Recognizing Surfaces from 3D Curves

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 properties of curves and surfaces are 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 [17, 16, 25], 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. 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.

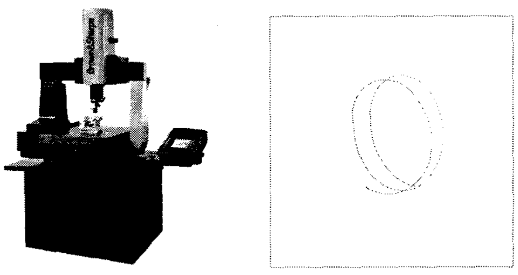

Figure 1: High-accuracy measuring device and a curve it measured on a cylinder.

In [2], 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 [14, 8]. 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

[*] 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

0-8186-8821-1/98 $10.00 © 1998 IEEE

type, determine the surface. This is the *algebraic approach* [19, 11]. 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. 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 [4, 3, 6, 24, 21, 10].

In Section 4 we present a method to solve the problem, which uses the differential properties of the curve (tangent and normal vectors and the curvature), and 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 [3, 21, 18] but not the surfaces on which they lie.

2 The Algebraic Approach

Implicit polynomials can be used to describe 2D and 3D objects. Some works which address the fitting of implicit polynomials are [22, 1, 12, 23, 20]. One can then use polynomial invariants to recognize the objects [19, 11, 7, 9]. Let us shortly describe how a sphere, cone, cylinder and torus can be recognized using such invariants. Note that the first three objects can be fitted with a quadratic, and the torus with a quartic. Suppose, then, that we succeeded to fit data with a quadratic. Write it as

$$XAX^t + (v, X) + s = 0 \qquad (1)$$

where A is a 3×3 matrix, v a vector in $\mathcal{R}^3$, and s a scalar. It is easy to verify that

- If the object is a sphere, A has three positive and identical eigenvalues. It is then trivial to extract the sphere's center and radius.

- If the object is a cylinder, A has two positive and identical eigenvalues, and one zero eigenvalue; also, the axis of the cylinder is in the direction of the eigenvector with zero eigenvalue, and it is trivial to extract its radius.

- If the object is a cone, A has two identical positive eigenvalues and one negative eigenvalue. The axis of the cone is in the direction of the eigenvector with the negative eigenvalue. It is then trivial to extract the cone's opening angle and apex.

- If the object is a torus, its general equation is

  $$\begin{aligned} E_{tor} = \quad & ((x-a)^2 + (y-b)^2 + (z-c)^2 + R^2 - r^2)^2 \\ & -4R^2((x-a)^2 + (y-b)^2 + (z-c)^2 \\ & -((x-a)n_1 + (y-b)n_2 + (z-c)n_3)^2) \end{aligned}$$

  where (a, b, c) is its center point, (n_1, n_2, n_3) a unit vector perpendicular to the plane over which the torus lies, and R (r) are the major (minor) radii.

  It's trivial to extract a, b, c from E_{tor} (for instance, differentiating E_{tor} three times by x gives $24x - 24a$). To extract r and R, note that substituting $\{x = a, y = b, z = c\}$ in E_{tor} gives $r^4 + R^4 - 2\,R^2r^2$, and substituting $\{x = a, y = b, z = c\}$ in $\frac{\partial^2 E_{tor}}{\partial x^2} + \frac{\partial^2 E_{tor}}{\partial y^2} + \frac{\partial^2 E_{tor}}{\partial z^2}$ gives $-12\,R^2 - 12\,r^2 + 8\,R^2{n_1}^2 + 8\,R^2{n_2}^2 + 8\,R^2{n_3}^2 = -4R^2 - 12r^2$. It is trivial to extract R and r from these two identities. After R, r, a, b, c have been recovered, it is trivial to recover (n_1, n_2, n_3).

2.1 Number of Points Needed

Experiments on curve data show that a relatively high number of points is necessary to achieve reliable algebraic fitting. For instance, for the cylinder data we have used (Figure 1), more than 200 points are required for a reliable fit. When 200 points were used, there was a difference of 22% between the true radius of the cylinder and the radius of the fitted cylinder. At least 300 points are necessary to obtain a reliable fit. Apparently, the fact that the points lie on a curve, which is a "one dimensional entity", results in singularities when trying to fit it with an implicit polynomial which, by its nature, is appropriate for fitting "two dimensional entities".

On the other hand, when using the differential invariants proposed here, a far smaller number of points was necessary; usually, invariants were computed using 10 points or so.

3 Mathematical Preliminaries

In the sequel, a few concepts from geometry and algebra are required. We proceed to define them and state some of their important properties.

Locally, a surface S in 3D Euclidean space is a differentiable image of an open set $\mathcal{O}$ in $\mathcal{R}^2$. Formally, it is the set of triplets $\{(x(u,v), y(u,v), z(u,v)) \,/\, (u,v) \in \mathcal{O}\}$. The *tangent plane* to S at the point $((x(u,v), y(u,v), z(u,v))$ is the plane spanned by (x_u, y_u, z_u) and (x_v, y_v, z_v). The *normal* to S at (u,v) is the unit vector pointing at the direction of

$(x_u, y_u, z_u) \times (x_v, y_v, z_v)$; it is therefore perpendicular to the tangent plane.

In the sequel, we shall use the fact that if C_1 and C_2 are curves which intersect on S, then the normal to S at their intersection point is a unit vector at the direction of the vector product of their tangent vectors. This holds unless these tangent vectors are parallel.

The intersection of S with any plane containing N is called a *normal section* of S. Note that the normal section is determined by a unit vector v in the tangent plane, which is the direction at which the plane containing N intersects the tangent plane. Thus, we may speak of a normal section at the direction v.

The curvature of a normal section is called the *normal curvature*. The maximal such curvature, k_1, and the minimal, k_2, are called the *principle curvatures* of S. Let us denote their directions by $\vec{k_1}$ and $\vec{k_2}$. It can be proved that they are orthogonal and that, if $v = \vec{k_1}\cos(\theta) + \vec{k_2}\sin(\theta)$, then the normal curvature at the direction v equals

$$k_1 \cos^2(\theta) + k_2 \sin^2(\theta) \tag{2}$$

The product $K = k_1 k_2$ is called the *Gaussian curvature*, and the mean $H = \frac{k_1+k_2}{2}$ is the *mean curvature.*

Suppose a curve C lies on the surface S. Then, if its curvature is κ_C, and the normal curvature of S at the direction of C's tangent vector is κ_S, then

$$\kappa_S = \kappa_C \cos(\theta) \tag{3}$$

where θ is the angle between N_S, the normal to S, and N_C, the normal to C.

4 Our Approach

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 θ is the angle between N_S and N_C, then $\kappa_S = \kappa_C \cos(\theta)$, where the value of κ_S (the curvature of the normal section on the surface) is determined by the principal curvatures κ_1 and κ_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.

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 θ, 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 κ_1, κ_2 are the principal curvatures for the surface at M, and β is the angle between the tangent to the curve and $\vec{\kappa_2}$, the second principal direction.

Given two curves we have two equations for the surface normal curvature, with three unknowns – κ_1, k_2, and β:

$$\begin{aligned} \kappa_{N_S 1} &= \kappa_1 \sin^2(\beta) + \kappa_2 \cos^2(\beta) \\ \kappa_{N_S 2} &= \kappa_1 \sin^2(\beta + \phi) + \kappa_2 \cos^2(\beta + \phi), \end{aligned} \tag{4}$$

where ϕ 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 κ_2 and the principal directions $\vec{\kappa_1}, \vec{\kappa_2}$. The cylinder's radius is $R = \frac{1}{\kappa_2}$, and the orientation of its axis is $\vec{\kappa_1}$. 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 θ between N_S and N_C. When we are given only one curve, θ 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 θ_1 and β_1. When they are given, the cylinder is uniquely defined. Note that $\vec{\kappa_1}$ 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 $\vec{\kappa_1}$ the axis of the cylinder as functions of θ_1 and β_1:

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

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

$$\vec{\kappa_1} = T_C \sin(\beta_1) + (T_C \times N_S)\cos(\beta_1) \qquad (5)$$

And a point on the axis is:

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

Given an additional point, its β_2 and θ_2 can be recovered as follows:

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

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 $\vec{\kappa_1}$; therefore,

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

which gives us two equations in two unknowns, which can be solved for the values of θ_1 and β_1.

These two points give the equation of the cylinder that passes through them and satisfies the given constraints. In addition, from (5)

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

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 κ_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 [15].

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 2(a). One of these results and the original data are shown in 2(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.

6 Conclusions

A novel method to recognize some surfaces, given curve(s) on them, was presented. The method recognizes the type of surface on which the curve lies by finding relationships between the differential properties of the curves and the surfaces on which they lie.

The method can use 3D curves derived from stereo and structured light; it is particularly useful when given the output of measuring devices which produce

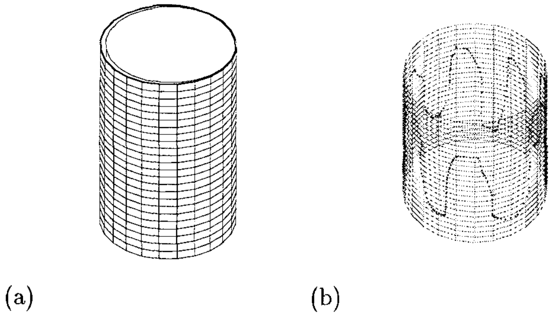

Figure 2: (a) The five recovered cylinders with the lowest LSE. (b) The recovered shape of the cylinder and the original data points.

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.

An additional method using curve invariants has been presented in [13].

References

[1] C. Bajaj, I. Ihm, and J. Warren. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Transaction on Graphics*, 12(4):327–347, 1993.

[2] 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.

[3] 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.

[4] A.M. Bruckstein, N. Katzir, M. Lindenbaum, and M. Porat. Similarity-invariant signatures for partially occluded planar shapes. *IJCV*, 7:271–285, 1992.

[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. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3D object recognition and pose. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:971–992, October 1991.

[8] D.A. Forsyth. Recognizing algebraic surfaces from their outlines. In *International Conference on Computer Vision*, pages 476–480, Berlin, May 1993.

[9] D.A Forsyth, J.L Mundy, A. Zisserman, and C.M Brown. Projectively invariant representations using implicit algebraic curves. In *European Conference on Computer Vision*, pages 427–436, June 1990.

[10]. A. Gueziec and N. Ayache. Smoothing and matching of 3D curves. *Interational Journal of Computer Vision*, 12(1):79–104, 1994.

[11] D. Keren. Using symbolic computation to find algebraic invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1143–1149, 1994.

[12] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:38–53, January 1994.

[13] D. Keren, E. Rivlin, I. Shimshoni, and I Weiss. Recognizing surfaces using curve invariants and differential properties of curves and surfaces. In *IEEE Trans. on Robotics and Automation*, pages 3375–3381, 1998.

[14] 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.

[15] J.J. Moré, B.S. Garbow, and K.E. Hillstrom. User guide for MINPACK-1. ANL-80-74, Argonne National Laboratories, 1980.

[16] J. Mundy and A. Zisserman. *Geometric Invariance in Machine Vision.* MIT Press, 1992.

[17] 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.

[18] T. Pajdla and L. VanGool. Matching of 3d curves using semi-differential invariants. In *ICCV*, pages 390–395, 1995.

[19] 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.

[20] S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3D object modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1183–1196, 1994.

[21] Moons T., Pauwels, E.J., VanGool, L.J., and A. Oosterlinck. Foundations of semi-differential invariants. *IJCV*, 14(1):25–47, 1995.

[22] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:1115—1138, November 1991.

[23] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:287–303, 1994.

[24] 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.

[25] I. Weiss. Geometric invariants and object recognition. *IJCV*, 10:3:201–231, June 1993.