Shape Recovery from Passive Locally Dense Tactile Data

Michael Erdmann
School of Computer Science and the Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
me@cs.cmu.edu

1 Introduction

As a robotic hand grasps an object of unknown shape, the object may slip in the robot's fingers. If the robot can infer the local shape as the object slips, the robot may be able to readjust its hand and fingers so as to better grasp the object.

This paper considers the problem of inferring local contact geometry from passive tactile information. The term "passive" means that the information results from motions of the object that are not necessarily under the control of the robot. To juxtapose, generally the term "active sensing" means that a robot actively explores an object, for instance with visual and/or tactile sensors that move around the object. This paper is not concerned with such active exploration. Instead, our goal is to determine how local shape geometry may be inferred from purely passive information. Shape recovery based on passive information will be useful both for dealing with unexpected events, as in the slippery grasping example above, and for more elaborate manipulation strategies, such as active exploration.

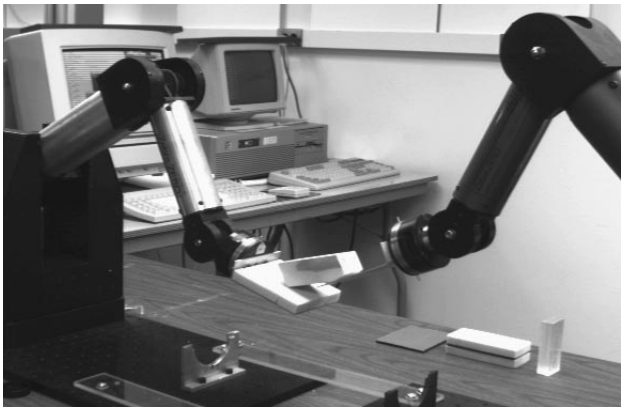

Figure 1: Two robots, with flat palms attached to each wrist, manipulate an irregular block.

1.1 Motivation and Context

This paper concentrates on planar objects manipulated by flat linear fingers, which we refer to as "palms". The motivation for this setup comes from our earlier work on nonprehensile palmar manipulation [Erdmann 1995] (see Figure 1). In that work, two robots manipulated objects by rolling and sliding the objects between linear palms. The objects' shape, mass properties, and friction were known in advance. Thus an automatic planning system could produce open-loop (sensorless) strategies for manipulating the objects from given start configurations to desired goal configurations.

Sensorless palms provide an existence proof of the basic manipulation strategies one can expect a robot to perform. Of particular interest to us previously was the ability to effectively turn friction on or off at a contact, by reorienting a palm. The ability to control friction allowed the robots

to reposition their hands via sliding without effectively releasing the object, that is, without dropping or reorienting the object.

In a natural setting one cannot expect to know the object shape or friction coefficients precisely, or even expect the coefficients of friction to be constant over time. Instead, one might endow the palms with tactile sensors and hope to infer geometric and frictional characteristics of the object from the tactile sensors. For instance, contact location sensors might be used to infer the local object geometry and nibbed sensors might be used to detect impending slip.

In this paper we will focus on acquiring local geometry from contact location sensors. Specifically, we envision a linear palm[1] endowed with a tactile sensor that can report the location of any contacts with the palm. We note in passing that this is a fairly traditional view of a tactile sensor; many more elaborate sensors have been studied and built. The main issue for us is density of information. We will assume that our sensor can report contact locations to infinite precision. In practice, even achieving sub-millimeter precision at reasonable cost is beyond the state of the art today.

1.2 Basic Results

1.2.1 Assumptions

We consider planar objects in contact with linear tactile sensors, called "palms". Our assumptions are:

1. The objects are locally convex at the contacts. This assumption ensures that contact is possible locally with the palms.
2. A given object makes contact with a given palm at no more than one location. This is a natural extension of the local convexity assumption. Strictly convex shapes satisfy this assumption.
3. The objects are locally smooth. In other words, at each contact, the object's outward normal is anti-parallel to the palm's outward normal.
4. Each sensor reports the location of contact on the sensor with infinite precision.
5. The palms maintain contact passively. By this we mean that the palms do not move in an exploratory fashion; at most they move compliantly to maintain contact with the object.

[1] In 2D, a linear palm is just a straight line segment; in 3D, it would be a flat surface.

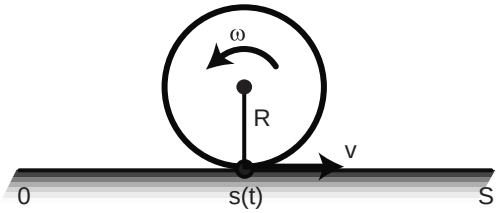

Figure 2: Relationship between local object curvature, object motion, and sensory data.

1.2.2 Results

Our basic question is:

Given a set of linear tactile sensors in passive contact with an object, can we recover the local shape of the object at the contacts?

The answer has three parts:

1. A single tactile sensor is generally unsatisfactory. We need to know too much to infer the local contact shape.
2. Two tactile sensors are satisfactory, given the angular rate at which the object is turning. In other words, if we know the angular velocity of the object, then we can recover its local geometry.
3. Three tactile sensors are generally sufficient to recover both the turning rate of the object and its local geometry. There are some non-generic exceptions, such as objects of constant local breadth.

1.2.3 Discussion of the Results

The negative result for a single sensor is not surprising. What is interesting, are the different remedies.

Figure 2 shows a circle of radius R on a flat horizontal palm. The circle maintains contact with the palm. The

location of the contact point at time t is $s(t)$; its time derivative is $\dot{s}(t)$. The instantaneous contact point on the circle translates with horizontal velocity v; the circle rotates with angular velocity ω. The relationship between these terms is:

$$\dot{s} = v - \omega R. \tag{1}$$

This equation applies to any smooth object with instantaneous radius of curvature R at the contact point. The equation is a special case of the contact equations found in [MacMillan 1936], [Cai and Roth 1986], and [Montana 1988].

The problem of local shape recovery is quite simply: Determine the radius of curvature R. If R can be determined for each point along some curve segment, then that curve segment can be fully reconstructed. This is a consequence of the Frenet formulas for planar curves (see, for instance, [O'Neill 1997]).

The problem thus is to infer R from $s(t)$, using Equation (1). Unfortunately, Equation (1) is only one equation in four variables. We can sense $s(t)$ and thus compute $\dot{s}(t)$, but the velocity terms v and ω cause difficulty.

If v and ω are known, and if ω is non-zero, then one can easily solve for R. Otherwise, there is insufficient information. One positive instance occurs when one knows that the object is rolling without slipping and one also knows the object's angular rotation rate. In that case, $v = 0$, and thus R is just $-\dot{s}/\omega$.

Much of the work on active tactile sensing is motivated by this last positive observation. Rather than allow the object to roll and slide over a stationary sensor, one instead moves the sensor around the object (usually the object is held stationary). Thus one has complete control over ω and v; in particular, one can rotate the sensor on the object without slip and thus calculate the object's radius of curvature, much as we just did.[2]

The advantage of this active approach is knowledge by control of ω and v. The disadvantage is that one may not actually be able to achieve the desired ω and v, since the shape of the object is unknown. Thus the sensor may slip, even though one may be trying to prevent slip in order to enforce $v = 0$. These issues aside, even if control over ω and v is perfect, one is still faced with the more general problem of how to infer R from $s(t)$ when the object being manipulated can move on its own.

Our approach in this paper is to add additional sensory palms. Each such palm provides a further constraint on the variables in Equation (1). For instance, when we add a second palm, the velocities ω and v can no longer be completely arbitrary. Instead, they must satisfy the constraint that contact be maintained with both palms. Of course, we do not know exactly what that constraint is, since we also do not know the shape of the object at the second palm. However, the tactile information at the second palm provides a second equation like Equation (1). Intuitively speaking, we have thus gone from one equation and three unknowns to two equations and three unknowns. We can think of these three unknowns as the radii of curvature at the two contact points and the angular velocity of the part. Thus if we somehow know the angular velocity of the part, then we can recover the local geometries at the contacts.

Finally, when we add a third, compliant, palm, we can further constrain the unknown ω. The compliance in the third palm means that we do not actually get quite as strong a constraint as we did when we added the second palm. However, if we keep track of the location of the third palm relative to the first two as it moves, then we do have essentially an identical constraint. Thus, with some exceptions, we can recover the local contact geometries.

1.2.4 Discussion of the Assumptions

As will become clear, the assumptions are not terribly important, but they simplify the development. With regard to assumption (1), one would generalize to nonconvex objects by using nonlinear sensors. With regard to assumption (2), multiple contacts with a single palm actually provide more information than does a single contact. The smoothness assumption (3) in some sense makes shape inference more difficult, as we shall see. For instance, suppose we have two contacts with two palms, that is, one contact with each palm. Suppose further that we *know* one of the contacts to consist of an object vertex and a palm. Then the local geometry of the unknown part at the second contact with the second palm can be recovered much more easily than it can when both contacts are locally smooth.

Assumption (4) has the standard drawbacks. Finally, here

[2] Usually one uses a circular tactile sensor rather than a linear one, and a more general version of Equation (1), but the underlying idea is the same.

are three comments regarding the "passivity" assumption (5):

First, our main reason for the passivity assumption is to make clear that we will only be considering information that comes directly from contact with the tactile sensors, as opposed to any knowledge of the sensors' absolute motion in space. In an actual manipulation example all palms might be moving in space, while manipulating an object. Our results still hold in that case. However, to tie things down a bit, let us elaborate the passivity assumption with the following specific requirements:

1. If the sensing apparatus consists of a single palm, then that palm does not move at all. The object is free to rotate and slide in any manner it chooses, while maintaining contact with the palm. Relative to the palm, the object has two degrees of freedom.
2. If the sensing apparatus consists of two palms, then the two palms do not move at all. Again, the object is free to rotate and slide in any manner it chooses, while maintaining contact with both palms. Relative to the two palms, the object has a single degree of freedom.
3. If the sensing apparatus consists of three of more palms, then two palms are stationary while the remaining palms move compliantly, so as to maintain contact with the object. Again, relative to the two stationary palms, the object has one degree of freedom.

Second, we could insist that the sensors always remain completely stationary, even when three or more palms are involved, but this would be pointless. With three or more contacts, generically, the object would be immobilized. We are interested in the cases in which the object does slide or slip, and hope to infer information from such motions.

Third, we tacitly assume that the object does indeed maintain contact with the sensors. In practice, there would probably be a mix of times involving contact and times in which contact is broken. We are here concerned with those, possibly small, intervals of time in which contact exists continuously.

1.3 Roadmap

We will review previous related work in the next section. Then we will begin the development by first considering sensors arranged at right angles to each other. All the basic ideas will be illustrated by this simple setup. At the end of the paper we will write out the general equations for general arrangements of sensors.

We will first consider the very special case of an object in vertex contact with one palm and smooth contact with another palm. This special case will lead us to review and extend the standard definition of the radius function of a convex object. With that new tool in hand, we will then generalize to smooth contacts at both palms, and finally to contacts with three palms.

2 Related Work

Prior work related to this paper falls roughly into six categories: contact kinematics, algorithmic shape recovery, local shape estimation, global shape and/or pose inference, active sensing, and tactile sensor technologies.

2.1 Contact Kinematics

Much of the research on shape recovery points back to the works of Cai and Roth and Montana on contact kinematics. Four relevant papers are: [Cai and Roth 1986], [Cai and Roth 1987], [Montana 1988], [Montana 1995]. These papers contain kinematic analyses for fairly arbitrary shapes in contact with each other. The papers derive relationships between the local motions of the contact points on the shapes and the motions of the shapes relative to each other in world coordinates. We cited the simplest of these in the Introduction.

The resulting kinematic equations may be used to infer unknown geometric properties from observed geometric properties. For instance, [Montana 1988] proposes a strategy for determining the local curvature of a shape by repeatedly rolling a sensor, such as a cylindrical or spherical tactile sensor, over the shape. Much of the subsequent work on curvature estimation has taken this idea as its starting point.

We feel that the basic results of the current paper are implicit in the derivations of [Cai and Roth 1986], even if the tools are different. Towards the end of that paper, Cai and Roth consider the problem of manipulating a known object with two fingers. In particular, they propose a numerical

method for computing the pose of an object, along with its angular velocity and angular acceleration, from the contact information sensed at the fingers.

In order to see the high-level connection of that method with our work in this paper, let us oversimplify a little. We may think of the contact kinematics as specifying a relationship

$$F(R_1, R_2, \omega) = 0,$$

where R_1 and R_2 are the radii of curvature of the object at the two contact locations, one with each finger, and ω is the instantaneous angular velocity of the object. We are purposefully suppressing some variables in this oversimplified description, such as the location of the fingers, the motions of the fingers, the location of the contact points on the fingers, and time.

In [Cai and Roth 1986], the object shape is known. Moreover, the pose of the object may be obtained from tactile sensory information, so we may think of R_1 and R_2 as known. Thus, effectively, Cai and Roth use the relationship F and tactile information at the fingers to determine ω from known R_1 and R_2. In other words, they perform the inference:

$$(R_1, R_2) \stackrel{F}{\Longrightarrow} \omega. \tag{2}$$

In principle it should also be possible to use F and tactile information for the opposite inference:

$$\omega \stackrel{F}{\Longrightarrow} (R_1, R_2). \tag{3}$$

In other words, it should be possible to recover the shape of an unknown object from its angular velocity and passive tactile information, using the kinematic relationship F.

Performing this inference is the main topic of the current paper, although our relationship F will look different from that presented in [Cai and Roth 1986]. Given ω, we will present a method for determining the local curvature at the contact points, based on passive tactile information obtained from contact with two palms. Finally, we will add a third palm, so as to infer ω as well.

One final comment. It may seem curious that inference (2) maps from 2D to 1D while inference (3) maps from 1D to 2D. The explanation of this paradox lies in the information we have suppressed. Cai and Roth only need tactile information from one of the two contact points in order to infer ω from R_1 and R_2, whereas we will require tactile information from both contact points to infer R_1 and R_2 from ω.

2.2 Algorithmic Shape Recovery

Closely related to the techniques in this paper is the paper by [Rao and Goldberg 1994], which assesses the information contained in the *diameter function* of an object. [Sinden 1985] discusses related information provided by chords inscribed in an object. The diameter function of a planar object is the distance between the jaws of a parallel-jaw gripper grasping the object, expressed as a function of the relative orientation between the gripper and the object.

The diameter function $D(\theta)$ of an object is closely related its *radius function* $r(\theta)$. In order to define the radius function, let us first define some auxiliary terms. By a *supporting line* we mean a line touching the object, with the object wholly on one side or other of the line. By an *oriented supporting line at angle* θ, denoted $\ell(\theta)$, we mean a supporting line with an *exterior* side and an *interior* side. The normal to this line is given by the direction vector $(\cos\theta, \sin\theta)$; this normal vector points towards the exterior side of the line; the object of interest must lie wholly on the other side, that is, on the interior side of the line or on the line itself.

The radius function measures the distance from some fixed interior point of an object to its oriented supporting lines. Specifically, $r(\theta)$ is the distance from this fixed point to the oriented supporting line $\ell(\theta)$. Note that $D(\theta) = r(\theta) + r(\theta + \pi)$ regardless of the choice of fixed interior point.

It is well-known that the shape of a convex object with piecewise smooth boundary is uniquely determined by its radius function, computed relative to any fixed interior point. See also [Santaló 1976]. A natural question to ask is whether the diameter function of an object contains as much information. For general shapes, the answer is no, since there are objects of constant diameter other than circles [Reuleuax 1876]. [Rao and Goldberg 1994] observe that even when the class of objects is restricted to be polygonal, infinitely many objects can share the same diameter function. They then consider the problem

of distinguishing between a known finite set of polygonal objects based on sparse diameter information. In a similar spirit, [Akella 1996] uses the radius function to distinguish between a finite set of polygonal shapes. In both cases, the objective is to find a discriminating tree of minimal height. A node in this tree represents a single sampling of the diameter or radius function. These techniques are useful in pose estimation during parts orienting.

In this paper we will be using a generalization of the radius function (see Sections 3.1 and 4.1). Since we do not know the object shape or its motion *a priori*, we cannot choose a fixed interior point. Thus we cannot compute the radius function. However, we do have access to the vector between the two contact points, and from this we will build a generalized two-dimensional support function. This will serve us as well as would a radius function, permitting us to reconstruct the object shape. This two-dimensional support function is a generalization of the radius function in the following sense: Suppose that one of the contacts involves a vertex of the object rather than a smooth curve. Then the (yet to be defined) 2D generalized support function will contain precisely the same information as would a radius function constructed by using that vertex as the fixed interior point.

Additional related algorithmic methods include the work on shape recovery from probing and projection operations. [Cole and Yap 1987] investigate the minimum number of *finger probes* required to reconstruct the shape of an unknown convex polygon. A finger probe entails choosing a ray in space, then moving along this ray, and recording the location at which the ray encounters the unknown object. [Boissonnat and Yvinec 1992] extend this basic finger probe to return not only the contact location but also the contact normal; they give algorithms for reconstructing a class of nonconvex polygons and polyhedra from such extended finger probes.

Another type of probe is the *line probe*. A line probe consists of choosing a line normal, then sweeping the corresponding line parallel to its normal, and recording the location of the line at which it touches the unknown object. In terms of our previous notation, a line probe entails choosing an angle θ, then determining the oriented supporting line $\ell(\theta)$ for the object. Generally, the sensor returns the location of the line in space, but not the location of the contact point of the object on the line. For a survey of algorithmic probing problems and results see [Skiena 1989].

Both finger and line probes constitute types of active tactile sensing. In each case, a robot actively moves a finger or palm around an object, touching it at various locations.

[Lindenbaum and Bruckstein 1992] use parallel line probes and parallel finger probes to reconstruct convex polygons; they derive upper and lower bounds on the required number of such parallel probes. Their line probes are generalized line probes, i.e., lines that rotate around a specific point until they make contact with the unknown object. [Li 1988] considers line probes and projection probes to reconstruct the shape of unknown convex polygons. A *projection probe* is simply a pair of line probes that approach the unknown object from opposite directions; effectively the probe "projects" the object onto the common perpendicular of the two line probes. [Lindenbaum and Bruckstein 1994] give algorithms for the approximate reconstruction of arbitrary convex planar shapes using line probes. That work is interesting in particular because it moves away from the restricted class of polygonal shapes. Finally, [Kölzow, Kuba, and Volčič 1989] reconstruct convex shapes from sets of cross-sections.

2.3 Local Shape Estimation

As we indicated earlier, the rigid body contact kinematics and applications in [Montana 1988] inspired many researches to investigate methods for estimating local curvature (and thus local shape). We will mention a few here. [Fearing and Binford 1988] use a cylindrical tactile sensor to determine principal curvatures, normal force, and contact location from strain measurements, i.e., from a theory of elasticity. [Chen, Rink, and Zhang 1996] describe an unknown surface locally in terms of a general quadratic form, then fit experimental tactile information to this form using active exploration and least-squares. [Charlebois, Gupta, and Payandeh 1996, 1997] describe two different methods for curvature estimation. The first entails rolling a sensor over the unknown surface, *à la* Montana; the paper analyzes the sensitivity of this strategy to sensor noise. The second entails dragging a set of fingers over the surface, then fitting the resulting contact data to a B-spline description of the surface patch. Any given patch is used to guide the exploration further. Along a different vein, [Kaneko, Higashimori, and

Tsuji 1997] describe a strategy for estimating local surface shape for nonconvex objects, by pulling probes out of concavities.

One issue in curvature estimation is sensitivity. There is reason to be cautious. [Charlebois, Gupta, and Payandeh 1996] investigate the sensitivity of curvature estimates based on rolling to the accuracy with which the turning velocity is known, the distance rolled, and the tactile sensor resolution. One insidious problem is the interaction between the sensor resolution and the distance rolled. Rolling a larger distance will overcome low sensor resolution, but will lead to increased inaccuracies in the curvature estimate for objects of non-constant curvature. [Nicolson and Fearing 1995] examine the sensitivity of curvature estimates based on elastic sensor models. [Zhang, Maekawa, and Tanie 1996] perform a sensitivity analysis and observe that rolling motions may be inadequate for determining curvature when the object's curvature is much larger or smaller than that of the sensor. The discussion at the end of that paper, comparing elastic models to rigid body models, is interesting as well.

2.4 Global Shape and/or Pose Inference

We have already indicated that techniques used for unknown shape recovery may also be used to distinguish between different shapes taken from some restricted class of possible shapes. Merely restricting the class of possible shapes to polygons or polyhedra is often a profitable strategy. In industrial settings, the class may be much smaller yet. For instance, in parts orienting, one may simply need to distinguish between the different possible poses of a single part. In some cases, this set of possible poses may even be finite. Examples include the stable resting configurations of a polygonal part under gravity or in the grasp of a frictionless parallel jaw gripper. We have already pointed to [Rao and Goldberg 1994] and [Akella 1996] in this context. Related work includes the grasping strategies of [Goldberg, Mason, and Erdmann 1991] and [Goldberg 1993].

The work on pose and model recognition from tactile information has its origins in [Gaston and Lozano-Pérez 1984] and [Grimson and Lozano-Pérez 1984]. These papers build interpretation trees that use sparse tactile data to distinguish between different models and poses of 3D objects known to lie in some class of possible models. [Fearing 1990] discusses shape recognition of a certain class of generalized cylinders from sparse tactile information. [Caselli, Magnanini, Zanichelli, and Caraffi 1996] and [Beccari, Caselli, and Zanichelli 1997] describe techniques for recognizing convex polyhedra from sparse tactile data, using volumetric representations and active tactile exploration.

Another type of planar pose recognition strategy entails sampling a finite set of locations in the plane to distinguish between different possible objects and poses. [Jia and Erdmann 1996a] and [Romanik and Salzberg 1992] discuss this problem from an algorithmic perspective, examine its complexity, and present optimal strategies and simulations. [Jia and Erdmann 1996a] also consider the problem of distinguishing object poses by inscribing the objects in cones. A cone may be thought of as a pair of generalized line probes, each rotating around the same point. The paper shows experimentally that generally two cone inscriptions are sufficient to determine an object's pose.

The pose recognition strategies discussed above have two characteristics: (i) sensing is sparse and (ii) the object is fixed in space. [Jia and Erdmann 1996b] consider a problem on the other end of the spectrum. Specifically, the hand moves and purposefully disturbs the object. As a result, the contact point on the hand traces out some curve as a function of time. An assumed high-resolution tactile sensor embedded in the hand records this curve. The robot then employs techniques from nonlinear observer theory to recover the object pose from the tactile information and knowledge of the hand/object dynamics.

2.5 Active Sensing

There has been considerable work on active sensing. We will only mention a few papers here. A seminal paper is [Allen and Michelman 1990]. The paper discusses an implemented system, consisting of a Utah-MIT hand mounted on a PUMA arm, with Interlink tactile sensors attached to the fingertips. The system performs exploratory operations such as enveloping grasps to recover shape, fitting the resultant tactile data to a superquadric description of the unknown shape. The paper also presents several other exploratory procedures, such as contour following. [Bay 1989] proposes

an active exploration method, in which tactile data from multiple fingers is used to estimate a surface equation in a Kalman-filtering style. Exploration consists of rotating the hand, while using active force compliance to maintain contact between the fingertips and the unknown surface.

A number of researchers have considered the problem of planning exploratory strategies for obtaining useful tactile information. Both [Roberts 1990] and [Ellis 1992] present strategies that take account of the geometric constraints implied by the tactile information acquired thus far. Their methods determine the "best next" sensory operation based on these geometric constraints. Ellis considers sensory operations analogous to finger probes, while Roberts considers surface tracing moves. In recent work [Okamura, Turner, and Cutkosky 1997] describe a system with two types of fingers: (i) fingers that manipulate the object, and (ii) fingers with embedded tactile sensors that roll and slide over the object surface. Exploration is divided into phases. During each phase some subset of the fingers manipulate the object, while another subset traces over the object's surfaces. These second fingers record sensory information as they move, and they move to locations from which they can profitably act as the manipulating fingers during the next phase. The actual motions during a phase may be determined by the information gained in the previous phase.

2.6 Tactile Sensor Technologies

We point to [Howe and Cutkosky 1992] for an excellent survey of tactile sensor technologies. Some interesting papers that have appeared since that article include [Bicchi, Salisbury, and Brock 1993] on estimating contact locations from force measurements, [Howe and Cutkosky 1993] on sensing fine surface features with dynamic tactile sensors, and [Tremblay and Cutkosky 1993] on sensing incipient slip and estimating friction.

3 Two Orthogonal Sensors and a Known Vertex

First we will consider the problem depicted in Figure 3. This figure shows a convex planar object of unknown shape in contact with two linear tactile sensors. The sensors are arranged at right angles to each other, parallel to the vertical and horizontal. The sensor lines intersect at some point O. The object makes single point contact with each sensor. The vertical sensor reports back a value s_1, the horizontal sensor reports back a value s_2. These values represent the distances of the two contact points from the point O.

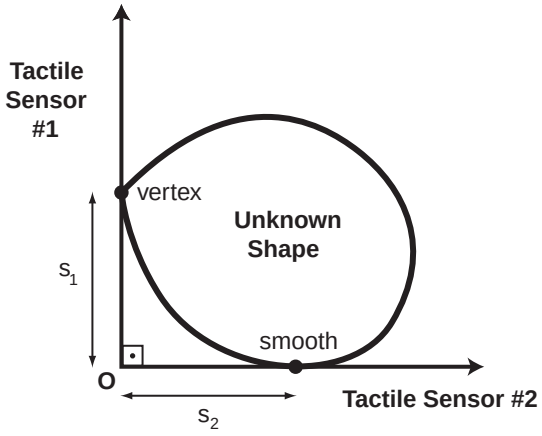

Figure 3: Vertex contact with one tactile sensor; smooth contact with another.

We will assume that the contact between the object and the vertical sensor entails contact of an object vertex with the sensor, while the contact with the horizontal sensor entails contact of a smooth portion of the object boundary with the sensor. Our task is to determine the shape of the object near the smooth contact, as the object rotates.

We note in passing that the vertex contact with the vertical sensor violates our assumption that the object be locally smooth at all the contacts. However, in order to simplify the development, this is a convenient and useful violation. Knowing that the contact is a vertex provides a lot of information, and will help us develop the basic tools. Thereafter we will treat the case of two smooth contacts, and then three smooth contacts. Finally, it will be clear that our results should apply with small modifications to any convex planar object whose boundary is piecewise smooth.

3.1 Radius and Support Functions

A convenient tool for analyzing the previous contact case will be the radius function. Let us review the basics of radius

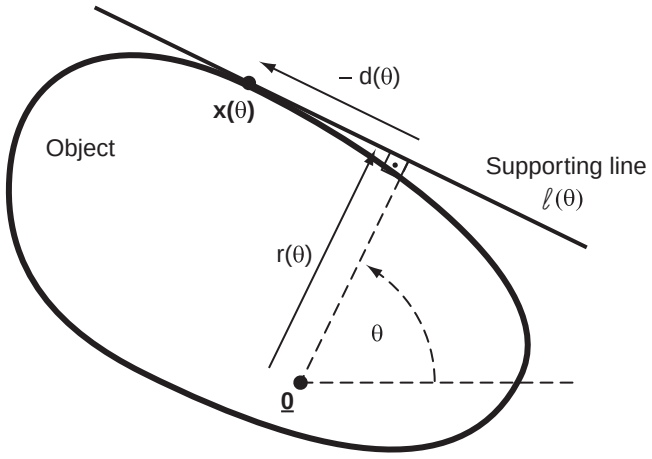

Figure 4: The radius function $r(\theta)$ measures the distance of an object's supporting lines from a given interior point.

functions. Much of this material is taken from [Santaló 1976], pp. 2–3. Figure 4 shows the basic picture for a strictly convex planar object. We assume that the object has a smooth boundary. We also assume, for convenience, that in body coordinates the origin $\underline{0}$ is interior to the object (or possibly on its boundary), and we will construct the object's radius function relative to this origin as fixed point.

For every angle θ we can construct the oriented supporting line $\ell(\theta)$ as defined in Section 2.2. The line $\ell(\theta)$ makes tangential contact with the shape at some point, which we will denote $\mathbf{x}(\theta)$. We also write $\mathbf{n}(\theta)$ for the outward normal to $\ell(\theta)$ and $\mathbf{t}(\theta)$ for the tangent to $\ell(\theta)$, chosen so that $[\mathbf{t}, \mathbf{n}]$ constitutes a right-handed frame. Observe that $\mathbf{n}(\theta) = (\cos\theta, \sin\theta)$ and $\mathbf{t}(\theta) = (\sin\theta, -\cos\theta)$. Observe also the following:

- The object's outward normal at the point $\mathbf{x}(\theta)$ is $\mathbf{n}(\theta)$, by the smoothness assumption. In other words, $\mathbf{x}(\theta)$ parameterizes the boundary of the shape as a function of normal angle θ, with θ varying over $[0, 2\pi]$.

- As θ increases, the point $\mathbf{x}(\theta)$ moves in the direction $-\mathbf{t}(\theta)$ with some non-negative velocity $v(\theta)$.

- The parameterization velocity $v(\theta) = ||\mathbf{x}'(\theta)||$ is the radius of curvature of the shape boundary at the point $\mathbf{x}(\theta)$. (This follows, for instance, from the Frenet formulas [O'Neill 1997]).

We now define the number $r(\theta)$ to be the distance of the line $\ell(\theta)$ from the origin $\underline{0}$, as shown in Figure 4. Thus

$$r(\theta) = \mathbf{x}(\theta) \cdot \mathbf{n}(\theta).$$

Although not usually part of the radius function proper, it will be useful for our shape recovery strategies to also define $d(\theta)$ as the signed distance of the contact point $\mathbf{x}(\theta)$ from the foot of the supporting line $\ell(\theta)$. Specifically,

$$d(\theta) = \mathbf{x}(\theta) \cdot \mathbf{t}(\theta).$$

Some terminology: $r(\theta)$ is variously called a *radius function* or *support function* in the literature. We will refer to $r(\theta)$ as a *radius function*, and to the pair $(r(\theta), d(\theta))$ as a *contact support function*.

Given a radius function $r(\theta)$ over some domain $[\theta_s, \theta_f]$, one recovers the object shape between $\mathbf{x}(\theta_s)$ and $\mathbf{x}(\theta_f)$ as follows. For each θ in the open interval (θ_s, θ_f), $r(\theta)$ defines a supporting line $\ell(\theta)$. The envelope of this collection of lines is the desired shape. If we write the implicit equation for $\ell(\theta)$ as $L(x, y, \theta) = 0$, then the required envelope is simply the simultaneous solution to the two equations

$$L(x, y, \theta) = 0, \tag{4}$$

$$\frac{\partial}{\partial\theta} L(x, y, \theta) = 0, \tag{5}$$

written as $\mathbf{x}(\theta) = (x(\theta), y(\theta))$, with $\theta \in (\theta_s, \theta_f)$.

Specifically, substituting the line equation determined by $r(\theta)$ for $L(x, y, \theta)$, we find that Equations (4) and (5) become:

$$x\cos\theta + y\sin\theta - r(\theta) = 0, \tag{6}$$

$$-x\sin\theta + y\cos\theta - r'(\theta) = 0. \tag{7}$$

Solving for x and y in terms of θ, we determine the curve parameterization as

$$x(\theta) = r(\theta)\cos\theta - r'(\theta)\sin\theta,$$

$$y(\theta) = r(\theta)\sin\theta + r'(\theta)\cos\theta.$$

Some observations:

- The solution $\mathbf{x}(\theta)$ is well-defined. In other words, given any smooth radius function $r(\theta)$, the system (4)–(5) has a unique solution. Thus each smooth radius function defines a unique smooth shape.

- From the above equations it is easy to see that $d(\theta) = -r'(\theta)$. In other words, $r(\theta)$ already contains information about the distance function $d(\theta)$. That is why many people refer to $r(\theta)$ as a support function, without direct reference to $d(\theta)$. However, in our case our tactile sensors will be measuring $r(\theta)$ and $d(\theta)$, or functions closely related to r and d, independently. We will then use the fact that $d(\theta) = -r'(\theta)$ to determine the angular velocity at which the object is rotating.

- It is also straightforward to see that

$$\mathbf{x}'(\theta) = -\left[r(\theta) + r''(\theta)\right]\, \mathbf{t}(\theta).$$

  Thus by our earlier comments, $r + r''$ is both the velocity of the point $\mathbf{x}(\theta)$ as it moves along the object boundary with θ-rate 1 as well as the radius of curvature of the boundary at the point $\mathbf{x}(\theta)$. This establishes the well-known fact that a smooth 2π-periodic function $r(\theta)$ is the radius function of a smooth strictly convex shape if and only if $r + r''$ is a strictly positive function.

- These results extend easily to piecewise smooth shapes. Suppose an object has a vertex $\mathbf{v}$. Suppose the object boundary incident on $\mathbf{v}$ from the clockwise side has limiting outward normal angle θ_1 and that from the counterclockwise side has limiting outward normal angle θ_2. The second derivative of r will be discontinuous at θ_1 and θ_2, but over the open interval (θ_1, θ_2), $r(\theta) + r''(\theta) = 0$, and $\mathbf{x}(\theta) = \mathbf{v}$. This shows how to extend the results above to piecewise smooth, strictly convex, shapes.

- The results further extend to shapes that are not strictly convex, that is, shapes which contain straight edges, as do polygons. In that case r' will contain jump discontinuities. These are mirrored precisely in the jumps of $d(\theta)$, as the point $\mathbf{x}(\theta)$ instantaneously moves from one end of an edge to the other without the outward normal changing angle θ. (See Exercise 3 on page 11 of [Santaló 1976]).

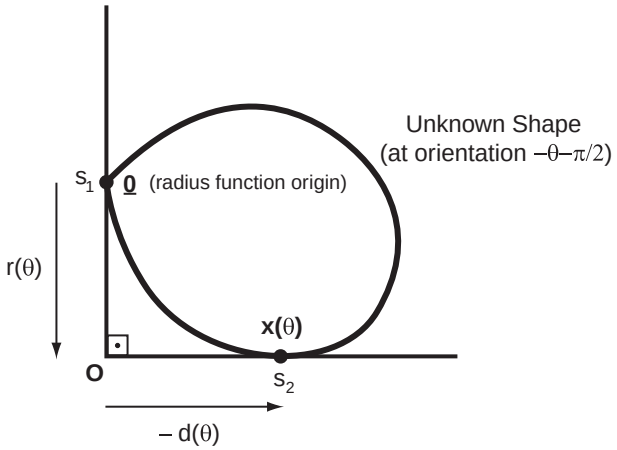

Figure 5: A vertex contact serves as the origin of a radius function.

- Finally, in principle we could extend these results to nonconvex shapes, by permitting more complicated radius functions for which $r+r''$ is sometimes negative.

3.2 Shape Recovery

The connection between Figures 3 and 4 is given by Figure 5. In the definition of the radius function proper, one moves the supporting line $\ell(\theta)$ around the object, which itself remains stationary. In contrast, for shape recovery from passive tactile sensing, the object rotates of its own volition while the sensors remain stationary. Aside from this difference in relative motion, the sensors constitute supporting lines just like the supporting lines in the definition of the radius function. Indeed, we will use the horizontal sensor as the equivalent of the supporting line $\ell(\theta)$ in Section 3.1. By observing the motion of the contact points on both sensors, we will reconstruct the contact support function $(r(\theta), d(\theta))$.

There are two issues. First, we must pick some known point on the object, that remains known to us as the object rotates, relative to which we can define the contact support function $(r(\theta), d(\theta))$. Since the object shape is unknown, the only such point is the vertex contact between the object and the vertical sensor. As shown in Figure 5 we therefore place the object origin $\underline{0}$ at this vertex contact. We can then

denote the contact between the object and the horizontal sensor by $\mathbf{x}(\theta)$. The curve $\mathbf{x}(\theta)$ defines the object boundary just as it did in the definition of the radius function in Section 3.1. We can think of the orientation of the object relative to the sensors as parameterized by $-\theta$. In other words, as the object rotates clockwise, θ increases, which is to say, the contact $\mathbf{x}(\theta)$ between the object and the horizontal sensor moves counterclockwise around the object boundary. Given these conventions, we see that the sensor value s_1 is the same as $r(\theta)$, while the sensor value s_2 is the same as $-d(\theta)$.

Second, there remains one detail to be resolved. We need to compute the angular velocity of the object in order to infer the contact support function $(r(\theta), d(\theta))$ from the sensor values $(s_1, -s_2)$. After all, s_1 and s_2 are really functions of time, not of object orientation.

3.3 Misgauging the Rotation Rate

This section constitutes a quick examination of the effect that misgauging an object's rotation rate has on the shape recovery process. For instance, if we were only to look at sensor value s_1 and make the direct inference $s_1 \Rightarrow r(\theta)$, then we would very likely misjudge the rotation rate of the object.

We consider here a very simple object, namely a square with edge length 2 centered at the origin. We focus only on the first quadrant of this square and its associated radius function. That radius function, computed relative to the center of the square, is

$$r(\theta) = \sqrt{2}\cos\left(\theta - \frac{\pi}{4}\right), \quad \text{with} \quad 0 \le \theta \le \frac{\pi}{2}. \quad (8)$$

Observe that r is smooth over the open θ-interval $(0, \pi/2)$ and that $r + r''$ is identically zero over this interval. Indeed, the solution $\mathbf{x}(\theta)$ determined by Equations (6) and (7) is just the constant vertex $(1, 1)$, as one would hope. Figure 6 shows the appearance of this vertex explicitly as the envelope of the supporting lines determined by the radius function (8).

In terms of our sensors from Section 3.2, $r(\theta)$ is the same radius function we should see if the contacts with both sensors happen to be vertices separated by distance $\sqrt{2}$. The center of the square above plays the role of the vertex that is in contact with the vertical sensor, while the square's vertex $(1, 1)$ plays the role of the vertex that is in contact with the horizontal sensor.

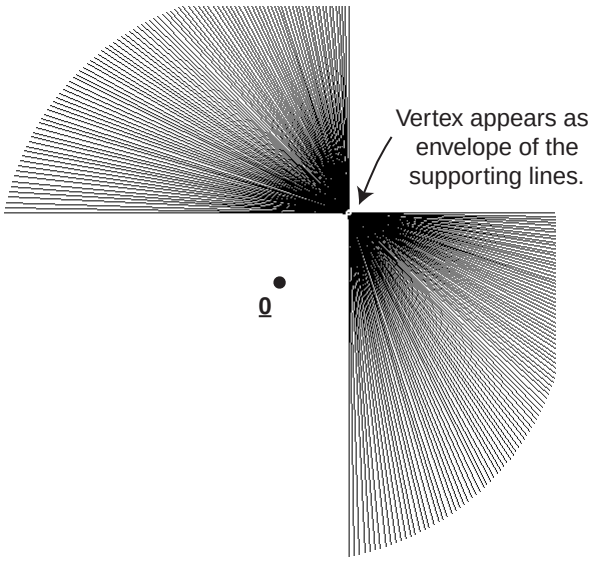

Figure 6: The first quadrant of a square appears as the envelope of its supporting lines.

As for the information required for shape recovery, we would need to know that the contact with the vertical sensor is a vertex, in order to apply the shape recovery techniques of this larger section (Section 3). We would *not* need to know that the contact with the horizontal sensor is a vertex. That fact would establish itself naturally from our (yet to be defined) calculations. In order for these calculations to be successful, however, we would need to recover the proper rotation rate of the object, as we shall now see.

Figure 7 shows what happens when the object rotates more slowly than we think. Figure 8 shows what happens when the object rotates more quickly than we think.

In Figure 7 the object is rotating at one third the rate that we think. Thus we are mistakenly working with the radius function $r_{\text{slow}}(\theta) = \sqrt{2}\cos(\frac{\theta}{3} - \frac{\pi}{4})$. Notice how nice and smooth the object has become. Indeed, in the limit, if the object is stationary but we think it is moving at constant angular velocity, we would wind up mistakenly recovering an arc of a circle. The circle's radius would simply be the value of the radius function for the given configuration of

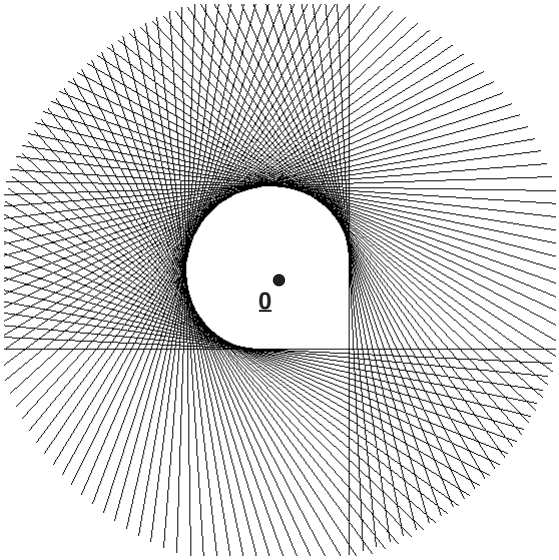

Figure 7: If the sensors misinterpret the object's rotation rate as faster than correct, the envelope of the supporting lines tends toward a circular shape.

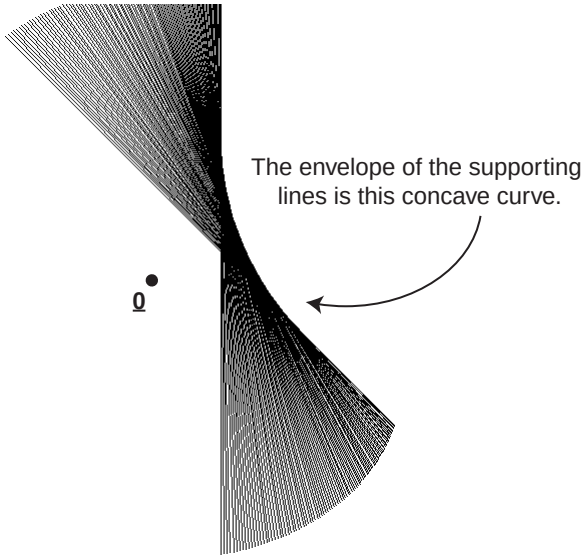

Figure 8: If the sensors misinterpret the object's rotation rate as slower than correct, the envelope of the supporting lines tends toward a nonconvex shape.

the unmoving object.

In Figure 8 the object is rotating at twice the rate that we think. Thus we are mistakenly working with the radius function $r_{\text{fast}}(\theta) = \sqrt{2}\cos(2\theta - \frac{\pi}{4})$. The object has now, mistakenly, become nonconvex. The envelope of the supporting lines defined by r_{fast} is the smooth upward bending curve in the upper right of the figure, *with the object lying towards the lower left*. In other words, the recovered shape is nonconvex.

In passing, we note that this last observation suggests an approach for handling nonconvex shapes. The radius function r_{fast} is a perfectly good radius function of a nonconvex shape. It just happens to be different from the correct radius function of the square.

Our task in the next subsection is to avoid these mistaken shapes, by properly recovering the rotation rate of the object.

3.4 Determining the Rate of Rotation

We use ' $\dot{}$ ' to represent differentiation with respect to time t and ' $'$ ' to represent differentiation with respect to angle θ. We denote the rotation rate of the object by $-\omega$. In other words, $\omega(t) = \dot{\theta}(t)$. (The negative sign appears because clockwise rotation of the object corresponds to increasing θ counterclockwise in the boundary characterization $\mathbf{x}(\theta)$.)

Recall that $s_1(t) = r(\theta(t))$ and $s_2(t) = -d(\theta(t))$. Also, recall that $d(\theta) = -r'(\theta)$. Therefore we see that:

$$\begin{aligned} \dot{s}_1(t) &= \omega(t)\, r'(\theta(t)) \\ &= -\omega(t)\, d(\theta(t)) \\ &= \omega(t)\, s_2(t). \end{aligned} \tag{9}$$

This last equation allows us to compute $\omega(t)$ at any instant in time from our sensor readings s_1 and s_2. So long as ω is never zero this means that we can solve, perhaps by numerical integration, for time as a function of angle, i.e., $t = t(\theta)$. In turn, that means we can compute the true radius function $r(\theta)$ by the rule $r(\theta) = s_1(t(\theta))$. From r we can then recover the true object shape using the envelope technique discussed in Section 3.1.

Some comments regarding details:

- Equation (9) is only useful to us if $s_2(t)$ is non-zero. $s_2 = 0$ means that the contact point with the horizontal

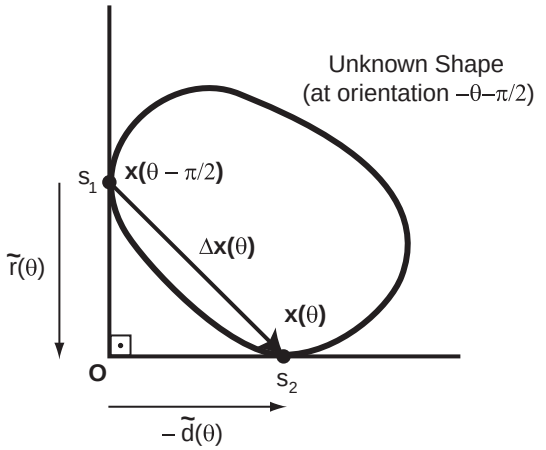

Figure 9: Smooth contacts with two tactile sensors.

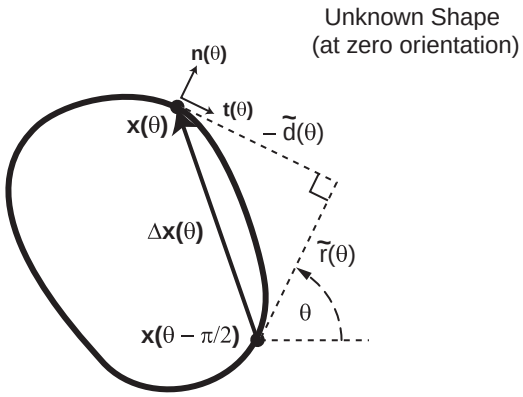

Figure 10: The generalized support function expresses the vector between the two contact points in local coordinates.

palm is at the intersection O of the two sensors. For strictly convex objects this cannot occur, unless both contact points coincide. For general shapes, the condition $s_2(t) \neq 0$ should be kept in mind.

- If $\dot{s}_1(t) = 0$ at some point in time then either $\omega(t) = 0$ or $s_2(t) = 0$. If indeed ω is ever zero, then shape recovery should be broken up into distinct intervals of time, over any one of which $\omega(t)$ is non-zero.

- For the special case in which the object is rotating at some constant unknown rate $-\omega$, Equation (9) can be used repeatedly at several points in time to estimate ω in a least squares sense. There exits a straightforward summation formula that combines this least-squares estimate with numerical differentiation to yield ω. Once ω has been found, the radius function r is simply $r(\theta) = s_1((\theta - \theta_0)/\omega)$, where θ_0 is the object orientation at time zero. There is of course no harm in arbitrarily setting θ_0 to be zero.

4 Two Orthogonal Sensors and Smooth Contacts

In this section we will generalize the results of Section 3. Figure 9 shows the basic setup. We will continue to assume that we have two orthogonal sensors each in single-point contact with a strictly convex object of unknown shape. We will now assume that both contacts are smooth. As a result we can no longer pick a fixed point on the object in a simple way, as we did in Section 3, to define a radius function $r(\theta)$. Instead, we will be working with a generalization of the full contact support function $(r(\theta), d(\theta))$.

In this section we will assume that the rotation rate of the object is known, and thus assume it to be unity. In the next section we will show how to relax this assumption by adding a third sensor.

As before, we will designate the contact between the object and the horizontal sensor by $\mathbf{x}(\theta)$. As before, the curve $\mathbf{x}$ is a parameterization of the object boundary by normal angle. We will also designate the contact between the object and the vertical sensor by $\mathbf{x}(\theta - \pi/2)$. This is a consistent designation, since the contacts are assumed smooth, and since the sensors are oriented at right angles.

Relative to body coordinates, Figure 9 becomes Figure 10. This figure is the generalized analogue of Figure 4 from Section 3, and suggests how we will proceed. In particular, since we have no natural fixed point $\underline{0}$ relative to which we can construct a radius function, we will use a varying point, namely $\mathbf{x}(\theta - \pi/2)$. This gives rise to the vector $\Delta\mathbf{x}(\theta)$ pointing from $\mathbf{x}(\theta - \pi/2)$ to $\mathbf{x}(\theta)$. We will look at the

projections of this vector onto the contact normal $\mathbf{n}(\theta)$ and tangent $\mathbf{t}(\theta)$. This is precisely the information delivered by the two tactile sensors. The question is whether this information is enough to recover the local shape of the object at the contact point $\mathbf{x}(\theta)$.

4.1 Generalized Support Function

With this motivation in mind, we now define for every smooth strictly convex planar shape a *generalized support function* $(\tilde{r}(\theta), \tilde{d}(\theta))$ by the rules:

$$\tilde{r}(\theta) = \big[\mathbf{x}(\theta) - \mathbf{x}(\theta - \pi/2)\big] \cdot \mathbf{n}(\theta), \quad (10)$$

$$\tilde{d}(\theta) = \big[\mathbf{x}(\theta) - \mathbf{x}(\theta - \pi/2)\big] \cdot \mathbf{t}(\theta), \quad (11)$$

where $\mathbf{n}(\theta) = (\cos\theta, \sin\theta)$ and $\mathbf{t}(\theta) = (\sin\theta, -\cos\theta)$.

We will also find it useful to look at the velocity of the $\mathbf{x}(\theta)$ parameterization of the boundary curve. Specifically, let us define

$$v(\theta) = \mathbf{x}'(\theta) \cdot (-\mathbf{t}(\theta)).$$

Then $v(\theta)$ is the velocity of the point $\mathbf{x}(\theta)$ as it moves counterclockwise around the boundary curve of our unknown shape, i.e., as θ increases. Observe also that $v(\theta)$ is the radius of curvature of the boundary curve at the point $\mathbf{x}(\theta)$, since the curve is parameterized by normal angle.

As we saw in Section 3, if $\mathbf{x}(\theta - \pi/2)$ happens to be a vertex rather than a smooth contact, then $(\tilde{r}, \tilde{d})$ is just (r, d). In the earlier case, we were fortunate to observe the homogeneous relationship $r'(\theta) + d(\theta) = 0$. For generalized support functions a non-homogeneous term, namely the object's radius of curvature, appears on the right hand side of this equation. This term will allow us to recover the object shape. Specifically, let us compute $\tilde{r}'(\theta)$ and $\tilde{d}'(\theta)$ and see what we get.

$$\begin{aligned}
\tilde{r}'(\theta) &= \frac{d}{d\theta}\Big\{\big[\mathbf{x}(\theta) - \mathbf{x}(\theta - \pi/2)\big] \cdot \mathbf{n}(\theta)\Big\} \\
&= \big[\mathbf{x}'(\theta) - \mathbf{x}'(\theta - \pi/2)\big] \cdot \mathbf{n}(\theta) \\
&\qquad - \big[\mathbf{x}(\theta) - \mathbf{x}(\theta - \pi/2)\big] \cdot \mathbf{t}(\theta) \\
&= \mathbf{x}'(\theta) \cdot \mathbf{n}(\theta) - \mathbf{x}'(\theta - \pi/2) \cdot \mathbf{n}(\theta) - \tilde{d}(\theta) \\
&= 0 - v(\theta - \pi/2) - \tilde{d}(\theta),
\end{aligned}$$

where the last line follows from standard curve geometry, specifically because $\mathbf{x}'(\phi)$ is parallel to $-\mathbf{t}(\phi)$ for any angle ϕ. Similarly,

$$\begin{aligned}
\tilde{d}'(\theta) &= \frac{d}{d\theta}\Big\{\big[\mathbf{x}(\theta) - \mathbf{x}(\theta - \pi/2)\big] \cdot \mathbf{t}(\theta)\Big\} \\
&= \big[\mathbf{x}'(\theta) - \mathbf{x}'(\theta - \pi/2)\big] \cdot \mathbf{t}(\theta) \\
&\qquad + \big[\mathbf{x}(\theta) - \mathbf{x}(\theta - \pi/2)\big] \cdot \mathbf{n}(\theta) \\
&= \mathbf{x}'(\theta) \cdot \mathbf{t}(\theta) - \mathbf{x}'(\theta - \pi/2) \cdot \mathbf{t}(\theta) + \tilde{r}(\theta) \\
&= -v(\theta) - 0 + \tilde{r}(\theta).
\end{aligned}$$

In short, we find that

$$\tilde{r}'(\theta) + \tilde{d}(\theta) = -v(\theta - \pi/2), \quad (12)$$

$$\tilde{d}'(\theta) - \tilde{r}(\theta) = -v(\theta). \quad (13)$$

In retrospect, we now better understand the observation $r' + d = 0$ when the object makes vertex contact with the vertical sensor. After all, if the object has a vertex, then $\mathbf{x}(\theta - \pi/2)$ must be constant for some range of angles, i.e., the boundary curve must sit at that vertex for some range of angles. Consequently, $v(\theta - \pi/2)$ must be zero for the range of orientations during which the object makes vertex contact with the vertical sensor.

Moreover, we now see that Equation (12) actually allows us to discern whether the contact with the vertical sensor is a vertex contact. Similarly, Equation (13) allows us to discern whether the contact with the horizontal sensor is a vertex contact. Keep in mind, however, that this deduction is only possible if we know the rotation rate of the object. If we do not know the rotation rate of the object, then an unknown scale factor appears in the derivative terms of (12) and (13), preventing us from testing for homogeneity.

4.2 Shape Recovery

Assuming we know the rotation rate of the object (we take it to be unity), Equations (12) and (13) allow us to recover the shape of the object at both contact points. We focus here on the contact with the horizontal sensor. Specifically, the sensor readings $(s_1(t), s_2(t))$ over some time interval provide us directly with the generalized

support function $(\tilde{r}(\theta), \tilde{d}(\theta))$ over some corresponding angle interval. Equation (13) then allows us to compute $v(\theta)$ over that angle interval. In turn, this allows us to write down a differential equation for $\mathbf{x}(\theta)$ over that range of angles. The differential equation is simply:

$$\mathbf{x}'(\theta) = -v(\theta)\,\mathbf{t}(\theta). \tag{14}$$

Written out in coordinates, this becomes

$$\begin{aligned} \frac{dx}{d\theta}(\theta) &= \left\{\tilde{d}'(\theta) - \tilde{r}(\theta)\right\} \sin\theta, \\ \frac{dy}{d\theta}(\theta) &= \left\{\tilde{r}(\theta) - \tilde{d}'(\theta)\right\} \cos\theta. \end{aligned}$$

These are very simply differential equations, in that x and y do not appear on the right hand side. In other words, we can determine x and y by ordinary integration with respect to θ. Doing so removes the need to differentiate the function $\tilde{d}$ in the first place. We obtain

$$\boxed{\mathbf{x}(\theta) = \tilde{d}(\theta)\,\mathbf{t}(\theta) - \int \mathbf{c}(\theta)\mathbf{A}(\theta)d\theta,} \tag{15}$$

where all vectors are to be thought of as row vectors. With that convention in mind, $\mathbf{c}(\theta)$ is just the contact support function, $\mathbf{c}(\theta) = (\tilde{r}(\theta), \tilde{d}(\theta))$, and $\mathbf{A}(\theta)$ is the orthogonal matrix

$$\mathbf{A}(\theta) = \begin{pmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{pmatrix}.$$

In short, Equation (15) is our shape recovered.

5 Three Orthogonal Sensors

We now address the problem of determining the rotation rate of the unknown shape. We will use a third linear tactile sensor, as shown in Figure 11. In this section we will continue to use orthogonally mounted sensors. In the next section we will generalize our results to arbitrarily positioned sensors.

Figure 11 shows three sensors. Two of these are fixed, while the third complies vertically to any motions of the object. Thus the three sensors provide two pairs of sensor readings, i.e., four sensor values total. For simplicity we continue to assume that the object has a strictly convex smooth boundary. Thus we may designate the three contact points by $\mathbf{x}(\theta)$, $\mathbf{x}(\theta - \pi/2)$, and $\mathbf{x}(\theta - \pi)$, where $\mathbf{x}$ is an (unknown) parameterization of the object boundary in terms of normal angle. As before, we are interested in determining the local shape of the object at $\mathbf{x}(\theta)$. From Section 4 we know how to recover this local shape from tactile information provided by Sensors 1 and 2 alone, assuming we know the object's rotation rate.

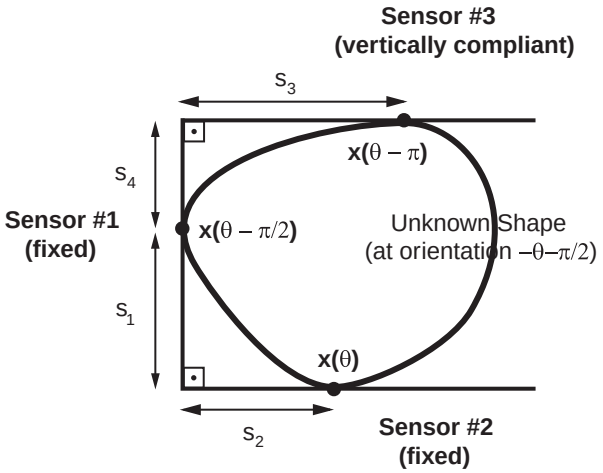

Figure 11: Three orthogonal sensors (two fixed, one vertically compliant) can usually recover both the rotation rate and the shape of the object.

Our basic strategy is as follows: We have two pairs of sensor readings that give us information about the local shape at the point $\mathbf{x}(\theta - \pi/2)$. Specifically, Equation (12) is an explicit formula for the local shape at $\mathbf{x}(\theta - \pi/2)$, computed from tactile information gained by Sensors 1 and 2. Similarly, a rewritten version of Equation (13) is another formula for the local shape at $\mathbf{x}(\theta - \pi/2)$, this time computed from tactile information gained by Sensors 3 and 1. Individually, each of these equations requires that we know the rotation rate of the object. Taken together, these two relationships provide a constraint on what the rotation rate must be.

In the remainder of this section we fill in the arithmetic details of the strategy just outlined. And we will briefly discuss exceptions for which this strategy fails.

5.1 Determining the Rotation Rate

As in Section 4, we let $(\tilde{r}(\theta), \tilde{d}(\theta))$ be the generalized support function of the unknown shape, given by Equations (10) and (11).

We now define the four sensor readings s_1, s_2, s_3, and s_4 as indicated in Figure 11. Specifically, s_1 and s_2 measure the locations of the contact points $\mathbf{x}(\theta - \pi/2)$ and $\mathbf{x}(\theta)$ relative to the intersection of the lines through Sensors 1 and 2. Similarly, s_3 and s_4 measure the locations of the contact points $\mathbf{x}(\theta - \pi)$ and $\mathbf{x}(\theta - \pi/2)$ relative to the intersection of the lines through Sensors 3 and 1.

The sensor readings are functions of time, while the generalized support function is a function of normal angle. We have the following relationship, much as we did in Section 4:

$$\begin{aligned} s_1(t) &= \tilde{r}(\theta(t)), && (16)\\ s_2(t) &= -\tilde{d}(\theta(t)), && (17)\\ s_3(t) &= \tilde{r}(\theta(t) - \pi/2), && (18)\\ s_4(t) &= -\tilde{d}(\theta(t) - \pi/2), && (19) \end{aligned}$$

where $\theta(t)$ indirectly describes the orientation of the object as a function of time t. Specifically, $\theta(t)$ is the angle of the outward normal at the contact point with Sensor 1, measured in body coordinates. Thus we can think of $-(\theta(t) + \pi/2)$ as the orientation of the object at time t, measured in sensor or world coordinates.

Differentiating with respect to time, we obtain:

$$\begin{aligned} \dot{s}_1(t) &= \omega(t)\, \tilde{r}'(\theta(t)), && (20)\\ \dot{s}_2(t) &= -\omega(t)\, \tilde{d}'(\theta(t)), && (21)\\ \dot{s}_3(t) &= \omega(t)\, \tilde{r}'(\theta(t) - \pi/2), && (22)\\ \dot{s}_4(t) &= -\omega(t)\, \tilde{d}'(\theta(t) - \pi/2), && (23) \end{aligned}$$

where $-\omega(t)$ is the unknown rotation rate of the object at time t. Equivalently, $\omega(t)$ is the rate in body coordinates at which the normal angle of the contact point with Sensor 1 is rotating. Our goal is to determine this rate ω.

Recall Equations (12) and (13). We will use Equation (12) directly, and recenter Equation (13) at $\theta - \pi/2$. This yields the following two relationships:

$$\begin{aligned} \tilde{r}'(\theta) + \tilde{d}(\theta) &= -v(\theta - \pi/2), && (24)\\ \tilde{d}'(\theta - \pi/2) - \tilde{r}(\theta - \pi/2) &= -v(\theta - \pi/2). && (25) \end{aligned}$$

Since the right hand sides of these two equations are the same, so are the left hand sides. Intuitively, sensor values s_1 and s_2 determine $v(\theta - \pi/2)$ from Equation (24), while sensor values s_3 and s_4 determine $v(\theta - \pi/2)$ from Equation (25). Equating the left hand sides, we see that:

$$\tilde{r}'(\theta) + \tilde{d}(\theta) = \tilde{d}'(\theta - \pi/2) - \tilde{r}(\theta - \pi/2). \quad (26)$$

Plugging in the relationship between the generalized support function $(\tilde{r}, \tilde{d})$ and the four sensor values $\{s_1, s_2, s_3, s_4\}$ described by Equations (16)–(23), and multiplying the entire equation (26) by $\omega(t)$, we see that:

$$\dot{s}_1(t) - \omega(t)\, s_2(t) = -\dot{s}_4(t) - \omega(t)\, s_3(t).$$

In other words,

$$\omega(t) \left[s_2(t) - s_3(t)\right] = \left[\dot{s}_1(t) + \dot{s}_4(t)\right], \quad (27)$$

and thus

$$\omega(t) = \frac{\dot{s}_1(t) + \dot{s}_4(t)}{s_2(t) - s_3(t)}. \quad (28)$$

5.2 Recovering the Shape

Equation (27) for smooth contacts is the analogue of Equation (9) determined in Section 3 for the case of one smooth and one vertex contact. From this equation we can obtain $\omega(t)$, thus placing us in a position to solve the shape integral (15) of Section 4.

As for details, there are several solution approaches. We briefly sketch one approach here. Whereas in Section 3.4 we suggested obtaining time as a function of angle, that is $t(\theta)$, in this section let us take the opposite approach and obtain angle as a function of time, that is, $\theta(t)$.

Recall the basic shape equation (14), which is a differential equation for $\mathbf{x}$ in terms of normal angle θ. We now have a second differential equation (28), which is a differential equation for θ in terms of time t. We can combine these two differential equations, to obtain a differential equation for

the boundary curve $\mathbf{x}$ parameterized as a function of time rather than angle:

First, from Equation (28) we have

$$\theta(t) \;=\; \int \frac{\dot{s}_1(t) + \dot{s}_4(t)}{s_2(t) - s_3(t)} dt. \tag{29}$$

If we now write $\mathbf{x}(t) = \mathbf{x}(\theta(t))$, then (14) becomes

$$\dot{\mathbf{x}}(t) \;=\; -v(\theta(t))\,\omega(t)\,\mathbf{t}(\theta(t)).$$

Combining Equations (13) and (28) with the information contained in Equations (16)–(23), we obtain

$$\begin{aligned} v(\theta(t))\,\omega(t) &= \tilde{r}(\theta(t))\,\omega(t) - \tilde{d}'(\theta(t))\,\omega(t) \\ &= s_1(t)\,\omega(t) + \dot{s}_2(t) \\ &= s_1(t)\,\frac{\dot{s}_1(t) + \dot{s}_4(t)}{s_2(t) - s_3(t)} + \dot{s}_2(t). \end{aligned} \tag{30}$$

Defining $\dot{s}(t)$ to be the final expression on the right hand side of (30), we obtain the following expression for $\mathbf{x}(t)$:

$$\boxed{\mathbf{x}(t) \;=\; -\int \dot{s}(t)\,\mathbf{t}(\theta(t))dt,} \tag{31}$$

where $\mathbf{t}(\theta(t)) \;=\; \big(\sin(\theta(t)), -\cos(\theta(t))\big)$ is the usual tangent vector and $\theta(t)$ is given by (29).

Equation (31) is our shape recovered, as a function of the set of four sensor signals $\{s_1(t), s_2(t), s_3(t), s_4(t)\}$.

5.3 Exceptions

We have skipped over one important detail. Equation (28) only allows us to solve for $\omega(t)$ if the expression $s_2(t) - s_3(t)$ is non-zero.

Suppose that $s_2(t) - s_3(t)$ is in fact zero at some time t. This tells us that the contact points with Sensors 2 and 3 lie directly above each other. Consequently, there is a unique instantaneous rotation center around which the object can rotate while maintaining all three contacts. In other words, instantaneously, the object can rotate without requiring Sensor 3 to move compliantly. Indeed, from Equation (27) we see that $\dot{s}_1(t) + \dot{s}_4(t)$ must be zero as well. $s_1(t) + s_4(t)$ is just the diameter of the object at its current orientation, as defined in Section 2.2. In other words, if $s_2(t) - s_3(t) = 0$, then the object is *of constant local breadth*.

Conversely, if the object is of constant local breadth, then $\dot{s}_1(t) + \dot{s}_4(t) = 0$ by definition. Equation (27) then tells us that either $\omega(t) = 0$ or again $s_2(t) - s_3(t) = 0$.

In short, the strategy of attaching a third orthogonal sensor to estimate the rotation rate of the object is successful if and only if the object is not of constant local breadth at the given orientation. This makes sense, for certainly we could not expect to estimate the rotation rate of, for example, a circle, from the motions of the contact points alone; they are stationary.

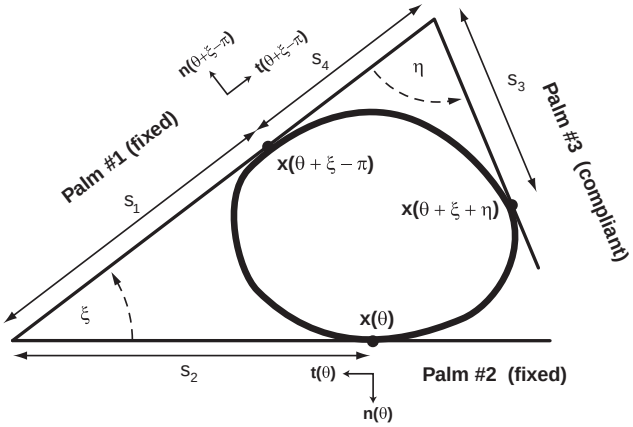

Figure 12: Three sensors (two fixed, one compliant) in arbitrary configurations can usually recover the shape and rotation rate of the object.

6 Three Arbitrarily-Positioned Linear Sensors

Figure 12 shows the setup for three arbitrarily positioned tactile sensors. As before, we assume that two of the sensors, namely those labeled "Palm 1" and "Palm 2" are fixed, while one sensor, labeled "Palm 3", complies to the motion of the object. Although, for the sake of simplicity, we have drawn the sensors as if they connect, this is not necessary. All that is required is three linear sensors at the contact points, lying on lines as drawn in the figure; the sensors themselves need

not have particularly large extent.

We assume that the robot knows the relative configurations of the palms to each other, but does not know the object shape. Our goal is to recover the object shape locally around its contact with Palm 1 (of course, we can also then recover the shape at the other two contacts).

First, we will assume contact only with two sensors, namely Palms 1 and 2. We will show how the information from these sensors is sufficient to recover the local shape of the object, assuming the rotation rate of the object is known. Second, we will show how all three sensors may be used to ascertain the rotation rate of the object. The basic approach is identical to the approach we have already described for orthogonal sensors, but there are some extra details. Where possible, we will skip over familiar details, jumping directly to the key results.

We continue to assume that our unknown shape has a smooth strictly convex boundary. We again parameterize that boundary curve by normal angle, written as $\mathbf{x}(\theta)$. Again, we note that the results generalize to piecewise smooth boundaries, and even, with appropriately modified sensors, to nonconvex boundaries.

6.1 Shape from Two Sensors

6.1.1 Contact Support Function

First, let us define yet one more variation of the contact support function, in order to account for the relative orientation of two sensors. In Figure 12, Palms 1 and 2 have relative orientation ξ. We assume that $0 < \xi < \pi$. We denote the contact point with the horizontal sensor, Palm 2, by $\mathbf{x}(\theta)$. Then the contact with Palm 1 must be $\mathbf{x}(\theta+\xi-\pi)$. We now define a generalized support function for such relative orientations, by the rules:

$$\tilde{r}_\xi(\theta) = \left[\mathbf{x}(\theta) - \mathbf{x}(\theta+\xi-\pi)\right] \cdot \mathbf{n}(\theta), \quad (32)$$

$$\tilde{d}_\xi(\theta) = \left[\mathbf{x}(\theta) - \mathbf{x}(\theta+\xi-\pi)\right] \cdot \mathbf{t}(\theta). \quad (33)$$

In other words, suppose we look at the vector pointing from the contact point with Palm 1 to the contact point with Palm 2. $\tilde{r}_\xi(\theta)$ is the projection of this vector onto the outward normal $\mathbf{n}(\theta)$ at Palm 2, while $\tilde{d}_\xi(\theta)$ is the projection of this vector onto the tangent $\mathbf{t}(\theta)$.

Note that for orthogonal sensors, $(\tilde{r}_{\pi/2}, \tilde{d}_{\pi/2})$ agrees with the old definition of $(\tilde{r}, \tilde{d})$ given in Equations (10) and (11). We further note that $(\tilde{r}_\xi, \tilde{d}_\xi)$ is not the only generalization of $(\tilde{r}, \tilde{d})$ that one could define. For instance, we will see below a slightly different version of this support function (see $(\hat{r}, \hat{d})$ later). Ultimately though, they all give the same data.

Differentiating Equations (32) and (33), we obtain a generalization of the relationships (12) and (13):

$$\tilde{r}'_\xi(\theta) + \tilde{d}_\xi(\theta) = -v(\theta+\xi-\pi)\,\sin\xi, \quad (34)$$

$$\tilde{d}'_\xi(\theta) - \tilde{r}_\xi(\theta) = -v(\theta) - v(\theta+\xi-\pi)\cos\xi. \quad (35)$$

6.1.2 Shape Recovered

Combining Equations (34) and (35) we obtain an explicit expression for the velocity $v(\theta)$ of the curve (or equivalently, the radius of curvature) at the contact point $\mathbf{x}(\theta)$:

$$v(\theta) = \left[\tilde{r}_\xi(\theta) - \tilde{d}'_\xi(\theta)\right] + \cot\xi\,\left[\tilde{r}'_\xi(\theta) + \tilde{d}_\xi(\theta)\right]. \quad (36)$$

We can put this into a familiar form by defining $(\hat{r}, \hat{d})$ as follows:

$$\hat{r}(\theta) = \tilde{r}_\xi(\theta) + \cot\xi\,\tilde{d}_\xi(\theta),$$

$$\hat{d}(\theta) = \tilde{d}_\xi(\theta) - \cot\xi\,\tilde{r}_\xi(\theta).$$

Then

$$\hat{d}'(\theta) - \hat{r}(\theta) = -v(\theta).$$

This equation looks just like Equation (13), with $(\tilde{r}, \tilde{d})$ replaced by $(\hat{r}, \hat{d})$. Thus we can recover the object shape using the integral formula (15), with $(\tilde{r}, \tilde{d})$ replaced by $(\hat{r}, \hat{d})$.

6.1.3 Geometric Interpretation of $(\hat{r}, \hat{d})$

The terms $\hat{r}(\theta)$ and $\hat{d}(\theta)$ have a simple geometric significance, as shown in Figure 13.

The point $\mathbf{COR}_{12}(\theta)$ denotes the instantaneous rotation center of the object while in contact with Palms 1 and 2 at the given orientation. $\hat{r}(\theta)$ is the distance of this rotation center from the horizontal sensor, Palm 2. $-\hat{d}(\theta)$ is the tangential distance of the rotation center from the intersection of the two sensors, i.e., from the point $\mathbf{O}$.

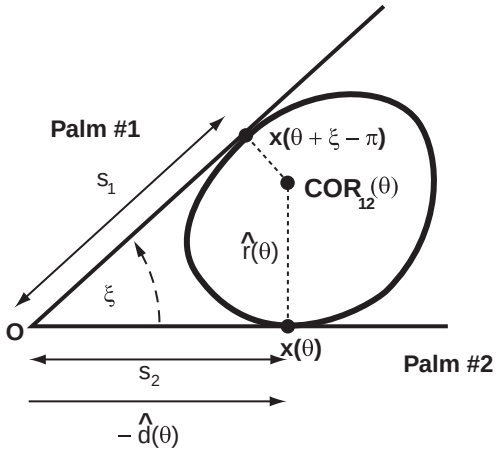

Figure 13: Another support function expresses the location of the center of rotation in local coordinates.

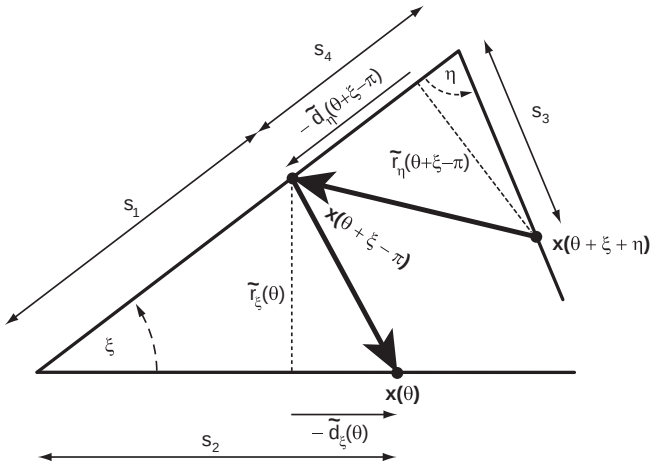

Figure 14: Relationship between sensor data and contact support functions.

6.2 Determining the Rotation Rate

We have seen how to recover the object shape from two sensors, assuming the object rotation rate is known. We now add a third palm, as in Figure 12, in order to determine the object rotation rate.

We now have two generalized support functions with which to work, namely $(\tilde{r}_\xi, \tilde{d}_\xi)$ and $(\tilde{r}_\eta, \tilde{d}_\eta)$. Here ξ is the relative orientation between Palms 1 and 2, while η is the relative orientation between Palms 3 and 1. As before, we assume $0 < \xi, \eta < \pi$. Note that the contact point with Palm 3 is $\mathbf{x}(\theta + \xi + \eta)$.

We now have two expressions for $v(\theta + \xi - \pi)$. The first comes from sensor data at Palms 1 and 2, and is simply Equation (34) rewritten:

$$v(\theta + \xi - \pi) = -\frac{\tilde{r}'_\xi(\theta) + \tilde{d}_\xi(\theta)}{\sin\xi}. \tag{37}$$

The second comes from sensor data at Palms 3 and 1, and is Equation (36) rewritten in terms of $(\tilde{r}_\eta, \tilde{d}_\eta)$ and recentered at $\theta + \xi - \pi$:

$$\begin{aligned} v(\theta + \xi - \pi) = & \\ & \left[\tilde{r}_\eta(\theta + \xi - \pi) - \tilde{d}'_\eta(\theta + \xi - \pi)\right] \\ + \quad & \cot\eta \, \left[\tilde{r}'_\eta(\theta + \xi - \pi) + \tilde{d}_\eta(\theta + \xi - \pi)\right]. \end{aligned} \tag{38}$$

Now, let us determine the relationship between the four sensed values $\{s_1(t), s_2(t), s_3(t), s_4(t)\}$ and the four support values $\{\tilde{r}_\xi(\theta), \tilde{d}_\xi(\theta), \tilde{r}_\eta(\theta + \xi - \pi), \tilde{d}_\eta(\theta + \xi - \pi)\}$. We assume that θ is an unknown function of time t, written as $\theta(t)$. Figure 14 is a recasting of Figure 12 with the object hidden but some extra labels shown. From this figure we see that:

$$\begin{aligned} s_1(t)\ \sin\xi &= \tilde{r}_\xi(\theta(t)), & (39)\\ s_1(t)\ \cos\xi - s_2(t) &= \tilde{d}_\xi(\theta(t)), & (40)\\ s_3(t)\ \sin\eta &= \tilde{r}_\eta(\theta(t) + \xi - \pi), & (41)\\ s_3(t)\ \cos\eta - s_4(t) &= \tilde{d}_\eta(\theta(t) + \xi - \pi). & (42) \end{aligned}$$

Differentiating with respect to time we obtain:

$$\begin{aligned} \dot{s}_1(t)\ \sin\xi &= \tilde{r}'_\xi(\theta(t))\,\omega(t), & (43)\\ \dot{s}_1(t)\ \cos\xi - \dot{s}_2(t) &= \tilde{d}'_\xi(\theta(t))\,\omega(t), & (44)\\ \dot{s}_3(t)\ \sin\eta &= \tilde{r}'_\eta(\theta(t) + \xi - \pi)\,\omega(t), & (45)\\ \dot{s}_3(t)\ \cos\eta - \dot{s}_4(t) &= \tilde{d}'_\eta(\theta(t) + \xi - \pi)\,\omega(t). & (46) \end{aligned}$$

Plugging these expressions into Equations (37) and (38), and multiplying by $\omega(t)$, yields the following two equations:

$$v(\theta(t)+\xi-\pi)\omega(t) = -\dot{s}_1(t) - \omega(t)\,\frac{s_1(t)\,\cos\xi - s_2(t)}{\sin\xi}, \tag{47}$$

$$v(\theta(t)+\xi-\pi)\omega(t) = \dot{s}_4(t) + \omega(t)\,\frac{s_3(t) - s_4(t)\,\cos\eta}{\sin\eta}. \tag{48}$$

Equating the right hand sides of (47) and (48), we finally arrive at:

$$\omega(t)\,\left[\frac{s_2(t) - s_1(t)\,\cos\xi}{\sin\xi} - \frac{s_3(t) - s_4(t)\,\cos\eta}{\sin\eta}\right] = \dot{s}_1(t) + \dot{s}_4(t). \tag{49}$$

Equation (49) allows us to solve for $\omega(t)$. Thereafter we can proceed much as we did at the end of Section 5.2. There is one additional detail. In the definition of the shape integral (31), $\dot{s}(t)$ was previously given by Equation (30). Using Equations (36) and (39)–(46), we see that $\dot{s}(t)$ now becomes

$$\dot{s}(t) = v(\theta(t))\omega(t) = \frac{s_1(t) - s_2(t)\,\cos\xi}{\sin\xi}\,\omega(t) + \dot{s}_2(t),$$

with $\omega(t)$ determined by Equation (49).

6.3 Exceptions

Finally, we should comment on the exceptions. Equation (49) only allows us to solve for $\omega(t)$ if the term

$$\left[\frac{s_2(t) - s_1(t)\,\cos\xi}{\sin\xi} - \frac{s_3(t) - s_4(t)\,\cos\eta}{\sin\eta}\right] \tag{50}$$

is non-zero. This term actually represents the distance between two instantaneous rotation centers of the object, namely $\mathbf{COR}_{12}$ and $\mathbf{COR}_{13}$. Here $\mathbf{COR}_{12}$ is the instantaneous rotation center due to contact with Palms 1 and 2, just as before. Similarly, $\mathbf{COR}_{13}$ is the instantaneous rotation center due to contact with Palms 1 and 3.

In order to see this, consider again Figures 13 and 14. From (40) we know that $s_2(t) - s_1(t)\cos\xi$ is simply the distance between the two contact points, as measured along Palm 2. Thus $[s_2(t) - s_1(t)\,\cos\xi]/\sin\xi$ is the distance of $\mathbf{COR}_{12}$ from Palm 1.

Similar reasoning shows that $[s_3(t) - s_4(t)\cos\eta]/\sin\eta$ is the distance of $\mathbf{COR}_{13}$ from Palm 1. Both centers of rotation lie on the same perpendicular from Palm 1. Thus (50) measures the (signed) distance between the centers of rotation.

Equation (49) now makes a lot of sense. It is telling us that the separation between Palms 2 and 3, as measured along Palm 1, is changing at a rate that is equal to the product of the rotation rate of the object and the distance between the two centers of rotation.

In particular, if the two centers of rotation coincide, then Palm 3 does not need to move compliantly as the object rotates. Conversely, if Palm 3 does not need to move compliantly, then either the object is not rotating or the two centers of rotation $\mathbf{COR}_{12}$ and $\mathbf{COR}_{13}$ coincide.

We can think of $s_1(t) + s_4(t)$ as a *generalized diameter function*. It is giving us the separation between two skew jaws of a gripper grasping our unknown object at the points $\mathbf{x}(\theta)$ and $\mathbf{x}(\theta+\xi+\eta)$, measured along a particular line in space. If the object locally has constant "diameter", as measured by this function, then we cannot recover the object's rotation rate, at least not with the given arrangement of the palms.

6.4 An Example

Figure 15 shows three tactile sensors touching an ellipse at an instant in time. The ellipse is rotating clockwise with constant angular acceleration. Two of the sensors are fixed, the third is compliant. The true ellipse appears as a thin dashed outline. The local shape recovered at the contacts appears as three thick curves. The ellipse rotated for 1 second total time. The data consisted of 1000 toplevel sample points. However, the internal integration routines required additional data at intermediate points, effectively multiplying by four the actual sample values required. Thus the sensors must deliver data at 4000 Hz, with infinite precision. The match obtained is fairly reasonable, although one can see slight discrepancies at the end of the sampling period. The maximum error was approximately 2.7%, occurring near the end of the sampling period. Here 100% corresponds to the length of the minor half-axis of the ellipse.

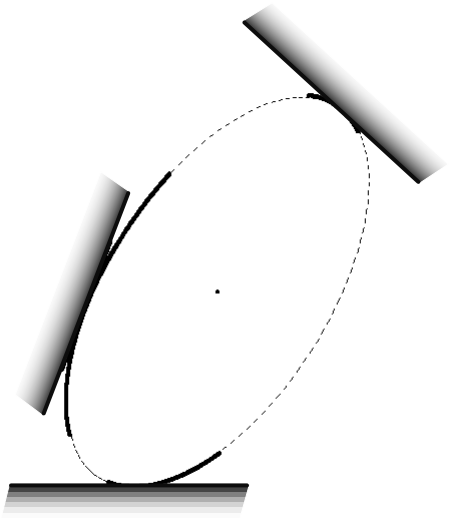

Figure 15: Three sensors touch a moving ellipse and recover its shape locally.

7 Acknowledgments

Many thanks to Yan-Bin Jia and Matt Mason for inspiration and engaging discussions about tactile sensing. Financial support for this research was provided in part by Carnegie Mellon University and in part by the National Science Foundation through the following grants: NSF Presidential Young Investigator award IRI-9157643 and NSF Grant IRI-9503648.

8 Bibliography

Akella, S. 1996. Robotic Manipulation for Parts Transfer and Orienting: Mechanics, Planning, and Shape Uncertainty. CMU-RI-TR-96-38. Ph.D. thesis. The Robotics Institute, Carnegie Mellon University.

Allen, P. K., and Michelman, P. 1990. Acquisition and Interpretation of 3-D Sensor Data from Touch. *IEEE Transactions on Robotics and Automation.* **6**(4):397–404.

Bay, J. S. 1989. Tactile Shape Sensing via Single- and Multi-Fingered Hands. *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, Vol 1, pp. 290–295.

Beccari, G., Caselli, S., and Zanichelli. F. 1997. Pose-Independent Recognition of Convex Objects from Sparse Tactile Data. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 3397–3402.

Bicchi, A., Salisbury, J. K., and Brock, D. L. 1993. Contact Sensing from Force Measurements. *International Journal of Robotics Research.* **12**(3):249–262.

Boissonnat, J. D., and Yvinec, M. 1992. Probing a Scene of Nonconvex Polyhedra. *Algorithmica.* **8**:321–342.

Cai, C. S, and Roth, B. 1986. On the Planar Motion of Rigid Bodies with Point Contact. *Mechanism and Machine Theory.* **21**(6):453–466.

Cai, C. S, and Roth, B. 1987. On the Spatial Motion of a Rigid Body with Point Contact. *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, North Carolina, pp. 686–695.

Caselli, S., Magnanini, C., Zanichelli, F., and Caraffi, E. 1996. Efficient Exploration and Recognition of Convex Objects Based on Haptic Perception. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 3508–3513.

Charlebois, M., Gupta, K., and Payandeh, S. 1996. Curvature Based Shape Estimation Using Tactile Sensing. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 3502–3507.

Charlebois, M., Gupta, K., and Payandeh, S. 1997. Shape Description of General, Curved Surfaces Using Tactile Sensing and Surface Normal Information. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 2819–2824.

Chen, N., Rink, R., and Zhang, H. 1996. Local Object Shape from Tactile Sensing. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 3496–3501.

Cole, R., and Yap, C. K. 1987. Shape from Probing. *Journal of Algorithms.* **8**(1):19–38.

Ellis, R. E. 1992. Planning Tactile Recognition Paths in Two and Three Dimensions. *International Journal of Robotics Research.* **11**(2):87–111.

Erdmann, M. A. 1995. An Exploration of Nonprehensile

Two-Palm Manipulation: Planning and Execution. *Seventh International Symposium on Robotics Research*, October 21–24, 1995, Herrsching, Germany, pp. 16–27.

Fearing, R. S. 1990. Tactile Sensing for Shape Interpretation. In *Dexterous Robot Hands*, Venkataraman, S .T., and Iberall, T., eds. Berlin: Springer-Verlag, pp. 209–238.

Fearing, R. S., and Binford, T. O. 1988. Using a Cylindrical Tactile Sensor for Determining Curvature. *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, Pennsylvania, pp. 765–771.

Gaston, P. C., and Lozano-Pérez, T. 1984. Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **PAMI-6**(3):257–266.

Goldberg, K. 1993. Orienting Polygonal Parts Without Sensors. *Algorithmica.* **10**(2,3,4):201–225.

Goldberg, K., Mason, M., and Erdmann, M. 1991. Generating Stochastic Plans for a Programmable Parts Feeder. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, pp. 352-359.

Grimson, W. E. L., and Lozano-Pérez, T. 1984. Model-Based Recognition and Localization from Sparse Range or Tactile Data. *International Journal of Robotics Research.* **3**(3):3–35.

Howe, R. D., and Cutkosky, M. R. 1992. Touch Sensing for Robotic Manipulation and Recognition. In *The Robotics Review 2*, eds. Khatib, O., Craig, J., and Lozano-Pérez, T., Cambridge, MA: MIT Press.

Howe, R. D., and Cutkosky, M. R. 1993. Dynamic Tactile Sensing: Perception of Fine Surface Features with Stress Rate Sensing. *IEEE Transactions on Robotics and Automation.* **9**(2):140–151.

Jia, Y.-B., and Erdmann, M. A. 1996a. Geometric Sensing of Known Planar Shapes. *International Journal of Robotics Research.* **15**(4):365–392.

Jia, Y.-B., and Erdmann, M. A. 1996b. Pose from Pushing. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 165–171.

Kaneko, M., Higashimori, M., and Tsuji, T. 1997. Pulling Motion Based Tactile Sensing for Concave Surface. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 2477–2484.

Kölzow, D., Kuba, A., and Volčič, A. 1989. An Algorithm for Reconstructing Convex Bodies from Their Projections. *Discrete Computational Geometry.* **4**:205–237.

Li, S.-Y. R. 1988. Reconstruction of Polygons from Projections. *Information Processing Letters.* **28**(5):235–240.

Lindenbaum, M., and Bruckstein, A. 1992. Parallel Strategies for Geometric Probing. *Journal of Algorithms.* **13**(2):320–349.

Lindenbaum, M., and Bruckstein, A. 1994. Blind Approximation of Planar Convex Sets. *IEEE Transactions on Robotics and Automation.* **10**(4):517–529.

MacMillan, W. D. 1936. *Dynamics of Rigid Bodies.* New York: Dover Publications. (1960 reprint of a 1936 McGraw-Hill edition.)

Montana, D. J. 1988. The Kinematics of Contact and Grasp. *International Journal of Robotics Research.* **7**(3):17–32.

Montana, D. J. 1995. The Kinematics of Multi-Fingered Manipulation. *IEEE Transactions on Robotics and Automation.* **11**(4):491–503.

Nicolson, E. J., and Fearing, R. S. 1995. The Reliability of Curvature Estimates from Linear Elastic Tactile Sensors. *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan, Vol 1, pp. 1126–1133.

Okamura, A. M., Turner, M. L., and Cutkosky, M. R. 1997. Haptic Exploration of Objects with Rolling and Sliding. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 2485–2490.

O'Neill, B. 1997. *Elementary Differential Geometry.* Second Edition. New York: Academic Press.

Rao, A. S., and Goldberg, K. Y. 1994. Shape from Diameter: Recognizing Polygonal Parts With a Parallel-Jaw Gripper. *International Journal of Robotics Research.* **13**(1):16–37.

Reuleuax, F. 1876. *The Kinematics of Machinery.* New York: Dover Publications. (1963 reprint of an 1876 Macmillan and Company edition.)

Roberts, K. S. 1990. Robot Active Touch Exploration: Constraints and Strategies. *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, Vol 2, pp. 980–985.

Romanik, K., and Salzberg, S. 1992. Testing Orthogonal Shapes. *Proceedings of the Fourth Canadian Conference on Computational Geometry*, pp. 216–222.

Santaló, L. A. 1976. *Integral Geometry and Geometric Probability.* Encyclopedia of Mathematics and its Applications. Volume 1. Reading, MA: Addison-Wesley.

Sinden, F. W. 1985. Shape Information from Rotated Scans. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **PAMI-7**(6):726–730.

Skiena, S. S. 1989. Problems in Geometric Probing. *Algorithmica.* **4**(4):599–605.

Tremblay, M. R., and Cutkosky, M. R. 1993. Estimating Friction Using Incipient Slip Sensing During a Manipulation Task. *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, Vol 1, pp. 429–434.

Zhang, H., Maekawa, H., and Tanie, K. 1996. Sensitivity Analysis and Experiments of Curvature Estimation Based on Rolling Contact. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 3514–3519.