

# Recognizing Surfaces Using Curve Invariants and Differential Properties of Curves and Surfaces

Daniel Keren\* Ehud Rivlin† Ilan Shimshoni† Isaac Weiss‡

## Abstract

A general paradigm for recognizing 3D objects is offered, and applied to some geometric primitives (spheres, cylinders, cones, and tori). The assumption is that a curve on the surface was measured with high accuracy (for instance, by a sensory robot). Differential invariants of the curve in one method and differential properties of curves and surfaces in the other are then used to recognize the surface. The motivation is twofold: the output of some devices is not surface range data, but such curves. So, surface invariants, which may be simpler in some cases, cannot always be obtained. Also, a considerable speedup is obtained by using curve data, as opposed to surface data which usually contains a much higher number of points.

## 1 Introduction and Previous Work

One task an intelligent system should be able to accomplish is *recognition*. Usually, a recognition system derives some characteristics of an object it examines, and tries to match them against similar characteristics in a database. Suppose, for instance, that one is dealing with 2D objects, and tries to recognize them, given their boundary. Typically, there is a finite database these boundaries are matched against; various *invariants* have been derived, some global and some local [13, 12, 18], to solve this problem. These are quantities that do not change under certain transformations (Euclidean, affine, projective), and therefore can be used to recognize an object even after it had been altered by such transformations.

Here, a different problem is addressed – recognizing a surface in 3D space, while the information we have is one-dimensional. Specifically, we assume that some measuring device has sampled a curve on the surface.

\*Department of Computer Science, University of Haifa, Haifa 31905, Israel, E.mail: dkeren@mathcs2.haifa.ac.il

†Department of Computer Science, Technion, Technion City 32000, Israel, E.mail: (ehudr,ilan)@cs.technion.ac.il

‡Center for Automation Research, University of Maryland, College Park, MD, 20742 USA, E.mail: weiss@cfar.umd.edu

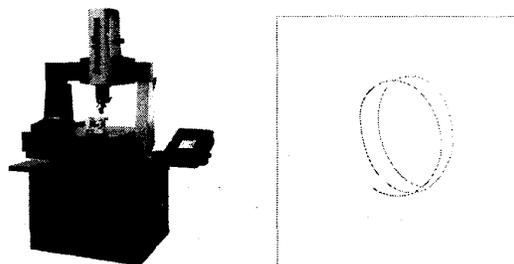


Figure 1: High-accuracy measuring device and a curve it measured on a cylinder.

Given the curve, the goal is to recognize the surface. Typical sensors which are the source of such curves are measuring devices, such as coordinate measuring machines, manufactured by the Brown & Sharpe Company (Figure 1), or the IBM RS/1 Cartesian robot. Such devices can measure 3D curves with very high accuracy (for instance, typical error range for a coordinate measuring machine is 0.01 mm). Another source of curves on a surface is stereo; if there are shapes (such as letters) on a surface, they can be used to recover curve data, but usually surface data will be much harder to obtain, due to the difficulty of solving the matching problem in smooth areas. We focus here on the case where surface (range) data is not available.

In [1], an algorithm is presented for determining the axis of a surface of revolution, using the information measured by a tactile sensor which can also estimate the two principle curvatures. Here, we do not assume that the two curvatures are given. In [5], the parameters of a cylinder are computed from structured light patterns.

Some previous work has addressed the problem of recognizing various surfaces given their occluding contours [10, 7]. However, the aggregate of possible curves on a surface is much larger than the aggregate of its occluding contours, and may contain far more complicated curves; for instance, the occluding contour of a sphere is always a circle, while there are a great many 3D curves – some of which have rather complicated

structure – on a sphere.

One way to proceed is straightforward: fit an implicit polynomial to the curve's points, and, from its type, determine the surface. This is the *algebraic approach* [15, 9]. However, this approach will fail if the curve does not lie on a single "primitive" (sphere, cylinder etc), but "crosses over" between two or more primitives (see Section 2 and Figure 2). In that case, the global algebraic fit will give us a meaningless result. A very rich theory of *local*, or *differential* invariants, was developed to solve this problem [3, 2, 6, 17, 16, 8].

Natural curve characteristics to use for recognition are curvature and torsion, as they do not change under rigid transformations. Since we're dealing with 3D data, a rigid transformation is usually a general enough model. So, the goal is to discover invariants depending on a curve's curvature and torsion, which will provide a necessary condition for it to lie on a cylinder (we have obtained results for some other surfaces, but will not include them here due to lack of space). These depend only on the curvature and torsion at a point on the curve.

In Section 4 we present a second method to solve the same problem, which uses not only the differential properties of the curve (tangent and normal vectors and the curvature), but their relationships to the differential properties of the surface. These differential properties measured at one or more points on the curve are used to recognize the type of primitive (sphere, cylinder, cone or torus) on which the those points lie and also recover their shape parameters. Similar techniques have been used for recognizing plane and space curves [2, 16, 14] but not the surfaces on which they lie.

## 1.1 Applying Invariants to Segmentation

Since the algebraic approach for recognition given a curve may fail, because it can pass through a few geometric primitives, one may try to segment the curve, using some notion of discontinuity, and then use algebraic techniques for each segment. We now show that this is not always easy, by constructing a curve which is infinitely differentiable, yet crosses over from a sphere to a cylinder. The curve is displayed in Figure 2. Next to it, we plot the curvature, torsion, curvature's derivative, and a spherical invariant for curves. It is interesting to see that, although the curvature and torsion are continuous, there is a very sharp break in the spherical invariant, at the point in which the curve crosses over from the sphere to the cylinder; this demonstrates that the kind of invariants presented

here can succeed where segmentation by "ordinary" differential properties (curvature, torsion etc.) fails.

## 2 Curve Invariants Method

In this section, a general overview of the method for deriving differential invariants for curves lying on surfaces is provided.

We wish to find conditions on the curvature and torsion of a curve  $C$  which will allow us to determine if it possibly lies on a certain geometric object  $OBJ$ , which is described by a generic implicit equation,  $P(x, y, z) = 0$ .

The method by which these conditions is derived proceeds as follows. First, we use the local canonical form [4] to write down an expression for  $C$  in the vicinity of a point  $M$  we have measured on  $OBJ$ ; we also assume that we have measured  $\kappa$ ,  $\tau$ , and their derivatives, as well as the Frenet trihedron at  $M$ . These are all determined from the derivatives of  $C$ ; so, if we have accurate measurements for  $C$  in the vicinity of  $M$ , we may directly calculate them. Since  $\kappa$  and  $\tau$  do not depend on the pose of the  $C$ , we are allowed to translate and rotate  $OBJ$  – and the curve on it – thus obtaining a new curve  $\mathcal{C}$ . Denote the rotated and translated object by  $OBJ_{new}$ .

Every condition on  $\tau$  and  $\kappa$  we derive for  $\mathcal{C}$  is, of course, also a condition for  $C$ . The reason we apply rigid transformations to  $OBJ$  is because these allow us to make assumptions on  $\mathcal{C}$ 's Frenet trihedron which result in simpler calculations; this will be explained in the sequel. Let  $\mathcal{P}(x, y, z)$  be the implicit equation defining  $OBJ_{new}$ .

Next, we substitute  $\mathcal{C}$ 's local canonical form into  $\mathcal{P}(x, y, z)$ ; This results in a Taylor series in  $s$ . This series *has to be identically zero*, because  $\mathcal{C}$  is contained in  $OBJ_{new}$ , and, therefore, has to satisfy the equation which defines  $OBJ_{new}$ . This gives us a set of equations – each for every coefficient in the Taylor series. Next, we eliminate from these equations everything *but*  $\mathcal{C}$ 's curvature and torsion. For one curve, we usually have to eliminate the Frenet trihedron. For two curves, we will show that the Frenet trihedrons are known and therefore need not be eliminated. In both cases, the elimination gives an expression that has to be zero; and this is the sought invariant.

## 3 The Case for a Cylinder

### 3.1 One Curve, Known Radius

We now proceed to derive differential invariants for a curve which lies on a cylinder. To the best of our

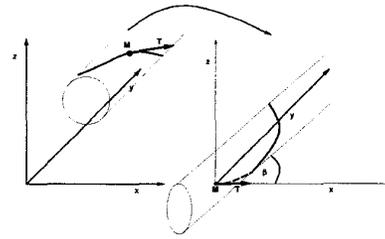
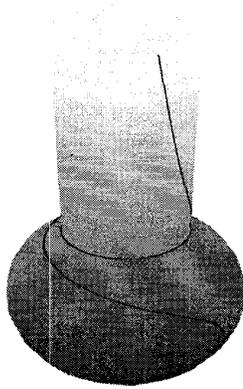


Figure 3: Rotating and translating the cylinder.

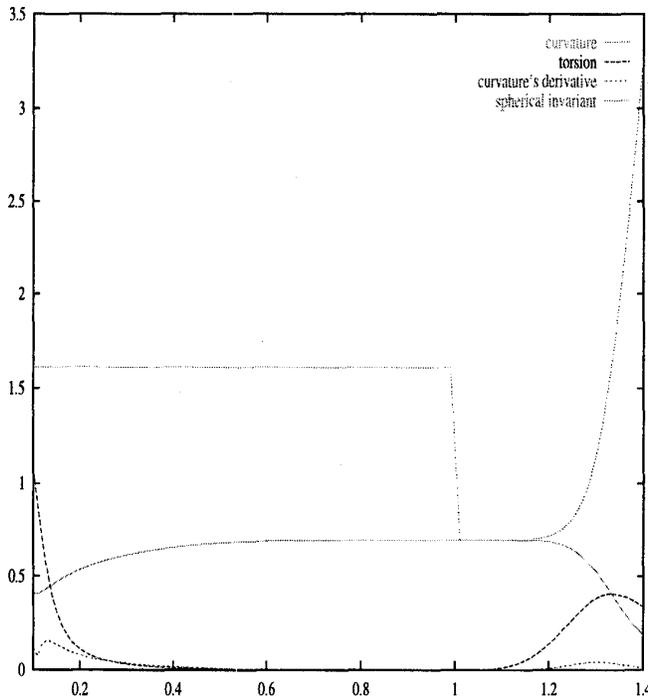


Figure 2: Demonstrating how invariants manage to detect when a curve crosses over from one geometric primitive (sphere) to another (cylinder), although the curvature, torsion etc. cannot detect this crossing over.

knowledge, such invariants have not been derived before.

Given a point  $M$  on a curve which lies on a cylinder, we can assume without loss of generality that the cylinder had been translated and rotated so that  $M$  is at the origin, and the cylinder lies on the  $XY$  plane (recall that this does not alter the curvature and torsion). Let us further assume that it had been rotated at some angle  $\beta$  so that the tangent vector at  $M$  is aligned with the  $X$ -axis (see Figure 3).

Hence,  $T = (1, 0, 0)$ , and the cylinder's equation becomes

$$(x \cos(\beta) + y \sin(\beta))^2 + (z - R)^2 - R^2 = 0 \quad (1)$$

It follows that  $N = (0, \cos(\alpha), \sin(\alpha))$  for some  $\alpha$ , and  $B = (0, -\sin(\alpha), \cos(\alpha))$ .

We now substitute these  $T, N, B$  in the local canonical form [4], which yields expressions for  $c(s)$ 's components as power series in the arc-length parameter  $s$ .

Plugging these into the cylinder's equation (1) gives a Taylor series in  $s$  which has to be identically zero. This expression is huge and we do not write it down here; we need only the coefficients of the powers of  $s$  between 0 and 5.

The coefficients of the constant and linear terms are identically zero.

For the other terms, we obtain the following expressions, after substituting  $\cos(\alpha) = x, \sin(\alpha) = y, \cos(\beta) = z, \sin(\beta) = w$ :

For the coefficient of  $s^2$

$$-2\kappa y R + 2z^2 = 0 \quad (2)$$

For the coefficient of  $s^3$

$$6zw\kappa x - 2\kappa' y R + 2\kappa\tau x R = 0 \quad (3)$$

For the coefficient of  $s^4$

$$8zw\kappa\tau y - 2\kappa''yR + 4\kappa'\tau xR + 2\kappa^3yR + 2\kappa\tau'xR + 2\kappa\tau^2yR - 6\kappa^2x^2z^2 + 8zw\kappa'x - 8z^2\kappa^2 + 6\kappa^2 = 0 \quad (4)$$

$$(-36k^6\tau^2R^2 - k^2\tau^2R^2\kappa'^2 + 6\tau\kappa'k^3\tau'R^2 + 3\kappa'^2R^2\kappa''k - 2k^4\tau^4R^2 - 5R^2\kappa'^4 + 72R^2\kappa'^2k^4 + 162k^6 - 3\tau^2R^2\kappa''k^3 - 162k^8R^2 - 27R^2\kappa''k^5)q^4 + 90\tau\kappa'Rk^4q^3 + (-3\kappa'^2R^2\kappa''k + 6R^2\kappa'^4 + 9k^2\tau^2R^2\kappa'^2 - 27R^2\kappa'^2k^4)q^2 - R^2\kappa'^4 = 0$$

In addition, we have an equation for the coefficient of  $s^5$  and

$$x^2 + y^2 - 1 = 0 \quad (5)$$

$$z^2 + w^2 - 1 = 0 \quad (6)$$

Assume now that the radius  $R$  is known. In that case, we have to eliminate  $x, y, z, w$  from Equations (2,3,4,5,6) (note that we need at least five equations in order to eliminate four unknowns). All our attempts to directly do this, using various packages for symbolic computations, have failed; however, it is possible to proceed as follows. First, solve the system consisting of the four simplest equations (2,3,5,6). Then, substitute the solution into (4).

Using the Maple symbolic computation program, it was possible to find a solution for Equations (2,3,5,6). This solution uses an auxiliary polynomial we denote by  $p_1(\delta)$ :

$$p_1(\delta) = 81\kappa^8\delta^8R^2 + (18R^2\kappa'^2\kappa^4 - 18\kappa^6\tau^2R^2 + 81\kappa^6 - 162\kappa^8R^2)\delta^6 + 36\kappa^4R\delta^5\kappa'\tau + (-81\kappa^6 + 81\kappa^8R^2 - 36R^2\kappa'^2\kappa^4 + \kappa^4\tau^4R^2 + 2\kappa^2\tau^2R^2\kappa'^2 + R^2\kappa'^4 + 18\kappa^6\tau^2R^2)\delta^4 - 36\kappa^4R\delta^3\kappa'\tau + (18R^2\kappa'^2\kappa^4 - 2\kappa^2\tau^2R^2\kappa'^2 - 2R^2\kappa'^4)\delta^2 + R^2\kappa'^4$$

Denote by  $q$  a root of  $p_1(\delta)$ . Then, the solution of (2,3,5,6) equals

$$z = R\sqrt{\frac{9\kappa^4q^4 - q^2\kappa^2\tau^2 + q^2\kappa'^2 - 9\kappa^4q^2 - \kappa'^2}{q(9q\kappa^2 + 2\kappa'\tau R)}}$$

$$= \frac{-9\kappa'R\kappa^4q^4 + \kappa'Rq^2\kappa^2\tau^2 + \kappa'^3Rq^2 - 9\kappa'R\kappa^4q^2 - \kappa'^3R + 9\kappa^4\tau q^3}{3q^2(9q\kappa^2 + 2\kappa'\tau R)\kappa^2\sqrt{\frac{9\kappa^4q^4 - q^2\kappa^2\tau^2 + q^2\kappa'^2 - 9\kappa^4q^2 - \kappa'^2}{q(9q\kappa^2 + 2\kappa'\tau R)}}}$$

$$y = -\frac{R(9\kappa^4q^4 - q^2\kappa^2\tau^2 + q^2\kappa'^2 - 9\kappa^4q^2 - \kappa'^2)}{q(9q\kappa^2 + 2\kappa'\tau R)\kappa}$$

$$x = q$$

Substituting these expressions into (4) and simplifying, we obtain the following identity

$$(18k^6\tau^2R^2 - 45R^2k'^2k^4 + 27R^2\kappa''k^5 + 162k^8R^2 - 81k^6)q^6 - (27k^5\tau'R - 54\tau\kappa'Rk^4)q^5 +$$

Let us denote this polynomial by  $p_2(q)$ .

Now, we know that  $p_1()$  and  $p_2()$  must have a common root; therefore, their resultant must be zero. This resultant is, therefore, an invariant for a curve lying on a cylinder.

Using elimination techniques we can write down the resultant of  $p_1()$  and  $p_2()$ . It is a determinant whose elements depend on the curvature and torsion; if the curve lies on a cylinder, this determinant has to be zero, and this is an invariant for a curve lying on a cylinder.

## 3.2 One Curve, Unknown Radius

### 3.2.1 Numerical Search for the Correct Radius

Suppose we do not know the radii of the cylinders in the database.

There are two ways to proceed. We can simply follow the trivial observation that, if we substitute the correct  $R$  into  $p_1()$  and  $p_2()$ , we will get two polynomials whose resultant is zero. We can therefore conduct a simple, one-dimensional search for  $R$  which minimizes this resultant.

Experience has shown us that this simple numerical algorithm works quite well. For example, in Figure 4, a plot for the logarithm of the resultant, for values of  $\kappa$  and  $\tau$  measured on a curve on a cylinder with a radius of 2, is displayed. We can clearly see a strong minimum at the correct radius.

### 3.2.2 Solve for the Correct Radius

The second method for the case in which the radius is unknown is to eliminate  $R, x, y, z, w$  from Equations 2,3,4,5,6.

This can be done by solving Equations 2,3,4,5,6, and substituting the solution in the equation for  $s^5$ ; if this gives zero, it means that these six equations have a common solution, which is a necessary condition for the curve to lie on a cylinder. It turns out that Equations 2,3,4,5,6 do have a relatively simple solution, expressed as follows.  $x$  is the root of a polynomial equation whose coefficients can be expressed as functions of  $\kappa$  and  $\tau$ . We skip the details, due to lack of space. After  $x$  is solved for, we can easily extract  $y$  from Equation 5. Then, after substituting the known

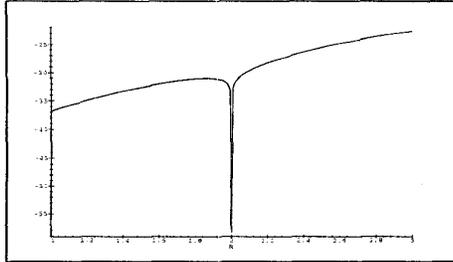


Figure 4: Logarithm of the resultant of  $p_1()$  and  $p_2()$ , as a function of  $R$ , for a point sampled from a curve on a cylinder of radius 2.

values of  $x$  and  $y$  in Equations 2,3,6 we can solve for the remaining unknowns –  $w$ ,  $z$  and  $R$ :

Define  $\epsilon$  to be

$$\kappa^2 \tau^2 x^2 - 2 k \tau x y \kappa' + y^2 \kappa'^2 + 9 \kappa^4 x^2 y^2$$

and then

$$\left\{ w = -\frac{\text{Root}(\epsilon \delta^2 - 9 \kappa^4 x^2 y^2)(k \tau x - y \kappa')}{3 \kappa^2 x y}, \quad (7) \right.$$

$$\left. z = \text{Root}(\epsilon \delta^2 - 9 \kappa^4 x^2 y^2), R = \frac{9 \kappa^3 y x^2}{\epsilon} \right\}$$

(by *Root* of an equation, we mean the root of the equation when viewed as an equation in  $\delta$ ). The equations in (7) are trivial to solve and involve only taking square roots.

### 3.2.3 Comparison of Methods

While the second method is straightforward and does not require any search (as opposed to the first method), it has the drawback of requiring the fifth derivative of the curve. The first method requires a numerical search for the correct radius, but uses only the fourth derivative of the curve. Depending on how accurate the measurements are, one may opt for using the first or the second method.

## 4 A Method Using Differential Properties of Surfaces

In this section we study curve invariants which use only curvature (this requires computing only the first

and second derivatives of the curve). We also assume that the only primitives the recognition system may encounter are spheres, cylinders, cones, and tori. When the information from one point is not enough to uniquely determine the object, we will use an additional point or two on the curve to help disambiguate the object.

Each of the classes of objects mentioned above have a small number of parameters which determine its shape. The sphere is defined by its center and radius (four parameters) and the cylinder, cone, and torus have 5, 6, and 7 parameters respectively.

In order to be able to recover the shape of primitives, constraints which involve the differential properties of the curve and shape parameters have to be derived, simple techniques for recovering the shape parameters from these constraints have to be found, and additional constraints are used to verify that the shape is correct.

For each point on a curve the proposed primitive must satisfy the following constraints:

- The point  $M$  must lie on the surface. This means that if  $P$  is the object's implicit equation,  $P(M) = 0$ .
- $T_C$ , the curve's tangent, must be orthogonal to the surface normal  $N_S$  at the point. Thus  $N_S \cdot T_C = 0$ .
- If  $\theta$  is the angle between  $N_S$  and  $N_C$ , then  $\kappa_S = \kappa_C \cos(\theta)$ , where the value of  $\kappa_S$  (the curvature of the normal section on the surface) is determined by the principal curvatures  $\kappa_1$  and  $\kappa_2$  and the angle between them and  $T_C$ .

Therefore, each point yields three equations which have to be satisfied. These equations can be used to verify hypotheses or to determine the value of unknown parameters.

When two curves intersect, at the intersection point only five constraints exist because the first constraint for the two curves is identical.

If additional points are not on a curve, and we don't have any differential properties associated with them, we still have the first condition (they have to satisfy the surface equation). In that case, we will need more points; this is a typical tradeoff for semi-differential invariants.

In all the cases considered, we will require at least as many constraints as unknown shape parameters and use the remaining (or additional) constraints to verify the shape hypothesis.

#### 4.1 Object Recognition from Two Intersecting Curves

Given two intersecting curves  $C_1$  and  $C_2$ , we extract  $T_1, N_1, B_1, \kappa_{c1}, T_2, N_2, B_2, \kappa_{c2}$  at the intersection point  $M$ . These are the Frenet trihedrons and the curvature for both curves respectively. Recall that  $N_S$ , the normal to the surface at  $M$ , equals  $T_1 \times T_2$ .

For each curve we compute  $\theta$ , the angle between  $N_S$  and the curve's normal. The surface normal curvature equals  $\kappa_{N_S} = \kappa_C \cos(\theta)$ , and  $\kappa_{N_S}(\beta) = \kappa_1 \sin^2(\beta) + \kappa_2 \cos^2(\beta)$ , where  $\kappa_1, \kappa_2$  are the principal curvatures for the surface at  $M$ , and  $\beta$  is the angle between the tangent to the curve and  $\kappa_2$ , the second principal direction.

Given two curves we have two equations for the surface normal curvature, with three unknowns -  $\kappa_1, \kappa_2$ , and  $\beta$ :

$$\begin{aligned} \kappa_{N_{S1}} &= \kappa_1 \sin^2(\beta) + \kappa_2 \cos^2(\beta) \\ \kappa_{N_{S2}} &= \kappa_1 \sin^2(\beta + \phi) + \kappa_2 \cos^2(\beta + \phi), \end{aligned} \quad (8)$$

where  $\phi$  the angle between  $T_1$  and  $T_2$  is known. Usually, it is impossible to solve such a system; however, if we know in advance that the geometric primitives can only be spheres, cylinders, cones, and tori, it is possible to identify them and extract their parameters.

If the given object is a cylinder, its parameters can be recovered as follows. As  $\kappa_1 = 0$ , the surface normal equations are reduced to two equations with two unknowns. Solving them, we can recover  $\kappa_2$  and the principal directions  $\kappa_1, \kappa_2$ . The cylinder's radius is  $R = \frac{1}{\kappa_2}$ , and the orientation of its axis is  $\kappa_1$ . A point on the axis is:

$$C = M + RN_S.$$

It is important to note that this does not prove that the object is a cylinder. That has to be verified using an additional point on the curve.

#### 4.2 Object Recognition from One Curve

When two intersecting curves are given, we are able to recover  $N_S$  and thus we know the angle  $\theta$  between  $N_S$  and  $N_C$ . When we are given only one curve,  $\theta$  is an unknown parameter which has to be recovered.

In the case of the cylinder we know that  $\kappa_1 = 0$  and  $\kappa_2 = \frac{1}{R}$ . Given a point  $M_1$  on the curve, the two unknowns are  $\theta_1$  and  $\beta_1$ . When they are given, the cylinder is uniquely defined. Note that  $\kappa_1$  is parallel to the axis of the cylinder, so it has to be the same for every point on the cylinder. We will now use these facts to define  $R$  and  $\kappa_1$  the axis of the cylinder as

functions of  $\theta_1$  and  $\beta_1$ :

$$R = \frac{\cos^2(\beta_1)}{\kappa_{C1} \cos(\theta_1)}.$$

$$N_S = \cos(\theta_1)N_C + \sin(\theta_1)B_C$$

$$\kappa_1 = T_C \sin(\beta_1) + (T_C \times N_S) \cos(\beta_1) \quad (9)$$

And a point on the axis is:

$$C_1 = M_1 + RN_{S1}$$

Given an additional point, its  $\beta_2$  and  $\theta_2$  can be recovered as follows:

$$\beta_2 = \arcsin(T_{C2} \cdot \kappa_1), \quad \theta_2 = \arccos\left(\frac{\cos^2(\beta_2)}{\kappa_{C2} R}\right).$$

From them we can recover the point on the axis  $C_2$  closest to the second point, and both points must lie on the cylinder's axis, which is parallel to  $\kappa_1$ ; therefore,

$$(C_1 - C_2) \times \kappa_1 = 0,$$

which gives us two equations in two unknowns, which can be solved for the values of  $\theta_1$  and  $\beta_1$ .

These two points give the equation of the cylinder that passes through them and satisfies the given constraints. In addition, from (9)

$$\beta_2 = \arccos((T_{C2} \times N_{S2}) \cdot \kappa_1),$$

which gives an additional constraint to verify that this is indeed a cylinder with the computed parameters.

### 5 Experimental Results

The algorithm for a single curve has been tested on real data received from the Brown & Sharpe Company using their coordinate measuring machines (Figure 1). The data is a curve measured on a cylinder. For each point on the curve  $T_C, N_C, B_C$ , and  $\kappa_C$  are estimated. Using the algorithm described above, the problem is reduced to solving for  $\cos(\theta_1)$  and  $\cos(\beta_1)$ , where all other parameters are expressed as functions of these unknown values. The correct values must satisfy four equations and have to satisfy the constraints that the absolute values of the cosine and sine of the various angles must be less than 1. The values of the unknowns are found using non-linear least squares optimization techniques. In this case we use the Levenberg-Marquardt procedure of the MINPACK library [11].

We chose at random 200 pairs of points and ran the minimization procedure on them using several initial conditions for the minimization. Even though the data is noisy, most pairs of points yielded results close to the correct shape. The results were sorted according to the least-squares error (LSE) of the four equations. We trace the five cylinders with the smallest LSE in Figure 5(a). One of these results and the original data are shown in 5(b). It is important to note that only the data on the two points and their derivatives mentioned above was used to recover the shape of the cylinder. Additional points can then be used, if desired, to get a better estimate for the shape.

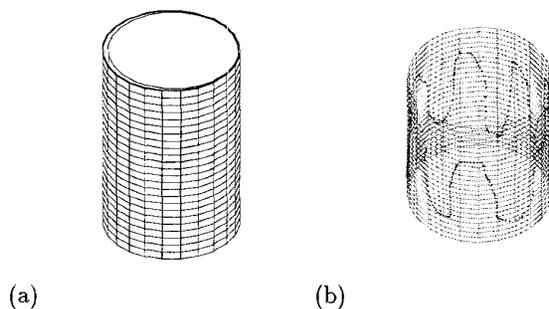


Figure 5: (a) The five recovered cylinders with the lowest LSE. (b) The recovered shape of the cylinder and the original data points.

## 6 Conclusions

Two novel methods to recognize some surfaces, given curve(s) on them, were presented. The first method exploits invariants which are computed on curves, that supply information on the type of surfaces the curve can possibly lie on. The second method achieves the same result by finding relationships between the differential properties of the curves and the surfaces on which they lie.

The methods can use 3D curves derived from stereo and structured light; it is particularly useful when given the output of measuring devices which produce such curves (for instance, sensory robots and coordinate measuring machines).

The main advantage of the proposed method compared to algebraic methods is in its local nature, which enables it to segment and recognize curves (and the surfaces they lie on), even if the curves lie on more than one geometric primitive. Also, it necessitates a far smaller number of curve points than the algebraic method, for recognizing a single primitive.

## References

- [1] M.D. Berkemeier and R.S. Fearing. Determining the axis of a surface of revolution using tactile sensing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:1079–1087, 1993.
- [2] A.M. Bruckstein, R.J. Holt, A.N. Netravali, and T.J. Richardson. Invariant signatures for planar shape recognition under partial occlusion. *IU*, 53:49–65, 1993.
- [3] A.M. Bruckstein, N. Katzir, M. Lindenbaum, and M. Porat. Similarity-invariant signatures for partially occluded planar shapes. *IJCV*, 7:271–285, 1992.
- [4] M.P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.
- [5] Z. Chen, T.L. Chia, and S.Y. Ho. Measuring 3D location and shape parameters of cylinders by a special encoding technique. *IEEE Trans. on Robotics and Automation*, 10:632–647, 1994.
- [6] M. Van Diest, L.J. Van Gool, T. Moons, and E.J. Pauwels. Projective invariants for planar contour recognition. In *European Conference on Computer Vision*, pages 527–534, 1993.
- [7] D.A. Forsyth. Recognizing algebraic surfaces from their outlines. In *International Conference on Computer Vision*, pages 476–480, Berlin, May 1993.
- [8] A. Guezic and N. Ayache. Smoothing and matching of 3D curves. *International Journal of Computer Vision*, 12(1):79–104, 1994.
- [9] D. Keren. Using symbolic computation to find algebraic invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1143–1149, 1994.
- [10] D.J. Kriegman and J. Ponce. On recognizing and positioning curved 3D objects from image contours. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:1127–1138, December 1990.
- [11] J.J. Moré, B.S. Garbow, and K.E. Hillstom. User guide for MINPACK-1. ANL-80-74, Argonne National Laboratories, 1980.
- [12] J. Mundy and A. Zisserman. *Geometric Invariance in Machine Vision*. MIT Press, 1992.
- [13] J. Mundy and A. Zisserman. *Applications of Invariants in Computer Vision II*. ESPRIT, ARPA/NSF, Ponta Delgada, Azores, October 1993. Conference organized by J. Mundy and A. Zisserman.
- [14] T. Pajdla and L. VanGool. Matching of 3d curves using semi-differential invariants. In *ICCV*, pages 390–395, 1995.
- [15] J. Subrahmonia, D. Cooper, and D.Keren. Practical reliable Bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:505–519, 1996.
- [16] Moons T., Pauwels, E.J., VanGool, L.J., and A. Oosterlinck. Foundations of semi-differential invariants. *IJCV*, 14(1):25–47, 1995.
- [17] L. VanGool, P. Kempenaers, and A. Oosterlinck. Recognition and semi-differential invariants. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, 1991.
- [18] I. Weiss. Geometric invariants and object recognition. *IJCV*, 10:3:201–231, June 1993.