Using Symbolic Computation to Find Algebraic Invariants

Daniel Keren

***Abstract*— Implicit polynomials have proved themselves as having excellent representation power for complicated objects, and there is growing use of them in computer vision, graphics, and CAD. A must for every system that tries to recognize objects based on their representation by implicit polynomials are *invariants*, which are quantities assigned to polynomials that do not change under coordinate transformations. In the recognition system developed at the Laboratory for Engineering Man-Machine Studies in Brown University (LEMS), it became necessary to use invariants which are *explicit and simple* functions of the polynomial coefficients. A method to find such invariants is described and the new invariants presented. This work addresses only the problem of finding the invariants; their stability is studied in another paper.**

I. What Are Invariants?

Invariants are quantities assigned to objects that do not change when the coordinate system undergoes transformations, and hence are good descriptors for recognition. The more general the transformation, the more difficult it is to find invariants. For instance, if one deals with shapes described solely by ellipsoids, and the transformations are Euclidean, the lengths of the various axis of the ellipsoids are invariants. If, however, affine or projective transformations are allowed, the axis are no longer invariants. Another way to classify invariants is by their scope. Roughly, there are *local* invariants—such as curvature—that are determined by the local behavior of the curve, or surface, and *global* invariants, which depend on all the points of the object. Recently, there have been interesting works on *semi-differential* invariants, which combine local and global features. It is beyond the scope of this work to go into a deeper analysis of these families of invariants and their respective advantages; for an extensive survey of invariants in computer vision, see [7], [10], [1].

This work focuses on finding *algebraic invariants*, which are related to the description of objects as the zero sets of implicit polynomials [6], [5], [2], [9]. These are global invariants which show great promise for recognition of complicated objects. In [9], an elegant theory is presented which allows to describe many such invariants, as eigenvalues of matrices constructed from the coefficients. However, in order to use invariants in the Bayesian based recognition system developed in the LEMS laboratory, we must have invariants that are expressed as *simple explicit* functions of the coefficients [8]. Roughly, this is because our recognition system is based on probabilistic analysis of the invariants, and at some stage a covariance matrix for them has to be constructed from the covariance matrix of the coefficients. For this and other reasons it is necessary to write down the invariants as explicit functions of the coefficients, and the simpler the better. Needless to say, the simpler the shape of the invariants the easier it is to analyze how stable they are in the presence of noise. Also, the more invariants a recognition system uses the more discriminatory power it has. The method described here is bound to find all the invariants of the type it is looking for.

Manuscript received September 11, 1992; revised January 24, 1994. Supported in part by NSF Grant #IRI-8715774 and NSF-DARPA Grant #IRI-8905436. Recommended for acceptance by Associate Editor R. Nevatia.

The author is with the Laboratory for Engineering Man/Machine Systems, Division of Engineering, Brown University, Providence, RI 02912 USA; e-mail: dk@lems.brown.edu.

IEEE Log Number 9405787.

Let us note in passing that, in order to use invariants for recognition purposes, their stability has to be studied. This is presented in a different paper [8]. That paper demonstrates how the invariants found in this work are used for recognition of 2-D and 3-D free-form objects.

We turn now to a short survey of polynomial invariants. Probably the first known invariants were those of the second degree polynomial $p_{20}x^2 + p_{11}xy + p_{02}y^2 + p_{10}x + p_{01}y + p_{00}$: if the coordinate system undergoes an affine transformation $(u,v)^t = A(x,y)^t + T$, where A is a non-singular matrix and T a vector, then the quantities $p_{20} + p_{02}$ and $4p_{20}p_{02} - p_{11}^2$ are multiplied by the square of the determinant of A; hence, they are invariants of the *leading form* $p_{20}x^2 + p_{11}xy + p_{02}y^2$ (in general, the leading form is the part of the polynomial that contains the monomials with the highest total degree, i.e., with the highest sum of powers of x and y). Such invariants are called *relative* invariants.

Classical work [3] concentrated on finding affine invariants, but only of the leading form. This is probably because the leading form does not change under translation, hence the problem of finding invariants is more approachable and the theory more elegant. In [3], an elegant solution—the *symbolic method*—is presented for writing down all the affine invariants of the leading form. However, there are other classes of invariants, such as Euclidean invariants, and invariants that depend on *all* the coefficients, that remain to be explored.

II. The Proposed Method for Finding Invariants: Mathematical Foundation

In order to find simple and explicit invariants of polynomials, the tool of symbolic computation was utilized. We have used the *Mathematica* package [11], running on a SPARC working station. Recently, symbolic computation is finding more applications to vision [6], [4].

The suggested method for finding invariants is simple: assume that the invariants are low-degree homogeneous polynomials in the coefficients. This is a reasonable assumption, as we know that the known invariants can be represented in such a manner. The degree of the homogeneous polynomials is called the *rank* of the invariant. Now, try to solve for the coefficients of this homogeneous polynomial. A more detailed explanation follows.

Formally, let a polynomial be denoted by $P(x,y) = \sum_{0\le i+j\le n}^{n} p_{ij}x^iy^j$. Let x and y be subjected to some kind of transformation $(u,v)^t = T(x,y)^t$, where T is determined by a certain number of parameters t_{ij} (six in the case of an affine transformation, three for Euclidean, and one for rotation). Then, $P(x,y)$ transforms into a polynomial $Q(u,v)$, where Q's coefficients, q_{ij}, are functions of the p_{ij}'s and t_{ij}'s. From here on, it will be more convenient to look at the coefficients p_{ij} and q_{ij} as being indexed by a single variable. So let us revise the notations as follows: $P(x,y)$ is determined by the coefficients $\{p_i\}_{i=1}^{i=N}$, and $Q(u,v)$ by the coefficients $\{q_i\}_{i=1}^{i=N}$, where each q_i is a function of the p_i's and the t_{ij}'s, and $N = \frac{1}{2}(d+1)(d+2)$, where d is the degree of $P(x,y)$.

Now, a particular algebraic structure for the invariant I is assumed—a homogeneous polynomial (or *form*) Ψ in the p_i's, say of second degree, so $I = \sum_{0\le i<j\le N}\Psi_{ij}p_ip_j$, which has to be equal to $\sum_{0\le i<j\le N}\Psi_{ij}q_iq_j$. Let us consider a simple kind of transformation T, e.g., rotation by an angle θ. In that case, each q_i is a function of the p_i's and θ; more specifically, it is a polynomial in the p_i's and

in $\cos(\theta)$ and $\sin(\theta)$. For instance, if $d = 2$, than the polynomial $p_{20}x^2 + p_{11}xy + p_{02}y^2 + p_{10}x + p_{01}y + p_{00}$ is transformed into the polynomial $q_{20}x^2 + q_{11}xy + q_{02}y^2 + q_{10}x + q_{01}y + q_{00}$, where

$$q_{20} = p_{20}\cos^2(\theta) + p_{11}\sin(\theta)\cos(\theta)$$
$$q_{11} = p_{11}(\cos^2(\theta) - \sin^2(\theta)) + 2p_{02}\cos(\theta)\sin(\theta) - 2p_{20}\cos(\theta)\sin(\theta),$$

etc.

Let us denote the relation between the p_i's and θ and the q_i's by Φ. Formally,

$$\Phi : \mathcal{R} \times \mathcal{R}^N \rightarrow \mathcal{R}^N.$$

Here $\Phi(\theta, p) = q$, where as before θ is the rotation angle, p the vector of coefficients of the polynomial $P(x, y)$, and q the vector of coefficients of $Q(u, v)$, where $(u, v)^t$ is the rotation of the (x, y) coordinate system by θ.

The following simple property of Φ is needed in the sequel: $\Phi(\theta_1 + \theta_2, p) = \Phi(\theta_1, \Phi(\theta_2, p))$. This is obvious, as rotation by $\theta_1 + \theta_2$ is equivalent to rotation by θ_1 followed by a rotation by θ_2.

As noted before, we are looking for an invariant $I = \sum_{0 \le i < j \le N} \Psi_{ij} p_i p_j$. Thus, the following has to hold for every coefficient vector p and every angle θ:

$$\Psi(\{p_i\}) = \sum_{0 \le i < j \le N} \Psi_{ij} p_i p_j = \sum_{0 \le i < j \le N} \Psi_{ij} q_i q_j = \Psi(\{q_i\}).$$

Theorem 1: For the above to hold—e.g., for the form Ψ to define an invariant—it is necessary and sufficient that for every p,

$$\left(\frac{\partial}{\partial \theta}\Psi[\Phi(\theta, p)]\right)_{\theta=0} = 0. \tag{1}$$

Proof: First, let us show that for every coefficient vector p and every angle θ_0,

$$\left(\frac{\partial}{\partial \theta}\Psi[\Phi(\theta, p)]\right)_{\theta=\theta_0} = 0.$$

This is equal to

$$\lim_{\theta \rightarrow 0} \frac{\Psi[\Phi(\theta_0 + \theta, p)] - \Psi[\Phi(\theta_0, p)]}{\theta} = \lim_{\theta \rightarrow 0} \frac{\Psi[\Phi(\theta, \Phi(\theta_0, p))] - \Psi[\Phi(\theta_0, p)]}{\theta}$$

denoting $\Phi(\theta_0, p) = p_0$, the above reduces to

$$\lim_{\theta \rightarrow 0} \frac{\Psi[\Phi(\theta, p_0)] - \Psi[p_0]}{\theta}$$

which, because $\Phi(0, p_0) = p_0$, equals

$$\left(\frac{\partial}{\partial \theta}\Psi[\Phi(\theta, p_0)]\right)_{\theta=0},$$

but the assumption was that this is zero.

Since the partial derivative of $\Psi[\Phi(\theta, p)]$ by θ is equal to zero everywhere, it does not depend on θ; hence, Ψ defines an invariant.

The case for more general transformations is similar; let T_α be any family of transformations, indexed by a real parameter α, satisfying $T_{\alpha_1 + \alpha_2} = T_{\alpha_1} \circ T_{\alpha_2}$; then the same proof extends to show that if $(\frac{\partial}{\partial \alpha}\Psi[\Phi(\alpha, p)])_{\alpha=0} = 0$ (using the same notation as for rotation), then Ψ defines an invariant.

Now, this property is satisfied by translation; but affine transformations include also stretching at the x and y directions, and these do not satisfy $T_{\alpha_1 + \alpha_2} = T_{\alpha_1} \circ T_{\alpha_2}$, but $T_{\alpha_1 \alpha_2} = T_{\alpha_1} \circ T_{\alpha_2}$. In order to overcome this, let us define stretching by a factor of $1 + \alpha$ by T_α; then, the following holds:

$$T_{\alpha_1 + \alpha_2 + \alpha_1 \alpha_2} = T_{\alpha_1} \circ T_{\alpha_2}.$$

Now, we have for every α_0

$$\left(\frac{\partial}{\partial \alpha}\Psi[\Phi(\alpha, p)]\right)_{\alpha=\alpha_0} = \lim_{\alpha \rightarrow 0} \frac{\Psi[\Phi(\alpha_0 + \alpha + \alpha\alpha_0, p)] - \Psi[\Phi(\alpha_0, p)]}{\alpha + \alpha\alpha_0} = \lim_{\alpha \rightarrow 0} \frac{\Psi[\Phi(\alpha, \Phi(\alpha_0, p))] - \Psi[\Phi(\alpha_0, p)]}{\alpha + \alpha\alpha_0}.$$

Denoting $\Phi(\alpha_0, p)$ by p_0, the above reduces to

$$\lim_{\alpha \rightarrow 0} \frac{\Psi[\Phi(\alpha, p_0)] - \Psi[p_0]}{\alpha + \alpha\alpha_0},$$

which is equal to

$$\frac{1}{1 + \alpha_0} \lim_{\alpha \rightarrow 0} \frac{\Psi[\Phi(\alpha, p_0)] - \Psi[p_0]}{\alpha}.$$

But this is simply

$$\frac{1}{1 + \alpha_0}\left(\frac{\partial}{\partial \alpha}\Psi[\Phi(\alpha, p_0)]\right)_{\alpha=0}.$$

Hence, in this case also, for Ψ to define an invariant it is enough that $(\frac{\partial}{\partial \alpha}\Psi[\Phi(\alpha, p)])_{\alpha=0} = 0$ for every p.

These derivatives are special cases of *Lie derivatives*, but are simple enough to be derived in a straight-forward manner without relying on the general theory. For a short introduction to Lie derivatives and their applications to invariant theory, see [2].

Since the group of affine transformations is generated by rotations, translations and stretchings in the x and y axis, it is enough to test that Ψ is invariant under each of these type of transformations to prove that it is an affine invariant. Thus, the technique described here, which handles only very simple transformations, really covers general linear transformations. All this generalizes easily to invariants of 3-D polynomials; in 3-D, we assume that objects undergo Euclidean transformations, which are generated by rotations and translations on the x, y, and z axis. Hence, to test if I is an invariant, it is enough to test that it doesn't change under rotations around the three axis or translations along them. This is done in exactly the same way as for the 2-D case.

III. Reducing the Invariant Problem to a Linear System

Next, the theorem of the previous section is used to find invariants of polynomials. All the invariants we shall find are homogeneous polynomials (or forms) in the coefficients of the polynomials—either all the coefficients or only those of the leading form. The method is best demonstrated with a simple example; suppose we are interested in finding some invariants under rotation of the second degree form in x and y, $p_{20}x^2 + p_{11}xy + p_{02}y^2$. For applying the method, some general shape for the invariant is assumed, and then one attempts to solve for its specific parameters. For example, let us assume the invariant has the following shape:

$$\Psi(p_{20}x^2 + p_{11}xy + p_{02}y^2) = Ap_{20}^2 + Bp_{11}^2 + Cp_{02}^2 + Dp_{20}p_{11} + Ep_{20}p_{02} + Fp_{11}p_{02},$$

i.e., it's a second degree form in the coefficients of the second degree form $p_{20}x^2 + p_{11}xy + p_{02}y^2$. According to Theorem 1, a necessary and sufficient condition for $\Psi()$ to define an invariant under rotation is that $(\frac{\partial}{\partial\theta}\Psi[\Phi(\theta, p)])_{\theta=0} = 0$, where p is any second degree form in x and y.

The goal is to transform the problem of finding A, B, C, D, E, F—which characterize I—to that of solving a linear system of equations in A, B, C, D, E, F. Let us assume the new coefficients are q_{20}, q_{11}, q_{02}, which are functions of p_{20}, p_{11}, p_{02} and θ. According to Theorem 1, the following should hold:

$$\lim_{\theta\to 0}\left[\frac{(Ap_{20}^2 + Bp_{11}^2 + Cp_{02}^2 + Dp_{20}p_{11} + Ep_{20}p_{02} + Fp_{11}p_{02})}{\theta} - \frac{(Aq_{20}^2 + Bq_{11}^2 + Cq_{02}^2 + Dq_{20}q_{11} + Eq_{20}q_{02} + Fq_{11}q_{02})}{\theta}\right] = 0.$$

In order to write this expression down, it is necessary to compute q_{20}, q_{11}, q_{02}. Since all powers of θ higher than 1 will be discarded—this is true because we divide by θ and then let it approach zero—it is legitimate to replace $\sin(\theta)$ and $\cos(\theta)$ by their first order Taylor approximations (θ and 1), since the higher order terms vanish anyway; this greatly simplifies the expressions for the new coefficients. Doing this results in the following expression for the limit above:

$$2Ap_{11}p_{20} + 4p_{02}p_{11}B - 4p_{11}p_{20}B - 2p_{02}p_{11}C + {p_{11}}^2D + 2p_{02}p_{20}D - 2{p_{20}}^2D + p_{02}p_{11}E - p_{11}p_{20}E + 2{p_{02}}^2F - {p_{11}}^2F - 2p_{02}p_{20}F.$$

Now, this has to be zero for every choice of p_{20}, p_{11}, p_{02}, resulting in the following set of six linear homogeneous equations in A, B, C, D, E, F: $-2 - D = 0$

$$D - F = 0$$
$$-2F = 0$$
$$2A - 4B - E = 0$$
$$4B - 2C + E = 0$$
$$D - F = 0,$$

which have the solution $\{A = C, B = C/2 - E/4, D = F = 0\}$. Substituting $C = 1$, $E = 0$ results in $A = 1$, $B = 1/2$, $C = 1$, $D = E = F = 0$, which correspond to the invariant $p_{20}^2 + \frac{1}{2}p_{11}^2 + p_{02}^2$. Substituting $C = 0$, $E = 1$ gives $A = 0$, $B = -1/4$, $C = D = 0$, $E = 1$, $F = 0$, resulting in the invariant $p_{20}p_{02} - \frac{1}{4}p_{11}^2$.

In this simple case, it is possible to do the calculations manually, but that would be impossible when looking for invariants of more complicated forms, especially when the rank of the invariant is higher than two. To the rescue comes the tool of *symbolic computation*. In this work, the *Mathematica* package was used. The algorithm for finding the invariants follows.

1) Write down the expression whose invariants are sought (in this work, either polynomials or forms were selected).
2) Replace the variables in that expression by new variables, which are a transformed version of the old ones. Use only the first approximation for the transformation. This results in a polynomial whose coefficients are polynomials in the original variables and the transformation parameters. This is repeated, as noted before, for all the generators of the transformation group in question.
3) After deciding on some shape for the invariant (say, a form in the coefficients—the method adopted in this work), expand the invariant in the original coefficients and also in the coefficients that are combinations of the original coefficients and the transformation parameters. Call these I_1 and I_2.
4) Take the expression $I_1 - I_2$. It is a polynomial in the original coefficients and the transformation parameters. Of course, it has to be zero. Take as many partial derivatives of this expression as necessary to get rid of the original coefficients and the transformation parameters; equate all these derivatives to zero. This results in a linear system in the coefficients that determine the invariants (a word of caution here: these are *not* the coefficients of the original polynomial, but the coefficients of the invariant, which is a polynomial in the original coefficients). Solve this system, and you have the invariants (of course, there may be no solutions; this will indicate that there are no invariants of the type you decided to look for).

One should note that the invariants are not necessarily independent. I currently know of no method that produces algebraic invariants that are guaranteed to be independent. After the invariants are found, symbolic computation can be used to discover dependencies among them. However, this is not guaranteed to work, especially if there are many invariants and their shape is complicated.

Dependency in the invariants can cause a bias in the recognition process, for the following reason. Suppose I_1, I_2, I_3 are invariants and $P(I_1, I_2, I_3) = 0$ for some polynomial P. Then, if one uses I_1 and I_2 for recognition, I_3 should be omitted, as it brings no new information. The recognition system developed at LEMS [8] can overcome this to some extent, as it computes a covariance matrix for the invariants and uses it as a weight measure when comparing invariants of two different objects. The question is how effective the covariance matrix is for detecting high-order dependencies between the invariants; this has to be studied further, but is outside the scope of this limited work, which is confined to finding the invariants.

The case demonstrated before—finding invariants of a second-degree form in two variables—is very easy. It results in only six equations, and the *Mathematica* 2.0 version solved it in less then two seconds (running on a SPARC 1 work station). Other cases, although exactly the same in principal, result in manipulating huge expressions and solving very large linear systems. In general, suppose we want to find the invariants of rank d for a polynomial of degree n in m variables. The number of coefficients for such a polynomial is $\binom{n+m}{m}$. The number of coefficients in the form composing the elusive invariant is

$$\binom{d + \binom{n+m}{m}}{d},$$

which might be very large. In our experiments, the highest we could go to was $n = 4$, $m = 3$, $d = 3$. The input consisted of 8,312 lines, and was generated using a lexical analyzer. Running time was about 5 hours.

If *affine* invariants are sought, the problem is easier, because most possible combinations of coefficients can be ruled out. If the *x-weight* (*y-weight*) of a monomial in the coefficients is defined as the sum of the powers of $x(y)$ that the monomial represents (for instance, the

x-weight of $p_{34}^2 p_{51}$ is 11, and the y-weight 9), it is easy to see that an affine invariant can consist only of monomials having the same x-weight and y-weight. This is because the invariant should change the same under stretchings in the x and y directions, and stretching in the $x(y)$ direction multiplies the coefficients by the stretching parameter raised to the x-weight (y-weight) power. This can be demonstrated by a simple example: suppose we look for affine invariants of a second-degree form, $p_{20}x^2 + p_{11}xy + p_{02}y^2$. Let us look for an affine invariant of the shape $I = Ap_{20}^2 + Bp_{11}^2 + Cp_{02}^2 + Dp_{20}p_{11} + Ep_{20}p_{02} + Fp_{11}p_{02}$. Suppose, now, that the x-axis has been stretched by a factor of α. I should be multiplied by the determinant of the transformation squared, which is α^2 (see, for instance, [3]). It is easy to see that the coefficients after the stretching are $\alpha^2 p_{20}^2$, αp_{11}, p_{02}. Hence, the following equality should hold (on the left is the determinant squared multiplied by the value of the invariant for the original form $p_{20}x^2 + p_{11}xy + p_{02}y^2$, and on the right is the value of invariant for the original form after it had been subjected to the stretching, $\alpha^2 p_{20}^2 x^2 + \alpha p_{11}xy + p_{02}y^2$):

$$\begin{aligned}\alpha^2(Ap_{20}^2 + Bp_{11}^2 + Cp_{02}^2 + Dp_{20}p_{11} + Ep_{20}p_{02} + Fp_{11}p_{02}) \\ = A\alpha^4 p_{20}^2 + B\alpha^2 p_{11}^2 + Cp_{02}^2 + D\alpha p_{20}p_{11} \\ + E\alpha^2 p_{20}p_{02} + F\alpha p_{11}p_{02}.\end{aligned}$$

Since this has to hold for every $\{p_{20}, p_{11}, p_{02}\}$ and α, it is obvious that $A = C = D = F = 0$. This argument generalizes in a trivial manner to polynomials of any degree and invariants of any rank.

It is interesting to observe that the process described here is bound to find *all* invariants of the type it assumes. This is in contrast to other methods, such as the symbolic method [3], which does not allow any control on the complexity of the resulting invariants. (Note that the "symbolic method" referred to here dates back to the previous century, and has nothing to do with the tool of symbolic computation used in this work).

IV. An Example

As an example, the invariants of a 3-D object in two different positions are presented. The object is an eggplant, and the 3-D data were gathered at the LEMS laboratory using an IBM RS/1 Cartesian robot. Fourth degree polynomials were fit to the data using the algorithm described in [5], and the seven invariants were computed from the coefficients.

As can be seen, the invariants are not identical. The reason for this is the inherent ambiguity in the polynomial fitting; the data, in this case, does not completely constrain the coefficients. The Bayesian method described in [8] was developed to take care of this ambiguity, by weighing the difference between two sets of invariants according to their covariance matrix.

For comparison, the fit and invariants for a pear are also displayed (Fig. 1(c)).

A. An Eggplant in the First Position

The implicit equation describing an eggplant as shown in Fig. 1(a) is

$$\begin{aligned}&0.000000159x^4 + 0.000000109x^3y - 0.000000052x^3z \\ &+ 0.000000528x^2y^2 - 0.000000176x^2yz + 0.000000469x^2z^2 \\ &+ 0.000000188xy^3 - 0.000000304xy^2z + 0.000000216xyz^2 \\ &+ 0.000000086xz^3 + 0.000000198y^4 - 0.000000104y^3z \\ &+ 0.000000144y^2z^2 - 0.000000092yz^3 + 0.000000562z^4 \\ &- 0.000003817x^3 + 0.000012272x^2y + 0.000001845x^2z \\ &- 0.000010695xy^2 + 0.000002500xyz - 0.000023821xz^2 \\ &+ 0.000023714y^3 + 0.000003443y^2z + 0.000015230yz^2 \\ &- 0.000000493z^3 - 0.000221265x^2 + 0.000126875xy \\ &- 0.000043034xz + 0.001085746y^2 - 0.000303012yz \\ &+ 0.000901378z^2 + 0.021760320x - 0.019226730y \\ &- 0.006822656z - 1.00 = 0\end{aligned}$$

and the (normalized) invariants are

$$\{-6.9 \quad -53.1 \quad 1,631 \quad -762.7 \quad -222.7 \quad 90.2 \quad 504.4\}.$$

B. The Eggplant in a Second Position

The implicit equation describing the eggplant in another position (see Fig. 1(b)) is

$$\begin{aligned}&0.000000320x^4 + 0.000000106x^3y + 0.000000268x^3z \\ &+ 0.000000059x^2y^2 + 0.000000008x^2yz + 0.000000420x^2z^2 \\ &+ 0.000000061xy^3 - 0.000000406xy^2z + 0.000000008xyz^2 \\ &+ 0.000000114xz^3 + 0.000000279y^4 - 0.000000331y^3z \\ &+ 0.000000617y^2z^2 - 0.000000279yz^3 + 0.000000209z^4 \\ &- 0.000006870x^3 + 0.000016088x^2y - 0.000000236x^2z \\ &- 0.000000797xy^2 + 0.000001425xyz + 0.000006098xz^2 \\ &+ 0.000026666y^3 - 0.000000823y^2z + 0.000017790yz^2 \\ &+ 0.000003685z^3 + 0.000661459x^2 - 0.000229126xy \\ &+ 0.001154826xz + 0.001013282y^2 - 0.000231176yz \\ &+ 0.000209293z^2 + 0.003136426x - 0.021879080y \\ &- 0.013379310z - 1.00 = 0\end{aligned}$$

and the (normalized) invariants are

$$\{-7.9 \quad -42.4 \quad 1,240.2 \quad -656.9 \quad -191.4 \quad 101.3 \quad 479.7\}.$$

C. A Pear

The implicit equation describing a pear (as shown in Fig. 1(C)) is

$$\begin{aligned}&0.00023825699x^4 - 0.00003152196x^3y + 0.00014727940x^3z \\ &+ 0.00015570311x^2y^2 + 0.00006637343x^2yz \\ &+ 0.00015663391x^2z^2 - 0.00017923310xy^3 \\ &- 0.00010124010xy^2z + 0.00003062547xyz^2 \\ &+ 0.00004791431xz^3 + 0.00015177330y^3y \\ &+ 0.00008492508y^3z + 0.00006513776y^2z^2 \\ &+ 0.00002481378yz^3 + 0.00001641268z^4 \\ &+ 0.00186856801x^3 - 0.00419494417x^2y \\ &+ 0.00615517283x^2z - 0.00143002998xy^2 \\ &+ 0.00347008789xyz + 0.00307465997xz^2 \\ &+ 0.00043983711y^3 + 0.00280278409y^2z \\ &+ 0.00125001499yz^2 + 0.00129544898z^3 \\ &- 0.18444170058x^2 + 0.13251049817xy \\ &- 0.01150386967xz - 0.11283490062y^2\end{aligned}$$

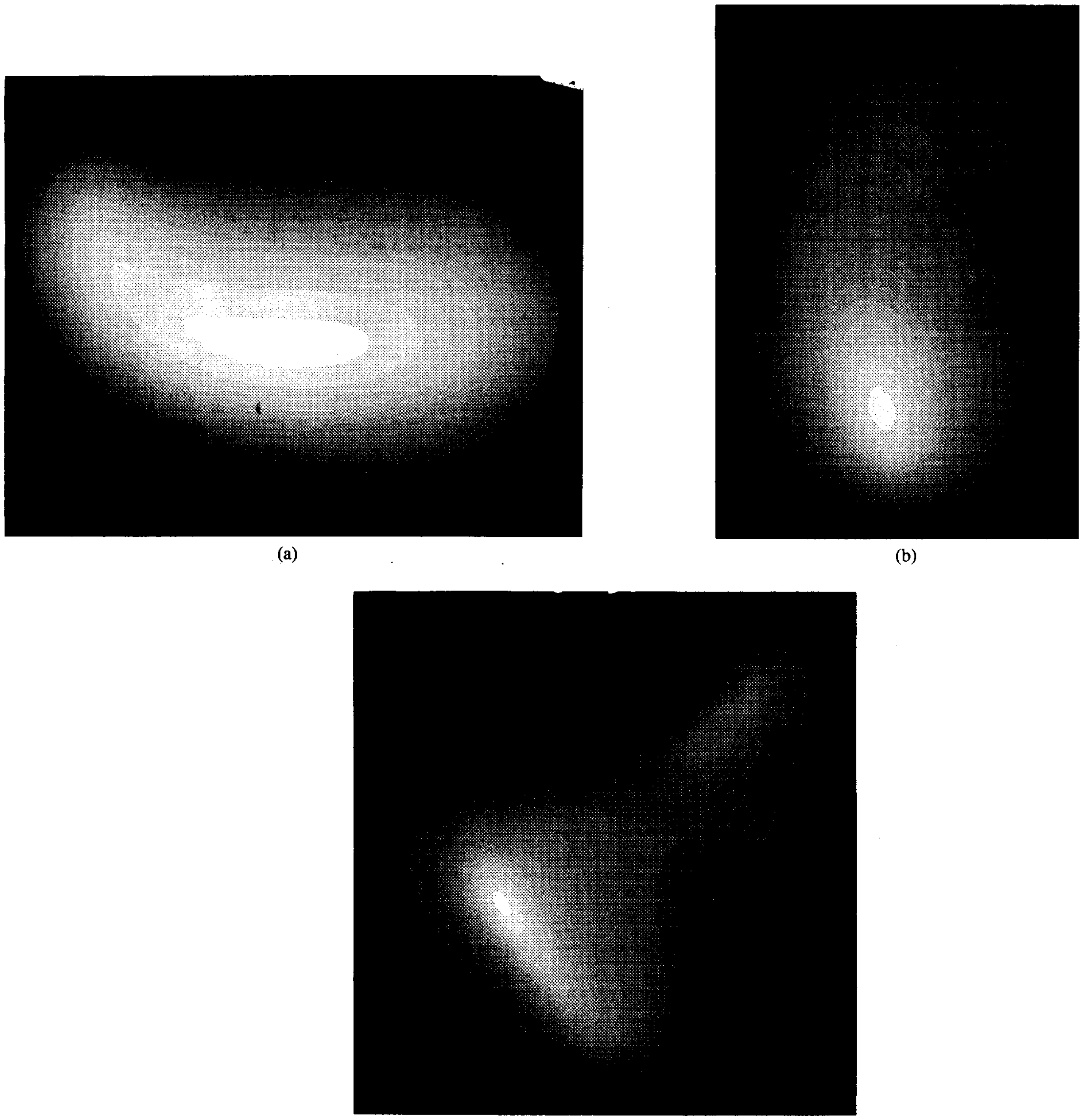

Fig. 1. (a) An eggplant. (b) The eggplant from a second viewpoint. (c) A pear.

$$- 0.07044252753yz - 0.05872813985z^2$$
$$- 0.41683548689x - 0.27396979928y$$
$$- 4.10123014450z - 1.00 = 0$$

and the (normalized) invariants are

$$\{3.8 \quad 32.5 \quad 840.2 \quad -356.9 \quad -154.4 \quad 56.3 \quad 303.7\}.$$

V. Application and Future Work

The invariants given in this paper are currently used in the Bayesian-based recognition system that is described in [8]. In the future, we hope to further explore other types of invariants and their distribution. An especially interesting problem is the one of recognizing curves, which are defined by a pair of polynomials (i.e., by the intersection of their zero-sets). The fact that the pair is not unique makes the problem of using invariants in this case mathematically challenging.

Appendix
Invariants Found Using Symbolic Computation

The method described above was used to find invariants of both general polynomials and forms. As the input files fed to *Mathematica* were too large to include here, only the invariants themselves will be given; suffice it to say that the method used to find these invariants is a straightforward extension of the method in which invariants of rank two of the second degree form in x and y were found. Since fourth degree implicit polynomials seem to be very suitable for describing a large variety of objects [5], [9], we have concentrated on finding invariants for them; also, some invariants for higher degree polynomials were found (up to degree eight).

Following are the Euclidean invariants for a fourth-degree polynomial in two variables, which were found using the method described in this work.

$$\begin{aligned}&6p_{04}^3 + 6p_{04}^2p_{22} + 6p_{04}p_{13}p_{31} + 6p_{04}p_{31}^2 - 6p_{04}^2p_{40} + 6p_{13}^2p_{40}\\&\quad - 12p_{04}p_{22}p_{40} + 6p_{13}p_{31}p_{40} - 6p_{04}p_{40}^2 + 6p_{22}p_{40}^2 + 6p_{40}^3;\end{aligned}$$

$$\begin{aligned}&6p_{04}p_{13}^2 - 16p_{04}^2p_{22} + 2p_{13}^2p_{22} - 12p_{04}p_{13}p_{31} + 4p_{13}p_{22}p_{31}\\&\quad - 18p_{04}p_{31}^2 + 2p_{22}p_{31}^2 - 18p_{13}^2p_{40} + 32p_{04}p_{22}p_{40}\\&\quad - 12p_{13}p_{31}p_{40} + 6p_{31}^2p_{40} - 16p_{22}p_{40}^2;\end{aligned}$$

$$\begin{aligned}&6p_{04}p_{22}^2 - 18p_{04}p_{13}p_{31} + 3p_{13}p_{22}p_{31} - 27p_{04}p_{31}^2 + 72p_{04}^2p_{40}\\&\quad + 72p_{04}^2p_{40} - 27p_{13}^2p_{40} + 96p_{04}p_{22}p_{40} + 6p_{22}^2p_{40}\\&\quad - 18p_{13}p_{31}p_{40} + 72p_{04}p_{40}^2;\end{aligned}$$

$$\begin{aligned}&6p_{04}p_{12}^2 - 9p_{03}p_{12}p_{13} + 9p_{02}p_{13}^2 + 9p_{13}^2p_{20} - 3p_{12}p_{13}p_{21}\\&\quad + 9p_{03}^2p_{22} - 24p_{02}p_{04}p_{22} + p_{12}^2p_{22} - 24p_{04}p_{20}p_{22}\\&\quad + 6p_{03}p_{21}p_{22} + p_{21}^2p_{22}4p_{02}p_{22}^2 - 4p_{20}p_{22}^2\\&\quad + 36p_{04}p_{12}p_{30} - 27p_{03}p_{13}p_{30} - 9p_{13}p_{21}p_{30} + 6p_{12}p_{22}p_{30}\\&\quad + 54p_{04}p_{30}^2 + 9p_{22}p_{30}^2 - 9p_{03}p_{12}p_{31} + 18p_{02}p_{13}p_{31}\\&\quad + 18p_{13}p_{20}p_{31} - 3p_{12}p_{21}p_{31} - 27p_{03}p_{30}p_{31} - 9p_{21}p_{30}p_{31}\\&\quad + 9p_{02}p_{31}^2 + 9p_{20}p_{31}^2 + 54p_{03}^2p_{40} - 144p_{02}p_{04}p_{40}\\&\quad - 144p_{04}p_{20}p_{40} + 36p_{03}p_{21}p_{40} + 6p_{21}^2p_{40} - 24p_{02}p_{22}p_{40}\\&\quad - 24p_{20}p_{22}p_{40};\end{aligned}$$

$$6p_{22}^3 - 27p_{13}p_{22}p_{31} + 81p_{04}p_{31}^2 + 81p_{13}^2p_{40} - 216p_{04}p_{22}p_{40};$$

$$\begin{aligned}&- 9p_{13}^2p_{20} + 3p_{12}p_{13}p_{21} + 6p_{04}p_{21}^2 - 2p_{12}^2p_{22} + 3p_{11}p_{13}p_{22}\\&\quad + 24p_{04}p_{20}p_{22} - 12p_{03}p_{21}p_{22} - 2p_{21}^2p_{22} + 8p_{02}p_{22}^2\\&\quad + 8p_{20}p_{22}^2 - 36p_{04}p_{12}p_{30} + 27p_{03}p_{13}p_{30} + 18p_{13}p_{21}p_{30}\\&\quad - 12p_{12}p_{22}p_{30} - 54p_{04}p_{30}^2 - 18p_{04}p_{11}p_{31} + 18p_{03}p_{12}p_{31}\\&\quad - 27p_{02}p_{13}p_{31} - 27p_{13}p_{20}p_{31} + 3p_{12}p_{21}p_{31} + 3p_{11}p_{22}p_{31}\\&\quad + 27p_{03}p_{30}p_{31} - 9p_{02}p_{31}^2 - 54p_{03}^2p_{40} + 144p_{02}p_{04}p_{40}\\&\quad + 6p_{12}^2p_{40} - 18p_{11}p_{13}p_{40} + 144p_{04}p_{20}p_{40} - 36p_{03}p_{21}p_{40}\\&\quad + 24p_{02}p_{22}p_{40};\end{aligned}$$

$$\begin{aligned}&6p_{13}^3 - 24p_{04}p_{13}p_{22} + 48p_{04}^2p_{31} + 6p_{13}^2p_{31} - 24p_{04}p_{22}p_{31}\\&\quad - 6p_{13}p_{31}^2 - 6p_{31}^3 + 48p_{04}p_{13}p_{40} + 24p_{13}p_{22}p_{40}\\&\quad - 48p_{04}p_{31}p_{40} + 24p_{22}p_{31}p_{40} - 48p_{13}p_{40}^2;\end{aligned}$$

$$\begin{aligned}&6p_{12}^2p_{20} - 18p_{11}p_{13}p_{20} + 36p_{04}p_{20}^2 - 3p_{11}p_{12}p_{21} + 27p_{10}p_{13}p_{21}\\&\quad - 18p_{03}p_{20}p_{21} + 6p_{02}p_{21}^2 + 6p_{11}^2p_{22} - 18p_{10}p_{12}p_{22}\\&\quad + 12p_{02}p_{20}p_{22} - 18p_{01}p_{21}p_{22} + 36p_{00}p_{22}^2 - 108p_{04}p_{10}p_{30}\\&\quad + 27p_{03}p_{11}p_{30} - 18p_{02}p_{12}p_{30} + 27p_{01}p_{13}p_{30} + 27p_{03}p_{10}p_{31}\\&\quad - 18p_{02}p_{11}p_{31} + 27p_{01}p_{12}p_{31} - 108p_{00}p_{13}p_{31} + 36p_{02}^2p_{40}\\&\quad - 108p_{01}p_{03}p_{40} + 432p_{00}p_{04}p_{40};\end{aligned}$$

Also, seven Euclidean invariants for a fourth-degree polynomial in three variables were discovered. Lack of space allows us to present only one of them. Those interested in the rest can e-mail the author at dk@lems.brown.edu for a listing of the other invariants and for other classes of invariants:

$$\begin{aligned}&36p_{004}^3 + 36p_{004}^2p_{022} + 36p_{004}p_{013}p_{031} + 36p_{004}p_{031}^2\\&\quad - 36p_{004}^2p_{040} + 36p_{013}^2p_{040} - 72p_{004}p_{022}p_{040}\\&\quad + 36p_{013}p_{031}p_{040} - 36p_{004}p_{040}^2 + 36p_{022}p_{040}^2\\&\quad + 36p_{040}^3 - 4p_{004}p_{112}^2 + 4p_{040}p_{112}^2 + 12p_{004}p_{103}p_{121}\\&\quad - 12p_{040}p_{103}p_{121} - 2p_{013}p_{112}p_{121} - 2p_{031}p_{112}p_{121}\\&\quad + 4p_{004}p_{121}^2 - 4p_{040}p_{121}^2 + 18p_{013}p_{103}p_{130}\\&\quad + 18p_{031}p_{103}p_{130} - 12p_{004}p_{112}p_{130} + 12p_{040}p_{112}p_{130}\\&\quad + 36p_{004}^2p_{202} + 16p_{004}p_{022}p_{202} + 6p_{013}p_{031}p_{202}\\&\quad + 6p_{031}^2p_{202} - 24p_{004}p_{040}p_{202} - 16p_{022}p_{040}p_{202}\\&\quad - 12p_{040}^2p_{202} + 2p_{103}p_{121}p_{202} + 4p_{112}p_{130}p_{202}\\&\quad + 6p_{130}^2p_{202} - 8p_{040}p_{202}^2 + 12p_{004}p_{013}p_{211}\\&\quad + 2p_{013}p_{022}p_{211} + 24p_{004}p_{031}p_{211} + 2p_{022}p_{031}p_{211}\\&\quad + 24p_{013}p_{040}p_{211} + 12p_{031}p_{040}p_{211} - 2p_{103}p_{112}p_{211}\\&\quad - p_{112}p_{121}p_{211} - 3p_{103}p_{130}p_{211} - 2p_{121}p_{130}p_{211}\\&\quad + 4p_{031}p_{202}p_{211} + 4p_{004}p_{211}^2 + 4p_{040}p_{211}^2\\&\quad - 12p_{004}^2p_{220} + 6p_{013}^2p_{220} - 16p_{004}p_{022}p_{220}\\&\quad + 6p_{013}p_{031}p_{220} - 24p_{004}p_{040}p_{220} + 16p_{022}p_{040}p_{220}\\&\quad + 36p_{040}^2p_{220} + 6p_{103}^2p_{220} + 4p_{103}p_{121}p_{220}\\&\quad + 2p_{112}p_{130}p_{220} - 16p_{004}p_{202}p_{220} - 8p_{022}p_{202}p_{220}\\&\quad - 16p_{040}p_{202}p_{220} + 4p_{013}p_{211}p_{220} - 8p_{004}p_{220}^2\\&\quad + 36p_{004}p_{103}p_{301} + 6p_{022}p_{103}p_{301} + 12p_{040}p_{103}p_{301}\\&\quad - 3p_{031}p_{112}p_{301} + 24p_{004}p_{121}p_{301} + 4p_{022}p_{121}p_{301}\\&\quad - 12p_{040}p_{121}p_{301} + 15p_{013}p_{130}p_{301} + 18p_{031}p_{130}p_{301}\\&\quad + 2p_{121}p_{202}p_{301} - 2p_{112}p_{211}p_{301} + 6p_{103}p_{220}p_{301}\\&\quad + 36p_{004}p_{301}^2 + 6p_{022}p_{301}^2 + 18p_{013}p_{103}p_{310}\\&\quad + 15p_{031}p_{103}p_{310} - 12p_{004}p_{112}p_{310} + 4p_{022}p_{112}p_{310}\\&\quad + 24p_{040}p_{112}p_{310} - 3p_{013}p_{121}p_{310} + 12p_{004}p_{130}p_{310}\\&\quad + 6p_{022}p_{130}p_{310} + 36p_{040}p_{130}p_{310} + 6p_{130}p_{202}p_{310}\\&\quad - 2p_{121}p_{211}p_{310} + 2p_{112}p_{220}p_{310} + 18p_{013}p_{301}p_{310}\\&\quad + 18p_{031}p_{301}p_{310} + 6p_{022}p_{310}^2 + 36p_{040}p_{310}^2\\&\quad - 36p_{004}^2p_{400} - 24p_{004}p_{022}p_{400} - 8p_{022}^2p_{400}\\&\quad + 12p_{013}p_{031}p_{400} - 120p_{004}p_{040}p_{400} - 24p_{022}p_{040}p_{400}\\&\quad - 36p_{040}^2p_{400} + 36p_{103}^2p_{400} + 4p_{112}^2p_{400}\\&\quad + 24p_{103}p_{121}p_{400} + 4p_{121}^2p_{400} + 24p_{112}p_{130}p_{400}\\&\quad + 36p_{130}^2p_{400} - 72p_{004}p_{202}p_{400} - 16p_{022}p_{202}p_{400}\\&\quad - 24p_{040}p_{202}p_{400} - 12p_{013}p_{211}p_{400} - 12p_{031}p_{211}p_{400}\\&\quad - 4p_{211}^2p_{400} - 24p_{004}p_{220}p_{400} - 16p_{022}p_{220}p_{400}\\&\quad - 72p_{040}p_{220}p_{400} + 16p_{202}p_{220}p_{400} + 36p_{103}p_{301}p_{400}\\&\quad + 12p_{121}p_{301}p_{400} + 12p_{112}p_{310}p_{400} + 36p_{130}p_{310}p_{400}\\&\quad - 36p_{004}p_{400}^2 - 12p_{022}p_{400}^2 - 36p_{040}p_{400}^2\\&\quad + 36p_{202}p_{400}^2 + 36p_{220}p_{400}^2 + 36p_{400}^3;\end{aligned}$$

REFERENCES

[1] J. Mundy and A. Zisserman, Eds.,*Conf. on Applications of Invariants in Computer Vision II*, Esprit, ARPA/NSF, Ponta Delgada, Azores, October 1993.

[2] 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. Machine Intell.*, vol. 13, no. 10, pp. 971–992, 1991.

[3] J. H. Grace and A. Young, *The Algebra of Invariants*. Cambridge, UK: Cambridge University Press, 1903.

[4] C. Jerian and R. Jain, "Polynomial methods for structure from motion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 12, pp. 1150–1167, 1990.

[5] D. Keren, D. B. Cooper, and J. Subrahmonia "Describing complicated objects by implicit polynomials," Tech. Rep. LEMS-93, Brown University, 1991; *IEEE Trans. Pattern Anal. Machine Intell.*, in press.

[6] D. J. Kriegman and J. Ponce, "On recognizing and positioning curved 3d objects from image contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 12, pp. 1127–1138, 1990.

[7] J. Mundy and A. Zisserman, *Geometric Invariance in Machine Vision*. Cambridge, MA: MIT Press, 1992.

[8] J. Subrahmonia, D. B. Cooper, and D. Keren, "Reliable object recognition using high dimensional implicit polynomials for 2d curves and 3d surfaces," Tech. Rep. LEMS-107, Brown University, 1992; *IEEE Trans. Pattern Anal. Machine Intell.*, submitted.

[9] 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. Machine Intell.*, vol. 13, no. 11, pp. 1115–1138, 1991.

[10] I. Weiss, "Geometric invariants and object recognition," *IJCV*, vol. 10, no. 3, pp. 201–231, 1993.

[11] S. Wolfram, *Mathematica, a System for Doing Mathematics by Computer*. Reading, MA: Addison-Wesley, 1988.