COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 65, No. 1, January, pp. 95–108, 1997
ARTICLE NO. IV960478

NOTE

Deformation Invariants in Object Recognition

EHUD RIVLIN AND ISAAC WEISS[1]

Center for Automation Research, University of Maryland, College Park, Maryland 20742

Received July 26, 1993; accepted August 15, 1995

We study invariance to transformations having two components. The first is an arbitrary large affine transformation. This approximates a viewpoint change. The second is a small, but otherwise general, non-linear deformation. Such a deformation can arise from several sources, including change in the object itself. For instance, we want to recognize an apple even if individual apples are slightly different from each other. While there are no true invariants in this case, we show that affine invariants are quasi-invariants of these quasi-affine transformations. This is true for both global and local invariants. The method was applied to a set of real images. © 1997 Academic Press

1. INTRODUCTION

The problem of object recognition involves finding a match between a given image of an object and some image information stored in a data base. However, there is great difficulty in matching visual data because there is no one-to-one correspondence between images and the object they describe. There are many possible images corresponding to one object, and we would like to recognize the same object regardless of any changes in its images.

This situation has led to the development of invariant techniques, namely finding descriptors of the shape which are invariant to various changes in the image and can be stored in a data base instead of the image itself. This paper deals with invariants of a wide class of transformations described below.

Changes in observed images comes from several sources. One such source is the point of view. A change in the point of view from which the object is observed can change its image drastically.

Another source of change is various deformations of an object that do not change its identity. If a dog gains weight, we still want to recognize it as the same dog. An even more widespread example is the boundary contours of various objects. Looking at a pear, we can see a certain boundary contour which is quite distinctive. Looking from a different point of view, we see a contour which is different from the first one. This is because it is formed by a different occluding boundary; i.e., we see different parts of the pear. However, we can still identify the distinctive pear shape. Looking at another pear, we again have a different contour but it still has a similar pear shape.

These examples lead us to be interested in a class of transformations which is beyond the viewpoint change. Under these transformations, there is a small deformation of the shape in addition to the viewpoint transformation. The question that immediately arises, of course, is, how small is "small," or how much do we allow the object to deform and still identify it as the same object? We do not attempt here to provide a general answer to this serious question. We assume that the deformation is small in some mathematical sense that allows us to obtain useful results.

Once the deformation is defined, we want to find invariants of it. In general, there are no true invariants of small deformations, because they do not form a group. A sequence of small deformations can become a large deformation. Therefore, if invariants of small deformations existed, they would also be invariants of a large deformation. Since the latter do not exist in general, the former invariants cannot exist either. However, one can speak of "quasi-invariants" [2]. These can be defined as quantities that change much less than any other descriptors of the shape and can thus be approximated as invariants.

We now describe in more detail the two components of our transformation. Much work has been done recently on general viewpoint (projective) transformations and their subsets. This research was started by Weiss [14] and has been continued by many others, e.g., [6, 9, 17, 18]. Invariants to these transformations were found and can be used

[1] The author is grateful for the support of the Air Force Office of Scientific Research under Grant F49620-92-J-0332, the Defense Advanced Research Projects Agency (ARPA Order No. 8459), and the U.S. Army Topographic Engineering Center under Contract DACA76-92-C-0009.

for indexing of images in data bases. Certain classes of deformations were dealt with by generalizing the Fourier–Mellin method [12].

In this paper we approximate viewpoint change by an affine transformation. This is a pretty good approximation if the object is far away from the camera. The slight distortion that may come from the more general projectivity can be regarded simply as a part of the deformation. Invariants to affine transformation are well understood. They were described, e.g., in [7, 15, 3, 10, 8]. Important subsets of this transformation group were also treated before. Work was done on Euclidean transformations (translation and rotation) whose invariants are length and curvature [5]. Scale transformation can be handled with the Fourier–Mellin transform [4].

The affine transformation is linear and includes all possible linear changes of an object: rotation, translation, shear, and scalings. To this we add the deformation in the form of a small, but otherwise general, non-linear part. This part makes the transformation a "quasi-affine" transformation. The added part may be local, changing from one point to another, or global, being constant over the whole shape. The shape can be a curve or a surface. The only requirement beyond smallness is "good behavior," in terms of continuity and differentiability.

As mentioned before, there are no real invariants of such deformations. However, we can find "quasi-invariants," which we define as shape descriptors whose change under the deformation is much smaller than the change in other descriptors, or smaller than the deformation itself. We prove the general result that affine invariants are quasi-invariants of quasi-affine transformation. The proof applies to all the cases mentioned above, namely to global or local descriptors of either curves or surfaces.

2. QUASI-AFFINE INVARIANTS

We show here the quite intuitive results that the invariants of the affine transformation group are quasi-invariant of the quasi-affine transformations.

Our transformation consists of two parts:

(1) The first part is an arbitrary large affine transformation. This can be expressed as

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}, \tag{1}$$

where α_i, β_{ij} are constants.

(2) The second part is a small, but otherwise arbitrary, non-linear deformation.

We show that under this transformation, affine invariants change less than any other descriptor of the shape.

The proof does not distinguish between "local" and "global" invariants. We assume the deformation is sufficiently differentiable. First we expand the new coordinates $d\tilde{x}^i$ as a Taylor series in terms of the old coordinates dx^i in the neighborhood of any arbitrary point x^i,

$$d\tilde{x} = \beta_{ij} dx^j + \gamma_{ijk} dx^j dx^k + \cdots, \tag{2}$$

with β_{ij}, γ_{ijk} being the first and second derivatives, calculated at x^i. For a global deformation, these quantities are constant. For a local deformation, they are functions of x^i. We use here the implied summation convention, namely that summation is implied over any index that appears in a term twice. For curves we have $i = 1, 2$, while for surfaces $i = 1, 2, 3$.

We can see that the first term above correspond to a general affine transformation, Eq. (1), with the β_{ij} being the matrix coefficients of the affine transformation. We define a *quasi-affine* transformation as one in which the higher-order terms in the expansion above, from γ_{ijk} and up, are small compared with the linear ones:

$$\beta_{ij}(x^i) \gg \gamma_{ijk}(x^i)$$

We assume that this is valid for all points x^i in the domain of interest.

We are now interested in the descriptors a_l of some given shape. The descriptors can be global, such as polynomial coefficients, or local, such as derivatives (or any combination thereof). Shape descriptors usually change when the coordinate system changes, and thus they are functions of the coordinate transformation coefficients β_{ij}, γ_{ijk} appearing in Eq. (2). Therefore, in investigating the transformation properties of the descriptors we are interested in the new descriptors $\tilde{a}_l$ as functions of β_{ij}, γ_{ijk}:

$$\tilde{a}_l = \tilde{a}_l(\beta_{ij}, \gamma_{ijk}).$$

The descriptors may depend on other variables also, such as the point coordinates but this will not concern us now. (We drop the tildes now.)

Invariants are calculated as functions of the given shape descriptors a_l. We do not need to specify this function. For global a_l the invariant I is global, and for local descriptors it is local. Assuming that the function I is well behaved, i.e., it has bounded derivatives up to sufficient order, the change in I can be expanded in a Taylor series. Using the chain rule of differentiation we obtain

$$\Delta I(a_l) = \frac{\partial I}{\partial a_l} \frac{\partial a_l}{\partial \beta_{ij}} \beta_{ij} + \frac{\partial I}{\partial a_l} \frac{\partial a_l}{\partial \gamma_{ijk}} \gamma_{ijk}. \tag{3}$$

The expansion is around the values β_{ij} and $\gamma_{ijk} = 0$.

The first term above is nothing but an affine transformation of I. If we assume now that I is an affine invariant, this term vanishes. We are left with

$$\Delta I(a_l) = \frac{\partial I}{\partial a_l} \frac{\partial a_l}{\partial \gamma_{ijk}} \gamma_{ijk};$$

i.e., the change in I is proportional to γ_{ijk}. We have assumed that these γ_{ijk} are smaller than β_{ij}. The β_{ij} represent the main change in a non-invariant descriptor a_l, i.e., a descriptor for which the first term in the Taylor expansion (3) does not vanish. Therefore the change in I is much smaller than the change in other descriptors. Similarly, I changes less than the deformation itself, which is also mainly proportional to β_{ij} (Eq. (2)). These properties are the one we use as the definition of the quasi-invariants, thus I is a quasi-invariant.

3. EXAMPLES OF AFFINE INVARIANTS

Here derive some affine invariants suitable for application.

There are many kinds of affine invariants and many methods of deriving them. We only describe some useful examples here. A more complete account can be found in Weiss's review paper [16].

We concentrate here on local invariants rather than global, for two reasons: (i) Local invariants are more immune to occlusion. If part of the shape is missing, the invariants associated with the remaining part are not affected. Global invariants, on the other hand, require the whole shape for their calculation. (ii) Local invariants offer a richer description of the shape because they describe each part independently. The global ones are limited to a few low-order descriptors such as moments.

Also of interest are "semi-local" invariants, which take into account some known point or line in addition to the given curve. The resulting joint invariants can be more robust than the pure local ones.

We can use local invariants for recognition as follows. At each point of the given curve, we find two local invariants, I_1, I_2. These can be plotted against one another to obtain an invariant signature curve. That is, we define a plane with coordinates I_1, I_2. For each point of the given curve we plot a point I_1, I_2 in the invariant plane. Going over all points of the given points, we obtain a curve in the invariant plane. This can be stored in a data base as an invariant signature and used for matching. It can be shown [7] that given this invariant curve, we can reconstruct the original curve up to transformations to which I_1, I_2 are invariant. Thus the invariant signature can identify a curve up to a viewpoint change, if we use I_1, I_2 which are invariant to this change.

There are various methods for deriving local invariants of curves. Closed form formulas can be found, e.g., in [7] and in the above references. Most of these methods rely on having a curve in an explicit representation, namely as two functions of a parameter t: $x(t)$, $y(t)$. The disadvantage here is that we do not have an invariant parameter t and one needs to eliminate this parameter in some way.

An alternative method is the implicit representation, in which a shape is represented as a constraint on the coordinates without a parameter:

$$f(a_k, x_i) = 0.$$

This is possible because the parameter is not actually part of the geometry of the shape. Since we do not have a curve parameter, we do not need to worry about invariance to the parameter change. From a practical point of view, it has an advantage for the accuracy of the fitting of the curve to the given data, for the following reason.

Usually we are not given a curve but a set of pixels to which we have to fit a curve. This requires some assumption about what is the best fit, and the explicit and implicit methods differ in their assumptions.

In the explicit method, the fitted functions are $x(t)$, $y(t)$, measuring distances parallel to the x, y axes. The assumption here is that these parallel distances are minimal. These distances are very unstable when the curves are almost parallel to the axes and can introduce substantial errors. We also need to obtain two fitted functions $x(t)$, $y(t)$ rather than one curve. In the implicit method, the assumption is that distances roughly *perpendicular to the shape* are minimal. Thus an implicit fit seems more natural. This eliminates the curve parameter before it enters the invariant expressions and adds to an accumulation of errors. In addition, the explicit method assumes the existence of some ordering among the data points so that a parameter can be assigned to them, which is not always the case.

The Canonical Method. Of the many ways to derive invariants, the canonical method seems the most intuitively simple and the most general. It was developed by Weiss for local and semi-local projective and affine invariants, in the explicit and implicit representations [16, 18, 11]. Here we describe its use for affine invariants in our implicit curve representation.

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 is independent of the original system, it is invariant. All quantities defined in it are thus invariant.

This is easy to illustrate in the Euclidean case. To find invariants at point $\mathbf{x}$, we first move the origin to this point. We then rotate the coordinates so that the new x axis is tangent to the curve at $\mathbf{x}$. This is our canonical system. In it we have $y = y' = 0$ at our point. the second derivative

y'' at $\mathbf{x}$ is now invariant since we obtain the same canonical system regardless of which system we started with. In fact, this y'' is the curvature at that point. We see that by determining some of the properties of the canonical system, the others are also determined and become invariant.

We have generalized this process to the affine and projective cases and found two local invariants, I_1, I_2, at each point [18, 11]. We use these invariants to plot an invariant signature curve as discussed before.

We summarize here the canonicalization method for local and semi-local affine invariants. In the above references, affine invaraints were found as a by-product of projective ones. Here we find them in a more direct and simpler way.

The invariants of the implicit curve at a point $\mathbf{x}_0$ are found with the help of an osculating curve at $\mathbf{x}_0$. We have already seen the use of the tangent to find Euclidean invariants. 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, an nth order osculating curve can be defined as having $n + 1$ (independent) points of contact with the original curve, and the condition on the derivatives can be written as

$$\frac{d^k}{dt^k}(f^*(x,y) - f(x,y)) = 0, \quad k = 0, \dots, n, \tag{4}$$

with f^* being the osculating curve, f the given curve, and n 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 (and affine) invariant.

In the calculation we do not need either the parameter or the above derivatives. The data quantities needed here are the coefficients a_i of the given curve f, which can be obtained by fitting f to the data points. We need enough coefficients to eliminate the transformation and leave us with some invariants. In principle, a cubic will do, having nine coefficients plus the point's position. In practice, however, we have found that a wide window is necessary for robustness to noise, and this requires a higher-order curve such as a quartic

$$f(x,y) = a_0 + a_1 x + \cdots + a_{14} y^4. \tag{5}$$

(Not all its coefficients need be independent.)

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 an implicit polynomial curve to it, say a cubic or a quartic. All the following stages are performed analytically.

— Derive a canonical, intrinsic coordinate system based invariantly on the properties of the shape itself, independently of the given coordinate system. By doing so, we eliminate all the unknown quantities of the original system (e.g., the viewpoint). To accomplish this, define an "auxiliary curve" which osculates the original fitted curve with a known order of contact. The canonical system is defined so that in it the osculating curve has a particularly simple, predetermined form. (In our affine case, it is a conic.)

— Transform the original fitted curve 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 invariants I_1, I_2 that are independent of the window size or the order of the fitted curve and that depend only on the shape itself.

- Plot one invariant against the other to obtain an invariant signature curve. This will be enough to characterize the curve up to the affine transformation.

In the following sections we will describe the above steps in more detail. The canonicalization methods are different than the ones we used in [18], where the affine case was a by-product of the more general projective case.

We choose the osculating curve as the simplest one that enables us to eliminate the affine parameters. The affine transformation has six parameters—translation (in x and y directions), rotation scale (in x and y directions), and shear (skewing). Invariants are obtained by eliminating these parameters from the image, and this is the purpose of moving to a canonical system.

The elimination process is summarized as follows (Fig. 1). Three of the parameters, translation and rotation, are eliminated by moving to a Euclidean canonical system as discussed before. Moving the origin to our given curve point eliminates translation, and using the tangent eliminates rotation. The other three parameters are eliminated by using a conic auxiliary curve. In the Euclidean canonical system this conic has to pass through the origin and be tangent to the x axis, so its general form is

$$f^* = c(x,y) = c_0 x^2 + c_1 y^2 + c_2 xy + y = 0. \tag{6}$$

We have a canonical x axis, and we now need a canonical y axis. We will use the affine normal, namely the conic diameter passing through our point. We now use a skewing (shear) transformation to make this line orthogonal to the x axis, obtaining an orthogonal conic. This is done by replacing x by $x = \bar{x} - c_2 y / 2c_0$, eliminating the coefficient c_2. The remaining coefficients are eliminated by scaling x, y, and we obtain a unit circle.

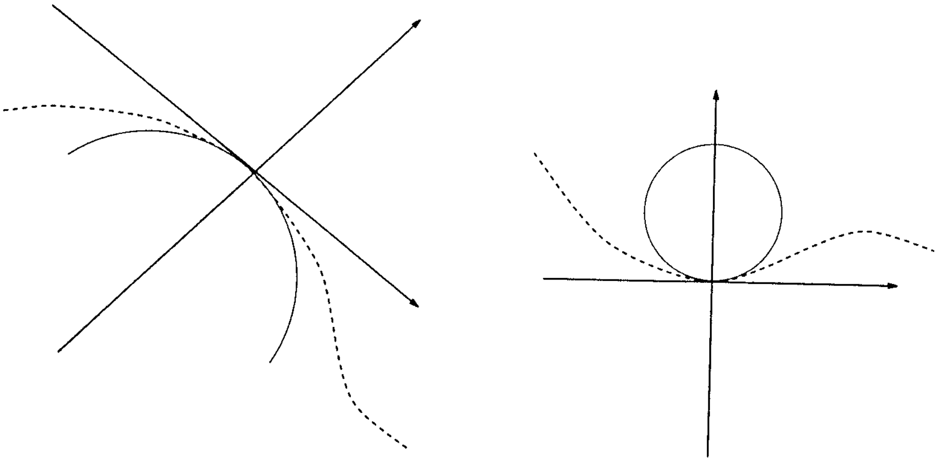

FIG. 1. (Left) An osculating conic. (Right) A canonical conic.

The detail of obtaining the above conic will be different for the different cases below. However, the above canonicalization of the resulting conic is the same in all cases (but different from [18], where we used a projective canonical system).

The Euclidean Canonical System. Here we detail the Euclidean canonization stage. As a convection, we denote the new coordinates after each canonicalization step by $\bar{x}$, $\bar{y}$ and drop the bars before going to the next step, and similarly for other quantities.

The first step is translation, 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, y_0^* which does lie on the curve; i.e., we solve Eq. (5) for 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, y_0^*. (We could simplify the solution by first translating so that $x_0 = 0$ and then solving for y_0^*.) We drop the star from y^*. 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, $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{x}_0$, substituting in (5), and rearranging. In this new system we have $\bar{a}_0 = 0$, which can be seen by simply substituting the point (0, 0) in Eq. (5).

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 $\bar{a}_1 = 0$ (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,

$$x = (\bar{x} + u_r\bar{y})/\nu, \quad y = (\bar{y} - u_r\bar{x})/\nu, \tag{7}$$

with the normalization $\nu = (1 + u_r^2)^{1/2}$. 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 $\mathbf{x}_0$ is now simply the second derivative, d^2y/dx^2. The arclength is $|dx|$ since $dy = 0$.

We will need to transform points and lines to this system. We list here for reference the relevant formulas. A point x_1, y_1 transforms to

$$\begin{aligned} \bar{x}_1 &= (x_1 - x_0 - u_r(y_1 - y_0))/(1 - u_r^2)^{1/2} \\ \bar{y}_1 &= (y_1 - y_0 + u_r(x_1 - x_0))/(1 + u_r^2)^{1/2}, \end{aligned} \tag{8}$$

while a line $b_0 + b_1x + b_2y$ is translated and rotated as

$$\begin{aligned} \bar{b} &= b_0 + b_1x_0 + b_2y_0 \\ \bar{b}_1 &= b_1 - u_r b_2 \\ \bar{b}_2 &= b_2 + u_r b_1. \end{aligned} \tag{9}$$

We again drop the bars from all quantities.

3.1. *Local Invariants*

Here we find two local affine invariants at each curve point.

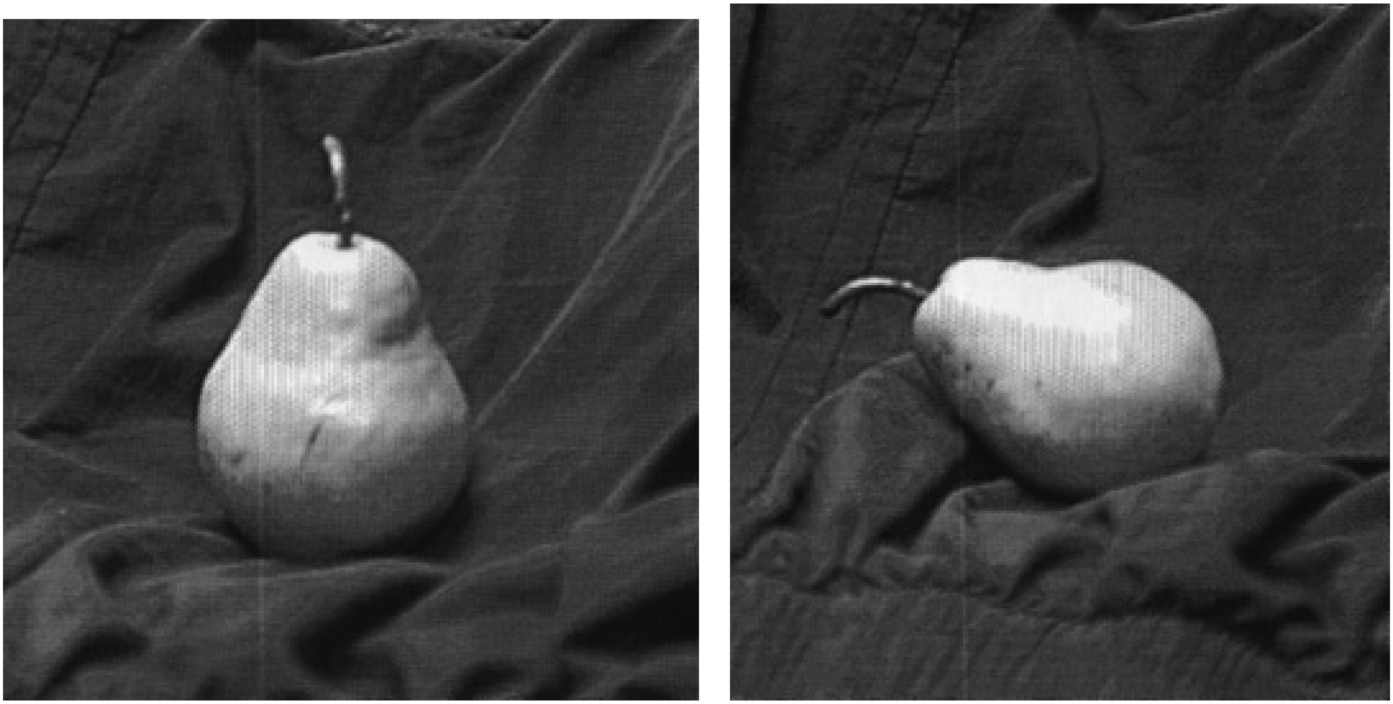

FIG. 2. Two views of a pear.

The Osculating Conic. We will now find the osculating conic f^* using the osculation condition, i.e. the equality of the first n derivatives of f, f^*, Eq. (4). The first derivative (and the zeroth) vanishes because of the tangency to the x axis. To determine the five coefficients c_i we need three more derivatives to be equal, i.e., up to the fourth one. The condition of equal derivatives ensures the locality of the treatment and also its invariance.

To proceed, we need to calculate the derivatives $d^n y/dx^n$ of the fitted curve f. This is done analytically from $f(x, y)$. To do this we use the fact that all the derivatives of f along the curve vanish, since f vanishes identically (Eq. (4)). The first 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.

Setting $a_2 = 1$ and denoting

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

we have

$$\begin{aligned} d_2 &= -a_3 \\ d_3 &= -a_6 - d_2 a_4 \\ d_4 &= -a_{10} - d_2 a_7 - d_2^2 a_5 - d_3 a_4. \end{aligned} \tag{10}$$

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

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

and substitute it in the conic expression, Eq. (6). Collecting terms with the same power x^n, we obtain five equations for the three c_i in terms of d_n. Their solution is

$$\begin{aligned} c_0 &= -d_2 \\ c_1 &= -(d_2 d_4 - d_3^2)/d_2^3 \\ c_2 &= -d_3/d_2. \end{aligned}$$

Having found the coefficients c_i, we set out to eliminate them. We define the *affine normal* as the conic diameter that passes through our point $\mathbf{x}_0$. First, we orthogonalize the axes, i.e., skew the system so that this affine normal

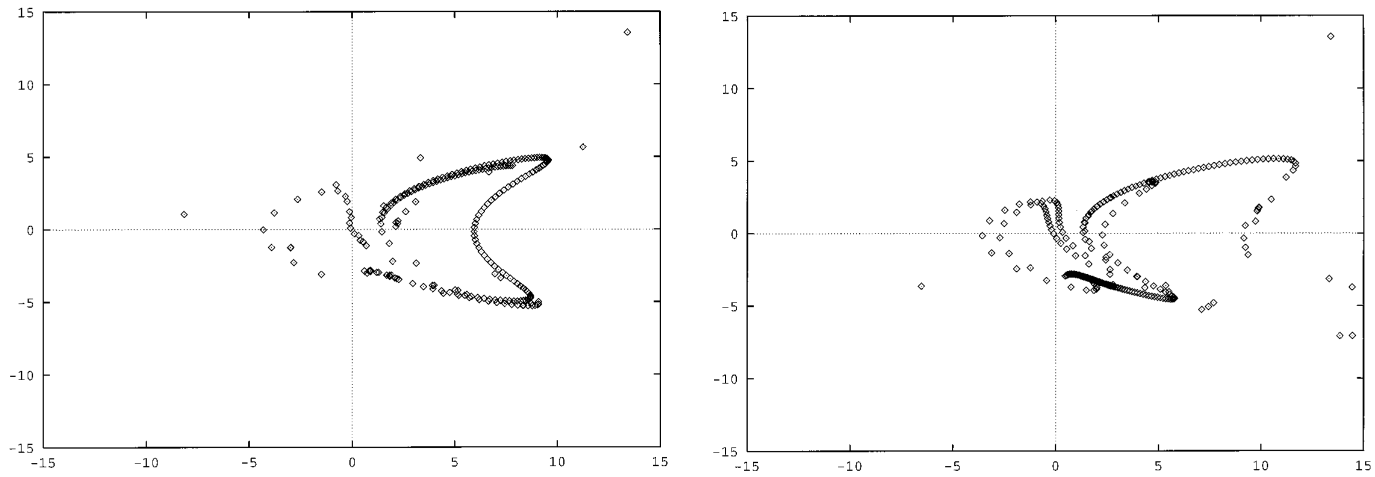

FIG. 3. Two affine signatures for the pears in Fig. 2.

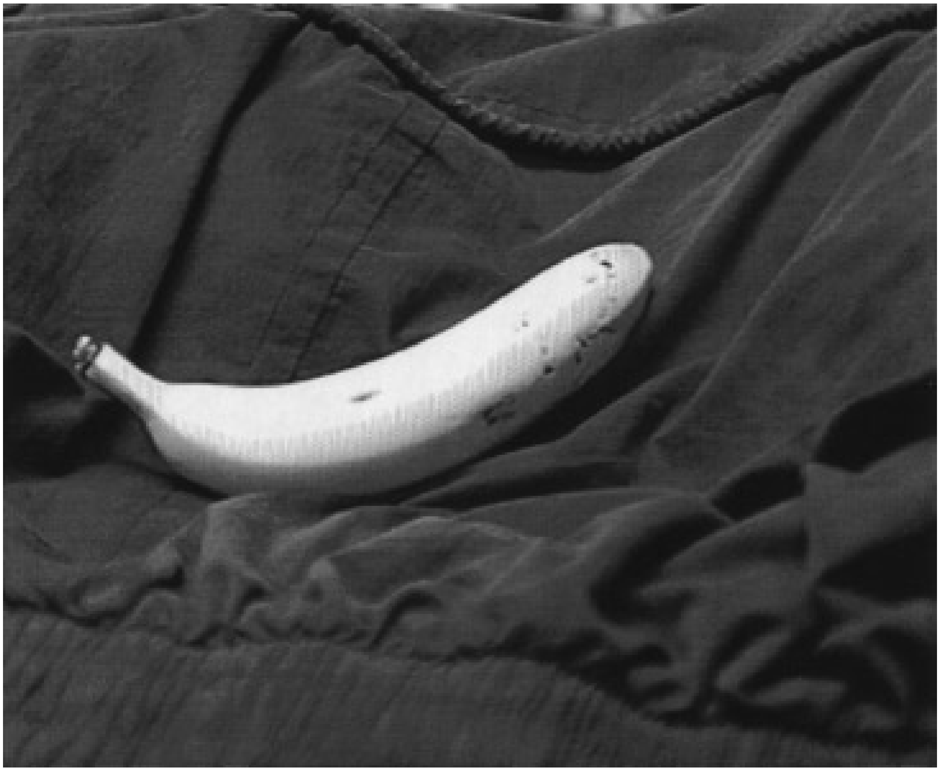

FIG. 4. A banana.

becomes perpendicular to the x axis. This will eliminate the term with c_2 in the conic. The skewing transformation is

$$x = \overline{x} + u_s y, \tag{11}$$

with $u_s = -c_2/2c_0$ being the skewing factor. y remains unchanged. Substituting the above equation in the conic (6) and rearranging, only c_1 is changed:

$$\overline{c}_1 = c_1 + c_0 u_s^2 + c_2 u_s.$$

We obtain the orthogonal conic (dropping the bars)

$$c_0 x^2 + c_1 y^2 + y = 0. \tag{12}$$

It is easy to eliminate the remaining coefficients by scaling the axes with the transformation

$$x = \overline{x}/s_x, \quad y = \overline{y}/s_y, \tag{13}$$

with

$$s_x = \sqrt{|c_0 c_1|}, \quad s_y = c_1.$$

We have thus obtained the unit circle or unit hyperbola

$$\pm x^2 + y^2 + y = 0$$

with the signal equal to the sign of $c_0 c_1$.

Local Affine Invariants. We now have an invariant canonical system but still no invariants. To obtain these, we transform the original fitted curve f, Eq. (5), to our canonical system. We collect all the transformations that were performed during the canonicalization 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 affinity (with factors u_s, s_x, s_y) on f. The coefficients of f will transform to new ones $\overline{a}_i$, which are not 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 $\overline{a}_i$ which best suit our needs.

To do this, we impose the condition of *locality,* namely that the invariants will not depend on the size of the fitted window but will be a property of the curve point itself. Thus we are looking for derivatives of the curve. The first four derivatives at $\mathbf{x}_0$ are already determined by the canonicalization process (as $d_0, \ldots, d_4 = 0, 0, -1, 0, 0$). Thus

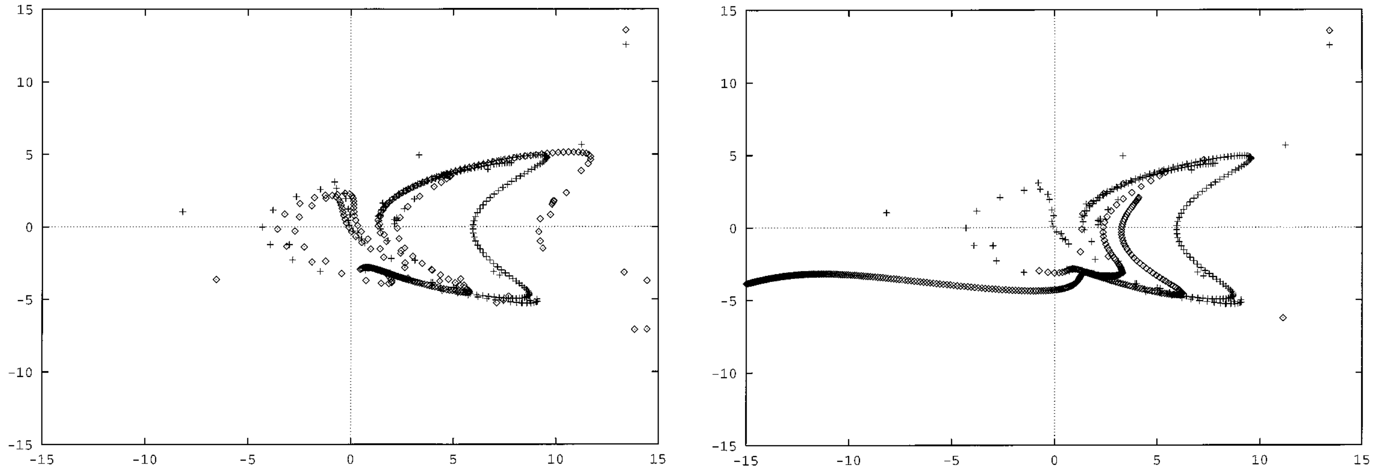

FIG. 5. (Left) The local affine signatures for the pears in Fig. 2 are presented on top of each other. (Right) The signature of the pear on the left-hand side of Fig. 2 is compared with the affine signature of a banana.

FIG. 6. Sample apples.

we need the fifth and sixth derivatives. These can be obtained in this particular system similarly to Eq. (10). With the above values of d_n we have (dropping the bars)

$$d_5 = a_{11} - a_8$$
$$d_6 = a_9 - a_{12} - a_4 d_5.$$

These quantities are our *local affine invariants.*

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.

3.2. *Semi-local Invariants*

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 [17] 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 in [1] and [13]. 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.

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 auxiliary osculating conic that will help us find the canonical system (Eq. (6)). The exact process of finding the conic differs for each case. However, the principles of invariance and locality must be maintained. In the following we will describe briefly the process for some different possible combinations. Each known feature point or line reduces the number of derivatives needed by two, because it eliminates two transformation factors.

The first step in all cases, as before, is to move to a Euclidean canonical system. The feature points and lines move according to Eqs. (8), (9). In the next step we find an orthogonal conic and scale it to a circle. The invariants will obtained differently, however. We will not need the higher derivatives of the previous case but the feature point/lines will become invariant quantities in the canonical system.

In each case the osculating conic is obtained in a somewhat different way. We will only give here the resulting conic coefficients, derived by Weiss in [18], [11]. Once the

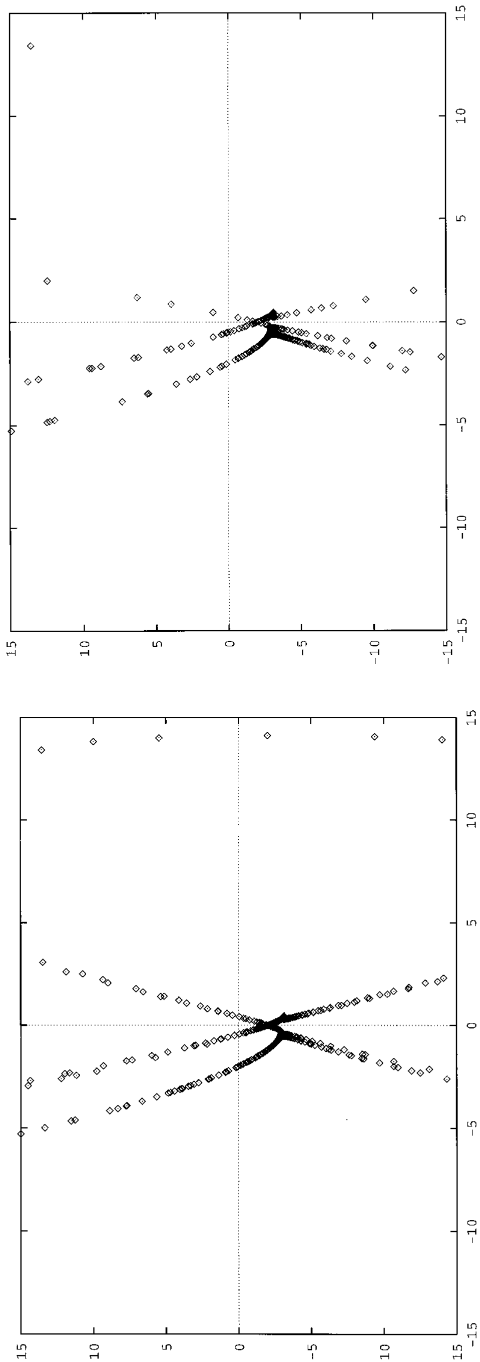

FIG. 7. Signatures for the apples in Fig. 6.

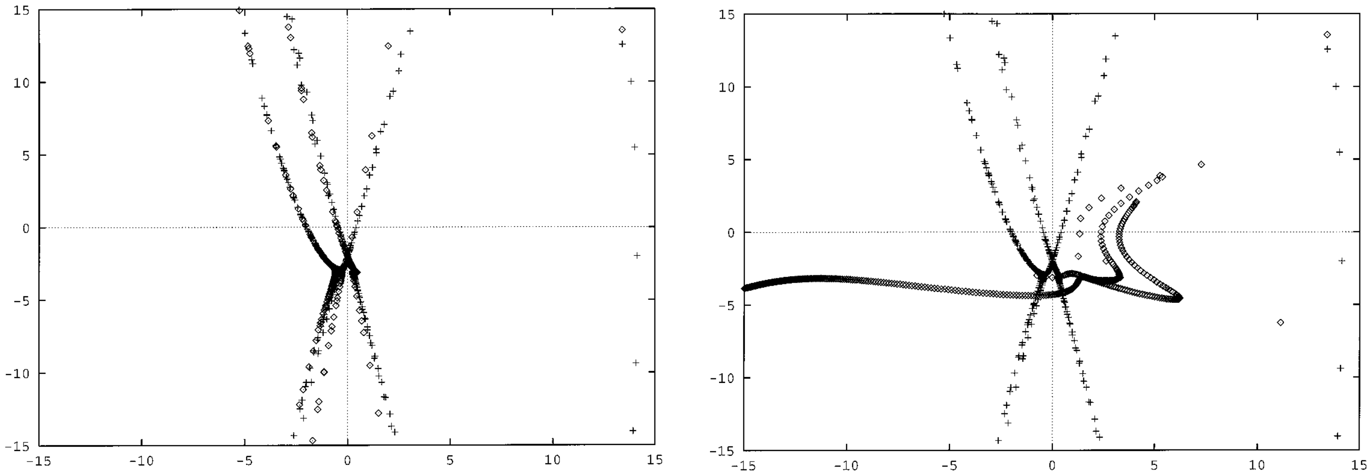

FIG. 8. (Left) The result of superimposing the two signatures of the apples in Fig. 6. (Right) The result of superimposing the signature of the apple on the left-hand side of Fig. 6 with that of the banana in Fig. 4.

conic is obtained, the canonicalization process is the same as described before (but different from the projective case described in the above reference).

- *A curve and one feature point:* We obtain the orthogonal conic, Eq. (12), as before. To find the invariant, we transform our feature point x_1, y_1 to our canonical system, combining all the transformations used before, Eqs. (8), (11), and (13) for the Euclidean, skewing, and scaling transformation. We obtain new $\bar{x}_1, \bar{y}_1$ which are now invariant because they are given in an invariant coordinate system.

Again we need a fourth-order contact here to find the conic, but we do not need the higher derivatives used before to find the invariants.

- *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. It is turned to a canonical unit circle in the same way. We then transform the given line $b_0 + b_1x + b_2y = 0$ to this system using Eq. (9) and substituting $\underline{(11)}$ and $\underline{(13)}$ in the line. The resulting line coefficients $\bar{b}_1/\bar{b}_0$ and $\bar{b}_2/\bar{b}_0$ are now invariants.

- *A curve and two feature points:* This case requires only the second derivative to determine the osculating conic (and the invariants), 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 which we then make canonical as before. We then use the line joining the points as a feature line, obtaining the previous case. The conic coefficients are

$$c_0 = -d_2$$

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

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

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

- *A curve and two feature lines:* Again we only need the second derivative. 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 make the conic canonical. We use the point of intersection of the lines as a feature points, bringing us to a previous case.

The conic coefficients are

$$c_0 = -d_2$$

$$c_2 = \frac{(b_0'^2b_1^2 - b_1'^2b_0^2)/2 + 2c_0(b_2b_0b_0'^2 - b_2'b_0'b_0^2)}{b_1b_0b_0'^2 - b_1'b_0'b_0^2}$$

$$c_1 = \frac{b_1^2 - 2c_2b_0b_1 + 4c_0b_0b_2 + c_2^2b_0^2}{4c_0b_0^2},$$

with b_i, b_i' being the coefficients of the two reference lines in the Euclidean canonical system. The invariants are as in the previous case.

- *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 w.r.t. the conic, an invariant construction. This provides sufficient conditions to determine the conic. We then make the conic canonical as before. After transforming the feature point and line to the canonical system, we can use their coefficients to obtain invariants. One way is to use draw a normal from the point to the line. The normal's coefficients are now invariants.

The conic coefficients are

$$c_1 = ((b_2 - 1)y_1 + 2c_0x_1^2 - b_1x_1)/(2y_1^2)$$

$$c_2 = -(2c_0x_1 - b_1)/y_1.$$

4. EXPERIMENTAL IMPLEMENTATION

In this section we show the local affine invariants obtained for objects undergoing quasi-affine deformations. We show that the local affine invariants quasi-invariant to these deformations and hence can be used as recognizers for classes of objects.

Our method for obtaining local affine invariants was applied to a set of real images of fruits. Segmentation was done by hand. 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 100 pixels wide around each contour point and fitted an implicit curve there, minimizing the affine square distances. The coefficients of this fitted curve were used to calculate the invariants.

Figure 2 shows two views of a pear. The occluding, or visible, contour of the pear is different in the second image (right) and is not a simple projection of the first one (left). Yet we can still recognize it as a pear. In effect these two contours could have easily come from two different pears. Thus, any signature that is common to both images can represent the whole class of pears. (This excludes extreme situations such as looking at the pear from directly above or below.) Assuming the pear is quite distant from the camera, the main component of the transformation here is affine, arising from the change of viewpoint, and there is also a relatively small, arbitrary non-linear deformations arising from the changing the occluding boundary. Thus the quasi-affine transformation fits this case perfectly. The resulting affine signature for the two images is presented in Fig. 3. A good match of the signatures is obtained. A check for a match is demonstrated in Fig. 5. The match is between the two pears' signatures (left) and between the pear signature and the one obtained from the banana from Fig. 4.

The matches between the signatures were determined by observation. Devising an automated matching method is an open problem in vision and deserves research in its own right. We only mention here one possibility. A method for an automatic matching of the signatures was successfully used by Wolfson [19] for the Euclidean case (curvature vs arclength): 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.

The same procedure was applied to the an apple presented in Fig. 6. Signatures were obtained for the same apple (in the middle of the pile in the right image). Figure 7 shows the signatures. A comparison with the signature for the banana is shown in Fig. 8.

It is apparent that the signatures of pears are similar to each other and so are signatures of apples (in spite of the different occluding boundaries from which they were calculated). At the same time, the signatures are distinct enough to differentiate a pear from an apple.

5. CONCLUSIONS

We have treated the problem of how to recognize objects which are deformed relative to some known image of them, e.g., an image stored in a data base. We assumed that the deformation is quasi-affine, namely it is mainly linear but also has a small (but otherwise general) non-linear component. This allows us to include both changes in the viewpoint and additional changes such as change in the occluding boundary. We have shown that affine invariants are quasi-invariants of our more general deformation; i.e., they change much less than other descriptors and can thus be approximated as invariants of the deformation. These invariants can be used for recognition by storing them in a data base as unique descriptors of the shapes instead of the shapes themselves. This allows us to perform matching without having to search for the correct deformation. We have applied the method to set of real images using local invariants, and it is equally applicable for global invariants.

REFERENCES

1. E. Barrett, P. Payton, N. Haag, and M. Brill, General methods for determining projective invariants in imagery *Comput. Vision Graphics Image Process.* **53,** 1991, 45–65.
2. T. O. Binford, Inferring surfaces from images, *Artificial Intelligence* **17,** 1981, 205–244.
3. A. M. Bruckstein, R. J. Holt, A. N. Netravali, and T. J. Richardson, Invariant signatures for planar shape recognition under partial occlusion *Comput. Vision Graphics Image Process.* **58,** 1993, 49–65.
4. D. Cassasent and P. Psaltis, Position, rotation and scale invariant optical correlation, *Appl. Opt.* **15,** 1976, 1975–1800.
5. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis,* Wiley, 1973.
6. D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, Invariant descriptors for 3-D object recognition and pose, *IEEE Trans. Pattern Anal. Mach. Intelligence* **13,** 1991, 971–991.
7. H. Guggenheimer, *Differential Geometry,* Dover, New York.
8. J. Hong and X. Tan, Recognize similarity between shapes under affine transformation, in *Proc. ICCV II, Tampa, Florida, 1988,* pp. 489–493.
9. J. Mundy and A. Zisserman, *Geometric Invariance in Machine Vision,* MIT Press, Cambridge, MA, 1992.
10. T. H. Reiss, Recognizing planar object using invariant image features, in *Lecture Notes in Computer Science,* Vol. 676, Springer-Verlag, Berlin, 1993.
11. E. Rivlin and I. Weiss, Local invariants for recognition, *IEEE Trans. Pattern Anal. Mach. Intelligence* **16,** 1995, 226–238.
12. J. Segman, J. Rubinstein, and Y. Y. Zeevi, The canonical coordinate method for deformation: Theoretical and computational consideration, *IEEE Pattern Anal. Mach. Intelligence* **14,** 1992, 1171–1183.
13. L. Van Gool, J. Wagemans, J. Vandeneede, and A. Oosterlinck, Similarity extraction and modeling, in *3rd Int. Conf. of Computer Vision,* 1990.
14. I. Weiss, Projective invariants of shapes, in *Proc. DARPA Image Understanding Workshop, Cambridge, MA,* pp. 1125–1134, 1988.
15. I. Weiss, Noise resistant invariant of curves, in *Geometric Invariance in Machine Vision* (J. L. Mundy and A. Zisserman, Eds.), MIT Press, Cambridge, MA, 1992.
16. I. Weiss, Geometric invariants and object recognition, *Int. J. Comput. Vision* **10,** 1993, 201–231.
17. I. Weiss, Noise resistant invariant of curves, *IEEE Trans. Pattern Anal. Mach. Intelligence* **14,** 1993, 943–948.
18. I. Weiss, Local projective and affine invariants, *Ann. Math. Artificial Intelligence* **13,** 1995, 203–225.
19. H. J. Wolfson, On curve matching, *IEEE Pattern Anal. Mach. Intelligence,* 1990, 483–489.