Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Tactile Recognition of Algebraic Shapes Using Differential Invariants*

Rinat Ibrayev Yan-Bin Jia
Department of Computer Science
Iowa State University
Ames IA 50011-1040
{rinat,jia}@cs.iastate.edu

Abstract

This paper studies the recognition and localization of 2-D shapes bounded by low-degree polynomial curve segments based on minimal tactile data. We have derived differential invariants for quadratic curves and two special classes of cubic curves. Such an invariant, independent of translation and rotation, is computed from the local geometry at *any* two points on the curve. Recognition of a curve class becomes verifying the corresponding invariant with more than one pairs of data points. Next, the actual curve is determined in its canonical parametric form using the same tactile data. Finally, the contact locations on the curve are computed, thereby localizing the shape completely relative to the touching hand. Simulation results support the working of the method in the presence of small noise, although real experiments need to be carried out in the future to demonstrate its applicability. The presented work distinguishes from traditional model-based recognition in its ability to simultaneously recognize as well as localize a shape from one of several classes, each consisting of a continuum of shapes.

1 Introduction

Human can feel the shape of an object through touch. Essentially, the action is performed to detect some geometric features on the object's surface which are then synthesized in the human brain. Typical geometric features include, for instance, smoothness, saliences, concavities, etc.

With the capability of touch sensing, the robot can also obtain shape information. Since tactile data are local (and one-dimensional in the case of point contact), seemingly they contain a very limited amount of geometric information. But how much information about the shape can the robot actually acquire?

Figure 1 illustrates a hand touching an object with two tactile fingers. Suppose with local movements of its fingers the hand could estimate some information such as the curvature at a couple or more points of contact. Then we would like answer the two questions posed in the figure.

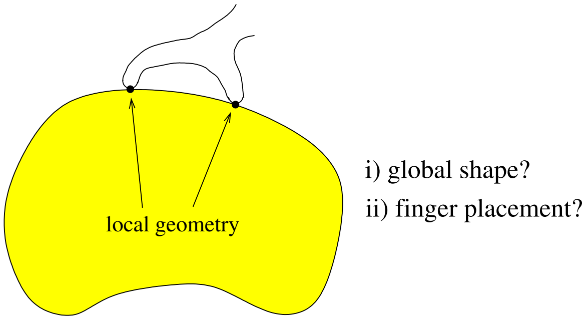

Figure 1: A robotic hand touching the boundary of an object to recognize its shape.

Even with limited information, it is possible for us to determine if the shape is from a certain class (such as the one of all ellipses). And if so, not only could we recover its exact description but also locate where the fingers are placed. To arrive at the above claims, in this paper we will develop a method based on differential invariants which are shape descriptors built upon local geometric variations.

The rest of the section addresses some related work on shape recognition and reconstruction, in both robotics and computer vision; and briefly goes over some basics of curve geometry. Sections 2 and 3 derive invariants for quadratic curves and two special classes of cubic curves. Section 4 follows by determining the locations of the data points on the curve which are used in the invariant computation. In Section 5, all the results are combined into the form of a recognition tree for quadratic and special cubic curves. Section 6 presents some simulation results.

1.1 Related Work

Shape recognition through touch has long built on the notion of interpretation tree which represents all possible correspondences between geometric features of an object with tactile data. The approach was introduced by Grimson and Lozano-Pérez [6] who identified and localized a 3-D polyhedron from a set of polyhedral models using tactile measurements of positions and surface normals. Fearing [4] described how a cylindrical tactile fingertip could recover the pose of a generalized convex cone

*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.

using constraint-based interpretation of a small amount of tactile data.

Montana [11] described a method for estimating local principal curvatures through the rolling of a spherical tactile sensor based on differential equations that govern the contact kinematics. Allen and Michelman [1] employed a Utah-MIT hand to obtain sparse contact points around an object and then fit (in a least-squares manner) a superquadric surface to the data as the reconstructed shape. Boissonnat and Yvinec [2] reconstructed the exact shape of a simple polygon through probing to obtain contact points and normals under some mild conditions. In their work [7], Jia and Erdmann studied how to observe the pose and motion of an object being pushed by a finger, drawing the solution from nonlinear observability theory.

A method based on the interpretation tree or least-squares fitting needs to recover the pose (position and orientation). This may become costly and often unnecessary since the object need only be localized relative to the hand in many situations. Differential invariants, meanwhile, capture intrinsic shape information and are independent of the pose. To recognize spheres, cylinders, cones, and tori, Keren *et al.* [9] constructed differential invariants using curvatures and torsions and their higher order derivatives along a surface curve computed from a large amount of tactile measurements.

In model-based vision, there are also two primary approaches. The first one hinges on the recovery of viewing parameters (thus the pose). Kriegman and Ponce [10] constructed the implicit shape equation from image contours using elimination theory and then solved for viewing parameters through fitting the equation to data points.

The second approach is to develop descriptors that are invariant to Euclidean transformation, perspective projection, or camera-dependent parameters [14]. Algebraic invariants are expressions in terms of the coefficients of a polynomial equation which is often found through fitting. Keren [8] and Forsyth *et al.* [5] introduced methods for finding this type of invariants and demonstrated on recognition of real objects. One drawback of algebraic invariants is the requirement of global shape data. The other drawback is that they do not necessarily work well on shapes that are not algebraic.

Differential invariants depend on local data and deal with situations such as occlusion well. They are functions of curvature and torsion and their derivatives. Calabi *et al.* [3] introduced "signature curves" invariant to Euclidean or affine transformation and described how to numerically approximate differential invariants. Weiss [15] looked into the construction of invariant signature, proposing a method for reliably obtaining higher order derivatives.

Semi-differential invariants combine global constraints and local information to ease the correspondence issue and also relieve the burden on estimating higher order derivatives. The theoretical foundation for this type of invariants was treated thoroughly by Moons *et al.* [12]. Pajdla and Van Gool [13] used semi-differential invariants which combine distance and angle information for matching curves extracted from range data in the presence of partial occlusion.

1.2 Geometric Basics

The touch sensor in contact with a 2-D 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 the parametrization, rotation, and translation. The touch sensor can measure the change of geometry with respect to arc length only. So we will use the derivative of curvature with respect to arc length:

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

The arc length between two points on the curve, if close to each other, can be approximated by their Euclidean distance. Using a straight jaw the robot can accurately measure the jaw rotation as the relative change in the tangential angle between two points. We approximate the curvature and its derivative by the finite difference quotients:

$$\begin{aligned} \kappa &\approx \frac{\phi(s+\Delta s) - \phi(s-\Delta s)}{2\Delta s}, \\ \kappa_s &\approx \frac{\phi(s+\Delta s) - 2\phi(s) + \phi(s-\Delta s)}{(\Delta s)^2}. \end{aligned}$$

2 Quadratics

All quadratic curves are classified into three classes: ellipses, hyperbolas, and parabolas. Together they are referred to as the conics. We will derive invariants for these three classes of conics in the following subsections.

2.1 Parabola

Parabolas are identified with all the curves parametrized by quadratic polynomials:

$$\begin{aligned} x &= a_2t^2 + a_1t + a_0, \\ y &= b_2t^2 + b_1t + b_0. \end{aligned}$$

One curve can be parametrized in many different ways. We are interested in the recovery of the shape of the curve. Hence we are not bounded by one particular parametrization. Moreover, the method we use to recognize the curve does not assume particular position and orientation. So we have the freedom to translate, rotate, and reparametrize the curve in order to get its simplest parametric form. The simplest form for the parabola is

$$\begin{aligned} x &= at^2, \\ y &= 2at, \qquad a \neq 0. \end{aligned}$$

From equations (1) and (2), we obtain

$$\begin{aligned} \kappa &= -\frac{1}{2a(t^2+1)^{3/2}}, && (3) \\ \kappa_s &= \frac{3t}{4a^2(t^2+1)^3}. && (4) \end{aligned}$$

Eliminating t from (3) and (4) leads to the following equation for the parabola:

$$\frac{1}{(2a)^{2/3}} = \kappa^{2/3}\left(\frac{\kappa_s^2}{9\kappa^4}+1\right) \quad \equiv \quad I_{\mathrm{p}}(\kappa, \kappa_s). \tag{5}$$

The expression $I_{\mathrm{p}}(\kappa, \kappa_s)$ is an invariant for the parabola. Since κ and κ_s values are measurable, from (5) we easily calculate the shape parameter a that describes the parabola.

2.2 Ellipse

Let us start with the standard parametrization:

$$\begin{aligned} x &= a\cos(t), \\ y &= b\sin(t), \qquad a, b > 0. \end{aligned}$$

The curvature and its derivative with respect to the arc length are:

$$\kappa = \frac{ab}{\left(a^2\sin^2(t)+b^2\cos^2(t)\right)^{3/2}}, \tag{6}$$

$$\kappa_s = \frac{-3ab\left(a^2-b^2\right)\sin(t)\cos(t)}{\left(a^2\sin^2(t)+b^2\cos^2(t)\right)^3}. \tag{7}$$

We can use equations (6), (7) and $\cos^2(t)+\sin^2(t)=1$ to eliminate $\sin(t)$ and $\cos(t)$, and end up with the following equation:

$$\frac{a^2+b^2}{(ab)^{4/3}} - \frac{1}{(ab\kappa)^{2/3}} - I_{\mathrm{p}}(\kappa, \kappa_s) = 0, \tag{8}$$

where I_{p} is an expression of κ and κ_s defined in (5). Since we have two unknowns a and b, we need at least two points on the ellipse.

Now we derive an invariant for the ellipse using two points. Let κ_i and κ_{si} be the curvature and its derivative at the ith point. Then we have two equations in the form of (8). Subtracting one of the them from the other, we obtain the following after a few more steps:

$$\begin{aligned} \frac{1}{(ab)^{2/3}} &= \frac{(\kappa_1\kappa_2)^{2/3}}{\kappa_1^{2/3}-\kappa_2^{2/3}}\Big(I_{\mathrm{p}}(\kappa_1, \kappa_{s1}) - I_{\mathrm{p}}(\kappa_2, \kappa_{s2})\Big) \\ &\equiv I_{\mathrm{c1}}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}). && (9) \end{aligned}$$

The expression (9) is the invariant that we seek. It stays constant regardless of which two points are used. The invariant I_{c1} alone cannot distinguish ellipses with the same product ab. So we derive a second invariant by substituting I_{c1} for $1/(ab)^{2/3}$ into equation (8):

$$\begin{aligned} \frac{a^2+b^2}{(ab)^{4/3}} &= I_{\mathrm{p}}(\kappa_1, \kappa_{s1}) + \frac{I_{\mathrm{c1}}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2})}{\kappa_1^{2/3}} \\ &= \frac{1}{\kappa_1^{2/3}-\kappa_2^{2/3}}\Big(\kappa_1^{2/3}I_{\mathrm{p}}(\kappa_1, \kappa_{s1}) - \kappa_2^{2/3}I_{\mathrm{p}}(\kappa_2, \kappa_{s2})\Big) \\ &\equiv I_{\mathrm{c2}}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}). && (10) \end{aligned}$$

From the two invariants we can compute the values of ab and a^2+b^2, and subsequently determine a and b.

2.3 Hyperbola

A hyperbola has the parametric form

$$\begin{aligned} x &= a\cosh(t) = a\,\frac{e^t+e^{-t}}{2}, \\ y &= b\sinh(t) = b\,\frac{e^t-e^{-t}}{2}, \qquad a, b > 0. \end{aligned}$$

As in the case of an ellipse, we eliminate t from the equations $\kappa = \kappa(t)$ and $\kappa_s = \kappa_s(t)$ and obtain the following equation for hyperbola:

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

where I_{p} is again defined in (5). We again use two points. From the two copies of equation (11) we derive

$$\begin{aligned} I_{\mathrm{c1}}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}) &= -\frac{1}{(ab)^{2/3}}, \\ I_{\mathrm{c2}}(\kappa_1, \kappa_2, \kappa_{s1}, \kappa_{s2}) &= \frac{a^2-b^2}{(ab)^{4/3}}. \end{aligned}$$

The above two invariants are the same 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.

2.4 General Invariant for Quadratic Curves

Both I_{c1} and I_{c2} are also invariants for a parabola, assuming values 0 and $1/(2a)^{2/3}$, respectively. 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 correlation between any two points on a conic.

3 Cubics

There is no classification of all cubic curves. So, it seems very difficult to construct one invariant that recognizes all of them. However, we would like to deal with cubic spline curves, whose continuity in curvature enables them to approximate any plane curve with almost no visual difference. The general parametric form for cubic spline segment is

$$\begin{aligned} x &= a_3t^3 + a_2t^2 + a_1t + a_0, \\ y &= b_3t^3 + b_2t^2 + b_1t + b_0, \end{aligned}$$

which has the equivalent canonical form

$$\begin{aligned} x &= t^2, \\ y &= at^3 + bt^2 + ct, \end{aligned}$$

where a, b, and c are the shape parameters. This section treats two subclasses of cubic spline polynomials — cubical and semi-cubical parabolas.

3.1 Cubical Parabola

This class of curves has the canonical form:

$$\begin{aligned} x &= t, \\ y &= at^3 + ct, \qquad a \neq 0. \end{aligned}$$

Unlike the conics case, we are not able to eliminate the parameter t from the expressions of curvature κ and its derivative κ_s. Instead, we will substitute t with the slope $\lambda = \frac{y'}{x'} = 3at^2 + c$, which leads to the following expressions for κ and κ_s:

$$\kappa^2 = \frac{12a(\lambda - b)}{(1+\lambda^2)^3}, \tag{12}$$

$$\kappa_s = \frac{6a(1+\lambda^2) - 36a\lambda(\lambda - c)}{(1+\lambda^2)^3}. \tag{13}$$

Using equations (12) and (13) we can solve for a and c:

$$a = \frac{\left(\kappa_s + 3\lambda\kappa^2\right)\left(1+\lambda^2\right)^2}{6} \equiv I_{\mathrm{cp1}}(\lambda, \kappa, \kappa_s), \tag{14}$$

$$c = \lambda - \frac{\kappa^2\left(1+\lambda^2\right)}{2\left(\kappa_s + 3\lambda\kappa^2\right)} \equiv I_{\mathrm{cp2}}(\lambda, \kappa, \kappa_s). \tag{15}$$

The expressions I_{cp1} and I_{cp2} are invariants of the cubical parabola provided that the slope λ can be determined. Denote by ϕ_i, κ_i and κ_{si} the tangential angle, curvature and its derivative at the ith point, respectively. Assuming that the robot can accurately measure the tangent rotation $\Delta\phi_{12} = \phi_2 - \phi_1$, we get the following equation relating the two slopes:

$$\lambda_2 = \frac{\lambda_1 + \delta_{12}}{1 - \lambda_1\delta_{12}}, \tag{16}$$

where $\delta_{12} = \tan\Delta\phi_{12}$. Since the value of I_{cp2} is constant, we have:

$$I_{\mathrm{cp2}}(\lambda_1, \kappa_1, \kappa_{s1}) = I_{\mathrm{cp2}}(\lambda_2, \kappa_2, \kappa_{s2}). \tag{17}$$

Eliminating λ_2 from (16) and (17) results in a quartic polynomial:

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

where

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

By solving the above quartic polynomial we find the value of λ_1, and then the value of λ_2 from (16). Evaluating the expressions I_{cp1} and I_{cp2} gives us the values of a and c, respectively.

3.2 Semi-Cubical Parabola

This class of curves is described by the equations:

$$\begin{aligned} x &= t^2, \\ y &= at^3 + bt^2, \qquad a \neq 0. \end{aligned}$$

The slope is $\lambda = y'/x' = 3at/2 + b$. So this time we reparametrize the curve using $t = \frac{2(\lambda - b)}{3a}$, and obtain the following:

$$a = \sqrt{-\frac{8\kappa^3(1+\lambda^2)^{5/2}}{9(\kappa_s + 3\lambda\kappa^2)}} \equiv I_{\mathrm{scp1}}(\lambda, \kappa, \kappa_s), \tag{18}$$

$$b = \lambda + \frac{\kappa^2(1+\lambda^2)}{\kappa_s + 3\lambda\kappa^2} \equiv I_{\mathrm{scp2}}(\lambda, \kappa, \kappa_s). \tag{19}$$

Again using two points, we can set up an equation:

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

This equation together with (16) yield a quartic polynomial in λ_1. Solving this polynomial will give us λ_1, and subsequently λ_2, a, and b. The invariants for this class of curves are I_{scp1} and I_{scp2}.

4 Locating Contact

The parameter value t determines the contact location on the curve with the touch sensor. For the quadratic and cubic curves discussed in Sects. 2 and 3, the expression for t is as follows:

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

In the case of a cubical parabola, the sign is determined based on the relative configuration of the two data points.

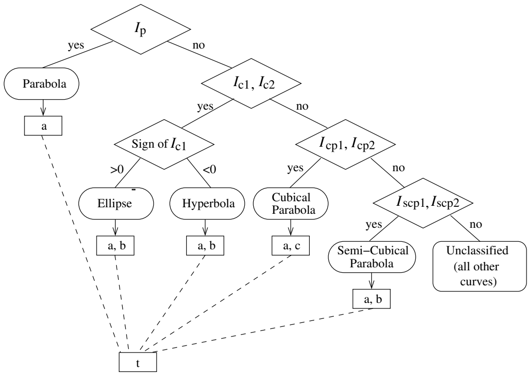

Figure 2: Recognition tree for quadratic and special cubic curves.

5 Recognition Tree

A general recognition strategy is illustrated in Figure 2. We estimate the values of κ and κ_s at as few as three points on the curve. Then we test the invariants down the tree to identify the curve type or determine that it is unclassified. Next, we recover the shape parameters of the curve. Finally, we compute the parameter value t, which determines the contact on the curve.

For example, consider the ellipse in Figure 3(a). The values of κ and κ_s are estimated at $t_1 = 0.36$, $t_2 = 1.86$, and $t_3 = 4.23$. I_p has values 0.8971 and 0.4030 at the first two points, so the curve is not a parabola. I_{c1} yields values 0.3447, 0.3446, and 0.3449 at the three resulting pair of points, from which we conclude that the curve is an ellipse. The recovered coefficients are $a \approx 2.8609$, and $b \approx 1.7275$. The parameter values were also computed correctly. Similarly, we have successfully recognized a cubical parabola as shown in Figure 3 (b).

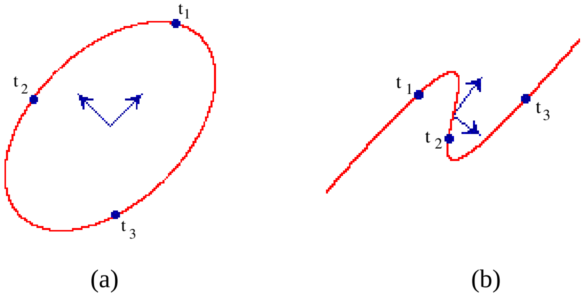

Figure 3: Recognition of two shapes based on local geometry at three points. (a) An ellipse with $a = 2.8605$ and $b = 1.7263$; (b) a cubical parabola with $a = 3.2543$ and $b = -2.3215$.

6 Simulations

The first group of simulations were conducted to verify the invariants of each curve class presented in this paper. One shape out of each class was chosen, and 100 values of invariant were calculated based on randomly generated points. The results are summarized in Table 1.

inv.	I_p	I_{c1}(ell.)	I_{c1}(hyp.)	I_{cp1}	I_{scp2}
real	0.2198	0.1760	-0.1222	6.9963	6.5107
min	0.2168	0.1711	-0.1369	6.7687	6.3945
max	0.2230	0.1790	-0.1147	7.0289	6.5834
mean	0.2198	0.1756	-0.1225	6.9355	6.5154

Table 1: Invariant verification on five specific shapes. The first row shows the real values of the invariants. The following three rows display the min, max, and mean of 100 values computed on perturbed data.

Estimation errors of κ and κ_s were due to linear approximation. They showed up in Table 1 as the discrepancies between actual invariant values and their estimates. Although three points on the curve are enough to recognize it, it would be more reliable to calculate the invariant at more points and take the mean value.

Having verified the invariants, we empirically demonstrated that the invariant of one curve class would not hold for another. This is necessary for the recognition strategy to work. Since all quadratic curves share the invariant I_{c1}, there are only three curve classes. We tested the invariants of one curve class against the data from another. The results are summarized in Table 2.

╲ inv. data ╲	I_{c1}	I_{cp2}	I_{scp2}
quadratics (ellipse)	——	-11.97(min) -15.46(max) -0.04(mean) 2.53(stdev)	-265.80 5.83 -3.22 26.75
cubical parabola	-6.38 -0.04 -0.73 1.22	——	7.80 65.22 29.17 17.19
semi-cub. parabola	-22.84 28.37 3.37 6.76	8.54 19.03 13.76 3.07	——

Table 2: Applying data from one curve on the invariant of a different class. Each cell displays the summary over 100 values.

From the table we see that when an invariant is applied to curves outside the curve class it was derived for, it has different values for different points. So, each invariant only holds for its own curve class.

Next, we looked into how well a given curve can be recognized. In other words, we examined how much the recovered parameters $\bar{a}$ and $\bar{b}$ would differ from the real ones a and b. For measurement, we calculated the relative errors of recovered parameters with respect to real ones as $\sqrt{\left(\frac{a-\bar{a}}{a}\right)^2 + \left(\frac{b-\bar{b}}{b}\right)^2}$. The calculations used 100 different shapes from each family. For each recovered shape the relative error was calculated. The results are summarized in Table 3.

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

Table 3: Relative error on estimation of a and b. Summary over 100 different curves for each class.

From Table 3 we can see that on the average the relative errors are around 1%. These errors depend on how well we estimate the curvature and its derivative. Finite differencing was used. An improvement would be to approximate the osculating circle and use the inverse of its radius, as introduced in [3].

7 Conclusion

We have introduced an invariant-based method that aims at unifying shape recognition, recovery, and pose estimation through touch. Differential and semi-differential invariants have been developed for several classes of low-degree algebraic curves. Each invariant characterizes the geometric correlations on a curve as determined inherently by the corresponding curve class. The canonical parametrization of the actual curve can meanwhile be recovered from the curvature and its derivative at as few as three points. The task of shape reconstruction is thus simplified. Furthermore, locations of contact on the curve, that is, the parameter values, can be estimated from the same data.

The small data requirement by our method makes it desirable for the application of touch sensors. 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. In such a situation, the data points plugged into each invariant need to be from the same segment. So the total amount of data to be obtained is linear in the number of segments on the object's boundary.

We are working on an extension of the results to closed cubic splines which, given their curvature continuity, can approximate any 2-D curved shapes very well. The extension will have strong implications in recognizing general 2-D shapes (and images).

We would like to find out the robustness of the introduced invariants in the presence of sensor noise as well as the errors due to numerical difference in curvature approximation. A reliable noise model needs to built. In the future, we would like to move on to the simultaneous recognition, reconstruction, and localization of 3-D curved shapes. This will be a much more challenging task.

References

[1] P. K. Allen and P. Michelman. Acquisition and interpretation of 3-D sensor data from touch. *IEEE Trans. Robot. and Automat.*, 6(4):397–404, 1990.

[2] J. D. Boissonnat and M. Yvinec. Probing a scene of nonconvex polyhedra. *Algorithmica*, 8(3):321–342, 1992.

[3] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker. Differential and numerically invariants signature curves applied to object recognition. *Int. J. Comp. Vision*, 26(2):107–135, 1998.

[4] 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.

[5] 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 Analys. and Mach. Intell.*, 13(10), 1991.

[6] W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *Int. J. Robot. Res.*, 3(3):3–35, 1984.

[7] Y.-B. Jia and M. Erdmann. Pose and motion from contact. *Int. J. Robot. Res.*, 18(5):466–490, 1999.

[8] D. Keren. Using symbolic computation to find algebraic invariants. *IEEE Trans. Pattern Analys. and Mach. Intell.*, 16(11):1143–1149, 1994.

[9] 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.

[10] D. J. Kriegman and J. Ponce. On recognizing and positioning curved 3-D objects from image contours. *IEEE Trans. Pattern Analys. and Mach. Intell.*, 12(12):1127–1137, 1990.

[11] D. J. Montana. The kinematics of contact and grasp. *Int. J. Robot. Res.*, 7(3):17–32, 1988.

[12] T. Moons, E. J. Pauwels, L. J. van Gool, and A. Oosterlinck. Foundations of semi-differential invariants. *Int. J. Comp. Vision*, 14(1):25–47, 1995.

[13] T. Pajdla and L. Van Gool. Matching of 3-d curves using semi-differential invariants. In *Proc. IEEE Int. Conf. Comp. Vision*, pp. 390–395, 1995.

[14] I. Weiss. Geometric invariants and object recognition. *Int. J. Comp. Vision*, 10(3):207–231, 1993.

[15] I. Weiss. Noise-resistant invariants of curves. *IEEE Trans. Pattern Analys. and Mach. Intell.*, 15(9):943–948, 1993.