Algorithmic Foundations of Robotics V, spr, p.293–310, 2003

Reconstructing the Shape and Motion of Unknown Objects with Active Tactile Sensors*

Mark Moll[1] and Michael A. Erdmann[2]

[1] Rice University, Houston TX 77005, USA
[2] Carnegie Mellon University, Pittsburgh PA 15213, USA

Abstract. We present a method to simultaneously reconstruct the shape and motion of an unknown smooth convex object. The object is manipulated by planar palms covered with tactile elements. The shape and dynamics of the object can be expressed as a function of the sensor values and the motion of the palms. We present a brief review of previous results for the planar case. In this paper we show that the 3D case is fundamentally different from the planar case, due to increased tangent dimensionality. The main contribution of this paper is a shape-dynamics analysis in 3D, and the synthesis of shape approximation methods via reconstructed contact point curves.

1 Introduction

Robotic manipulation of objects of unknown shape and weight is very difficult. To manipulate an object reliably a robot typically requires precise information about the object's shape and mass properties. Humans, on the other hand, seem to have few problems with manipulating objects of unknown shape and weight, even without visual feedback. During the manipulation of an unknown object the tactile sensors in the human hand give enough information to find the pose and shape of that object. At the same time humans can infer some mass properties of the object to determine a good grasp. Manipulation and sensing continuously interact with each other. These observations are an important motivation for our research. Our long-term goal is to develop combined manipulation and sensing strategies that allow robots to interact more robustly with unknown or uncertain environments. In this paper we present a model that integrates manipulation and tactile sensing. We derive equations for the shape and motion of an unknown object as a function of the motion of the manipulators and the sensor readings.

Figure 1 illustrates the basic idea for planar shapes. There are two palms that each have one rotational degree of freedom at the point where they connect, allowing the robot to change the angle between palm 1 and palm 2 and between the palms and the global frame. As the robot changes the palm angles it keeps track of the contact points through tactile elements on the palms.

* This work was supported in part by the National Science Foundation under grant IIS-9820180. The research was performed while the first author was with Carnegie Mellon University.

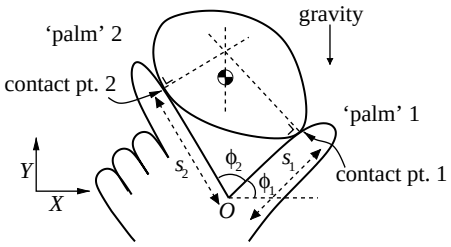


Fig. 1. A smooth convex object resting on two fingers that are covered with tactile sensors.

In the next section we will give an overview of the related work in shape reconstruction using tactile sensors. In section 3 we will briefly review planar shape reconstruction, which has been described in more detail in previous work [18, 19]. In sections 4 through 6 we will derive expressions for the shape and motion of an unknown 3D object being manipulated by three palms. In section 7 we will present some simulation results. Section 8 describes how we can use the curves traced out by the contact points to construct bounds and approximations of the entire shape. In section 9 we discuss how the model can be extended to arbitrary palm shapes. Finally, in section 10 we conclude and outline directions for future research.

2 Related Work

With tactile exploration the goal is to build up an accurate model of the shape of an unknown object. One early paper by Goldberg and Bajcsy [11] described a system requiring very little information to reconstruct an unknown shape. The system consisted of a cylindrical finger covered with 133 tactile elements. The finger could translate and tap different parts of an object.

Often the unknown shape is assumed to be a member of a parametrized class of shapes, so one could argue that this is really just shape recognition. Nevertheless, with some parametrized shape models, a large variety of shapes can still be characterized. In [10], for instance, results are given for recovering generalized cylinders. Allen and Roberts [2] model objects as superquadrics. Roberts [23] proposed a tactile exploration method for polyhedra. In [6] tactile data are fit to a general quadratic form. Finally, Liu and Hasegawa [15] use a network of triangular B-spline patches.

Allen and Michelman [1] presented methods for exploring shapes in three stages, from coarse to fine: grasping by containment, planar surface exploring and surface contour following. Montana [20] described a method to estimate curvature based on a number of probes. Montana also presented a control law for contour following. Charlebois et al. [4, 5] introduced two different tactile exploration methods. The first method is based on rolling a finger around the object to estimate the curvature using Montana's contact equations. Charlebois et al. analyze the sensitivity of this method to noise. With the second method a B-spline surface is fitted to the contact points and normals obtained by sliding multiple fingers along an unknown object.

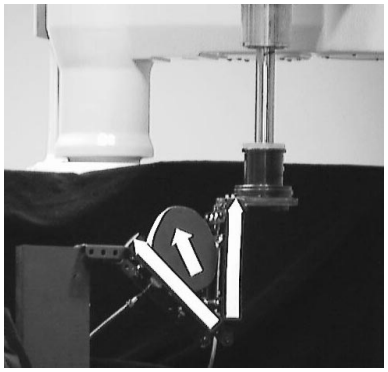
Marigo et al. [16] showed how to manipulate a known polyhedral part by rolling it between the two palms of a parallel-jaw gripper. Bicchi et al. [3] extended these results to tactile exploration of unknown objects with a parallel-jaw gripper equipped with tactile sensors. The two palms of the gripper roll the object without slipping and track the contact points. Using tools from regularization theory they produce spline-like models that best fit the sensor data. The work by Bicchi and colleagues is different from most other work on tactile shape reconstruction in that the object being sensed is not immobilized. With our approach the object is not immobilized either, but whereas Bicchi and colleagues assumed pure rolling we assume pure sliding.

A different approach is taken by Kaneko and Tsuji [13], who try to recover the shape by pulling a finger over the surface. With this finger they can also probe concavities. In [22, 21] the emphasis is on detecting fine surface features such as bumps and ridges. Sensing is done by rolling a finger around the object. Okamura et al. [21] show how one can measure friction by dragging a block over a surface at different velocities, measure the forces and solve for the unknowns.

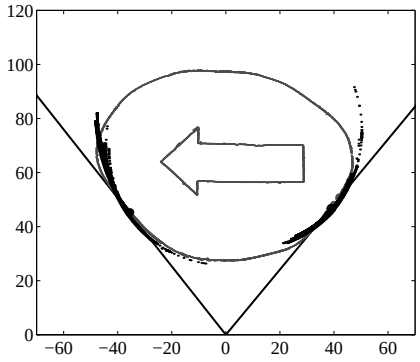
Much of our work builds forth on [9]. There, the shape of planar objects is recognized by three palms; two palms are at a fixed angle, the third palm can translate compliantly, ensuring that the object touches all three palms. Erdmann [9] derives the shape of an unknown object with an unknown motion as a function of the sensor values. In our work we no longer assume that the motion of the object is completely arbitrary. Instead, we model the dynamics of the object as it is manipulated by the palms. Only gravity and the contact forces are acting on the object. As a result we can recover the shape with fewer sensors. By modeling the dynamics we need one palm less in 2D. In this paper we will show that 3D objects can be reconstructed with three palms. In the long term we plan to develop a unified framework for reconstructing the shape and motion of unknown objects with varying contact modes.

3 Planar Shape Reconstruction

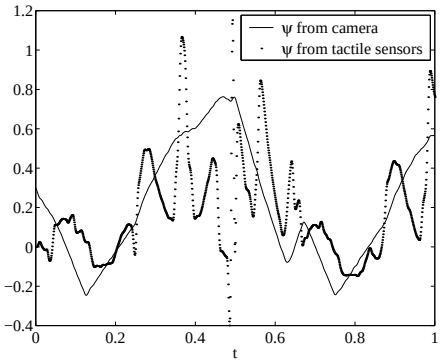
In this section we describe a method for reconstructing the shape and motion of an unknown planar object. Let the unknown shape be parametrized by the function $\boldsymbol{x} : [0, 2\pi] \to \mathbb{R}^2$, such that $\boldsymbol{x}(\theta)$ is the vector from the center of mass to the point on the surface where the outward-pointing normal $\boldsymbol{n}(\theta)$ is equal to $(\cos\theta, \sin\theta)^T$. Let the tangent $\boldsymbol{t}(\theta)$ be equal to $(\sin\theta, -\cos\theta)^T$ so that $[\boldsymbol{t}, \boldsymbol{n}]$ constitutes a right-handed frame. The projections of $\boldsymbol{x}(\theta)$ onto the normal and tangent are written as $r(\theta)$ and $d(\theta)$, respectively. The function r is called the *radius function*. To reconstruct $\boldsymbol{x}$ it is sufficient to reconstruct r: given the values of r at the contact points, the palm configurations, and palm angles we can solve for the values of the d function. Moreover, we have that $r'(\theta) = \boldsymbol{x}'(\theta) \cdot \boldsymbol{n}(\theta) - x(\theta) \cdot \boldsymbol{t}(\theta) = -d(\theta)$.



(a)



(b)



(c)

Fig. 2. Experimental Results. (**a**) Experimental setup. (**b**) Partially reconstructed shape. (**c**) Orientation measured by the vision system and the observed orientation

The rate of change of a sensor value is due to the relative motion of the object and the change of the contact point on the surface of the object. So if we would know the motion of the object, we could solve for the change of the contact point on the surface of the object. We can obtain the motion of the object in two different ways. One way is to assume the dynamics are quasistatic. At any point in time the object will be in force/torque balance. This constraint has the following geometric interpretation: the center of mass lies on a vertical line that passes through the lines of force at the contact points. This allows us to solve for the X-coordinate of the center of mass of the object. Assuming the object does not break contact, the object has only one degree of freedom, and thus we can solve for the motion of the object. This approach is described in more detail in [18]. Figure 2 shows some experimental results of shape reconstruction based on this model for the dynamics.

Another way to solve for the motion of the object is by analyzing the full dynamics. Based on the wrenches exerted by the palms on the object we can solve for the acceleration and angular acceleration of the object. Let $\boldsymbol{q}$ be a vector describing the state of the system. It contains the values of the

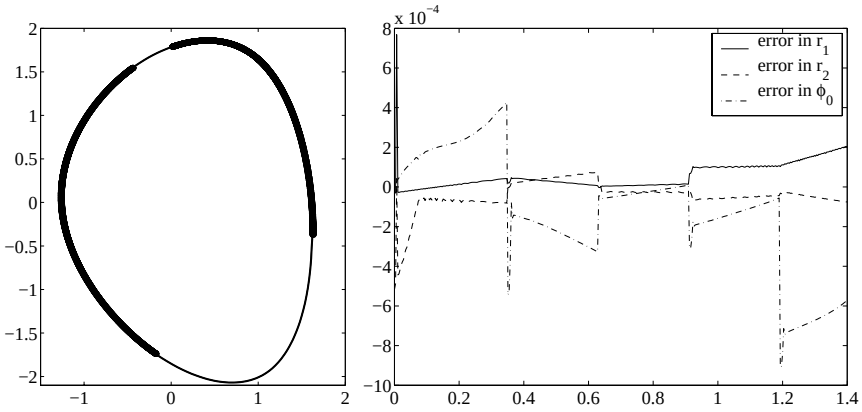


Fig. 3. Shape reconstruction with an observer based on Newton's method. (**a**) The reconstructed shape. (**b**) The absolute error in the radius function values, r_1 and r_2, and the orientation of the object, ϕ_0

radius function at the contact points, the rotational velocity of the object, the sensor values, and palm angles. Then we can write the solution for the shape and motion as a system of differential equations of the following form [19]:

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}) + \tau_1 \boldsymbol{g}_1(\boldsymbol{q}) + \tau_2 \boldsymbol{g}_2(\boldsymbol{q}), \tag{1}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{q}), \tag{2}$$

where $\boldsymbol{f}$, $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are vector fields, and $\boldsymbol{h}$ is called the *output function*. The output function returns the sensor readings. The vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are called the *input vector fields* and describe the rate of change of our system as torques are being applied on palm 1 and palm 2, respectively, at their point of intersection. The vector field $\boldsymbol{f}$ is called the *drift vector field*. It includes the effects of gravity. In [19] we describe this system in more detail and prove observability. If a system is observable, it is possible to construct a state estimator (called an observer) that, given an initial estimate near the true state, quickly converges to the true state as time progresses. In [17] we describe an observer that uses Newton's method to minimize the error in the estimate. This observer is due to [25]. The error is defined as the integral over a certain time interval of the squared norm of the difference in predicted and real output function values. Figure 3 shows some simulation results obtained using this observer. In this particular case the angle between the palms was fixed and the motion of the palms was simply a counter-clockwise rotation.

4 Three-Dimensional Shape Reconstruction

Naturally we would like to extend the results from the planar case to three dimensions. Although the same general approach can also be used for the 3D

case, there are also some fundamental differences. First of all, the difference is not just one extra dimension, but three extra dimensions: a planar object has three degrees of freedom and a 3D object has six. Analogous to the planar case we can derive a constraint on the position of the center of mass if we assume quasistatic dynamics. This constraint gives us the X and Y coordinate of the center of mass. However, assuming quasistatic dynamics is not sufficient to solve for the motion of an unknown object: if the object stays in contact with the palms it has *three* degrees of freedom, but the quasistatic dynamics give us only *two* constraints. Another difference from the planar case is that in 3D we cannot *completely* recover an unknown shape in finite time, since the contact points trace out only curves on the surface of the shape.

In 3D the surface of an arbitrary smooth convex object can be parameterized with spherical coordinates $\boldsymbol{\theta} = (\theta_1, \theta_2) \in [0, 2\pi) \times [-\frac{\pi}{2}, \frac{\pi}{2}]$. For a given $\boldsymbol{\theta}$ we define the following right-handed coordinate frame

$$\boldsymbol{t}_1(\boldsymbol{\theta}) = \begin{pmatrix} -\sin\theta_1 \\ \cos\theta_1 \\ 0 \end{pmatrix}, \ \boldsymbol{t}_2(\boldsymbol{\theta}) = \begin{pmatrix} -\cos\theta_1 \sin\theta_2 \\ -\sin\theta_1 \sin\theta_2 \\ \cos\theta_2 \end{pmatrix}, \ \boldsymbol{n}(\boldsymbol{\theta}) = \begin{pmatrix} \cos\theta_1 \cos\theta_2 \\ \sin\theta_1 \cos\theta_2 \\ \sin\theta_2 \end{pmatrix}. \quad (3)$$

The function $\boldsymbol{x} : S^2 \to \mathbb{R}^3$ describing a smooth shape can then be defined as follows. The vector $\boldsymbol{x}(\boldsymbol{\theta})$ is defined as the vector from the center of mass to the point on the shape where the surface normal is equal to $\boldsymbol{n}(\boldsymbol{\theta})$.

The *contact support function* $(r(\boldsymbol{\theta}), d(\boldsymbol{\theta}), e(\boldsymbol{\theta}))$ is defined as

$$r(\boldsymbol{\theta}) = \boldsymbol{x}(\boldsymbol{\theta}) \cdot \boldsymbol{n}(\boldsymbol{\theta}), \qquad d(\boldsymbol{\theta}) = \boldsymbol{x}(\boldsymbol{\theta}) \cdot \boldsymbol{t}_1(\boldsymbol{\theta}), \qquad e(\boldsymbol{\theta}) = \boldsymbol{x}(\boldsymbol{\theta}) \cdot \boldsymbol{t}_2(\boldsymbol{\theta})$$

The function $r(\boldsymbol{\theta})$ is called the *radius function*. The radius function completely describes the surface. The other two components of the contact support function appear in the derivatives of the radius function:

$$\frac{\partial r}{\partial \theta_1} = \frac{\partial \boldsymbol{x}}{\partial \theta_1} \cdot \boldsymbol{n} + \boldsymbol{x} \cdot \frac{\partial \boldsymbol{n}}{\partial \theta_1} = 0 + (\boldsymbol{x} \cdot \boldsymbol{t}_1) \cos\theta_2 = d \cos\theta_2, \quad (4)$$

$$\frac{\partial r}{\partial \theta_2} = \frac{\partial \boldsymbol{x}}{\partial \theta_2} \cdot \boldsymbol{n} + \boldsymbol{x} \cdot \frac{\partial \boldsymbol{n}}{\partial \theta_2} = 0 + (\boldsymbol{x} \cdot \boldsymbol{t}_2) = e. \quad (5)$$

By definition, the partial derivatives of $\boldsymbol{x}$ lie in the tangent plane, so the dot product of the partials with the normal is equal to 0. Let $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2})^T$ denote the surface parameters for contact point i. Below we will drop the argument $\boldsymbol{\theta}_i$ and replace it with a subscript i where it does not lead to confusion. For instance, we will write $\boldsymbol{n}_i$ for $\boldsymbol{n}(\boldsymbol{\theta}_i)$, the surface normal at contact point i in body coordinates.

The palms are modeled as three planes. The point of intersection of these planes is the origin of the world frame. Let us assume we can rotate each plane around the line of intersection with the horizontal plane through the origin. For each palm we can define a right-handed frame as follows. Let $\bar{\boldsymbol{n}}_i$ be the normal to palm i (in world coordinates) pointing toward the object, let $\bar{\boldsymbol{t}}_{i2}$ be the tangent normal to the axis of rotation and pointing in the positive

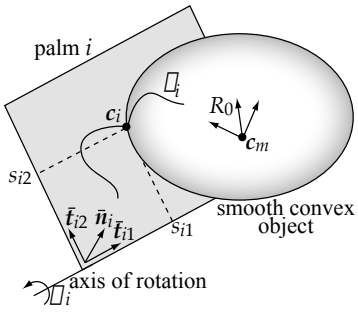


Fig. 4. Illustration of the notation.

Z direction and let $\bar{\boldsymbol{t}}_{i1}$ be $\bar{\boldsymbol{t}}_{i2} \times \bar{\boldsymbol{n}}_i$. Then $R_i = [\bar{\boldsymbol{t}}_{i1},\bar{\boldsymbol{t}}_{i2},\bar{\boldsymbol{n}}_i]$ is a right-handed frame for palm i. The configuration of the palms is completely described by the three rotation matrices R_1, R_2, and R_3. Let $\boldsymbol{s}_i$ denote the coordinates in palm frame i of contact point $\boldsymbol{c}_i$, so that $R_i \boldsymbol{s}_i = \boldsymbol{c}_i$. Note that the third component of $\boldsymbol{s}_i$ is always zero; by definition the distance of the contact point along the normal is zero. See Fig. 4 for an illustration.

The position and orientation of the unknown object are described by $\boldsymbol{c}_m$ and R_0, respectively. The center of mass is located at $\boldsymbol{c}_m$. The object coordinate frame defined by R_0 is chosen such that it coincides with the principal axes of inertia. The inertia matrix $\mathcal{I}$ can then be written as

$$\mathcal{I} = m \begin{pmatrix} \varrho_x^2 & 0 & 0 \\ 0 & \varrho_y^2 & 0 \\ 0 & 0 & \varrho_z^2 \end{pmatrix},$$

where m is the mass of the object, and the ϱ's correspond to the radii of gyration. We will write $\boldsymbol{\beta}_i$ for the curve traced out on the surface of the object by contact point i. So $\boldsymbol{\beta}_i(t) = \boldsymbol{x}\big(\boldsymbol{\theta}_i(t)\big)$.

5 Local Shape

The derivation of the shape and motion proceeds as follows. In this section we derive expressions for the contact point velocities in body coordinates. These expressions depend on the sensor values, the motions of the palms, and the motion of the object. In the next section we will solve for the motion of the object by analyzing the dynamics. We integrate the contact point velocities to obtain curves that describe the local shape of the object. We can recover the contact point velocities by considering the distances between the contact points and the rates at which they change. We derive additional constraints on the contact point velocities and curvature by considering the acceleration constraints induced by the position constraints.

We can write the constraint that the object maintains contact with each palm as

$$\boldsymbol{c}_i = \boldsymbol{c}_m + R_0 \boldsymbol{\beta}_i, \qquad i = 1, 2, 3. \tag{6}$$

The velocity of contact point i is therefore

$$\dot{\boldsymbol{c}}_i = \dot{\boldsymbol{c}}_m + \boldsymbol{\omega}_0 \times R_0 \boldsymbol{\beta}_i + R_0 \dot{\boldsymbol{\beta}}_i. \tag{7}$$

The difference between two contact point velocities is

$$\begin{aligned} \dot{\boldsymbol{c}}_i - \dot{\boldsymbol{c}}_j &= \boldsymbol{\omega}_0 \times R_0(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j) + R_0(\dot{\boldsymbol{\beta}}_i - \dot{\boldsymbol{\beta}}_j) && (8)\\ &= \boldsymbol{\omega}_0 \times (\boldsymbol{c}_i - \boldsymbol{c}_j) + R_0(\dot{\boldsymbol{\beta}}_i - \dot{\boldsymbol{\beta}}_j). && (9) \end{aligned}$$

Since we assume the object is smooth, we have that $\boldsymbol{n}_i \cdot \dot{\boldsymbol{\beta}}_i = 0$. Furthermore, the palm normals and object are related by the object orientation matrix:

$$\bar{\boldsymbol{n}}_i = -R_0 \boldsymbol{n}_i, \tag{10}$$

since $\bar{\boldsymbol{n}}_i$ is in world coordinates and $\boldsymbol{n}_i$ is in object coordinates. We can combine these constraints to solve for $\dot{\boldsymbol{\beta}}_i$:

$$\begin{aligned} \bar{\boldsymbol{n}}_i \cdot R_0 \dot{\boldsymbol{\beta}}_i &= 0 && (11)\\ \bar{\boldsymbol{n}}_j \cdot R_0 \dot{\boldsymbol{\beta}}_i &= \bar{\boldsymbol{n}}_j \cdot \big(\dot{\boldsymbol{c}}_i - \dot{\boldsymbol{c}}_j - \boldsymbol{\omega}_0 \times (\boldsymbol{c}_i - \boldsymbol{c}_j)\big) && (12)\\ \bar{\boldsymbol{n}}_k \cdot R_0 \dot{\boldsymbol{\beta}}_i &= \bar{\boldsymbol{n}}_k \cdot \big(\dot{\boldsymbol{c}}_i - \dot{\boldsymbol{c}}_k - \boldsymbol{\omega}_0 \times (\boldsymbol{c}_i - \boldsymbol{c}_k)\big), && (13) \end{aligned}$$

such that i, j, and k are distinct. Let Q be the defined as the 3×3 matrix with entries q_{ji}:

$$q_{ji} = \bar{\boldsymbol{n}}_j \cdot \big(\dot{\boldsymbol{c}}_i - \dot{\boldsymbol{c}}_j - \boldsymbol{\omega}_0 \times (\boldsymbol{c}_i - \boldsymbol{c}_j)\big). \tag{14}$$

Then we can write the solution for $\dot{\boldsymbol{\beta}}_1$, $\dot{\boldsymbol{\beta}}_2$, and $\dot{\boldsymbol{\beta}}_3$ more compactly as

$$\boxed{\big(\dot{\boldsymbol{\beta}}_1 \; \dot{\boldsymbol{\beta}}_2 \; \dot{\boldsymbol{\beta}}_3\big) = B^{-1} Q,} \tag{15}$$

where $B = \big(\bar{\boldsymbol{n}}_1 \; \bar{\boldsymbol{n}}_2 \; \bar{\boldsymbol{n}}_3\big)^T R_0$. As long as the palms are in general position, B will be invertible. Equation (15) describes the curves traced out by the contact points on the surface of the object (in body coordinates) as a function of the motion of the palms, the sensor values and the motion of the object. Note that these curves are not independent of each other. We know the configurations of the palms and the sensor values. If we also know one of the curves and the motion of the object, we can reconstruct the other curves. Below we will show that we can reconstruct all three curves by solving for the values of the radius function along the curves. Using (4) and (5), the derivative with respect to time of the radius function at contact point i is

$$\dot{r}_i = \frac{\partial r}{\partial \theta_{i1}} \dot{\theta}_{i1} + \frac{\partial r}{\partial \theta_{i2}} \dot{\theta}_{i2} = \begin{pmatrix} d_i \cos\theta_{i2} \\ e_i \end{pmatrix} \cdot \dot{\boldsymbol{\theta}}_i. \tag{16}$$

We will rewrite the right-hand side of this equation as a function of the motion of the palms and the object, and the values of the radius function at

the contact points. Using the position constraints we can rewrite d_i and e_i as a function of the configuration of the palms and r_i. We can write $\boldsymbol{\beta}_i$ as

$$\boldsymbol{\beta}_i = r_i \boldsymbol{n}_i + d_i \boldsymbol{t}_{i1} + e \boldsymbol{t}_{i2}. \tag{17}$$

The vector between contact point i and contact point j is then

$$\begin{aligned} \boldsymbol{c}_i - \boldsymbol{c}_j &= R_0(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j) \\ &= R_0(r_i \boldsymbol{n}_i + d_i \boldsymbol{t}_{i1} + e \boldsymbol{t}_{i2} - r_j \boldsymbol{n}_j + d_j \boldsymbol{t}_{j1} + e \boldsymbol{t}_{j2}). \end{aligned}$$

By rearranging terms we can obtain the following solution for the d's and e's:

$$\begin{pmatrix} d_1 \\ e_1 \\ d_2 \\ e_2 \\ d_3 \\ e_3 \end{pmatrix} = \begin{pmatrix} & & & & 0 & 0 \\ \boldsymbol{t}_{11} & \boldsymbol{t}_{12} & -\boldsymbol{t}_{21} & -\boldsymbol{t}_{22} & 0 & 0 \\ & & & & 0 & 0 \\ & & 0 & 0 & & \\ \boldsymbol{t}_{11} & \boldsymbol{t}_{12} & 0 & 0 & -\boldsymbol{t}_{31} & -\boldsymbol{t}_{32} \\ & & 0 & 0 & & \end{pmatrix}^{-1} \begin{pmatrix} R_0^T(\boldsymbol{c}_1 - \boldsymbol{c}_2) - (r_1 \boldsymbol{n}_1 - r_2 \boldsymbol{n}_2) \\ R_0^T(\boldsymbol{c}_1 - \boldsymbol{c}_3) - (r_1 \boldsymbol{n}_1 - r_3 \boldsymbol{n}_3) \end{pmatrix}. \tag{18}$$

'Hidden' in the tangent vectors are the $\boldsymbol{\theta}_i$'s. Using (3) we can write $\boldsymbol{\theta}_i$ as a function of the palm surface normal $\boldsymbol{n}_i$:

$$\boldsymbol{\theta}_i = \begin{pmatrix} \arctan(n_{i2}, n_{i1}) \\ \arcsin n_{i3} \end{pmatrix}. \tag{19}$$

The relationship between the normal $\boldsymbol{n}_i$ and the orientations of the palms and object is given by (10). The expression for $\dot{r}_i$ also contains $\dot{\boldsymbol{\theta}}_i$. By considering the derivative of the normal $\boldsymbol{n}_i$ we can obtain simple expressions for $\dot{\boldsymbol{\theta}}_i$. On the one hand we have that

$$\dot{\boldsymbol{n}}_i = \frac{\partial \boldsymbol{n}}{\partial \theta_{i1}} \dot{\theta}_{i1} + \frac{\partial \boldsymbol{n}}{\partial \theta_{i2}} \dot{\theta}_{i2} = \dot{\theta}_{i1} \cos\theta_2 \boldsymbol{t}_{i1} + \dot{\theta}_{i2} \boldsymbol{t}_{i2}. \tag{20}$$

But we can also obtain $\dot{\boldsymbol{n}}_i$ by differentiating (10):

$$\boldsymbol{\omega}_i \times \bar{\boldsymbol{n}}_i = \dot{\bar{\boldsymbol{n}}}_i = -\boldsymbol{\omega}_0 \times R_0 \boldsymbol{n}_i - R_0 \dot{\boldsymbol{n}}_i \Rightarrow \quad \dot{\boldsymbol{n}}_i = R_0^T \big((\boldsymbol{\omega}_0 - \boldsymbol{\omega}_i) \times \bar{\boldsymbol{n}}_i \big). \tag{21}$$

Here $\boldsymbol{\omega}_i$ is the rotational velocity of palm i. Combining these two expressions for $\dot{\boldsymbol{n}}_i$ we can write $\dot{\boldsymbol{\theta}}_i$ as

$$\dot{\boldsymbol{\theta}}_i = \big(\boldsymbol{t}_{i1} / \cos\theta_{i2} \; \boldsymbol{t}_{i2} \big)^T \, R_0^T \big((\boldsymbol{\omega}_0 - \boldsymbol{\omega}_i) \times \bar{\boldsymbol{n}}_i \big). \tag{22}$$

Let us now consider the constraints on the acceleration of the object induced by the three point contact assumption. This will provide us with an additional constraint on $\boldsymbol{\beta}_i$ and will give us some more insight into how the 3D case is fundamentally different from the planar case. By differentiating (7) we obtain the following constraint on the acceleration:

$$\ddot{\boldsymbol{c}}_i = \boldsymbol{a}_0 + \boldsymbol{\alpha}_0 \times R_0 \boldsymbol{\beta}_i + \boldsymbol{\omega}_0 \times (\boldsymbol{\omega}_0 \times R_0 \boldsymbol{\beta}_i + 2 R_0 \dot{\boldsymbol{\beta}}_i) + R_0 \ddot{\boldsymbol{\beta}}_i, \tag{23}$$

where $\boldsymbol{a}_0$ and $\boldsymbol{\alpha}_0$ are the acceleration and angular acceleration of the object. (We will solve for $\boldsymbol{a}_0$ and $\boldsymbol{\alpha}_0$ in the next section by analyzing the dynamics.) Observe that from differentiation of the smoothness constraint $\dot{\boldsymbol{\beta}}_i \cdot \boldsymbol{n}_i = 0$ it follows that $\ddot{\boldsymbol{\beta}}_i \cdot \boldsymbol{n}_i = -\dot{\boldsymbol{\beta}}_i \cdot \dot{\boldsymbol{n}}_i$. We can therefore rewrite the acceleration constraint in the normal direction as a constraint on $\dot{\boldsymbol{\beta}}_i$. First, we rewrite the terms containing $\dot{\boldsymbol{\beta}}_i$ and $\ddot{\boldsymbol{\beta}}_i$:

$$\begin{aligned}
&\bar{\boldsymbol{n}}_i \cdot (\boldsymbol{\omega}_0 \times 2R_0\dot{\boldsymbol{\beta}}_i) + \bar{\boldsymbol{n}}_i \cdot R_0\ddot{\boldsymbol{\beta}}_i \\
&\quad = 2(\bar{\boldsymbol{n}}_i \times \boldsymbol{\omega}_0) \cdot R_0\dot{\boldsymbol{\beta}}_i + R_0(\dot{\boldsymbol{n}}_i \cdot \dot{\boldsymbol{\beta}}_i) \\
&\quad = 2(\bar{\boldsymbol{n}}_i \times \boldsymbol{\omega}_0) \cdot R_0\dot{\boldsymbol{\beta}}_i + ((\boldsymbol{\omega}_0 - \boldsymbol{\omega}_i) \times \bar{\boldsymbol{n}}_i) \cdot R_0\dot{\boldsymbol{\beta}}_i \\
&\quad = (\bar{\boldsymbol{n}}_i \times (\boldsymbol{\omega}_0 + \boldsymbol{\omega}_i)) \cdot R_0\dot{\boldsymbol{\beta}}_i.
\end{aligned}$$

The constraint on $\dot{\boldsymbol{\beta}}_i$ is therefore

$$\begin{aligned}
&(\bar{\boldsymbol{n}}_i \times (\boldsymbol{\omega}_0 + \boldsymbol{\omega}_i)) \cdot R_0\dot{\boldsymbol{\beta}}_i = \\
&\qquad\qquad \bar{\boldsymbol{n}}_i \cdot \big(\ddot{\boldsymbol{c}}_i - \boldsymbol{a}_0 - \boldsymbol{\alpha}_0 \times R_0\boldsymbol{\beta}_i - \boldsymbol{\omega}_0 \times (\boldsymbol{\omega}_0 \times R_0\boldsymbol{\beta}_i)\big). \quad (24)
\end{aligned}$$

Let us now consider what how the acceleration constraint describes certain time-independent shape properties of the contact curves. The *Darboux frame field* [24] can be used to describe curves on a surface. For the curve $\boldsymbol{\beta}_i$ the Darboux frame field is defined by the unit tangent T of $\boldsymbol{\beta}_i$, the surface normal U restricted to $\boldsymbol{\beta}_i$, and $V = U \times T$. The normal U coincides with $\boldsymbol{n}_i$. Note that the normal of the *curve* does not necessarily coincide with the normal of the *surface*. Similar to the Frenet frame field, the derivatives of T, V, and U can be expressed in terms of T, V, and U:

$$\dot{T} = v(\qquad \kappa_g V + \kappa_n U) \quad (25)$$

$$\dot{V} = v(-\kappa_g T \qquad + \tau_g U\ ) \quad (26)$$

$$\dot{U} = v(-\kappa_n T - \tau_g V \qquad\ ) \quad (27)$$

Here $v = \|\dot{\boldsymbol{\beta}}_i\|$ is the velocity of the curve, κ_g the *geodesic curvature*, κ_n the *normal curvature*, and τ_g the *geodesic torsion*. The geodesic curvature at a point describes the 'bending' of the curve in the tangent plane of the surface at that point. The normal curvature at a point describes the 'bending' of the curve in the surface normal direction. Using this frame field we can write $\ddot{\boldsymbol{\beta}}_i$ as

$$\ddot{\boldsymbol{\beta}}_i = \dot{v}T + v\dot{T} = \dot{v}T + v^2(\kappa_g V + \kappa_n U). \quad (28)$$

So by taking the dot product with the normal on both sides of the acceleration constraint we can obtain a constraint on the normal curvature of the curve. In the planar case the velocity of the curve is equal to the radius of curvature, and the acceleration constraint determines the (normal) curvature at the contact points. In the 3D case, the acceleration constraint puts a constraint on the normal curvature at the contact points. But now we have two extra

curve shape parameters, κ_g and τ_g, which are equal to zero in the planar case. In other words, in 3D the contact point curves are less constrained than in 2D.

6 Dynamics

The solution for the shape of the object depends on the motion of the object. We can solve for the motion by writing out the dynamics equations for the system formed by the palms and the object. If the object remains in contact with the palms, the object has only three degrees of freedom. It is thus sufficient to solve for the angular acceleration. We obtain the rotational velocity by integration. Let I_i be the moment of inertia of palm i around its axis of rotation, α_i the angular acceleration around that axis, τ_i the torque produced by palm i's motor at the axis of rotation, and f_i the magnitude of the contact force. Then the motion of palm i is described by

$$I_i\alpha_i = \tau_i - f_i s_{i2}, \tag{29}$$

where s_{i2} is the second component of $\boldsymbol{s}_i$. From the definition of the palm frame it follows that s_{i2} measures the distance to the axis of rotation (see also Fig. 4). The net force and net torque on the object are given by Newton's and Euler's equations:

$$\boldsymbol{F}_0 = m\boldsymbol{a}_0 = \boldsymbol{F}_g + \textstyle\sum_{i=1}^{3} f_i\bar{\boldsymbol{n}}_i \tag{30}$$

$$\boldsymbol{\tau}_0 = \mathcal{I}'\boldsymbol{\alpha}_0 + \boldsymbol{\omega}_0 \times \mathcal{I}'\boldsymbol{\omega}_0 = \textstyle\sum_{i=1}^{3} \boldsymbol{\tau}_{\boldsymbol{c}_i} \tag{31}$$

where

$$\mathcal{I}' = R_0\mathcal{I}R_0^T \tag{32}$$

$$\boldsymbol{\tau}_{\boldsymbol{c}_i} = (R_0\boldsymbol{\beta}_i) \times (f_i\bar{\boldsymbol{n}}_i) = -f_i R_0(\boldsymbol{\beta}_i \times \boldsymbol{n}_i). \tag{33}$$

From these equations we can solve for the angular acceleration of the object, $\boldsymbol{\alpha}_0$:

$$\boldsymbol{\alpha}_0 = -\mathcal{I}'^{-1}\big(\boldsymbol{\omega}_0 \times \mathcal{I}'\boldsymbol{\omega}_0 + \textstyle\sum_{i=1}^{3} \frac{\tau_i - I_i\alpha_i}{s_{i2}} R_0(\boldsymbol{\beta}_i \times \boldsymbol{n}_i)\big). \tag{34}$$

Let us assume we can control the palms to move at a constant rotational velocity. The angular acceleration terms α_i will then disappear. We can summarize the simultaneous solution for the shape and motion of an unknown smooth convex object manipulated by three flat palms with the following system of differential equations

$$\dot{r}_i = (d_i \cos\theta_{i2},\, e_i)^T \cdot \dot{\boldsymbol{\theta}}_i, \qquad i = 1, 2, 3 \tag{35}$$

$$\dot{R}_0 = \hat{\boldsymbol{\omega}}_0 R_0 \tag{36}$$

$$\dot{\boldsymbol{\omega}}_0 = -\mathcal{I}'^{-1}\big(\boldsymbol{\omega}_0 \times \mathcal{I}'\boldsymbol{\omega}_0 + \textstyle\sum_{i=1}^{3} \frac{\tau_i}{s_{i2}} R_0(\boldsymbol{\beta}_i \times \boldsymbol{n}_i)\big). \tag{37}$$

Here $\hat{\boldsymbol{\omega}}_0$ is the matrix form of the cross product, i.e., the 3×3 matrix such that $\hat{\boldsymbol{\omega}}_0 \boldsymbol{p} = \boldsymbol{\omega}_0 \times \boldsymbol{p}$ for any vector $\boldsymbol{p}$. Equation (35) describes the shape of the object at the contact points. Equations (36) and (37) describe the dynamics of the object. We can replace the variables d_i, e_i, θ_{i2}, $\dot{\boldsymbol{\theta}}_i$, and $\boldsymbol{\beta}_i$ with the solutions given in (18), (19), (22) so that the system of differential equations only depends on the values of the radius function at the contact points, palm configurations and sensor values. This allows us to integrate the system of differential equations given the palm configurations and sensor values.

7 Simulation Results

We have written a program to simulate the motion of an arbitrary smooth convex object supported by three planes. To reconstruct the shape we need to integrate out the system of differential equations given by (35)–(37). There are several reasons why straightforward integration is not likely to produce good results. We can improve the results in the following two ways. First, we should use quaternions [7] for orientations to improve numerical stability and avoid singularities of three-parameter representations of orientations. Second, we should try to enforce *all* the constraints on the contact point curves given by (15) and (24). We use a prediction-correction numerical integration method. We use the difference between prediction and correction as an error measure when searching for the initial conditions of the system of differential equations.

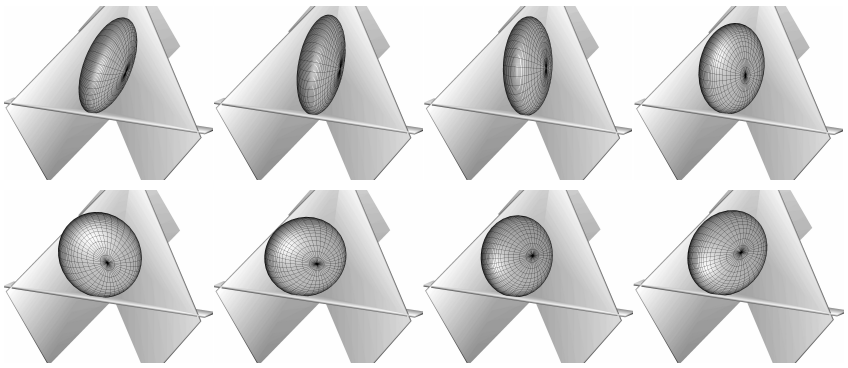


Fig. 5. An object rolling and sliding on immobile palms with gravity and contact forces acting on it. The object is given some initial rotational velocity. It is shown at $t = 0, 0.6, \ldots, 4.2$.

Figure 5 shows the motion of an ellipsoid supported by three palms. The palms are at 60 degree angles with the horizontal plane. The rotational axes are at 60 degree angles with each other. The radii of the axes of the ellipsoid are 2.5, 2, and 1. One problem with simulating this system without friction is that if the palms are moving they continuously increase the kinetic energy of

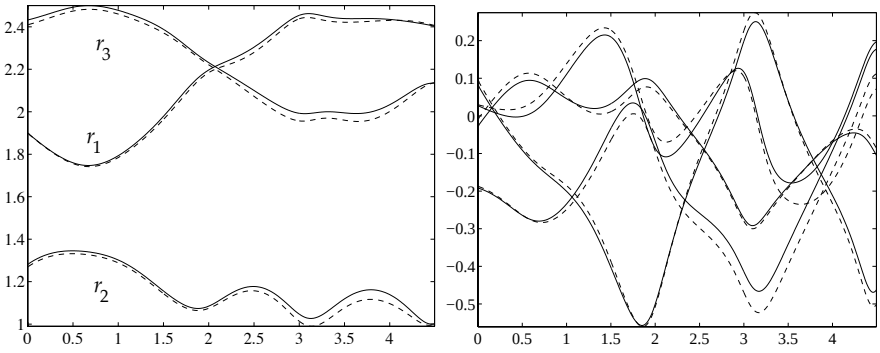


Fig. 6. Differences between real and observed shape and motion. Real values are plotted as solid lines, observed values as dashed lines. (**a**) The radius function values at the contact points. (**b**) The four quaternion components of the rotational velocity

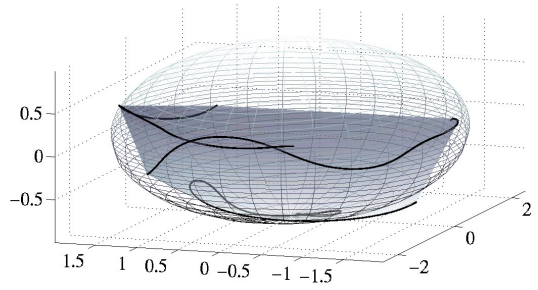


Fig. 7. The convex hull of the contact curves gives a lower bound on the volume occupied by the object. The true shape of the object is shown as a wire-frame model.

the object. In our simulations the ellipsoid breaks contact before a significant part of the object is recovered. To get around this problem, we keep the palms in the same position and give the ellipsoid some initial velocity. In Fig. 5 the object has an initial rotational velocity of $(0, 0.3, 0.3)^T$. In the search for initial conditions we assume the initial rotational velocity is known. This is, of course, not very realistic. If we would model friction, then moving the palms would become feasible and we could start the system with the object at rest. In that case the rotational velocity would be zero. The reconstructed shape and motion are shown in Fig. 6.

8 Shape Approximations

Given the recovered motion and contact curves we can also give a lower and an upper bound on the volume occupied by the entire shape. Since we assume the object is convex, the convex hull of the contact curves in the object frame is a lower bound on the shape. Figure 7 shows the lower bound on the shape obtained from the example in the previous section. We can obtain an upper bound by observing that at each contact point the corresponding palm plane

introduces a half-space constraint: the object has to lie entirely on one side of the plane. Clearly, the intersection of half-spaces along the contact curves in the object frame forms an upper bound on the shape. Note that we can obtain slightly tighter upper and lower bounds if we are given bounds on the curvature of the object.

These bounds could form the basis for a manipulation strategy. Having bounds makes it possible to speculate about the outcomes of actions. Suppose we would like to get an estimate of the entire shape. One objective for a planner could be to minimize the difference between the upper bound and lower bound. A simple 'greedy' planner would always try to move the contact points toward the largest distance between upper and lower bound. A smarter planner would also take into account the path length, such that the palms would minimize nearby differences between the bounds before trying to minimize faraway differences. This assumes we have some reasonable metric to measure distance between palm/object configurations.

9 Arbitrary Palm Shapes

In this paper we have written the shape and motion of an unknown object as a function of the contact point positions $\boldsymbol{c}_i$ and the contact point velocities $\dot{\boldsymbol{c}}_i$, all in world coordinates. We wrote out these positions and velocities in terms of poses of the palms and the sensor values. Effectively, we have used world coordinates to decouple the solution for the shape and motion of the unknown object from the shape and motion of the palm. This procedure allows us to use differently shaped palms without changing the solution for the shape and motion of the unknown object. Suppose we were to use spherical palms. For example, Fig. 8 shows two spherical palms holding an object. The contact point on palm i can be written as

$$\boldsymbol{c}_i = b_i R_i \begin{pmatrix} \cos s_{i1} \cos s_{i2} \\ \sin s_{i1} \cos s_{i2} \\ \sin s_{i2} \end{pmatrix}, \tag{38}$$

where b_i is the radius of sphere/palm i, and (s_{i1}, s_{i2}) are the spherical sensor coordinates. We can obtain expressions for $\dot{\boldsymbol{c}}_i$ by differentiating the right-hand side of equation 38 with respect to R_i and $\boldsymbol{s}_i$. The solution for the dynamics of the object would change slightly with spherical palms, but the solution for the shape and motion of the object would remain structurally

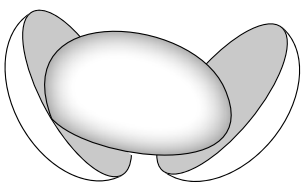


Fig. 8. Two spherical palms holding an object.

the same. Equations 14 and 15 show clearly how the solution for the shape of the object depends on the motion of the contact points.

Changing the palm shapes raises many new questions. Intuitively, it seems that spherical palms would be 'better' than flat palms, but can we quantify this? We can use the shape of the palms to get tighter upper bounds on the shape of the unknown object. Instead of taking intersections of half-spaces, we can now use intersections of spheres as an upper bound. It also seems that with spherical palms we would have to move the palms less to reach a point on the surface.

If we allow some of the palms to be convex, we could actually recover concavities on the object provided the radius of curvature along the concavities is larger than the radius of the palm. Allowing the object to have concavities would invalidate our bounds from the previous section. We can define a new upper bound as the intersection of half-spaces at those points on the contact curves that are also on the boundary of the convex hull of the curves. We can also define other approximate representations of the shape of the object that may be closer to the true shape, such as spline interpolations, α-shapes [8] or principal surfaces [12, 14].

10 Discussion

We have described a method for reconstructing the shape and motion of unknown smooth convex objects with tactile sensors. We required no immobilization of the object or even control of its exact motion. Instead, we used nonprehensile manipulation with planar palms covered with tactile elements to simultaneously manipulate and sense an object. The motion and shape of the object can be written as a function of the motion of the palms and the sensor values. This work will have applications in grasping (partially) unknown objects and tactile exploration of unknown environments.

In this paper we assumed there was no friction and that the unknown object being manipulated by the palms was smooth and convex. In future work we are planning to remove these assumptions. In the presence of friction the normal forces will no longer be along the normal, but at an angle with the normal. If the contact point is moving this angle is proportional to the coefficient of friction. If we know the coefficient of friction, we can again solve for the motion of the object. If a contact point is not moving (i.e., the contact force is inside the friction cone), but one of the other contact points is, we may be able to write the shape and motion of that contact point relative to the fixed contact point. Non-smoothness of the object results in some cases in additional constraints or at least non-smoothness in the sensor values, that can be used to detect corners and edges.

Removing the assumptions makes the system more complex, but the basic formulation remains the same. One question that arises if we allow various contact modes is whether we can detect which contact mode the system is

in. We can run observers for each contact mode in parallel and have a higher level algorithm decide which observer is (most likely) to have the smallest error in the state.

The contact kinematics lead naturally to a description of the shape of an object in terms of the curvature at the contact points. One disadvantage of this approach is that the shape has an infinite number of degrees of freedom. Although this allows us to reconstruct *any* smooth convex shape, in many cases it may be sufficient to have a 'reasonable' approximation. With future work we would like to find finite dimensional models for shape that admit a wide range of shapes and can be constructed efficiently in an incremental way. We may be able to apply some of the techniques developed for reconstructing the shape of immobilized objects.

Finally, we would like to construct a state-estimation method for the 3D case with observer-like properties. Since we currently do not have a state-space description of the system, constructing an observer is not possible. However, we derived more constraints on the shape and motion of the unknown than there are unknowns. These extra constraints can be used to adjust our state estimate. Borrowing ideas from reduced-order observers, we may be able construct something similar for our system. Alternatively, we can use the three additional constraints to estimate the moments of inertia, which we had previously assumed to be known.

References

1. Allen, P. K. and Michelman, P. (1990). Acquisition and interpretation of 3-D sensor data from touch. *IEEE Trans. on Robotics and Automation*, 6(4):397–404.
2. Allen, P. K. and Roberts, K. S. (1989). Haptic object recognition using a multi-fingered dextrous hand. In *Proc. 1989 IEEE Intl. Conf. on Robotics and Automation*, pp. 342–347, Scottsdale, AZ.
3. Bicchi, A., Marigo, A., and Prattichizzo, D. (1999). Dexterity through rolling: Manipulation of unknown objects. In *Proc. 1999 IEEE Intl. Conf. on Robotics and Automation*, pp. 1583–1588, Detroit, Michigan.
4. Charlebois, M., Gupta, K., and Payandeh, S. (1996). Curvature based shape estimation using tactile sensing. In *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*, pp. 3502–3507.
5. Charlebois, M., Gupta, K., and Payandeh, S. (1997). Shape description of general, curved surfaces using tactile sensing and surface normal information. In *Proc. 1997 IEEE Intl. Conf. on Robotics and Automation*, pp. 2819–2824.
6. Chen, N., Rink, R., and Zhang, H. (1996). Local object shape from tactile sensing. In *Proc. 1996 IEEE Intl. Conf. on Robotics and Automation*, pp. 3496–3501.
7. Chou, J. C. (1992). Quaternion kinematic and dynamic differential equations. *IEEE Trans. on Robotics and Automation*, 8(1):53–64.

8. Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72.
9. Erdmann, M. A. (1999). Shape recovery from passive locally dense tactile data. In *Robotics: The Algorithmic Perspective*. A.K. Peters.
10. Fearing, R. S. (1990). Tactile sensing for shape interpretation. In *Dexterous Robot Hands*, chapter 10, pp. 209–238. Springer Verlag, Berlin; Heidelberg; New York.
11. Goldberg, K. Y. and Bajcsy, R. (1984). Active touch and robot perception. *Cognition and Brain Theory*, 7(2):199–214.
12. Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406):502–516.
13. Kaneko, M. and Tsuji, T. (2001). Pulling motion based tactile sensing. In *Algorithmic and Computational Robotics: New Directions*, pp. 157–167. A. K. Peters.
14. LeBlanc, M. and Tibshirani, R. (1994). Adaptive principal surfaces. *Journal of the American Statistical Association*, 89(425):53–64.
15. Liu, F. and Hasegawa, T. (2001). Reconstruction of curved surfaces using active tactile sensing and surface normal information. In *Proc. 2001 IEEE Intl. Conf. on Robotics and Automation*, pp. 4029–4034.
16. Marigo, A., Chitour, Y., and Bicchi, A. (1997). Manipulation of polyhedral parts by rolling. In *Proc. 1997 IEEE Intl. Conf. on Robotics and Automation*, pp. 2992–2997.
17. Moll, M. (2002). *Shape Reconstruction Using Active Tactile Sensors*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.
18. Moll, M. and Erdmann, M. A. (2001). Reconstructing shape from motion using tactile sensors. In *Proc. 2001 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 691–700, Maui, HI.
19. Moll, M. and Erdmann, M. A. (2002). Dynamic shape reconstruction using tactile sensors. In *Proc. 2002 IEEE Intl. Conf. on Robotics and Automation*, pp. 1636–1641.
20. Montana, D. J. (1988). The kinematics of contact and grasp. *Intl. J. of Robotics Research*, 7(3):17–32.
21. Okamura, A. M., Costa, M. A., Turner, M. L., Richard, C., and Cutkosky, M. R. (1999). Haptic surface exploration. In *Experimental Robotics VI*, Sydney, Australia. Springer Verlag.
22. Okamura, A. M. and Cutkosky, M. R. (2001). Feature detection for haptic exploration with robotic fingers. *Intl. J. of Robotics Research*, 20(12):925–938.
23. Roberts, K. S. (1990). Robot active touch exploration: Constraints and strategies. In *Proc. 1990 IEEE Intl. Conf. on Robotics and Automation*, pp. 980–985.
24. Spivak, M. (1999). *A Comprehensive Introduction to Differential Geometry*, volume 3. Publish or Perish, Houston, Texas, third edition.
25. Zimmer, G. (1994). State observation by on-line minimization. *Intl. J. of Control*, 60(4):595–606.