Semi-Differential Invariants for Recognition of Algebraic Curves

Yan-Bin Jia and Rinat Ibrayev

Iowa State University, Ames, IA 50011, USA

Abstract. This paper studies the recognition of low-degree polynomial curves based on minimal tactile data. Differential and semi-differential invariants have been derived for quadratic curves and special cubic curves that are found in applications. Such an invariant, independent of translation and rotation, is computed from the local geometry at up to three points on a curve. Recognition of the curve reduces to invariant verification with its canonical parametric form determined along the way. In addition, the contact locations are found on the curve, thereby localizing it relative to the touch sensor. Simulation results support the method in the presence of small noise. Preliminary experiments have also been carried out. The presented work distinguishes itself from traditional model-based recognition in its ability to simultaneously recognize and localize a shape from one of several classes, each consisting of a continuum of shapes, by the use of local data.

1 Introduction

Human can feel a shape through touch. Essentially, the action is performed to detect some geometric features on the shape which are then subconsciously synthesized in the brain. Typical geometric features include, for instance, smoothness, saliences, concavities, etc.

Supported by touch sensing, the robot can also obtain shape information without the help of a vision system. Such shape inference is important, for example, when camera occlusion becomes inevitable or when motion is involved. Since tactile data are local (and one-dimensional for point contact), seemingly they convey only a limited amount of geometric information. But how much shape knowledge can the robot really acquire then?

Fig. 1 illustrates a hand with two tactile fingers touching an object. Suppose through local movements the fingers are able to estimate information such as the curvatures at several points of contact. And suppose the shape is known to be from a finite family of parametric curves. Then we would like to *recognize the shape as well as determine the finger placement.*

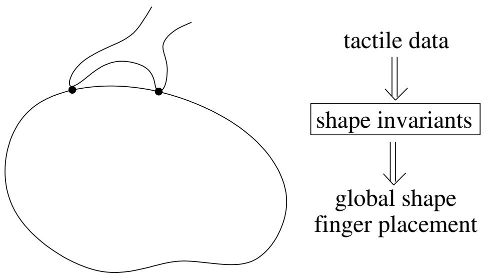


Fig. 1. A robotic hand touching an object to recognize its shape.

This problem draws several distinctions from traditional model-based recognition. First, every model here is not a real shape but rather a *continuum* of shapes parametrized in the same form. Second, we would like to keep the sensor data to the minimum. This is because a touch sensor, unlike a vision system, does not generate global shape data. Third, we hope to determine where the tactile data were obtained on the shape.

The characteristics of our problem naturally suggest an approach based on differential and semi-differential invariants. Such invariants of a shape are independent of its position and orientation, the computation of which is often a burden. In this paper, we are interested in invariants that are also *independent of point locations on a shape at which they are evaluated.* Given the local nature of touch sensing, such shape descriptors should be computable from measurements at just a few points. Our investigation will be focused on quadratic and cubic spline curves.

1.1 Related Work

There are two primary recognition strategies in model-based vision. The first one hinges on the recovery of viewing parameters (thus the pose). Kriegman and Ponce [12] constructed implicit shape equations from image contours and then solved for viewing parameters through data fitting. The second approach is to develop descriptors that are invariant to Euclidean, affine, or projective transformation, or to camera-dependent parameters [15,19].

Algebraic invariants are expressions of the coefficients of polynomial equations describing curved shapes. The foundation was due to Cayley, Sylvester, Young, and among others, Hilbert [8], who offered a procedure that constructs all independent algebraic invariants for a given curve or surface. In real applications, polynomials are fit to image data and their coefficients are extracted for invariant evaluation. Keren [10] and Forsyth *et al.* [6] presented efficient methods for finding algebraic invariants and demonstrated on recognition of real objects. Civi *et al.* [4] also conducted object recognition experiments with algebraic invariants of Euclidean, affine, and projective groups.

One drawback of algebraic invariants is the requirement of global shape data. This is almost impossible to provide by a touch sensor, or by a vision system in case of occlusion.

Differential invariants depend on local data and deal with situations like occlusion well. Up till now, vision- and invariant-based recognition has focused on differential invariants that are independent of various transformation groups but not of point locations on a shape. Calabi *et al.* [3] introduced the “signature curve” that is invariant to Euclidean or affine transformation. Rivlin and Weiss [18] derived differential invariants for a shape by applying to its quartic fit the same transformation that turns an osculating curve (a cubic) into the canonical form.

Semi-differential invariants combine global constraints and local information to ease the correspondence issue faced by non-invariant-based methods and also relieve the burden on estimating higher order derivatives for differential invariants. The theoretical foundation for this type of invariants was presented by Moons *et al.* [14]. Pajdla and Van Gool [16] used simple semi-differential invariants to match curves extracted from range data in the presence of partial occlusion.

In touch sensing, shape recognition has long built on the notion of "interpretation tree", which represents all possible correspondences between geometric features of an object and tactile data. Grimson and Lozano-Pérez [7] identified and localized a 3-D polyhedron from a set of models using tactile measurements of positions and surface normals. Fearing [5] described how a cylindrical tactile fingertip could recover the pose of a generalized convex cone from a small amount of data.

Allen and Michelman [1] fit a superquadric surface to sparse data obtained by a Utah-MIT hand around an object as its reconstructed shape. Moll and Erdmann [13] showed how to simultaneously estimate the shape and motion of an unknown convex object from tactile readings on multiple manipulating palms under frictionless contact.

A method based on the interpretation tree or least-squares fitting needs to recover the pose. This may be costly and often unnecessary. Not until very recently did differential invariants start to find applications with touch sensing. For spheres, cylinders, cones, and tori, Keren *et al.* [11] constructed descriptors in terms of curvatures and torsions and (up to their third order) derivatives estimated at points on one or two curves embedded in these surfaces.

In this work, we have extended our recent result [9] and derived semi-differential invariants that not only recognize the classes but also recover the algebraic descriptions of quadratic and certain cubic curves from curvature and derivative measurements.

2 Curve Invariants

The touch sensor in contact with a 2D object can "feel" its local geometry, which is described by the curvature. At the contact point denote by ϕ the tangential angle formed by the tangent of the boundary curve $\alpha(t) = (x(t), y(t))$ with the x-axis. The *curvature* κ is the rate of change of ϕ with respect to arc length s, that is,

$$\kappa = \frac{d\phi}{ds} = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}}. \tag{1}$$

Curvature is independent of parametrization, rotation, and translation. We are also interested in the derivative of curvature with respect to arc length:

$$\kappa_s = \frac{d\kappa}{dt}\frac{dt}{ds} = \frac{\kappa'(t)}{(x'^2 + y'^2)^{1/2}}. \tag{2}$$

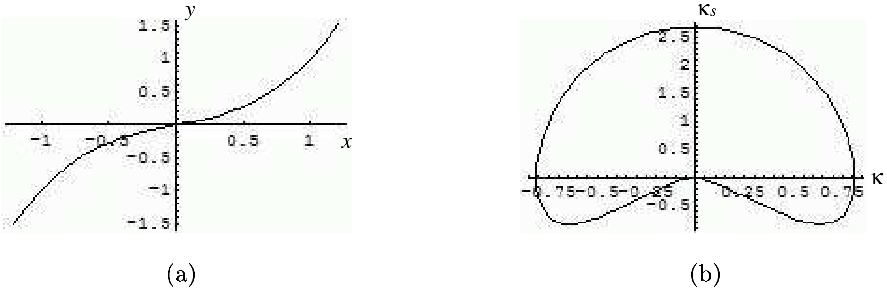
Section 7 will look at how curvature and derivative can be reliably estimated from real data. Until then we just assume that *these two quantities are measurable.*

2.1 Signature Curve

A *differential invariant* of a curve is a real-valued function that depends on the curve and its derivatives but not on a specified transformation group or parametrization. The transformation group considered in this paper is the plane Euclidean group $SE(2)$. More intuitively, the value of a differential invariant depends on the point location on the curve but not on the curve's rotation and orientation.

Theorem 1. *Every differential invariant of a plane curve is a function of the curvature κ and its derivatives with respect to arc length.*

The *Euclidean signature curve* of a curve $\alpha(t)$ is the set of all points $(\kappa(t), \kappa_s(t))$ evaluated along the curve. An example is shown in Fig. 2. The following result is well known [3]:



(a) (b)

Fig. 2. (a) A cubical parabola $y = 0.6x^3 + 0.4x$; (b) its signature curve.

Theorem 2. *Two smooth curves are equivalent up to an Euclidean transformation if and only if their signature curves are identical.*

The above result has led to the development of shape recognition methods [18,17,3] based on matching signature curves. Construction of the signature curve, nevertheless, requires global shape information, which the touch sensor does not provide. So our aim is to make use of the local geometry at a small number of points to perform the recognition task.

2.2 Semi-Differential Invariants as Curve Descriptors

Suppose the curve $\alpha(t) = (x(t), y(t))$ is known to be from a family. Often we can derive a *canonical parametric form* of the family through proper rotation, translation, and reparametrization. This canonical form should have minimum number of indeterminates (other than t) to parametrize the family. These independent indeterminates are referred to as *shape parameters* and denoted by $a_1, \ldots, a_n$. For instance, the class of all ellipses are parametrized with the semimajor axis a and the semiminor axis b. The canonical form expresses the curve in a coordinate system determined by its geometry.

Since the parameter t, which specifies the location of contact (with the touch sensor), is not measurable, we try to eliminate it from the expressions (1) and (2). This can always be done through computing the resultant, yielding an equation for the signature curve:

$$f[a_1, \ldots, a_n](\kappa, \kappa_s) = 0. \tag{3}$$

The simplest case is when the function f can be split into two parts and rewritten as

$$I(\kappa, \kappa_s) = g(a_1, \ldots, a_n).$$

Then I is an expression whose value depends on the shape of the curve not on any specific point at which it is evaluated. It is thus an invariant for the curve, or a *curve invariant.* That the expression assumes the same value at different points is a necessary condition for an unknown curve to be from the family. The family of parabolas will be given as an example in Sect. 3.1.

When a curve family has n shape parameters, we need n independent differential invariants to uniquely identify a curve from the family. If only one point on the curve is considered, this requires up to the nth derivative of the curvature. Numerical computation of high order derivatives is very unreliable. The solution is to trade the order of derivative for extra points. So we consider the curvatures and derivatives at n points and derive *semi-differential invariants.* They are functions of the $2n$ curvatures and derivatives but assume values depending on $a_1, \ldots, a_n$ only. For some curves, such as ellipses and hyperbolas (Sects. 3.2–3.3), semi-differential invariants can be found via algebraic manipulation.

However, derivation of semi-differential curve invariants appears to be very difficult, if not impossible, for many curves. It is more likely that we have to solve for the shape parameters $a_1, \ldots, a_n$ using curvature and derivative estimates at $m \geq n$ points. Viewed differently, the semi-differential invariants now have values equal to the shape parameters but their evaluation can only be done through solution or minimization.

From now on we focus on curves parameterized by polynomials. From (1) and (2) we derive two polynomial equations in t:

$$\kappa^2 \left(x'^2 + y'^2\right)^3 - (x'y'' - x''y')^2 = 0, \tag{4}$$

$$\begin{aligned} &\frac{\kappa_s}{\kappa^2} \left(x'y'' - x''y'\right)^2 - \left(x'y''' - x'''y'\right)\left(x'^2 + y'^2\right) \\ &\qquad + 3(x'x'' + y'y'')(x'y'' - x''y') = 0. \end{aligned} \tag{5}$$

Note that κ and κ_s are measurable and thus treated as constants. Suppose $x(t)$ and $y(t)$ have degrees d_x and d_y, respectively. The two polynomials on the left hand sides of (4) and (5) may have a resultant with degree as high as $d_x + d_y + 8\max\{d_x, d_y\} - 12$ in $a_1, \dots, a_n$. This is not computationally tractable once d_x or d_y exceeds 3.

2.3 Semi-Signature Curve

We can lower the degree of the resultant polynomial in $a_1, \dots, a_n$ by considering the slope $\lambda = y'/x'$. This can be rewritten as a polynomial equation:

$$\lambda x' = y'. \tag{6}$$

Assume that $x' \neq 0$ at every point measured by the touch sensor. Equations (1) and (5) are rewritten as

$$\kappa x'^2 (1+\lambda^2)^{3/2} - (y'' - x''\lambda) = 0, \tag{7}$$

$$\frac{\kappa_s/\kappa^2 + 3\lambda}{1+\lambda^2}(y'' - x''\lambda)^2 + 3x''(y'' - x''\lambda) - (y''' - x'''\lambda)x' = 0. \tag{8}$$

With the slope λ treated as a separate variable, we compute the resultants of (6) with (7) and (8), respectively, to eliminate t and obtain

$$f_1[a_1, \dots, a_n](\lambda, \kappa, \kappa_s) = 0 \quad \text{and} \quad f_2[a_1, \dots, a_n](\lambda, \kappa, \kappa_s) = 0.$$

The two functions f_1 and f_2 have degrees not exceeding $3\max\{d_x, d_y\} - 3$ and $3\max\{d_x, d_y\} - 5$, respectively, in $a_1, \dots, a_n$.

What is the reason for using the slope λ determined by the canonical form? Though not able to measure the slope, the touch sensor can measure the tangent rotation from one point to another. The slope λ_1 at the first point and the slope λ_i at the ith point are related as

$$\lambda_i = \frac{\lambda_1 + \tan \varDelta\theta_{1i}}{1 - \lambda_1 \tan \varDelta\theta_{1i}}, \qquad \theta_{1i} \text{ tangent rotation.} \tag{9}$$

So only one new variable λ_1 has been introduced.

We refer to the point set $\{(\lambda, \kappa, \kappa_s)\}$ as the *semi-signature curve* of $\alpha(t)$. An example is shown in Fig. 3. Its projection onto the κ-κ_s plane is the signature curve. The semi-signature curve is dependent on the chosen canonical parametrization, more specifically, on the orientation of the original curve under the parametrization. Now we may employ similar methods to derive expressions in terms of $\lambda, \kappa, \kappa_s$ but whose values depend on the shape parameters only. These "pseudo-invariants" together with (9) are solved for the slopes first, and then the shape parameters. The details will be described in Sect. 4 on cubical and semicubical parabolas and cubic spline curves.

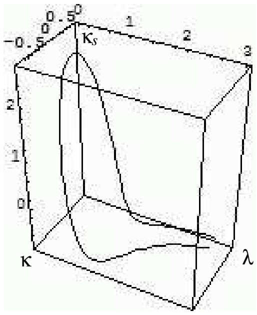


Fig. 3. The semi-signature curve of the cubical parabola in Fig. 2.

3 Quadratics

It is well known that every quadratic curve is one of three types: the ellipse, the hyperbola, and the parabola. Together the three types of curves are referred to as the *conics*. We start with deriving some invariants for these curves.

3.1 Parabola

Parabolas are identified with all the curves parametrized by quadratic polynomials: $x = a_2t^2 + a_1t + a_0$ and $y = b_2t^2 + b_1t + b_0$, where $a_2b_1 - a_1b_2 \neq 0$. Every parabola has a canonical parametrization up to rotation and translation:

$$x = at^2 \quad \text{and} \quad y = 2at, \qquad a > 0.$$

We obtain the curvature and its derivative with respect to arc length:

$$\kappa = -\frac{1}{2a(t^2+1)^{3/2}} \quad \text{and} \quad \kappa_s = \frac{\kappa'}{v(t)} = \frac{3t}{4a^2(t^2+1)^3}.$$

Eliminating t from the above equations leads to an equation that describes the signature curve of the parabola:

$$\frac{1}{(2a)^{2/3}} = \kappa^{2/3}\left(\frac{\kappa_s^2}{9\kappa^4} + 1\right) \equiv I_{\mathrm{p}}(\kappa, \kappa_s). \tag{10}$$

The expression $I_{\mathrm{p}}(\kappa, \kappa_s)$ has value independent of t. It is an invariant which has a one-to-one correspondence to the shape of the parabola. Fig. 4 illustrates three parabolas distinguished by I_{p}. Since κ and κ_s are measurable, from (10) we can determine the shape parameter a.

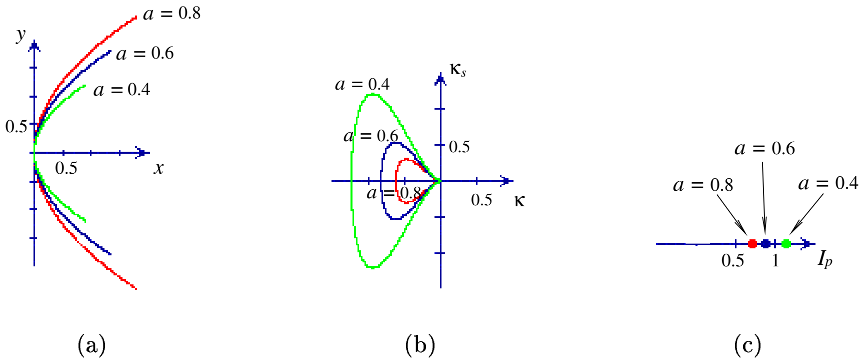


Fig. 4. (a) Three parabolas in the form $y^2 = 4ax$; (b) their signature curves $\{(\kappa, \kappa_s)\}$; (c) corresponding values of the invariant I_p. The invariant is evaluated using any point on a signature curve.

3.2 Ellipse

Let us start with the canonical parametrization:

$$x = a\cos(t) \quad \text{and} \quad y = b\sin(t), \qquad a, b > 0.$$

The curvature and its derivative with respect to arc length are

$$\kappa = \frac{ab}{\left(a^2\sin^2(t) + b^2\cos^2(t)\right)^{3/2}} \quad \text{and} \quad \kappa_s = \frac{-3ab\left(a^2 - b^2\right)\sin(t)\cos(t)}{\left(a^2\sin^2(t) + b^2\cos^2(t)\right)^3}.$$

Elimination of t from the above two equations results in an equation describing the signature curve (see Fig. 5(b)):

$$\frac{a^2 + b^2}{(ab)^{4/3}} - \frac{1}{(ab\kappa)^{2/3}} - I_{\mathrm{p}}(\kappa, \kappa_s) = 0, \tag{11}$$

where I_{p} is an expression of κ and κ_s defined in (10). Since there are two unknown quantities a and b, at least two points on the ellipse are required.

Let κ_i and κ_{si}, $i = 1, 2$, be the curvature and its derivative at the ith point. Then we end up with two equations in the form of (11). Subtracting one of them from the other yields the following (assuming $\kappa_1 \neq \kappa_2$):

$$\begin{aligned} \frac{1}{(ab)^{2/3}} &= \frac{(\kappa_1\kappa_2)^{2/3}}{\kappa_1^{2/3} - \kappa_2^{2/3}}\Big(I_{\mathrm{p}}(\kappa_1, \kappa_{s1}) - I_{\mathrm{p}}(\kappa_2, \kappa_{s2})\Big) \\ &\equiv I_{\mathrm{c}1}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}). \end{aligned} \tag{12}$$

The expression $I_{\mathrm{c}1}$ is a semi-differential invariant, since it involves the geometry at more than one points. Its value $1/(ab)^{2/3}$ is independent of the two points that are used.

The invariant $I_{\mathrm{c}1}$ alone cannot distinguish ellipses with the same product ab, or equivalently, with the same area. So we find a second invariant by substituting $I_{\mathrm{c}1}$ for $1/(ab)^{2/3}$ into the second term of the equation (11):

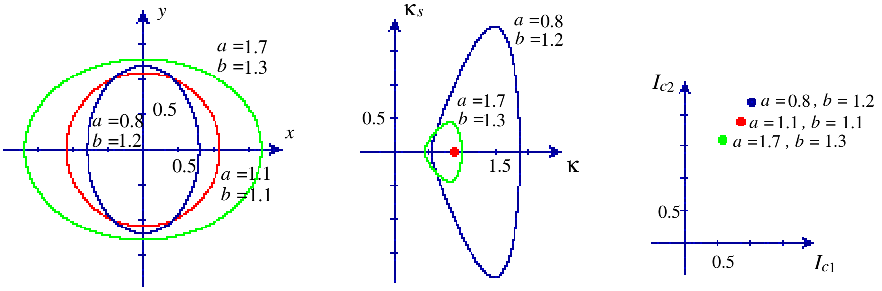
$$\begin{aligned} \frac{a^2 + b^2}{(ab)^{4/3}} &= \frac{1}{\kappa_1^{2/3} - \kappa_2^{2/3}}\Big(\kappa_1^{2/3} I_{\mathrm{p}}(\kappa_1, \kappa_{s1}) - \kappa_2^{2/3} I_{\mathrm{p}}(\kappa_2, \kappa_{s2})\Big) \\ &\equiv I_{\mathrm{c}2}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}). \end{aligned} \tag{13}$$

A one-to-one correspondence exists between the tuples $(I_{\mathrm{c}1}, I_{\mathrm{c}2})$ and (a, b). Fig. 5 compares two ellipses and a circle distinguished by the invariants $I_{\mathrm{c}1}$ and $I_{\mathrm{c}2}$. From the two invariants we easily recover the values of a and b.

3.3 Hyperbola

A hyperbola has the canonical parametric form

$$x = a\cosh(t) = a\,\frac{e^t + e^{-t}}{2} \quad \text{and} \quad y = b\sinh(t) = b\,\frac{e^t - e^{-t}}{2}, \quad a, b > 0.$$



(a) (b) (c)

Fig. 5. (a) Three ellipses in the form $x^2/a^2 + y^2/b^2 = 1$; (b) their signature curves (the one for the circle with $a = b = 1.1$ degenerates into a point $(1/1.1, 0)$); (c) corresponding values of the invariant pair (I_{c1}, I_{c2}). The invariants are evaluated using any two points on the same signature curve.

Similar to the case of an ellipse, we are able to eliminate t from the equations $\kappa = \kappa(t)$ and $\kappa_s = \kappa_s(t)$ and obtain the following:

$$\frac{a^2 - b^2}{(ab)^{4/3}} + \frac{1}{(ab\kappa)^{2/3}} - I_{\mathrm{p}}(\kappa, \kappa_s) = 0, \tag{14}$$

where I_{p} is again defined in (10). Taking the curvatures and derivatives at two points on the hyperbola, from the two copies of equation (14) we derive

$$I_{c1}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}) = -\frac{1}{(ab)^{2/3}} \quad \text{and} \quad I_{c2}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}) = \frac{a^2 - b^2}{(ab)^{4/3}}.$$

These two invariants are in the same forms as for an ellipse but their values are in different expressions of a and b. In particular, I_{c1} is always negative for the hyperbola.

The invariants I_{c1} and I_{c2} completely determine the hyperbola. Computation of a and b from them is very straightforward.

3.4 Invariants for Conics

Both I_{c1} and I_{c2} are also semi-differential invariants for a parabola, assuming values 0 and $1/(2a)^{2/3}$, respectively. With curvature and derivative information at any two different points, the sign of I_{c1} tells the type of a conic. When the invariant is positive the curve is an ellipse, when it is negative the curve is a hyperbola, and when it is zero the curve is a parabola. The invariants I_{c1} and I_{c2} thus describe the correlations between any two points on a conic.

4 Cubics

There is no classification of all cubic curves. So it seems very difficult to construct invariants that recognize all of them. However, we are interested in cubic splines, whose continuity in curvature enables them to model curved shapes in graphics and geometric modeling. Every segment of a cubic spline has the general and canonical parametric forms as follows:

$$\begin{array}{lll} x = a_3t^3 + a_2t^2 + a_1t + a_0, & & x = t^2, \\ y = b_3t^3 + b_2t^2 + b_1t + b_0; & \text{equivalently,} & y = at^3 + bt^2 + ct. \end{array} \tag{15}$$

In the canonical form on the right, a (> 0), b, c are the shape parameters.

This section starts with two subclasses of cubic splines — cubical and semi-cubical parabolas — and then moves on to general cubic splines.

4.1 Cubical Parabola

This class of curves has the canonical parametric form:

$$x = t \quad \text{and} \quad y = at^3 + ct, \qquad a > 0.$$

Figs. 2 and 3 plot an example and its signature and semi-signature curves. Unfortunately, it is not obvious how to eliminate the parameter t from the expressions of the curvature κ and its derivative κ_s. So we employ the method in Sect. 2.3 and utilize the slope $\lambda = \frac{y'}{x'} = 3at^2 + c$. First, we obtain

$$\kappa^2 = \frac{12a(\lambda - c)}{(1+\lambda^2)^3} \qquad \text{and} \qquad \kappa_s = \frac{6a(1+\lambda^2) - 36a\lambda(\lambda - c)}{(1+\lambda^2)^3}.$$

Note that we can obtain a and c using λ, κ, and κ_s:

$$a = \frac{\left(\kappa_s + 3\lambda\kappa^2\right)\left(1+\lambda^2\right)^2}{6} \quad \equiv \quad I_{\text{cp1}}(\lambda, \kappa, \kappa_s), \tag{16}$$

$$c = \lambda - \frac{\kappa^2\left(1+\lambda^2\right)}{2\left(\kappa_s + 3\lambda\kappa^2\right)} \quad \equiv \quad I_{\text{cp2}}(\lambda, \kappa, \kappa_s). \tag{17}$$

The expressions I_{cp1} and I_{cp2} map any point on the semi-signature curve to the shape parameters a and c, respectively. They are invariants of the curve provided that the slope λ can be determined.

Measure the tangent rotation $\Delta\theta_{12}$ from point 1 to point 2. Write $\delta_{12} = \tan\Delta\theta_{12}$. Since the value of c should be same, we have

$$I_{\text{cp2}}(\lambda_1, \kappa_1, \kappa_{s1}) = I_{\text{cp2}}(\lambda_2, \kappa_2, \kappa_{s2}). \tag{18}$$

Substitution of (9) with $i = 2$ into (18) results in a quartic polynomial:

$$d_4\lambda_1^4 + d_3\lambda_1^3 + d_2\lambda_1^2 + d_1\lambda_1 + d_0 = 0, \tag{19}$$

which can be solved for λ_1 (and hence λ_2). These coefficients are

$$\begin{aligned} d_0 &= \kappa_{s1}\left(\kappa_2^2\left(5\delta_{12}^2 - 1\right) + 2\kappa_{s2}\delta_{12}\right) + \kappa_1^2\left(3\kappa_2^2\delta_{12} + \kappa_{s2}\right), \\ d_1 &= 2\delta_{12}\left(\kappa_{s1}\left(3\kappa_2^2 - \kappa_{s2}\delta_{12}\right) + 2\kappa_1^2\left(3\kappa_2^2\delta_{12} + \kappa_{s2}\right)\right), \\ d_2 &= \kappa_{s1}\left(\kappa_2^2\left(5\delta_{12}^2 - 1\right) + 2\kappa_{s2}\delta_{12}\right) + \kappa_1^2\left(18\kappa_2^2\delta_{12} - \kappa_{s2}\left(5\delta_{12}^2 - 1\right)\right), \\ d_3 &= 2\delta_{12}\left(\kappa_{s1}\left(3\kappa_2^2 - \kappa_{s2}\delta_{12}\right) + 2\kappa_1^2\left(3\kappa_2^2\delta_{12} + \kappa_{s2}\right)\right), \\ d_4 &= 5\kappa_1^2\delta_{12}\left(3\kappa_2^2 - \kappa_{s2}\delta_{12}\right). \end{aligned}$$

4.2 Semi-Cubical Parabola

The canonical parametrization for this class of curves takes the form

$$x = t^2 \quad \text{and} \quad y = at^3 + bt^2, \qquad a > 0.$$

Reparametrize the curve with the slope is $\lambda = y'/x' = 3at/2 + b$ and obtain

$$a = \sqrt{-\frac{8\kappa^3(1+\lambda^2)^{5/2}}{9(\kappa_s + 3\lambda\kappa^2)}} \;\equiv\; I_{\text{scp1}}(\lambda, \kappa, \kappa_s), \tag{20}$$

$$b = \lambda + \frac{\kappa^2(1+\lambda^2)}{\kappa_s + 3\lambda\kappa^2} \;\equiv\; I_{\text{scp2}}(\lambda, \kappa, \kappa_s). \tag{21}$$

Using two points, we can set up an equation:

$$I_{\text{scp2}}(\lambda_1, \kappa_1, \kappa_{s1}) = I_{\text{scp2}}(\lambda_2, \kappa_2, \kappa_{s2}).$$

This equation together with (9) again yield a quartic polynomial in λ_1. The invariants for this class of curves are I_{scp1} and I_{scp2}.

4.3 Cubic Spline

It is time now to turn to general cubic splines described by (15). Different from its two subclasses, we cannot replace t with the slope λ in the expressions of the curvature and its derivative. So we resort to solving equations (6)–(8), which are simplified to the following:

$$3at^2 + 2(b-\lambda)t + c = 0, \tag{22}$$

$$Lt^2 - 3at - (b-\lambda) = 0, \tag{23}$$

$$6at + M\Big((b-\lambda)^2 - 3ac\Big) + 3(b-\lambda) = 0, \tag{24}$$

where $L = 2(1+\lambda^2)^{3/2}\kappa$ and $M = \frac{\kappa_s + 3\kappa^2\lambda}{(1+\lambda^2)\kappa^2}$. We substitute c in (24) with (22):

$$9a^2Mt^2 + 6a\Big(1 + M(b-\lambda)\Big)t + M(b-\lambda)^2 + 3(b-\lambda) = 0. \tag{25}$$

Next, the resultant of equations (23) and (25) is computed to eliminate t:

$$81Ma^4 + 18L\Big(1 + 3M(b-\lambda)\Big)a^2 + L^2(b-\lambda)\Big(M(b-\lambda) + 3\Big)^2 = 0. \tag{26}$$

Since M can get very large when κ is small, we divide the left hand side of (26) by $81Ma^4$ and denote the resulting expression as the function $g(a, b, \lambda)$.

With curvatures and derivatives estimated at $l \geq 3$ points, the shape parameters a, b, and the slope λ_1 at the first point can be estimated through a least-squares optimization:

$$\min_{a,b,\lambda_1} \sum_{i=1}^{l} g(a, b, \lambda_i)^2,$$

where λ_is depend on λ_1 according to (9). To determine the third parameter c, we just need to eliminate the t^2 and t terms from (22)–(24).

5 Locating Contact

The parameter value t determines the contact location on the curved shape with the touch sensor. Since the tangent at the contact is measurable, t also determines the relative pose of the shape to the sensor. We have the following:

$$t = \begin{cases} \frac{\kappa_s}{3\kappa^2}, & \text{if parabola;} \\ \sin^{-1}\left(\sqrt{\frac{\left(\frac{ab}{\kappa}\right)^{2/3} - b^2}{a^2 - b^2}}\right), & \text{if ellipse;} \\ \sinh^{-1}\left(\sqrt{\frac{\left(\frac{ab}{\kappa}\right)^{2/3} - b^2}{a^2 + b^2}}\right), & \text{if hyperbola;} \\ \pm\sqrt{\frac{\lambda - b}{3a}}, & \text{if cubical parabola;} \\ \frac{2(\lambda - b)}{3a}, & \text{if semi-cubical parabola;} \\ -\frac{M\left((b-\lambda)^2 - 3ac\right) + 3(b - \lambda)}{6a}, & \text{if cubic spline.} \end{cases}$$

6 Simulation

In simulation, we approximate the curvature and its derivative by finite difference quotients using arc length s and tangential angle ϕ:

$$\kappa \approx \frac{\phi(s + \Delta s) - \phi(s - \Delta s)}{2\Delta s} \quad \text{and} \quad \kappa_s \approx \frac{\phi(s + \Delta s) - 2\phi(s) + \phi(s - \Delta s)}{(\Delta s)^2},$$

The arc length between two points on the curve, close to each other, is approximated by their Euclidean distance. The rotation of the tangent from one point to another uses the exact value since it can be measured quite accurately in practice. Given the errors of finite differences, it is not very meaningful to introduce simulated noise, which may either reduce or magnify such errors.[1]

6.1 Verification of Invariants

The first group of simulations were conducted to verify the invariants of the three conics, cubical parabolas, and semi-cubical parabolas. The results are summarized in Table 1, where estimation errors of κ and κ_s accounted for the discrepancies between the actual values of the invariants and their estimates.

In the second group of simulations shown in Table 2, we demonstrate that *an invariant for one of the curve classes varies for another.*

Finally, Table 3 reveals how much the recovered shape parameters $\bar{a}, \bar{b}, \bar{c}$ differ from the real ones a, b, c. From the table we see that on the average the relative errors are around 1% except for cubic splines.

[1] A method introduced in [3] approximates the osculating circle with one that passes through three local points. This curvature estimation scheme was extended in [2]. These methods are able to generate slightly better estimates than finite differencing but the simulation outcomes would not have been altered.

inv.	I_p	I_{c1} (12)		I_{c2} (13)		I_{cp1}	I_{cp2}	I_{scp1}	I_{scp2}
	(10)	ellipse	hyperbola	ellipse	hyperbola	(16)	(17)	(20)	(21)
real	0.2198	0.1857	−0.2678	1.2055	0.3222	6.9963	2.6127	1.3730	6.5107
min	0.2168	0.1801	−0.2729	1.1749	0.2937	6.7687	1.7312	1.4111	6.3945
max	0.2230	0.1863	−0.2655	1.2083	0.3615	7.0289	3.1684	1.4447	6.5834
mean	0.2198	0.1852	−0.2675	1.2035	0.3210	6.9355	2.5022	1.4220	6.5154

Table 1. Invariant verification on five specific curves. Each invariant, labeled with its defining equation, is evaluated 100 times using points randomly selected from the corresponding curve. The shape parameters of each curve are easily recoverable from the definitions of its invariants.

\ data inv.\	conic (ellipse)	cubical parabola	semi-cub. parabola	cubic spline
I_{c1}	——	−6.38(min) −0.04(max) −0.73(mean) 1.22(stdev)	−22.84 28.37 3.37 6.76	−45.24 −4.94 −16.50 10.86
I_{cp2}	−11.97 −15.46 −0.04 2.53	——	8.54 19.03 13.76 3.07	11.66 1721.04 55.52 217.34
I_{scp2}	−265.80 5.83 −3.22 26.75	7.80 65.22 29.17 17.19	——	−150.68 1715.73 38.97 182.24

Table 2. Evaluating three invariants on data obtained from curves of different classes. Each cell displays the summary over 100 evaluations on a curve.

	ellip.	hyper.	par.	cub. par.	semi-cub. par.	cubic spline
min	0.02%	0.10%	0.01%	0.02%	0.04%	0.61%
max	7.99%	9.71%	3.35%	7.49%	8.09%	29.23%
mean	0.40%	1.15%	0.36%	0.83%	1.23%	11.27%

Table 3. Relative errors $\sqrt{((a-\bar{a})/a)^2 + \left((b-\bar{b})/b\right)^2 + ((c-\bar{c})/c)^2}$ on estimating shape parameters. Summary over 100 curves from each class (only 25 curves from the cubic spline class) randomly generated under uniform distributions of its shape parameters within prescribed ranges.

6.2 Curve Recognition

We now use invariants to recognize a curve out of the six classes of quadratic and cubic curves. For example, consider the ellipse in Figure 6(a). The values of κ and κ_s are estimated at $t_1 = 0.36$, $t_2 = 1.86$, and $t_3 = 4.23$. Invariant I_p has values 0.8971 and 0.4030 at the first two points, so the curve is not a parabola. Invariant I_{c1} yields values 0.3447, 0.3446, and 0.3449 at the three resulting pair of points, from which we infer that the curve is an ellipse. The recovered shape parameters (from I_{c1} and I_{c2}) are $a \approx 2.8609$ and $b \approx 1.7275$.

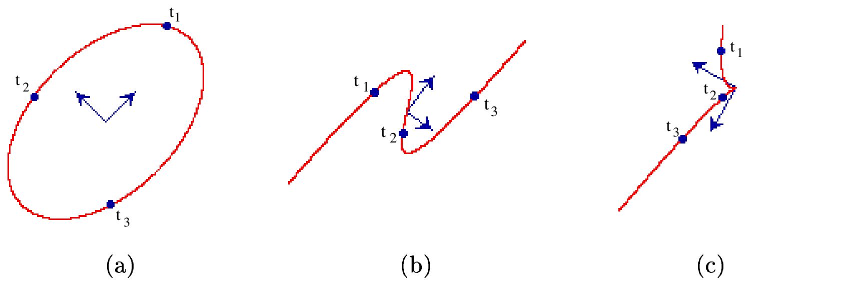


Fig. 6. Recognition of three shapes based on local geometry at three points. (a) An ellipse with $a = 2.8605$ and $b = 1.7263$; (b) a cubical parabola with $a = 3.2543$ and $b = -2.3215$; and (c) a semi-cubical parabola with $a = 2.5683$ and $b = 1.4102$.

For the cubical parabola in Figure 6(b), a test on the invariant I_{c1} has failed. So we know that the curve is not quadratic. Hypothesizing cubical parabola, we solve for λ_1 from (19). Subsequent tests on invariant I_{cp1} (for shape parameter a) yield values 3.2244, 3.2536, and 3.1872, and on invariant I_{cp2} (for b) yield values -2.3237, -2.3237, and -2.3972. The recognition of the semi-cubical parabola in Fig. 6(c) is similar.

7 Preliminary Experiments

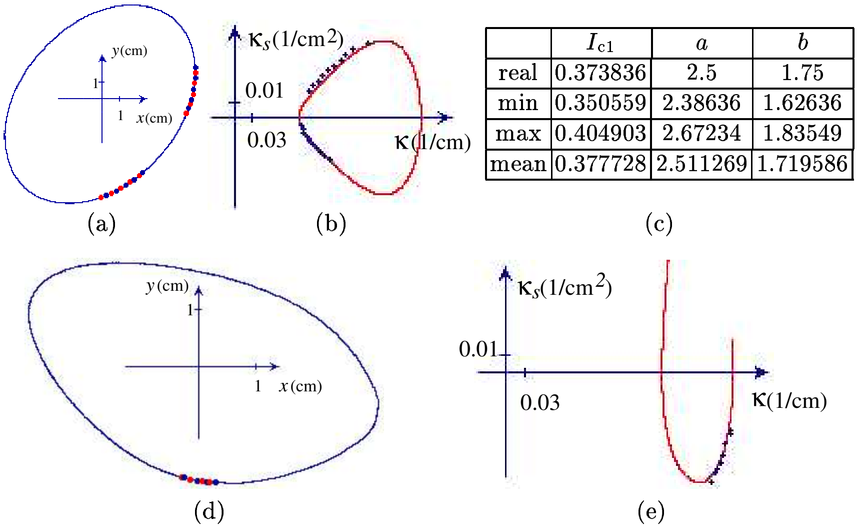
The key for the applicability of our invariant-based approach lies in obtaining reliable estimates of curvature κ and its derivative κ_s from real data. The tactile data used in our experiments were generated by a joystick sensor mounted on an Adept Cobra 600 robot. Despite the Adept's high precision, we found that difference quotients and other estimation methods [3,2] based on the Taylor expansion were still too sensitive to small measurement errors.

To reliably estimate κ at a point, say, p_0, the sensor measures n points (including p_0) in its neighborhood on the shape. Then a quadratic curve is fit over the point sequence. Differentiating this curve at p_0 gives us the curvature.

Similarly, we estimate curvature $\hat{\kappa}_1$ at a second point p_1 following p_0. This is done by sensing l more points along the shape while removing the first l points from the sequence (l is very small so p_0 remains in the new sequence). The arc length $\hat{s}_1$ between p_0 and p_1 is estimated by numerically integrating the new curve fit. Continuing this process, we generate a sequence of pairs $(0, \hat{\kappa}_0), (\hat{s}_1, \hat{\kappa}_1), \ldots, (\hat{s}_{m-1}, \hat{\kappa}_{m-1})$. Now, fit a quadratic curve over this sequence and differentiate it to obtain curvature derivative estimates at the m points. Fig. 7 shows the results for an ellipse and a cubic spline shape.

8 Discussion

We have introduced an invariant-based method that aims at unifying shape recognition, recovery, and pose estimation with tactile information. Differen-



	I_{c1}	a	b
real	0.373836	2.5	1.75
min	0.350559	2.38636	1.62636
max	0.404903	2.67234	1.83549
mean	0.377728	2.511269	1.719586

Fig. 7. Estimating κ and derivative κ_s from tactile data generated by a joystick sensor: (a) an ellipse and 20 sample points; (b) estimates $(\hat{\kappa}, \hat{\kappa}_s)$ at these points plotted against the signature curve; (c) the invariant I_{c1} evaluated at 80 pairs of the same points; (d) a closed cubic spline with 8 sample points from one of its segment; (e) estimates (κ, κ_s) and the signature curve of the segment.

tial and semi-differential invariants have been developed for several classes of low-degree algebraic curves. Evaluated over a few points, these invariants capture intrinsic information about a curve. They allow us to recover the shape parameters of the curve as well as obtain the locations of contact where the data are supposed to be taken.

Although only quadratic curves and special cubic curves are treated, it is straightforward to extend the results to objects bounded by segments of these types. The invariant-based scheme also extends to a curve in implicit form, which can be used along with curvature and its derivative to eliminate the two coordinates. Analytical feasibility of the scheme lies in deriving an equation that describes the signature (or semi-signature) curve. For a curve in 3D, such invariants will have to involve torsion (and possibly its derivatives).

Future work includes improvement on the robustness of curvature and derivative estimation, invariant design for more general plane curves, and extension to 3D surfaces.

Acknowledgment Support for this research has been provided in part by Iowa State University, and in part by the National Science Foundation through a CAREER award IIS-0133681. The authors would like to thank David Kriegman for pointing to differential invariants for shape recognition, and Liangchuan Mi for providing the tactile data used in the experiments. Also thanks to the reviewers for their helpful comments.

References

1. P. K. Allen and P. Michelman. Acquisition and interpretation of 3-D sensor data from touch. *IEEE Trans. Robot. and Automation*, 6(4):397–404, 1990.
2. M. Boutin. Numerically invariant signature curves. *Intl. J. Comp. Vision*, 40(3):235–248, 2000.
3. E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker. Differential and numerically invariant signature curves applied to object recognition. *Intl. J. Comp. Vision*, 26(2):107–135, 1998.
4. H. Civi, C. Christopher, and A. Ercil. The classical theory of invariants and object recognition using algebraic curve and surfaces. *J. Math. Imaging and Vision*, 19:237–253, 2003.
5. R. S. Fearing. Tactile sensing for shape interpretation. In S. T. Venkataraman and T. Iberall (eds.), *Dextrous Robot Hands*, pp. 209–238. Springer-Verlag, 1990.
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 Analysis and Machine Intell.*, 13(10), 1991.
7. W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *Intl. J. Robot. Res.*, 3(3):3–35, 1984.
8. D. Hilbert. *Theory of Algebraic Invariants.* Cambridge University Press, 1993.
9. R. Ibrayev and Y.-B. Jia. Tactile recognition of algebraic shapes using differential invariants. In *Proc. IEEE Intl. Conf. Robot. and Automation*, pp. 1548-1553, 2004.
10. D. Keren. Using symbolic computation to find algebraic invariants. *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(11):1143–1149, 1994.
11. D. Keren, E. Rivlin, I. Shimshoni, and I. Weiss. Recognizing 3D objects using tactile sensing and curve invariants. *J. Math. Imaging and Vision*, 12(1):5–23, 2000.
12. D. J. Kriegman and J. Ponce. On recognizing and positioning curved 3-D objects from image contours. *IEEE Trans. Pattern Analysis and Machine Intell.*, 12(12):1127–1137, 1990.
13. M. Moll and M. A. Erdmann. Reconstructing the shape and motion of unknown objects with active tactile sensors. In J.-D. Boissonnat *et al.* (eds.), *Algorithmic Foundations of Robotics V*, pp. 293–309. Springer-Verlag, 2004.
14. T. Moons, E. J. Pauwels, L. J. van Gool, and A. Oosterlinck. Foundations of semi-differential invariants. *Intl. J. Comp. Vision*, 14(1):25–47, 1995.
15. J. L. Mundy and A. Zisserman. Introduction — towards a new framework for vision. In J. L. Mundy and A. Zisserman (eds.), *Geometric Invariance in Computer Vision*, pp. 1–39. The MIT Press, 1992.
16. T. Pajdla and L. Van Gool. Matching of 3-D curves using semi-differential invariants. In *Proc. IEEE Intl. Conf. Comp. Vision*, pp. 390–395, 1995.
17. E. J. Pauwels, T. Moons, L. J. van Gool, P. Kempenaers, and A. Oosterlinck. Recognition of planar shapes under affine distortion. *Intl. J. Comp. Vision*, 14(1):49–65, 1995.
18. E. Rivlin and I. Weiss. Local invariants for recognition. *IEEE Trans. Pattern Analysis and Machine Intell.*, 17(3):226–238, 1995.
19. I. Weiss. Geometric invariants and object recognition. *Intl. J. Comp. Vision*, 10(3):207–231, 1993.