_3a_4 \qquad (8)$$

$$d_5 = -d_2a_{11} - d_2^2a_8 - d_3a_7 - 2d_2d_3a_5 - a_4d_4 \qquad (9)$$

$$d_6 = -d_2^2a_{12} - d_3a_{11} - d_2^3a_9 - 2d_2d_3a_8$$
$$-d_4a_7 - a_4d_5 + (-2d_2d_4 - d_3^2)a_5. \qquad (10)$$

Given these derivatives we find the coefficients c_n of the nodal cubic as follows. We write the nodal cubic as

$$y(x) = \sum_{n=0}^{6} d_nx^n$$

and substitute it in the cubic expression, (4). Collecting terms with the same power x^n we obtain five equations for the five c_i in terms of d_n. Their solution is

$$c_0 = -d_2 \qquad (11)$$

$$c_1 = \frac{d_2^4\left(d_4d_6 - d_5^2\right) + d_2^3\left(-d_3^2d_6 + 2d_3d_4d_5 - 2d_4^3\right) + 3d_2^2d_3^2d_4^2 - 3d_2d_3^4d_4d_3^6}{d_2^7d_5 - 3d_2^6d_3d_4 + 2d_2^5d_3^3}$$

$$c_2 = \frac{d_2^3(d_3d_6 - d_4d_5) + d_2^2\left(d_3d_4^2 - 3d_3^2d_5\right) + 5d_2d_3^3d_4 - 3d_3^5}{d_2^5d_5 - 3d_2^4d_3d_4 + 2d_2^3d_3^3} \qquad (12)$$

$$c_3 = \frac{d_2^3d_6 + d_2^2\left(-5d_3d_5 - d_4^2\right) + 13d_2d_3^2d_4 - 7d_3^4}{d_2^3d_5 - 3d_2^2d_3d_4 + 2d_2d_3^3} \qquad (13)$$

$$c_4 = -\frac{d_2^3d_6 + d_2^2\left(-4d_3d_5 - d_4^2\right) + 10d_2d_3^2d_4 - 5d_3^4}{d_2^4d_5 - 3d_2^3d_3d_4 + 2d_2^2d_3^3}. \qquad (14)$$

Having found the coefficients c_i, we set out to eliminate them. First, we orthogonalize the axes, i.e., skew the system so that the two nodal tangents become perpendicular. This will eliminate the term with c_4 in the nodal cubic. Our skewing transformation is

$$x = \bar{x} + u_sy$$

with $u_s = -c_4$ being the skewing factor. y remains unchanged. Substituting the above equation in the cubic (4) and rearranging we obtain new coefficients

$$\bar{c}_1 = -c_0c_4^3 + c_3c_4^2 - c_2c_4 + c_1 \qquad (15)$$

$$\bar{c}_2 = 3c_0c_4^2 - 2c_3c_4 + c_2 \qquad (16)$$

$$\bar{c}_3 = c_3 - 3c_0c_4. \qquad (17)$$

We again drop the bars from c_i and x.

One advantage of the orthogonalization is that it makes it possible to decouple the next transformations, i.e., slantings and scalings in the x and y directions. We can now proceed with these transformations in any order to eliminate the remaining c_i.

We next scale the axes with the scaling factors s_x, s_y:

$$x = \bar{x}/s_x, \quad y = \bar{y}/s_y \tag{18}$$

where

$$s_x = c_0^{2/3} c_1^{1/3}, \; s_y = c_0^{1/3} c_1^{2/3}.$$

Substituting this in the orthogonalized cubic we obtain

$$\bar{x}^3 + \bar{y}^3 + \bar{c}_2 \bar{x}\bar{y}^2 + \bar{c}_3 \bar{x}^2 \bar{y} + \bar{x}\bar{y} = 0 \tag{19}$$

$$\bar{c}_2 = \frac{c_2}{s_y}, \quad \bar{c}_3 = \frac{c_3}{s_x}.$$

These quantities are *local affine invariants* because we have reached an affine canonical system. We have used all possible affine transformations (translation, rotation, skewing, scaling) to eliminate all the possible affine transformation factors and arrive at the above form of the cubic so the remaining coefficients are uniquely defined regardless of which system we started with.

A projective canonical system is obtained by eliminating the last two coefficients using slants, which are purely projective, in the x and y directions. To do that, we drop the bars from the last cubic form (20) and express x, y there in terms of new coordinates, the projective canonical $\bar{x}, \bar{y}$:

$$x = \frac{\bar{x}}{1 + \sigma_x \bar{x} + \sigma_y \bar{y}} \quad y = \frac{\bar{y}}{1 + \sigma_x \bar{x} + \sigma_y \bar{y}} \tag{20}$$

with the x- and y-slant factors

$$\sigma_x = -c_3, \qquad \sigma_y = -c_2.$$

This finally brings us to Descartes' folium (5).

This concludes the elimination of the cubic coefficients and brings us to the projective canonical system. This system was defined invariantly by the curve's intrinsic properties such as the shape of the osculating nodal cubic which is independent of the original coordinate system.

C. Projective Invariants

We now have an invariant canonical system and affine invariants, but still no projective invariants. To obtain them, we transform the original fitted curve f, equation (2), to our canonical system. We collect all the transformations that were performed during the canonization process. We have already translated and rotated f (with the factors x_0, y_0, u_r), and we will perform the rest of the transformations making up the projectivity (with factors u_s, σ_x, σ_y, s_x, s_y) on f. The coefficients of f will transform to new ones $\bar{a}_i$, which are now all invariants because they represent a fitted curve defined in the invariant system. The only remaining question is how to select functions of the invariants $\bar{a}_i$ which best suit our needs.

As mentioned before, the condition of locality dictates that we use derivatives of the curve rather than some arbitrary function of $\bar{a}_i$. The first six derivatives at x_0 are already determined by the canonization process (as

$$d_0, \ldots, d_6 = 0, 0, -1, 0, 0, 1, 0).$$

Thus we need the seventh and eighth derivatives. These can be obtained in this particular system similarly to (6) through (10). With the above values of d_n we have (dropping the bars)

$$d_7 = a_{13} - a_7 + 2a_5 \tag{21}$$

$$d_8 = -a_{14} - a_{11} + 2a_8 - a_4 d_7. \tag{22}$$

These quantities are our *local projective invariants.* They can serve as our I_1, I_2 in this case.

In conclusion, we have started with a curve fitted to data points around x_0, y_0, and after a series of transformations of this curve we have arrived at local invariants which are independent of the fitting details or the point of view. We can repeat the process for other points to obtain an invariant signature. No correspondence was needed.

D. Affine Normal and Canonical System

We have obtained affine canonical system as a by-product of the projective one. However, there is a more direct and simpler way. We only touch on it here, full details are given by Weiss [19]. Instead of the nodal cubic we use a conic passing through our point $\mathbf{x}_0$ (24). We draw a conic diameter through that point, an affine invariant operation. This line is termed the *affine normal.* Through affine transformation we make this line perpendicular to the tangent and reduce the conic to a simple unit circle or hyperbola, thus obtaining an affine canonical system.

E. Experimental Implementation

The above method was implemented to extract local invariants from a set of real images. Each image was processed to obtain a contour curve for the relevant object, using standard techniques of edge detection and thinning. We used a window about 50 pixels wide around each contour point and fitted an implicit curve there, minimizing the square distances with Singular Value Decomposition. The coefficients of this fitted curve were used to calculate the invariants.

Fig. 2 shows two views of a hanger. Effects of perspective distortion can be seen. Fig. 3 presents the hanger under occlusion. Fig. 4, Fig. 5, and Fig. 6 show the local invariants for the above-mentioned figures. A good match of the signatures is observed. The match is demonstrated in Fig. 7 where the two signatures from the different viewpoints are superimposed and almost overlap. The match is between the hanger in Fig. 4 and in Fig. 6, where it is partially occluded. Note that the number of points and their locations along the signature are different in the two signatures due to the different size and viewpoint. However the signature curve is invariant to this.

The matches between the signatures were done by observation. Devising an automated matching method is an open problem in vision and deserves research in its own right. We mention here only one possibility. A method for automatic matching of the signatures was successfully used by Wolfson [22], for the Euclidean case (curvature vs. arc length): draw a circle of radius ε around a point in one signature, and measure how much of the other signature enters inside that circle. This gives a measure of the local overlap between the two signatures, taking into account the noise level ε. Then move the circle along all points of the signature and repeat the process for each point. Add up the local similarity measurements to obtain a global measure of the similarity.

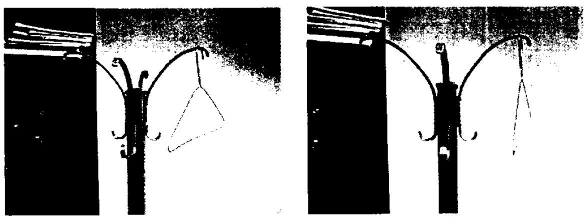


Fig. 2. Two views of a hanger.

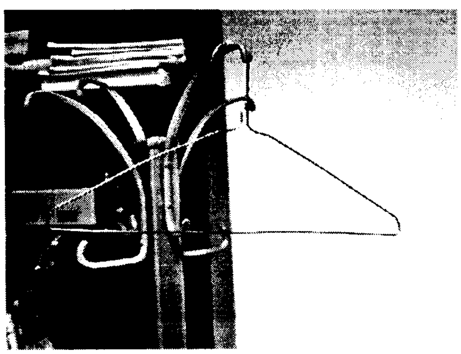


Fig. 3. The hanger under partial occlusion.

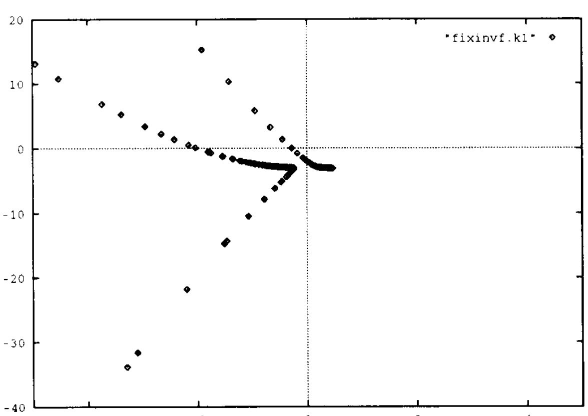


Fig. 4. The projective invariant signature for the first hanger image.

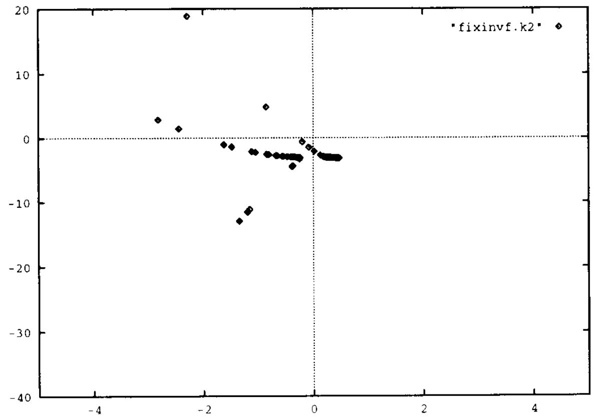


Fig. 5. The projective invariant signature for the second hanger image.

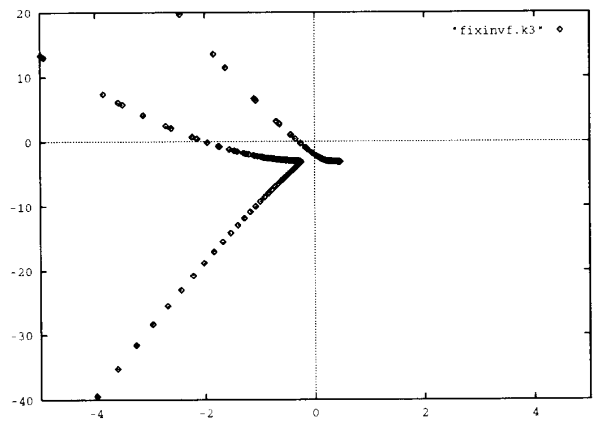


Fig. 6. The projective invariant signature for the occluded hanger image.

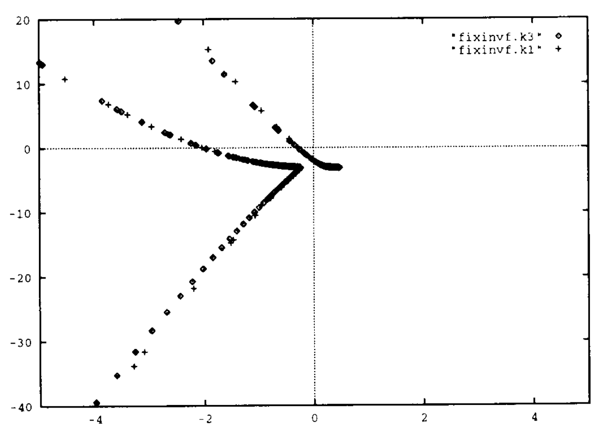


Fig. 7. The projective invariant signature for the occluded hanger image superimposed on the signature of the unoccluded hanger.

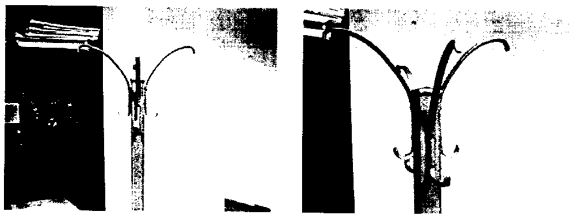


Fig. 8. Two views of a coat rack. The vertical lines in the curved piece of metal were used for invariants extraction.

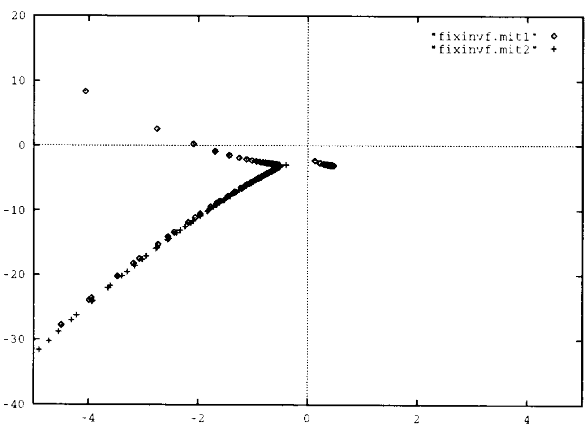


Fig. 9. The invariant signatures for the coat rack. The signatures are presented one on top of the other.

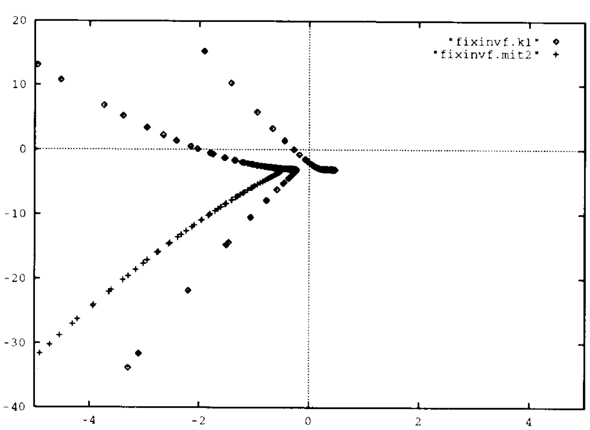


Fig. 10. The invariants signature for the second object (the coat rack) is presented on top of the signature of the first object (the hanger).

Fig. 8 shows a different object, a coat rack, from two different views. We tested the parts of the rack on which a coat hangs. These parts are somewhat similar in character to the hanger (under projectivity) but are still distinct from it. Accordingly, the signature has some similarity to the previous one but it is different enough to distinguish the hanger from the coat rack.

The invariant signatures are presented (one on top of the other) in Fig. 9 . The local invariants obtained from Fig. 8 are compared with those of the first hanger (Fig. 2). The result of this comparison is presented in Fig. 10.

IV. Local Invariants With Some Correspondence

While the previous process does not require correspondence, it leads to fitting rather high order curves which may be sensitive to noise. This problem is discussed by Weiss [20], and it is shown that one way of overcoming it is using a wide window.

Another approach to increasing robustness is to use some reference features, e.g., points or lines for which the correspondence is known. For example, a silhouette of an airplane can contain both curved parts and straight lines. We can use this information to eliminate some of the parameters of the projective or affine transformation, so there will be a need for fewer curve descriptors for the elimination of the remaining ones. Invariants involving both derivatives and reference points were found by Barrett et al and Van Gool et al [14]. However, they still use a curve parameter t which also has to be eliminated, and this reduces the robustness of their method.

The "parameterless" method described above is perfectly suited for this situation, and again leads to saving in the number of data quantities needed from the image and increased reliability. Here we use a canonical method similar to the correspondenceless case in order to find local invariants while avoiding the curve parameter. This makes the method more robust as there are fewer unknowns to eliminate. In addition to reference points used by previous methods, we are able to make use of reference lines, or combinations of points and lines.

The first stage is similar to the previous case: fit a high order curve over some window around some x_0, y_0 and then translate and rotate until the origin is at x_0, y_0 and the x axis is tangent to the curve. We need a smaller window than before and a lower order curve because we need lower derivatives.

Again we obtain an osculating curve that will help us find the canonical system. However, we do not need the nodal cubic; the conic, with three parameters, will suffice in all cases:

$$f^* = c(x, y) = c_0x^2 + c_1y^2 + c_2xy + y = 0. \qquad (23)$$

The exact process of finding the conic and the canonization process differs for each case. However, the principles of invariance and locality must be maintained. In the following we will describe briefly the process for the different possible combinations. Each known feature point or line reduces the number of derivatives needed by two, because it eliminates two transformation factors.

- **A Curve and One Feature Point**: We draw a line joining the given reference point x_1, y_1 with the curve point x_0, y_0 (Fig. 11). This is obviously a projectively invariant operation. We use this line as our new y axis. As before we skew the system so that this line becomes perpendicular to x. We thus obtain an orthogonal system which we can scale and slant as before.

To do this, we obtain an osculating conic to our fitted curve f. We need only fourth order contact, rather than sixth as before.

After fitting the conic, our goal will be to go over to a canonical system in which this conic is a unit parabola $x^2 + y = 0$, and the distance between the curve point and the reference point is unity (right-hand side of Fig. 11).

- **A Curve and One Feature Line**: We convert to the previous case by finding the polar point of the given line with respect to the osculating conic. Polarity of a point and a line is an invariant relation. Given a point, we can

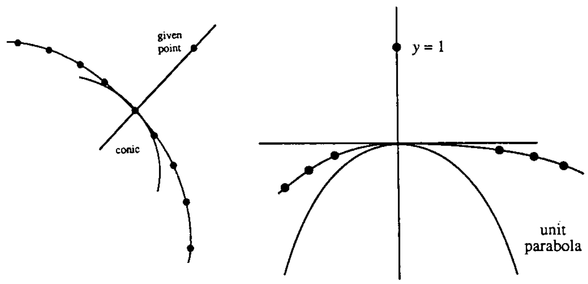


Fig. 11. Osculating conic (left), the canonical conic and a point (right).

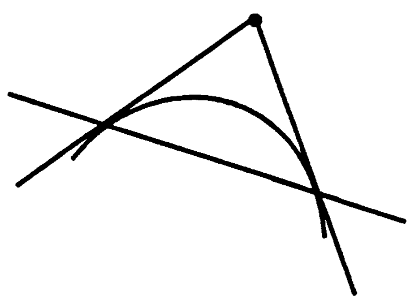


Fig. 12. Polar line and point.

draw from it two tangents to the conic, creating two points at which these tangents touch the conic. The line joining these two points is the polar line of the given point with respect to the conic (Fig. 12).

The conic is found in the same way as in the previous case, requiring osculation in the fourth derivatives. We are now in the same situation as in the previous case, having a conic and a point, and we can proceed to find invariants as before.

- **A Curve and Two Feature Points**: This case requires only the second derivative to determine the osculating conic, rather than the fourth as before. First find the conic that osculates the fitted curve with second order contact and also passes through the two reference points. This uniquely determines the conic. Then find the line that passes through the two reference points. This brings us to the same situation as before, namely, a conic plus a line, but with two fewer derivatives.
- **A Curve and Two Feature Lines**: This case too requires only the second derivative to determine the osculating conic, rather than the fourth as in the one point case. We first find the conic that osculates the fitted curve with second order contact and is also tangent to the two reference lines. We then find the intersection point of the reference lines. This brings us to the case of a conic plus a point that we dealt with earlier, but with two fewer derivatives.
- **A Curve, a Point, and a Line**: As before we require that the conic osculate the fitted curve up to second order contact. In addition we require that the reference line be polar to the reference point with respect to the conic. This provides sufficient conditions to determine the conic. Achieving this will bring us again to the situation of a conic plus a point, to be brought to a canonical system as before, again with two fewer derivatives.

In what follows we describe the above processes in detail, as well as some experimental results for some of the cases.

A. Transforming to a Euclidean Canonical System

In all of the above processes the reference points and lines need to be transformed to the Euclidean canonical system. For a feature point x_1, y_1 the transformation is

$$\bar{x}_1 = (x_1 - x_0 - u_r(y_1 - y_0))/(1 + u_r^2)^{1/2} \quad (24)$$

$$\bar{y}_1 = (y_1 - y_0 + u_r(x_1 - x_0))/(1 + u_r^2)^{1/2}. \quad (25)$$

(This involves the inverse of the rotation of the curve f, equation (3), because points transform with the inverse of a curve transformation.)

The reference (feature) line $b_0 + b_1 x + b_2 y$ is translated and rotated as

$$\bar{b}_0 = b_0 + b_1 x_0 + b_2 y_0 \quad (26)$$

$$\bar{b}_1 = b_1 - u_r b_2 \quad (27)$$

$$\bar{b}_2 = b_2 + u_r b_1. \quad (28)$$

We again drop the bars from all quantities.

B. A Curve and One Feature Point

We first find the first four derivatives of f using (6) through (8). From that we find the coefficients of the osculating conic in the same way we used for the nodal cubic. The result is

$$c_0 = -d_2 \quad (29)$$

$$c_1 = -(d_2 d_4 - d_3^2)/d_2^3 \quad (30)$$

$$c_2 = -d_3/d_2. \quad (31)$$

To orthogonalize the system, we want to obtain $\bar{x}_1 = 0$. This is achieved by skewing, with the skewing factor $u_s = x_1/y_1$. The orthogonalization changes the conic coefficients to

$$\bar{c}_1 = c_1 + c_0 u_s^2 + c_2 u_s \quad (32)$$

$$\bar{c}_2 = c_2 + 2c_0u_s. \quad (33)$$

We drop the bars from c_i. The reference point coordinates are now $(0, y_1)$.

For the affine case we need only scaling (19).

It is easy to obtain a distance of unity between the origin and the reference point by scaling the y axis with $s_y = \pm 1/y_1$. (The sign is set to be the same as the sign of c_0.) Scaling in the x direction is done by requiring $c_0 = 1$, which is achieved by

$$s_x = \sqrt{(c_0 s_y)}.$$

Substituting the scaling transformation (19) in the conic (24) we obtain (dropping bars)

$$x^2 + \frac{c_1}{s_y}y^2 + \frac{c_2}{s_x}xy + y = 0.$$

The two remaining coefficients, c_1/s_y and c_2/s_x, are affine invariants. (The conic now is not a unit parabola but has these two invariant coefficients.)

For projective invariants, we first have to slant the shape in the x and y directions. (This has to be done *before* scaling). The terms with c_1, c_2 are eliminated using the transformation (21) with the x, y slant factors being $\sigma_x = -c_2$, $\sigma_y = -c_1$.

As in the affine case we use the reference point for scaling, but now its distance has changed because of the slant. The new distance is now $y^p = y_1/(1 - \sigma_y y_1)$. We want to scale y so that this distance is unity, so $s_y = \pm 1/y^p$ (again with the sign of c_0).

At this point the conic is reduced to $c_0x^2 + y/s_y = 0$. To obtain a unit parabola and get rid of c_0 we scale in the x-direction with

$$s_x = \sqrt{(c_0 s_y)}.$$

We have thus obtained the projective canonical system. To obtain the invariants, we have to transform the original fitted curve f to this system. Again all the transformed a_i are invariants, but we need the ones that are local in nature and independent of the fitting details, namely, derivatives. Since we have used up the first four derivatives we need the fifth and sixth (two fewer than in the correspondenceless case). To obtain them we substitute in (9), (10) the canonical values $d_0, \ldots, d_4 = 0, 0, -1, 0, 0$ and obtain

$$d_5 = a_{11} - a_8 \quad (34)$$

$$d_6 = a_9 - a_{12} - a_4 d_5. \quad (35)$$

These are our local projective invariants.

C. A Curve and One Feature Line

The conic is found in the same way as in the previous case, requiring osculation in the fourth derivatives. The polar line is found as follows.

Given a point x_1^h in homogeneous coordinates, we can write the coefficients b_i of its polar line with respect to a homogeneous conic

$$C = c_0(x^h)^2 + c_1(y^h)^2 + c_2x^hy^h + y^hz^h = 0$$

as

$$b_0 = \left.\frac{\partial C}{\partial z^h}\right|_{x_1^h} = y_1^h \quad (36)$$

$$b_1 = \left.\frac{\partial C}{\partial x^h}\right|_{x_1^h} = 2c_0x_1^h + c_2y_1^h \quad (37)$$

$$b_2 = \left.\frac{\partial C}{\partial y^h}\right|_{x_1^h} = 2c_1y_1^h + c_2x_1^h + z_1^h. \quad (38)$$

(C is first differentiated and then the point coordinates x_1^h are substituted in the right-hand side.) In our case we know the line b_i and the conic C in the above equation, so we have a set of linear equations for the point x_1^h. Solving it we obtain

$$x_1^h = -b_1 + c_2b_0 \quad (39)$$

$$y_1^h = -2c_0b_0 \quad (40)$$

$$z_1^h = b_1c_2 - 2c_0b_2 + (4c_0c_1 - c_2^2)b_0. \quad (41)$$

Going back to regular coordinates we have the polar point in our Euclidean canonical system

$$x_1 = x_1^h / z_1^h, \quad y_1 = y_1^h / z_1^h.$$

We are now in the same situation as in the previous case, having a conic and a point in a Euclidean canonical system, and we can proceed to find invariants as before.

D. A Curve and Two Feature Points

We need here the formula for the conic coefficients given in terms of the second derivative and the reference points:

$$c_0 = -d_2 \quad (42)$$

$$c_1 = \frac{c_0(x_1x_2^2y_1 - x_1^2x_2y_2) - y_1(x_2y_2 - x_1y_2)}{x_2y_1^2y_2 - x_1y_1y_2^2} \quad (43)$$

$$c_2 = \frac{-c_0(x_2^2y_1^2 - x_1^2y_2^2) + y_1y_2^2 - y_1^2y_2}{x_2y_1^2y_2 - x_1y_1y_2^2} \quad (44)$$

with x_1, y_1, x_2, y_2 being the reference point coordinates in the Euclidean canonical system.

The first line above is the same condition on c_0 as in all previous cases. The next two lines are obtained by substituting the reference points in the conic (24) and solving for c_1, c_2.

The affine invariants are calculated from c_i as in the previous case. The projective invariants are now the third and fourth derivatives, two lower than before. Substituting $d_2 = -1$ in (6) we obtain for these

$$d_3 = -a_6 + a_4 \quad (45)$$

$$d_4 = -a_{10} + a_7 - a_5 - a_4d_3 \quad (46)$$

which are our local projective invariants.

E. A Curve and Two Feature Lines

The only new thing here is finding the conic. The tangents to a conic satisfy the equations of the "line conic," which is the dual of a regular conic. When representing the conic in matrix notation, the line conic matrix is the inverse of the point conic matrix. The inverse matrix of the one in (24) is

$$C^{-1} = \begin{pmatrix} 1 & 0 & -c_2 \\ 0 & 0 & 2c_0 \\ -c_2 & 2c_0 & c_2^2 - 4c_0c_1 \end{pmatrix}.$$

c_0 is determined as before by the second derivative d_2. The reference lines satisfy the equations $\mathbf{b}C^{-1}\mathbf{b}' = 0$, from which c_1, c_2 can be found. We obtain the conic

$$c_0 = -d_2 \tag{47}$$

$$c_2 = \frac{\left(b_0'^2 b_1^2 - b_1'^2 b_0^2\right)/2 + 2c_0\left(b_2 b_0 b_0'^2 - b_2' b_0' b_0^2\right)}{b_1 b_0 b_0'^2 - b_1' b_0' b_0^2} \tag{48}$$

$$c_1 = \frac{b_1^2 - 2c_2 b_0 b_1 + 4c_0 b_0 b + c_2^2 b_0^2}{4c_0 b_0^2} \tag{49}$$

with b_i, b'_i being the coefficients of the two reference lines in the Euclidean canonical system.

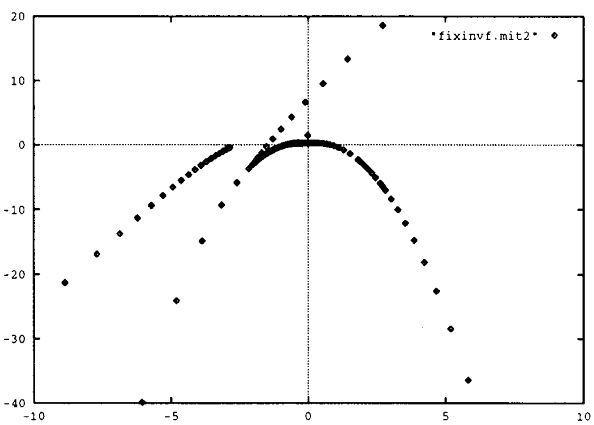


Fig. 13. The invariant signature for the coat rack image.

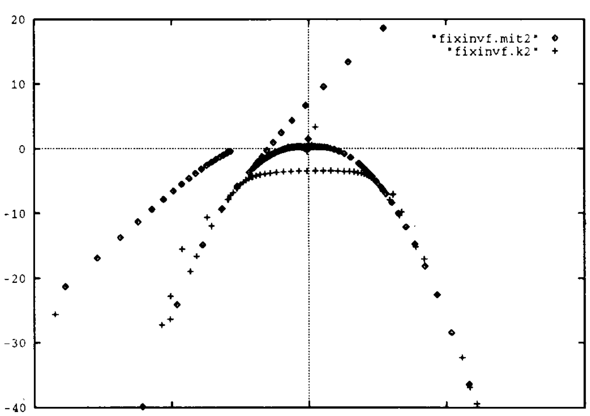


Fig. 14. The invariant signatures for the two images of the hanger and the coat rack presented on top of each other.

The affine and projective invariants are as in the previous case.

The images of the hanger and the coat rack were used to derive local signatures using two feature lines. The signatures obtained from the coat rack image are presented in Fig. 13. A comparison of the two signatures for the hanger and the coat rack is presented in Fig. 14.

The curve and two feature lines method was used to achieve affine invariants for the same objects. The results of the invariants computation are presented in Fig. 15.

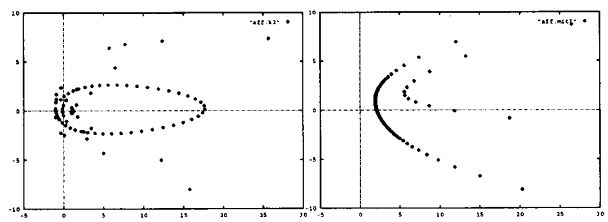


Fig. 15. The two affine invariant signatures for the hanger and the coat rack.

F. A Curve, a Point, and a Line

We require that the conic osculate the fitted curve up to second order contact. In addition we require that the reference line be polar to the reference point with respect to the conic. This provides sufficient conditions to determine the conic. Achieving this will bring us again to the situation of a conic plus a point, to be brought to a canonical system as before, again with two fewer derivatives.

As before, the osculation condition leads to $c_0 = -d_2$. Setting $z_1^h = 1$, the first of the polar equations, (37), leads to $y_1 = b_0$, and the line coefficients have to be normalized so that this equation is satisfied. This leads to the change

$$\overline{b_1} = b_1 y_1 / b_0, \qquad \overline{b_2} = b_2 y_1 / b_0$$

and we drop the bars. With this, the remaining two polar equations are

$$2c_0 x_1 + c_2 y_1 = b_1 \qquad 2c_1 y_1 + c_2 x_1 + 1 = b_2$$

which are satisfied by the conic coefficients

$$c_1 = ((b_2 - 1)y_1 + 2c_0 x_1^2 - b_1 x_1)/(2y_1^2) \tag{50}$$

$$c_2 = -(2c_0 x_1 - b_1)/y_1. \tag{51}$$

The affine and projective invariants are calculated as in the previous two cases.

V. Conclusions

We have presented a method for finding local projective and affine invariants and applied it to real images. Local invariants are useful for recognition even when part of the shape is missing or occluded. They will enable us to look for a match in a database when only a partial shape is given. This is in contrast to global methods such as algebraic invariants and moments. Our method of deriving these invariants consists of defining a canonical coordinate system by the intrinsic properties of the shape, independently of the given coordinate system. Since this canonical system is independent of the original one, it is invariant and all quantities defined in it are invariant. The canonical method is general and can be used locally or globally, implicitly or explicitly. We used the implicit curve representation since it enabled us to avoid fitting errors associated with the curve parameter.

We have applied the method to find local invariants of a general curve without any correspondence and curves with known correspondences of one or two feature points or lines. Experimental results for both cases are presented. Our experiments with real images have shown that by using our local implicit method we can find an invariant signature which is both insensitive to occlusion and relatively reliable. We have also demonstrated that these signatures, while unchanged under different viewpoints, do change for images of different objects. That is, they have enough descriptive power to distinguish between many different kinds of objects. Therefore they can be used for an automated object recognition system that can distinguish and identify objects regardless of the point of view from which they are observed.

ACKNOWLEDGMENTS

The support of the Defense Advanced Research Projects Agency (ARPA Order No. 8459) and the US Army Topographic Engineering Center under Contract DACA76-92-C-0009 is gratefully acknowledged.

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] I.N. Bronshtein, K.A. Semendyayev, *Handbook of Mathematics*. New York: Van Nostrand, 1985.

[3] A.M. Bruckstein, R.J. Holt, A.N. Netravali, and T.J. Richardson, "Invariant signatures for planar shape recognition under partial occlusion," *CVGIP:IU*, vol. 58, pp. 49–65, 1993.

[4] H. Burkhardt, A. Fenske, and H. Schulz-Mirbach, "Invariants for the recognition of planar contour and gray-scale images," ESPRIT Workshop "Invariants for Recognition," ECCV 92, 1992.

[5] F.S. Cohen, and J.-Y. Wang, "Modeling image curves using invariant 3D object curve models—a path to 3D recognition and shape estimation from image contours," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 16, pp. 1–12, Jan. 1994.

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

[7] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, "Invariant descriptors for 3D object recognition and pose," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 13, pp. 971–991, 1991.

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

[9] M. Halphen, "Sur les invariants différentiels des courbes gauches," *J. Ec. Polyt.*, vol. 28, no. 1, 1880.

[10] D. Keren, D. Cooper, and J. Subrahmonia, "Describing complicated objects by implicit polynomials," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 16, pp. 13–23, Jan. 1994.

[11] J.L. Mundy and A. Zisserman, "Appendix," *Geometric Invariance in Machine Vision*, J.L.Mundy and A. Zisserman, eds., Cambridge, Mass.: MIT Press, 1992.

[12] T.H. Reiss, "Recognizing planar object using invariant image features," *Lecture Notes in Computer Science*, no. 676. Berlin: Springer-Verlag, 1993.

[13] G. Taubin, "Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, Nov. 1991.

[14] L. Van Gool, J. Wagemans, J. Vandeneede, and A. Oosterlinck, "Similarity extraction and modeling," *3rd Int'l Conf. on Computer Vision*, pp. 530–534, 1990.

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

[16] I. Weiss, "Noise resistant invariants of shapes," Proc. *DARPA–ESPRIT Workshop on Invariance*, Reykjavik, Iceland, March 1991.

[17] I. Weiss, "Geometric invariants and object recognition," *Int'l J. Computer Vision*, vol. 10, no. 3, pp. 201–231, 1993.

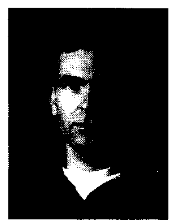
[18] I. Weiss, "Noise-resistant invariants of curves," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 15, no. 9, pp. 943-948, Sept. 1993.

[19] I. Weiss, "Local projective and affine invariants," *Annals of Mathematics and Artificial Intelligence*, Special Issue on Formal Methods in 2D Shape Analysis, to appear.

[20] I. Weiss, "High-order differentiation filters that work," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 16, no. 7, pp. 734-739, July 1994.

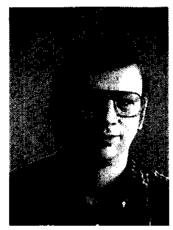
[21] E.J. Wilczynski, *Projective Differential Geometry of Curves and Ruled Surfaces*. Leipzig: Teubner, 1906.

[22] H.J. Wolfson, "On curve matching," *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol. 12, no. 5, pp. 483–489, 1990.



Ehud Rivlin received the BSc and MSc degrees in computer science and the MBA degree from the Hebrew University in Jerusalem and the PhD degree from the University of Maryland.

Currently, he is an assistant professor in the Computer Science Department at the Technion, Israel Institute of Technology. His current research interests are in machine vision and robot navigation.



Isaac Weiss received his PhD degree in physics from the Tel-Aviv University. He was subsequently a research scientist at New York University's Courant Institute of Mathematics and then at the Massachusetts Institute of Technology. He joined the Center for Automation Research of the University of Maryland in 1985.

His current research interests are computer vision, image processing, pattern recognition, and robotics.