2007

Surface patch reconstruction by touching

Jiang Tian
Iowa State University

Surface patch reconstruction by touching

by

Jiang Tian

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Yan-Bin Jia, Major Professor
David Fernndez-Baca
Greg R. Luecke

Iowa State University

Ames, Iowa

2007

DEDICATION

I would like to dedicate this thesis to my family and to my girlfriend Wenjun Li without whose support I would not have been able to complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis studies the reconstruction of unknown curved surfaces in 3D through contour tracking. The implementation involves a 2-axis joystick sensor and a 4-DOF Adept robot. The joystick's force sensing is combined with the Adept's high positional accuracy to yield precise contact measurements.

A surface patch in 3D can be rebuilt by tracking along three concurrent curves on the surface. These data curves lie in different planes and are acquired via planar contour tracking. The Darboux frame at the curve intersection is first estimated to reflect the local geometry. Then polynomial fitting is carried out in this frame. Minimization of the total (absolute) Gaussian curvature of the surface fit effectively prevents unnecessary folding otherwise expected to result from the use of touching data. Experiments have demonstrated high accuracy of reconstruction.

CHAPTER 1. INTRODUCTION AND RELATED WORK

1.1 Introduction

Reconstruction of objects with curved shapes is important in CAD/CAM. Such shapes are common in industry and in our daily life, and their visualization can be quite useful. In geometric modeling, dense 3D point data captured from the surface of an object are converted into a boundary representation CAD model (1). The first step of surface reconstruction is to use an appropriate method to capture 3D data with reasonable accuracy.

Data used to reconstruct a surface can be obtained from a camera, a range sensor, or a touch sensor. The major problem of camera data is occlusion, which makes it difficult to reconstruct a partially visible object. The occlusion problem also exists for a range sensor though it can capture 3D data in large scale. In contrast, tactile data do not have this problem because a touch sensor can reach where the camera or the range sensor cannot see.

With a touch sensor, there are several ways of acquiring geometric information about an object. One way is to uniformly sample data points over its surface. It is clear that this approach can control the surface fitting error to be within a threshold in the reconstruction stage. However, there is a major limitation on efficiency. Multiple probes are time consuming because the robot finger needs to move up and down repeatedly. Also, this does not help visualize the object model interactively at an acceptable speed. Therefore, uniform arrangement of tactile data loses the advantage of tactile data over range data.

In contrast, tracking out a few surface curves is much more efficient. Plus, it mimics moving the human finger across a surface several times to feel its shape. In (2), we proposed the use of a joystick sensor for tracking three concurrent curves on a surface and then reconstructing a local patch by fitting over these curves.

1.2 Related Work

The paper is about the reconstruction of curved surfaces in 3D. The physical operation involves tracking with a touch sensor, following strategies based on hybrid control. A surface patch is generated through fitting over the tracking data in a local frame which is set up through the estimation of principal curvatures.

1.2.1 Contour tracking

In this type of task, it is favorable to adopt the hybrid position/force control strategy proposed by Raibert and Craig (3), who combined force and torque information with position data to satisfy simultaneous position and force trajectory constraints.

Yoshikawa and Sudou (4) later incorporated on-line algorithm which estimates the local shape of the constrained surface. In this way, they made such dynamic hybrid control more practical. In (5), Im et al. assumed a frictionless surface and utilized the force/position control scheme. Their control strategy canceled the nonlinear terms of the projected end effector dynamics in the position and force controlled directions. The benefit is that the closed loop system is guaranteed to have uniformly bounded stability.

De Schutter and Van Brussel (6; 7) proposed a strategy for tracking 2D contours. In their work, orientation errors were detected and compensated based on velocities/forces of two task frames. A one-dimensional external force control loop was used for the purpose. In later research (8), for unknown environments, Baeten, Herman et al. proposed to combine visual servoing and force control.

1.2.2 Shape reconstruction

In CAD/CAM, robotics, and computer vision, polynomials are good enough to describe a lot of real objects (27). When using polynomial representation, fitting over real data is a common way for reverse engineering, i.e. shape reconstruction. For tactile data, polynomial fitting is also prevalent. In (9), Allen and Michelman use superquadric polynomial fitting to recover shape from touching data by robot hand. When data points' constraint is insufficient,

other constraints can be utilized, such as Gaussian Curvature (2). In (20), Ellis and Qin employed optimization to detect the surface shape from touching data, and they solved the optimization by Levenberg-Marquardt method.

1.2.3 Curvature Estimation

Methods for principal curvature estimation have been developed mostly for dense range data. They are not robust when applied on sparse tactile data. In (17), more than three normal curvatures were used to set up an overdetermined system to be solved for principal curvatures in a least-squares fashion. However, errors could still be significant because a circle approximation was used in estimating the normal curvature. A more robust estimation method (37) constructed a matrix at the point of interest. These two methods were later enhanced in (24) to better deal with noise.

Analytical methods (22; 25; 35) for curvature estimation generally fit over the range data in the neighborhood of the point of interest and then obtain the first and second fundamental forms through differentiation. Discrete methods based on surface triangulation such as in (29) may suffer from large estimation errors due to loss of differentiability.

Fearing and Binford (21) employed a cylindrical tactile fingertip mounted on the Stanford/JPL hand to estimate principal curvatures on quadric surfaces. In (26), two rounds of local fitting over tactile data from a 2D shape were applied to robustly estimate the curvature and its derivative with respect to arc length.

1.2.4 Geometric Modeling

Constructing a CAD model of an object typically consists of several steps (40). The first step is to capture dense 3D point data using a laser scanner and merge data from different views — a process refereed to as registration (12; 19). Then triangulation (14; 31; 10) is performed on the measured data points to reflect their adjacency relationships and the correct topology of the final model. The next step is to segment the triangulated mesh into meaningful regions (39; 18; 30). Free-form or analytic surfaces are then fit to these regions under constraints such as

smoothness, tangential continuity, and concentricity (15; 41). Such fitting applies to primary surfaces as well as feature surfaces with extrusions and of revolution. Finally, the fitting surfaces are joined together at explicit edges and vertices through intersection of adjacent surfaces (11).

CHAPTER 2. ARCHITRCTURE OF THE SYSTEM

2.1 Overview

The setup in Figure 2.1 is for tracking in the horizontal plane. To track a surface while constrained in an arbitrary plane, the tip of the joystick needs to be in contact with the object.

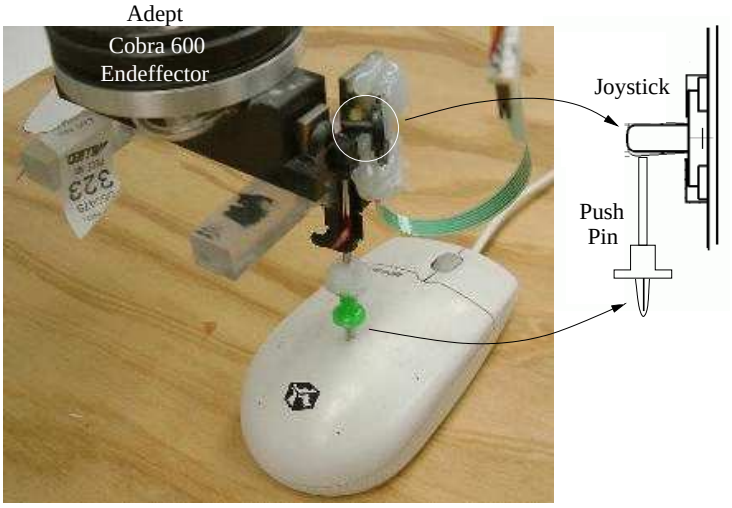


Figure 2.1 Setup for surface tracking and reconstruction.

The joystick is placed horizontal with a push pin underneath. The front end of the pin is in a plastic cover in order to easily slide on the tracked surface. In reaction to the contact force between the pin and the surface, the joystick beam bends slightly upward. The contact point is located based on the position of the robot's endeffector with some compensation for the beam bending (which is assumed to be proportional to the force reading).

We intend to use a hybrid strategy to carry out velocity control in the tangential direction at the contact while force control in the normal direction. Nevertheless, the Adept does not allow access to its joint torques, not to mention the control of them. So we implement force

control based on the readings from the 2-axis joystick. Another difficulty is that the Adept does not have the freedom of yaw and pitch to orient the push pin normal to the surface at contact. This has limited tracking mostly to the top portion of an object, where the force reading from the joystick does not deviate too much from the normal contact force.

2.1.1 Hardware and software

We build our system by a Adept Cobra robot. It has four DOFs. As far as the sensor is concerned, we use a joystick sensor from Interlink Inc. The Adept Cobra robot communicates with a host computer by TCP/IP. The joystick sensor communicates with the host computer by serial port.

2.2 Tracking from 3D to 2D

Our goal is to reconstruct a local surface patch by tracking data points from the object. Let p be a reference point on the object's. We need some local geometry description of the area surrounding the reference point. How to realize this? One solution is to sample data distributed uniformly in the interested data. But we would like to argue that this method by touch is slow. Our solution is to track along the intersections of three cutting planes through p and the surface. The intersections are three space curves, and the tracking data are discrete points along them.

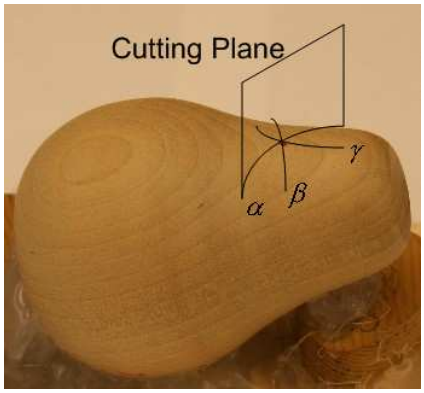


Figure 2.2 Three curves on surface.

The joystick sensor tracks the surface around the reference point p, while its motion is

constrained in a plane containing the point p. We call this the plane sampling plane. In this way, the 3D contour tracking problem is reduced to a 2D planar tracking problem. The three curves are denoted by α, β, γ.

Figure 2.2 displays a real object with three sampled curves on its surface. Denote by P_α the sampling plane that contains the curve α. We control the joystick track along the intersection curve α of P_α and the model. In this way we get one data curve. Similarly, control the sensor to track along the other two space curves β and γ through the reference point p that lie in the planes of sampling P_β and P_γ.

CHAPTER 3. SURFACE PATCH RECONSTRUCTION

3.1 Patch on a Surface

In the remainder of this thesis, we will show that local surface patches can be reconstructed from insufficient data generated by tracking. Let p be a point of interest on the surface of a curved object. We would like to obtain some geometry of a local patch which surrounds p on the object's surface. Planar contour tracking can be used as a subroutine for such reconstruction. We first let the robot to track along the intersection curves of the surface with several cutting planes through the reference point p. This is achievable by constraining the tracking motions to stay in these planes. The above tracking yields data points along multiple plane curves that are concurrent at p. The tangents at p to these curves can be estimated via local fitting. Their curvatures can also be reliably estimated. Next, we calculate the principal curvatures at p and set up a coordinate frame there. Under this frame we hope to generate a polynomial description of the surface area through controlled fitting over the curves.

3.1.1 Darboux Frame

Let the z-coordinate be represented by a polynomial function of degree d:

$$z(x, y) = \sum_{0 \le i+j \le d} a_{ij} x^i y^j.$$

That $z(0, 0) = 0$ immediately implies $a_{00} = 0$.

Next, we align the x-y plane with the tangent plane to the surface at p. Choose the direction of the z-axis to be along its outward normal. The two partial derivatives of σ are tangent to the surface and therefore lie in the x-y plane:

$$\sigma_x|_{(0,0,0)} = (1, 0, z_x)|_{(0,0,0)} = (1, 0, a_{10});$$

$$\sigma_y|_{(0,0,0)} \;=\; (0,1,z_y)|_{(0,0,0)} \;=\; (0,1,a_{01}).$$

The unit surface normal $N = (0,0,1)$ at p is parallel to the cross product $\sigma_x \times \sigma_y = (-a_{10}, -a_{01}, 1)$. Thus the coefficients a_{10} and a_{01} are both zero.

Any plane through p that is normal to the tangent plane will intersect the surface σ (at least locally) at a curve. The curvature κ_n of this curve at p is the *normal curvature* of σ in the tangent direction where the two planes intersect. It is well-known that κ_n achieves its minimum and maximum (in case they are not equal) in two orthogonal tangent directions. These two extrema are the *principal curvatures* κ_1 and κ_2, and the corresponding tangent directions are the *principal directions* d_1 and d_2.

We now choose the principal directions as the x- and y-axes so they are represented by $(1,0,0)$ and $(0,1,0)$, respectively. Figure 3.1 shows the resulting local coordinate system referred to as the *Darboux frame*.

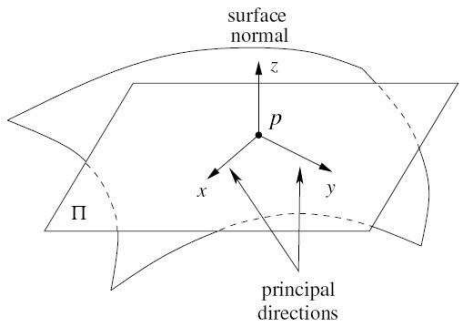


Figure 3.1 Darboux frame at p.

Express the two principal directions in terms of the partial derivatives:

$$d_1 = (1,0,0) = 1 \cdot \sigma_x|_{(0,0,0)} + 0 \cdot \sigma_y|_{(0,0,0)}$$

$$d_2 = (0,1,0) = 0 \cdot \sigma_x|_{(0,0,0)} + 1 \cdot \sigma_y|_{(0,0,0)}.$$

Gather the coefficients of σ_x and σ_y in the above two equations respectively into two vectors $v_1 = (1,0)$ and $v_2 = (0,1)$.

Now, we calculate the coefficients in the first fundamental form at $p = (0, 0, 0)$:

$$E = \sigma_x \cdot \sigma_x|_{(0,0,0)} = 1,$$

$$F = \sigma_x \cdot \sigma_y|_{(0,0,0)} = 0,$$

$$G = \sigma_y \cdot \sigma_y|_{(0,0,0)} = 1.$$

We also obtain the coefficients in the second fundamental form:

$$\begin{aligned} L &= \sigma_{xx} \cdot N = (0, 0, z_{xx}) \cdot (0, 0, 1) \\ &= z_{xx}|_{(0,0,0)} = 2a_{20}; \\ M &= z_{xy}|_{(0,0,0)} = a_{11}; \\ N &= z_{yy}|_{(0,0,0)} = 2a_{02}. \end{aligned}$$

It follows from differential geometry that

$$v_i \begin{pmatrix} L & M \\ M & N \end{pmatrix} v_j^T = \kappa_i v_i \begin{pmatrix} E & F \\ F & G \end{pmatrix} v_j^T = \begin{cases} \kappa_i & \text{if } (i-j) = 0; \\ 0 & \text{if } (i-j) \neq 0. \end{cases}$$

The product $K = \kappa_1 \kappa_2$ is the *Gaussian* curvature at p. Substitute the expressions for E, F, G, L, M, N, v_1, v_2 into the above equations. We thus determine three more coefficients in the function $z(x, y)$:

$$a_{20} = \frac{\kappa_1}{2}, \quad a_{02} = \frac{\kappa_2}{2}, \quad \text{and} \quad a_{11} = 0.$$

To summarize what we have done so far, in the Darboux frame at p, the local surface patch is described as below:

$$z(x, y) = \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right) + \sum_{d \geq i+j \geq 3} a_{ij} x^i y^j. \tag{3.1}$$

To reconstruct the patch surrounding p, we intend to fit the form (3.1) over data points obtained by a touch sensor. But how to estimate the principal curvatures and locate the principal directions?

3.1.2 Principal Curvatures Estimation

We use a touch sensor to track the surface around the reference point p while constraining the sensor motion in a plane through the point, which we call the *sampling plane*. The tracking data are discrete points along the intersection curve α of the sampling plane and the surface. The description of α is, of course, unknown just like the shape. For convenience, we identify these data points with the curve and called them the *data curve*. Denote the sampling plane as Π_α (see Figure 3.2).

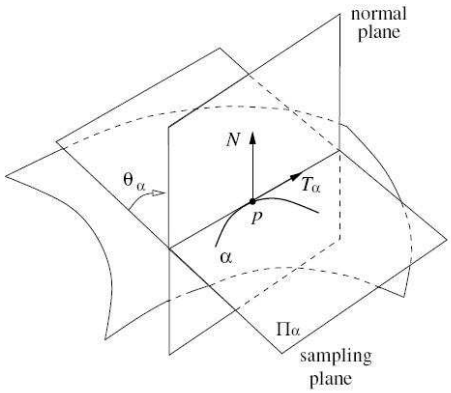


Figure 3.2 Data curve α lies where the sampling plane Π_α intersects the surface.

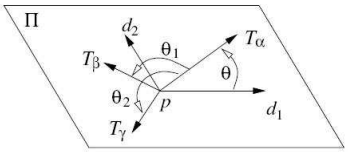


Figure 3.3 Tangent plane at the reference point p of sampling.

Then we fit over those data points very close to p and estimate the tangent T_α and the curvature κ_α at the point. A quadratic polynomial is used in the fitting because locally the curve resembles the osculating circle of α at p. The curvature (along with its derivative) can be estimated very accurately this way, as shown in (26).

Similarly, command the sensor to trace out two other surface curves β and γ through p that lie in different sampling planes Π_β and Π_γ, respectively. We also estimate the tangents

T_β and T_γ as well as the curvatures κ_β and κ_γ.

The surface normal N at p must be orthogonal to these tangent vectors. We estimate it through an optimization:

$$\min_{\|N\|=1} (N \cdot T_\alpha)^2 + (N \cdot T_\beta)^2 + (N \cdot T_\gamma)^2.$$

With N known, the tangent plane Π at p is determined. The vectors T_α and N define a normal plane through p. Let θ_α be the angle between this normal plane and the sampling plane Π_α. By a result from differential geometry (34, pp. 127-128), the normal curvature in the direction T_α is $\kappa'_\alpha = \kappa_\alpha \cos\theta_\alpha$. Similarly, we obtain the normal curvatures κ'_β and κ'_γ in the directions T_β and T_γ, respectively.

Let us now look at the tangent plane Π as shown in Figure 3.3.

Let θ be the angle from the principal direction d_1 to T_α, and θ_1 and θ_2 be the angles from T_α to T_β and T_γ, respectively. We need only consider $\theta \in [0, \frac{\pi}{2}]$ by a proper choice of d_1 (out of four possibilities). The angles θ_1 and θ_2 are easily determined from the tangents. The normal curvatures are then expressed in terms of the two principal curvatures (34, p. 137):

$$\begin{aligned} \kappa'_\alpha &= \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta, \\ \kappa'_\beta &= \kappa_1 \cos^2(\theta+\theta_1) + \kappa_2 \sin^2(\theta+\theta_1), \\ \kappa'_\gamma &= \kappa_1 \cos^2(\theta+\theta_2) + \kappa_2 \sin^2(\theta+\theta_2). \end{aligned} \tag{3.2}$$

There are three unknowns κ_1, κ_2, and θ in the above three equations.

Rewrite equations (3.2) into the following:

$$\begin{aligned} \kappa'_\alpha &= \frac{\kappa_1+\kappa_2}{2} + \cos(2\theta) \cdot \frac{\kappa_1-\kappa_2}{2}, \\ \kappa'_\beta &= \frac{\kappa_1+\kappa_2}{2} + \cos(2\theta+2\theta_1) \cdot \frac{\kappa_1-\kappa_2}{2}, \\ \kappa'_\gamma &= \frac{\kappa_1+\kappa_2}{2} + \cos(2\theta+2\theta_2) \cdot \frac{\kappa_1-\kappa_2}{2}. \end{aligned} \tag{3.3}$$

In the special case that $\kappa'_\alpha = \kappa'_\beta = \kappa'_\gamma$, two possibilities further arise.

(a) $\kappa_1 = \kappa_2$ So the reference point p is umbilic with constant normal curvature. Every direction in the tangent plane is a principal direction. We simply choose two orthogonal directions as d_1 and d_2.

(b) $\kappa_1 \neq \kappa_2$ So $\cos(2\theta) = \cos(2\theta + 2\theta_1) = \cos(2\theta + 2\theta_2)$, in which case $\theta_1, \theta_2 = \pi - 2\theta$ or $2\pi - 2\theta$. No unique solutions of κ_1 and κ_2 exist as the three equations in (3.3) are essentially one.

Situation (b) can be avoided by choosing sampling planes that violate the implicit relationship between θ_1 and θ_2.

The general case is when κ'_α, κ'_β, and κ'_γ are not all equal. Then one of the curvatures must be different from the other two. Assume that it is κ'_α. From (3.3) we have the following:

$$\begin{aligned} \kappa'_\alpha - \kappa'_\beta &= \Big(\cos(2\theta) - \cos(2\theta + 2\theta_1)\Big) \cdot \frac{\kappa_1 - \kappa_2}{2}, \\ \kappa'_\alpha - \kappa'_\gamma &= \Big(\cos(2\theta) - \cos(2\theta + 2\theta_2)\Big) \cdot \frac{\kappa_1 - \kappa_2}{2}. \end{aligned}$$

Divide both sides of the first equation above by those of the second:

$$\begin{aligned} \frac{\kappa'_\alpha - \kappa'_\beta}{\kappa'_\alpha - \kappa'_\gamma} &= \frac{\cos(2\theta) - \cos(2\theta + 2\theta_1)}{\cos(2\theta) - \cos(2\theta + 2\theta_2)} \\ &= \frac{\sin(2\theta + \theta_1)}{\sin(2\theta + \theta_2)} \cdot \frac{\sin\theta_1}{\sin\theta_2} \\ &= \frac{\sin\Big((2\theta + \theta_2) + (\theta_1 - \theta_2)\Big)}{\sin(2\theta + \theta_2)} \cdot \frac{\sin\theta_1}{\sin\theta_2}. \end{aligned}$$

From the last equation above we obtain

$$\tan(2\theta + \theta_2) = \frac{\sin(\theta_1 - \theta_2)}{\dfrac{\kappa'_\alpha - \kappa'_\beta}{\kappa'_\alpha - \kappa'_\gamma} \cdot \dfrac{\sin\theta_2}{\sin\theta_1} - \cos(\theta_1 - \theta_2)}.$$

So we obtain θ and the principal directions. The Darboux frame is thus determined. Substituting θ, θ_1, and θ_2 into the system (3.2) of linear equations, we can solve for the principal curvatures κ_1 and κ_2.

Subsequently, we transform the coordinates of all data points to the Darboux frame at p determined by d_1, d_2, and N.

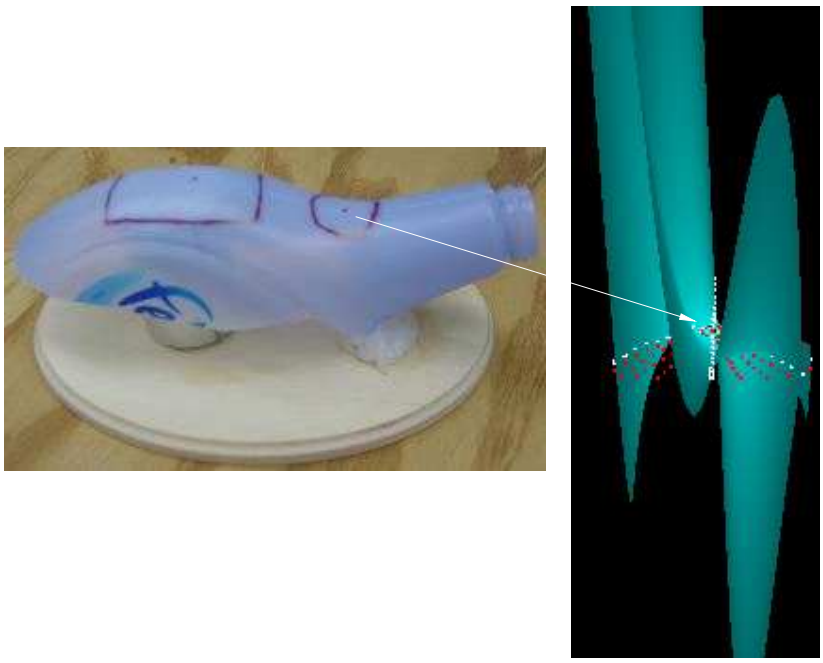
3.2 Surface Patch Reconstruction

The three data curves α, β, γ together serve as a "skeleton" for the patch to be reconstructed. In the newly estimated Darboux frame at p, we fit the z-coordinate polynomial (3.1)

over all the data points (x_i, y_i, z_i), $1 \le i \le n$, sampled along α, β, and γ. The degree of the polynomial is set to be $d = 4$. Quartic polynomials are capable of describing a wide range of real objects so that several subclasses are commonly used as shape models in computer vision and robotics. Write $a = (a_{30}, a_{21}, \ldots, a_{04})$ to include the nine polynomial coefficients. They are determined in a least-squares sense:

$$\min_a f(a) \quad \text{where} \quad f(a) = \sum_{k=1}^{n} (z(x_k, y_k) - z_k)^2. \tag{3.4}$$

The term $|z(x_k, y_k) - z_k|$ is an upper bound on the distance from the data point (x_k, y_k, z_k) to the patch (3.1) defined by a. The function $f(a)$ thus bounds the total squared distance from the data points to the patch.



(a) (b)

Figure 3.4 A broken plastic bottle and its reconstructed neck region.

We have sampled 57 points (white) along 3 concurrent curves, displayed in (b) in the figure, inside the marked neck region. These curves are concurrent at the reference point (marked inside the region), which is parabolic with negative Gaussian curvature. The principal curvatures are estimated as -0.0424 and 0.0172, respectively. The neck region of the bottle is reconstructed in the picture (b) according to (3.4).

We measure the average fitting error as

$$e = \frac{1}{n}\sum_{k=1}^{n}\Big|z(x_k, y_k) - z_k\Big|. \tag{3.5}$$

Although the error is small (0.0459mm) over the $n = 57$ data points, the reconstructed patch does not nearly resemble the bottle's neck region. It has "peaks" and "valleys". To confirm this quantitatively, we use 171 points (black) sampled along 9 extra curves through the same reference point. The average error (3.5) over these points then rises up dramatically to 13.0042mm.

3.2.1 Constraining the Fitting Surface

Naturally, we would like the reconstructed patch to look "smooth". That is, it should not have any "peaks", "valleys", or folds unless induced by the three data curves. This patch is assumed to be a "local" one where any drastic change of geometry between the three data curves is not expected. From the bottle example it is apparent that three data curves do not provide enough constraints on fitting. The objective function in (3.4) for fitting needs to include a term that can measure the "degree of folding" of the surface fit.

There seem to be at least three approaches to dealing with this issue. The first is to derive more information from the data curves to control the behavior of surface fitting. We could use directional derivatives estimated at all points along the data curves. The rationale is that the Monge patch (3.1) incorporates the tangent (first order) and curvature (second order) information at the reference point, but only the positions (zeroth order) of the other points are taken into account in the fitting (3.4). The second approach is to generate artificial data points by, say, interpolation, between the data curves.

After many trials, we have found that neither of these two approaches nor their combination can generate satisfactory shapes when verified against extra real data. Directional derivatives simply could not constrain the area of a surface fit between the data curves. Interpolation, on the other hand, tends to shape the surface fit with a bias imposed by the interpolation scheme itself. The areas between the three data curves are just too large for interpolation to work properly.

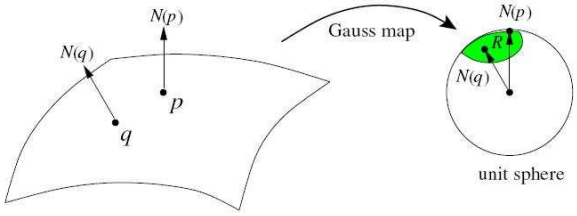


Figure 3.5 Gauss map from a patch to a unit sphere.

The third approach makes use of the *total Gaussian curvature*. It is the integral of the Gaussian curvature over a surface patch: $\int\int_\sigma K\,dA$. Geometrically, every point q on the patch maps to some point on the unit sphere that represents the unit surface normal $N(q)$. This is called the *Gauss map* as illustrated in Figure 3.5. All the image points constitute a shaded region R on the unit sphere, whose algebraic area is the total Gaussian curvature of the patch (33, p. 290).

The flatter the patch, the closer the total Gaussian curvature approaches zero. In the special case of a planar patch, the total Gaussian curvature becomes zero. The converse, however, is not necessarily true because Gaussian curvature may change its sign over the patch. When the sign varies, the Gauss map may fold the patch many times over the region R on the unit sphere. The total Gaussian curvature thus cannot tell how much the surface patch folds. To measure the "degree" of folding over R, we integrate the absolute Gaussian curvature over the patch.

Let D be the domain of the surface fit in the tangent plane at the reference point p (i.e., the xy-plane). The *total absolute Gaussian curvature* is defined as

$$\int\int_D |K(x,y)| \cdot \sqrt{1+z_x^2+z_y^2}\,dxdy. \tag{3.6}$$

The more the patch folds, the larger the integral. For example, the reconstructed patch in Figure 3.4 has total absolute Gaussian curvature 16.3086, which is too big for a small region on the bottle.

Given the values of the coefficient vector a of the polynomial fit (3.1), we evaluate the total absolute Gaussian curvature (3.6) numerically. For ease of computation, the patch domain D is chosen to be the convex hull of the projections of α, β, and γ onto the xy-plane. Discretize

the domain D into a grid of m points $(u_1, v_1), \ldots, (u_m, v_m)$ with uniform spacing h. The integral (3.6) is approximated by

$$g(a) = h^2 \cdot \sum_{j=1}^{m} \left(|K(u_j, v_j)| \cdot \sqrt{1 + z_x^2(u_j, v_j) + z_y^2(u_j, v_j)} \right). \tag{3.7}$$

Now, patch reconstruction over the n data points is done through optimizing a new objective function that incorporates the total absolute Gaussian curvature of the surface fit:

$$\min_a f(a) + \lambda n g(a). \tag{3.8}$$

This is equivalent to minimizing $\frac{1}{n} f(a) + \lambda g(a)$ except numerically more stable. The choice of the multiplier λ is independent of n. The first term in (3.8) constrains the patch to lie very close to the sampled points, while the second term prevents it from changing dramatically between the data curves. By a proper choice of λ, the resulting patch will be spanned "smoothly" by the three data curves.

3.2.2 Minimization

The nonlinear optimization (3.8) is carried out using the steepest descent method along the negative gradient of the objective function. Note that $g(a)$ depends on the signs of the Gaussian curvatures $K(u_j, v_j)$, $1 \le j \le m$, which may vary from the current coefficient estimate $a^{(l)}$ to the next one $a^{(l+1)}$. To cope with this issue, at $a^{(l)}$ for every grid point (u_j, v_j) we define

$$\delta_j = \begin{cases} 1 & \text{if } K(u_j, v_j) \ge 0; \\ -1 & \text{if } K(u_j, v_j) < 0. \end{cases}$$

Replacing $|K(u_j, v_j)|$ with $\delta_j K(u_j, v_j)$ in the definition (3.7) of $g(a)$ yields an equivalent function $\bar{g}(a)$ in some neighborhood of $a^{(l)}$ in the coefficient space. Performing the steepest descent along the negative gradient $-\nabla(f(a) + \lambda n \bar{g}(a))$ yields the next estimate $a^{(l+1)}$. If the signs of all Gaussian curvatures $K(u_j, v_j)$, $1 \le j \le m$, do not vary from $a^{(l)}$ to $a^{(l+1)}$, we have found a (local) minimum of (3.8). Otherwise, iteration continues. The minimization procedure is

detailed below.

1 initialize the coefficients $a^{(0)}$ of the polynomial (3.1)
2 $l \leftarrow 0$
3 **repeat**
4 **for** $j = 1$ **to** m
5 **if** $K(u_j, v_j) \geq 0$ **then** $\delta_j \leftarrow 1$
6 **else** $\delta_j \leftarrow -1$
7 $v \leftarrow -\nabla\Big(f(a) + \lambda n\bar{g}(a)\Big)$
8 $a^{(l+1)} \leftarrow$ steepest descent from $a^{(l)}$ along v
9 $l \leftarrow l + 1$
10 **until** for all $1 \leq j \leq m$ the sign of $K(u_j, v_j)$ under $a^{(l)}$
stays the same as that under $a^{(l-1)}$

The initial value of the coefficients, $a^{(0)}$, is obtained by unconstrained fitting (3.4) over the data points and some artificial points generated through linear interpolation. These artificial points are not used in the following optimization steps.

We use the algorithm to reconstruct the neck region of the plastic bottle in Figure 3.4. The multiplier λ is set to be 0.3. The result is shown in Figure 3.6.

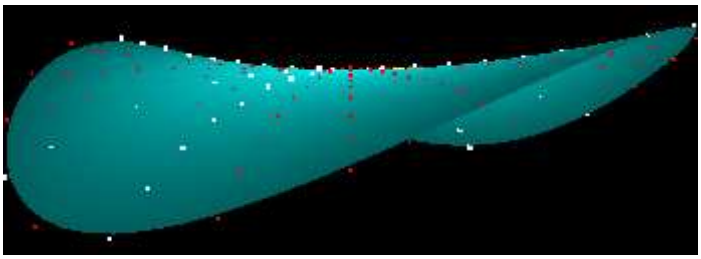


Figure 3.6 New patch reconstructed over the same neck region.

The same 9 data curves (shown in dark dots) from Figure 3.4 are used for verification purpose. The reconstructed patches in the two figures are compared in the first two rows of Table 3.1. By incorporating the total absolute Gaussian curvature, the average fitting error over these extra curves has reduced dramatically from 13.0042mm to 0.0826mm. The slight

reconstructed patch	principal curvatures (1/mm)	total absolute Gaussian curvature	average fitting error (3.5) (mm)	
			3 data curves (57 points)	9 verify. curves (171 points)
Figure 3.4 (b)	−0.0424, 0.0172	16.3026	0.0459	13.0042
Figure 3.6	same as above	0.0083	0.0838	0.0826
other	−0.0463, 0.0179	0.0026	0.1078	0.1574

Table 3.1 Statistics of the two reconstructed patches in Figures 3.4 and 3.6, respectively, and of a third patch derived over different data curves sampled from the same surface area.

increase in the average error over the 3 original curves is expected because of the extra term $\lambda ng(a)$ in (3.8) to minimize.

To test the robustness of the constrained fitting scheme (3.8), we also arbitrarily pick 3 out of the 9 extra curves for patch reconstruction. And the result is shown in the third row of Table 3.1. The last entry in this row is determined over points along the remaining 9 curves that are not used in the new fitting. The resulting patch is not shown because it has hardly any visual difference from Figure 3.6.

3.3 Experiments

Table 3.2 shows patch reconstruction results over six different shapes.

The first two columns of each row in the table display an object and its corresponding reconstructed patch. The third column lists the principal curvatures κ_1 and κ_2 at the marked reference points and the total absolute Gaussian curvatures (t.a.G.c.) of the reconstructed patches. The last two columns give the fitting errors defined in (3.5). The reference point is marked on the object, so is the region that contains the sampled data curves and (roughly) corresponds to the reconstructed patch. As before, the original data consists of points along three curves concurrent at the reference point. The extra data for verification consist of points along three other curves also through the same point.

The reconstructed patches have undergone some rotations for better display. The third column of each row includes the two estimated principle curvatures κ_1 and κ_2 at the reference

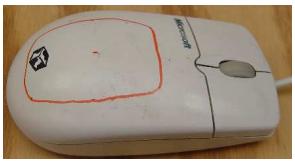
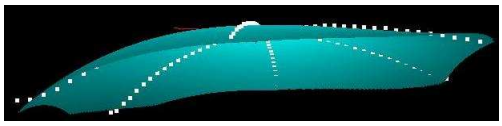
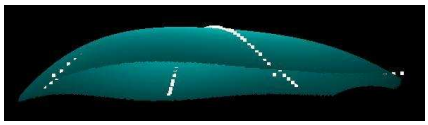
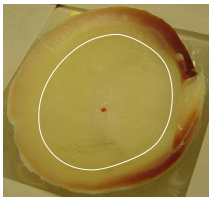
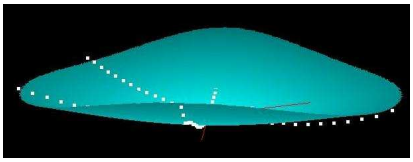
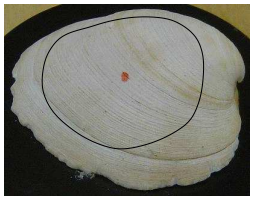
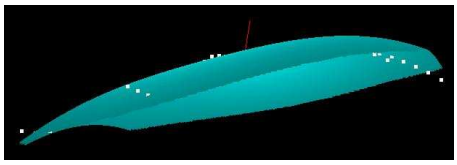
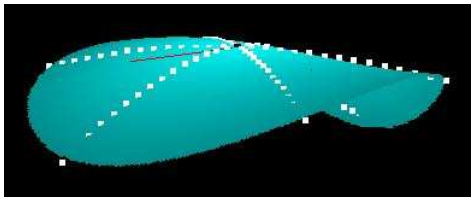
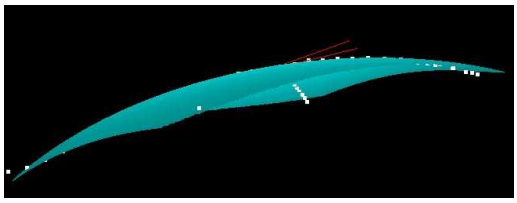
shape	reconstructed surface patch	κ_1, κ_2 (1/mm), t.a.G.c.	fit. error (mm)	
			original data	extra data
		0.0166 0.0231 0.1005	0.2165	0.1196
		0.0242 0.0074 0.1004	0.3392	0.3284
		−0.0208 −0.0395 0.2702	0.3801	0.5010
		0.0305 0.0077 0.0523	0.3815	0.2989
		0.0496 0.0079 0.2191	0.2271	0.2812
		0.0189 0.0138 0.0087	0.1633	0.2819

Table 3.2 Surface patches reconstructed over the marked areas on a mouse, a coconut, two shells, and two pebbles, respectively.

point, and the total absolute Gaussian curvature of the patch. Principal axes at the reference points are marked on the reconstructed patches. They are recognizable on those of the two shells and the two pebbles.

3.4 Validations

We first test the accuracy of the sampled data on several freeform objects. First of all, we sampled data points along three concurrent curves on the surface of each object, whose model has been generated by NextEngine's desktop scanner for the purpose of comparison.

Using the method from [5], the intersection point of the three data curves is located from the 3D model of the freeform object and the curves are registered onto the surface of the model. So we can display these three curves on the 3D mesh model of each object as shown in Figure 3.7.

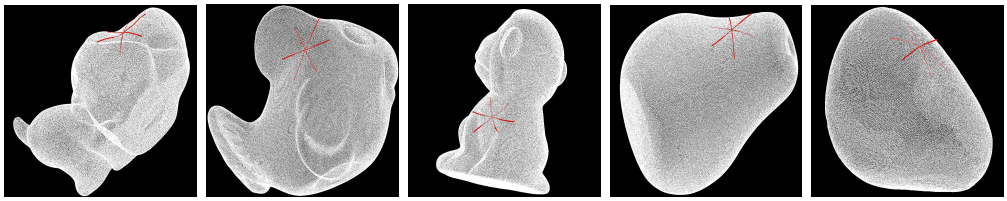


Figure 3.7 Five shape models, a dog, a fish, a monkey, a pear, a stone. Data points sampled along three concurrent curves on each model surface.

3.4.1 Accuracy of Reconstruction

We have tested the accuracy of the system by superposing the three data curves obtained from each object onto its mesh model. There is a corresponding point on the mesh that coincides with the curve intersection point through which the two tangent planes, one to the three curves and the other to the mesh model, are aligned. Carry out the following steps:

1. Use the common point as the origin. Use the aligned tangent planes as the x-y plane. Pick two orthogonal tangent directions as the x- and y-axes. As shown in Figure 3.8, we select the two principal directions in Darboux frame as the x- and y-axes.

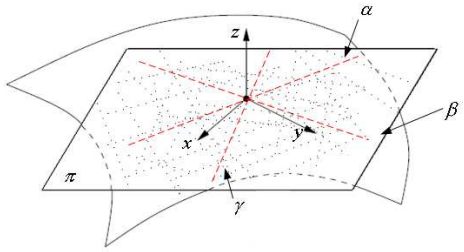


Figure 3.8 Projection of mesh model vertices and sampled curves data points onto the tangent plane.

2. Transform all mesh vertices into this Darboux frame (black dots as shown in Figure 3.8). The mesh vertices are initially in the global coordinates, we should transform them into the Darboux frame. This is realized by dot product of space vectors. Project the three curves onto the x-y plane (red lines as shown in Figure 3.8) to determine its correspondent vertices. For extracting the mesh vertices that correspond to some points on the reconstructed patch, we compute the convex hull of the line segments that are the curve projections, and choose all mesh vertices that project within the convex hull.

3. For every extracted vertex (x, y, z), evaluate the difference between its z coordinate and the corresponding data point's z coordinate in the curve. This is the error for the sampled data points on three curves.

Table 3.3 shows the calculated average errors of the sample data points from five objects.

model name	*averaged error* (mm)
Pear	0.153
Stone	0.169
Monkey	0.367
Fish	0.197
Dog	0.171

Table 3.3 Calculated error

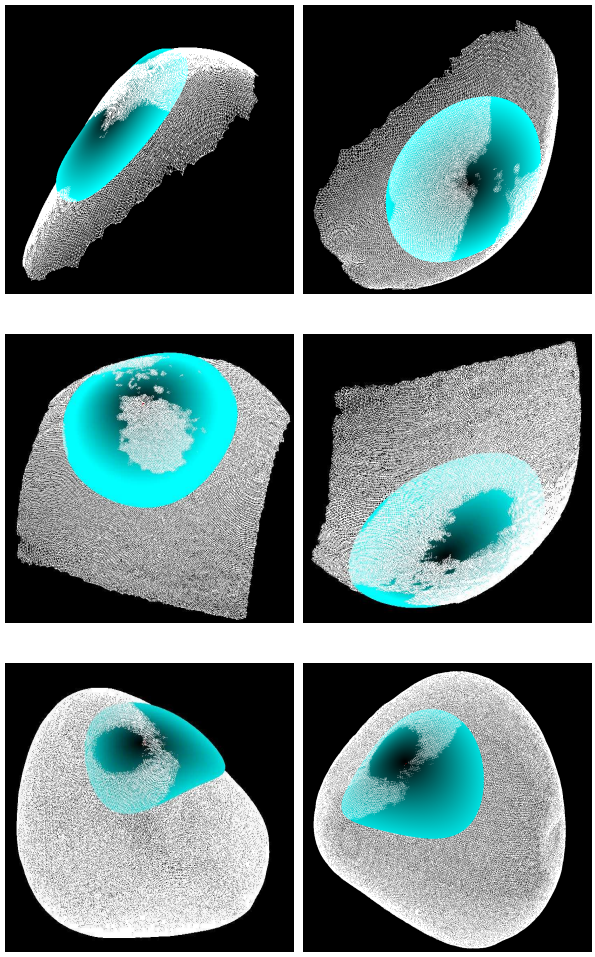


Figure 3.9 Three shape models, a pebble, a fruit, a stone. Two view of the reconstructed patch rendered on each model surface.

3.4.2 Local patch reconstruction results

Fig. 9 displays three reconstructed surface patches rendered on the corresponding models. Here, minimization of the total (absolute) Gaussian curvature of the surface fit effectively prevents unnecessary folding (2). Each row of the figure displays two views of a patch superposed on the corresponding model. We flip view one to get view two in order to show the good match of the reconstructed surface patch on the model.

CHAPTER 4. DISCUSSION AND FUTURE RESEARCH

This thesis investigates how to reconstruct curved shapes in 3D from contours generated by tracking with a 2-axis force sensor — a joystick. The joystick is mounted on a 4-DOF Adept Cobra 600 manipulator whose built-in position control is utilized to yield high accuracy data. Since the Adept robot does not allow direct access to its joint torques, force control is implemented with the joystick sensing.

The joystick as a touch sensor generates a sequence of points as it moves along the surface of an object. While the yielded data curves seem too sparse for building surfaces in 3D. We deal with this issue of insufficient data by choosing an arrangement of tracking trajectories, and applying some basic techniques in differential geometry. More specifically, the data curves intersect at a point around which the description of a local patch is simplified. The total absolute Gaussian curvature provides a solution to reconstruction over insufficient data, at least locally, as we have demonstrated. Its minimization while data fitting has effectively prevented unnecessary folding of the resulting patch.

The data curves on the surface do not need to be planar for the reconstruction purpose. But for space curves, curvature estimation is less reliable, which could affect the estimation of the principal curvatures and the Darboux frame at the reference point.

Probing a surface at a grid of points could lead to a more robust fitting result than tracking along three concurrent curves due to the better data arrangement in the former strategy. However, multiple probes are time consuming because the robot needs to move up and down repeatedly. In contrast, tracking out a few surface curves is much more efficient. Plus, it resembles moving the human finger across a surface several times to feel its shape.

The reconstruction procedure is influenced by the accuracies in the estimated principal

curvatures and directions at the reference point, which are in turn affected by the normal curvature estimates. Sensing errors occur with the sampled data points. Control errors may cause the tracking pin to slightly deviate from the sampling plane. However, the final fitting over all the data points is expected to reduce the influence of such sensing and control errors.

The objective of our work in surfaces constructed over range data are subjected to camera occlusion and sometimes lack in accuracy in practice. A situation may arise where very fine details about certain parts of a shape are needed. Tracking with a touch sensor controlled by a high precision robot poses a good solution.

4.1 Current improvement on Tracking

Due to its lack of two degrees of freedom, the Adept robot cannot orient the tracking tool normal to a surface everywhere. Thus it is unable to track in an area with steep slope or facing downward. We have most recently purchased a robotic hand from Barrett Technology, Inc. with three fingers and 4 DOFs. We are currently building our system by a Adept Cobra robot and a Barrett Hand. Both Adept Cobra robot and Barrett Hand have four DOFs. We mounted Barrett Hand on our Adept robot. Thus, we have eight DOFs in total. As far as the sensor is concerned, we use a joystick sensor from Interlink Inc. We mounted the sensor on one of the Barrett's finger. As shown in Figure 4.1, since we want to read in force sensitively, we use a lever to amplify the force. The Adept Cobra robot communicates with a host computer by TCP/IP. At the same time, the Barrett Hand communicates with the same host computer by Serial port. Because we have already used the serial port to communicate the joystick sensor, therefore we use a serial to USB adapter for the Barrett Hand. The joystick sensor communicates with the host computer by serial port.

For the software part, we basically have two main programs. Program one controls the Adept Cobra robot and joystick sensor, program two controls the Barrett hand. The communication between these two programs is by TCP/IP. After read in the force reading form the joystick sensor, program one processes the force reading and communicates with program two to adjust position and orientation of the whole system.

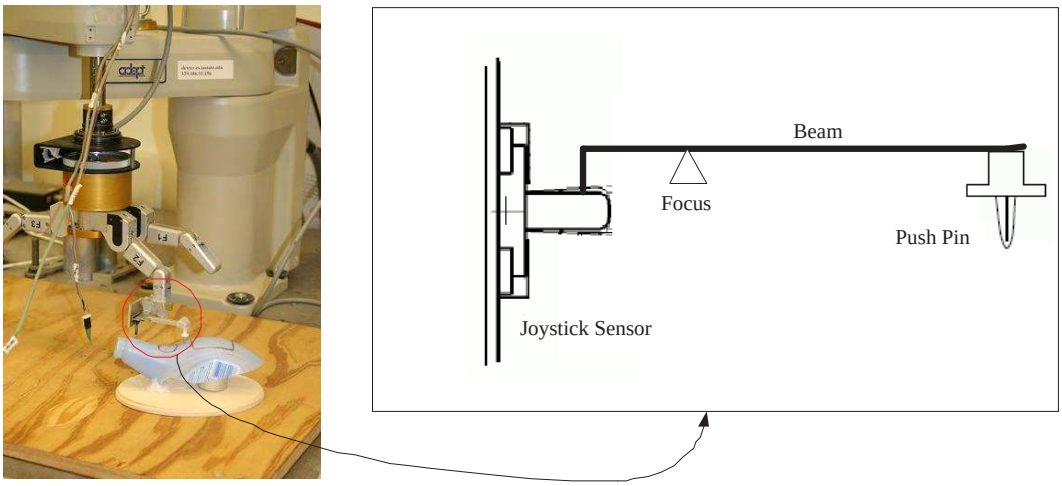


Figure 4.1 System architecture.

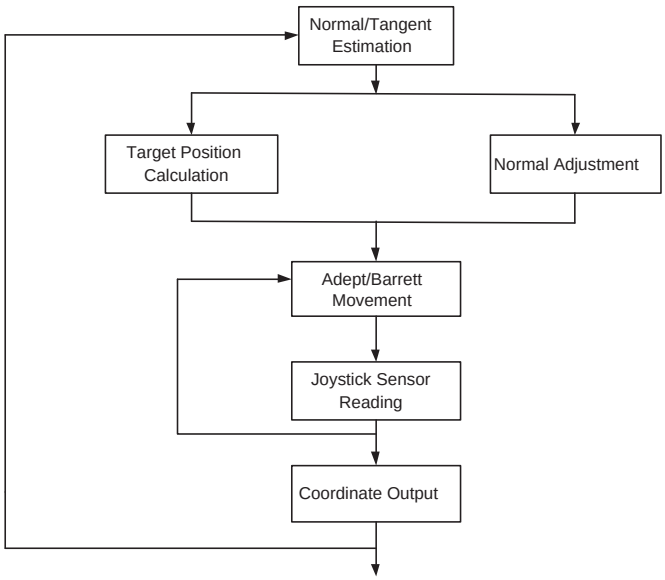


Figure 4.2 System flow diagram.

We are currently working on the system as shown in Figure 4.2, in the system, we begins from the tangent estimation. Then the target position was calculated. This signal was sent to the Adept robot and Barrett hand. They will do the required action. The force reading was loaded in from the joystick sensor, the position controller will calculate the position adjustment. The system iterates this procedure in the process of data sampling.

4.2 From Patches to Surfaces

In geometric modeling, a CAD model usually joins together multiple surfaces which reflect the correct topology. We can also view a complicated surface as a collection of local patches. Every patch is described in the Darboux frame at a reference point in its interior. In the future research, we will merge patches into a whole model. This will allow us to visualize the object while interacting with it.

BIBLIOGRAPHY

[1] Tamas, V., and Ralph, M. (2002). Reverse engineering. *Chapter 26, Handbook of Computer Aided Geometric Design, North-Holland.*

[2] Jia, Y., Mi, L., and Tian, J. (2006). Surface patch reconstruction via curve sampling. *The IEEE International Conference on Robotics and Automation, Orlando, FL.*

[3] Raibert, M.H. and Craig, J.J. (1981). Hybrid Position/Force Control of Manipulators. *Journal of Dynamics Systems, Measurements, and Control, Transaction of the ASME, Vol. 103, no. 2,*

[4] Yoshikawa, T., and Sudou, A. (1993). Dynamic Hybrid Position/Force Control of Robot Manipulators: On-Line Estimation of Unknown Constraint. *IEEE Transactions on Robotics and Automation, Volume 9, Issue 2, Page(s):220 - 226*

[5] Kyumann Im, Sacheul Jeong, Jeonghyu Lee, Woonchul Ham. (1996). Force/position control of robot manipulator via motion dynamics. *Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*

[6] J. De Schutter and H. Van Brussel. (1988 II). Compliant Robot Motion II. A Control Approach Based on External Control Loops. *The International Journal of Robotics Research, Vol. 7, No. 4*

[7] J. De Schutter and H. Van Brussel. (1988 II). Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks. *The International Journal of Robotics Research, Vol. 7, No. 4*

[8] Johan Baeten, Herman Bruyninckx, and J. De Schutter. (2003). Integrated Vision/Force Robotic Servoing in the Task Frame Formalism. *The International Journal of Robotics Research, Vol. 22, No. 10-11, 941-954*

[9] Peter K. Allen and Paul Michelman. Acquisition and interpretation of 3-D sensor data from touch. *IEEE Transactions on Robotics and Automation*, 6(4):397–404, 1990.

[10] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Computer Graphics Proceedings, SIGGRAPH*, pages 415–421, 1998.

[11] P. Benkö, R. R. Martin, and T. Várady. Algorithms for reverse engineering boundary rpresentation models. *Computer-Aided Design*, 33(11):839–851, 2001.

[12] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, 1996.

[13] Michael M. Blane, Zhibin Lei, Hakan ¸Civi, and David B. Cooper. The 3*l* algorithm for fitting implicit polynomial curves and surfaces to data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):298–313, 2000.

[14] Jean-Daniel Boissonnat. Geometric structure for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, 1984.

[15] R. M. Bolle and B. C. Vemuri. On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):1–13, 1991.

[16] M. Charlebois, K. Gupta, and S. Payandeh. Shape description of curved surfaces from contact sensing. *International Journal of Robotics Research*, 18(8):779–787, 1999.

[17] X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In *Proceedings of the European Conference on Computer Vision*, pages 739–743, 1992.

[18] Csákány and A. M. Wallace. Computation of local differential parameters on irregular meshes. In R. Cipolla and R. Martin, editors, *The Mathematics of Surfaces IX*, pages 19–33. Springer-Verlag, 2000.

[19] C. Dorai, G. Wang, A. K. Jain, and C. Mercer. Registration and integration of multiple views fro 3d model construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):83–89, 1998.

[20] R. E. Ellis and M. Qin. Singular-value and finite-element analysis of tactile shape recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2529–2535, 1994.

[21] R. S. Fearing and T. O. Binford. Using a cylindrical tactile sensor for determining curvature. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 765–771, 1988.

[22] P. J. Flynn and A. K. Jain. On reliable curvature estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 110–116, 1989.

[23] M. A. Halstead, B. A. Barsky, S. A. Klein, and R. B. Mandell. Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. In *Computer Graphics Proceedings, SIGGRAPH*, pages 335–341, 1996.

[24] Eyal Hameiri and Ilan Shimshoni. Estimating the principal curvatures and the Darboux frame from real 3-D range data. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(4):626–637, 2003.

[25] R. L. Hoffman and A. K. Jain. Segmentation and classification of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):608–620, 1987.

[26] Rinat Ibrayev and Yan-Bin Jia. Semi-differential invariants for tactile recognition of algebraic curves. *International Journal of Robotics Research*, 24(11):951–969, 2005.

[27] Daniel Keren, David Cooper, and Jayashree Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–53, 1994.

[28] Daniel Keren and Criag Gotsman. Fitting curves and surfaces with constrained implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):31–41, 1999.

[29] C. Lin and M. J. Perry. Shape description using surface triangulation. In *Proceedings of IEEE Workshop on Computer Vision*, pages 38–43, 1982.

[30] A. D. Marshall, G. Lukács, and R. R. Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):304–314, 2001.

[31] David Meyers and Shelley Skinner. Surfaces from contours. *ACM Transactions on Graphics*, 11(3):228–258, 1992.

[32] Mark Moll and Michael A. Erdmann. Reconstructing the shape and motion of unknown objects with active tactile sensors. In J.-D. Boissonnat et al., editor, *Algorithmic Foundations of Robotics V*, pages 293–309. Springer-Verlag, 2004.

[33] Barrett O'Neill. *Elementary Differential Geometry.* Academic Press, Inc., 1966.

[34] Andrew Pressley. *Elementary Differential Geometry.* Springer-Verlag, 2001.

[35] E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-D objects: five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):833–840, 1992.

[36] S. Sullivan, L.Sandford, and J. Ponce. Using geometric distance fits for 3-D object modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1183–1196, 1994.

[37] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 902–907, 1995.

[38] Gabriel Taubin, Fernando Cukierman, Steven Sullivan, Jean Ponce, and David J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):287–303, 1994.

[39] T. Várady, R. R. Martin, and J. Cox. Reverse engineering of geometric models — an introduction. *Computer-Aided Design*, 29(4):255–268, 1997.

[40] Tamas Varady and Ralph Martin. Reverse engineering. In G. Farin et al., editor, *Handbook of Computer Aided Geometric Design*, pages 651–681. Elsevier Science (North-Holland), 2002.

[41] N. Werghi, R. B. Fisher, C. Robertson, and A. Ashbrook. Object reconstruction by incorporating geometric constraints in reverse engineering. *Computer-Aided Design*, 31:363–399, 1999.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Yan-Bin Jia, for his guidance, help and support throughout this research. I also would like to thank my committee members David Fernandez-Baca and Greg R. Luecke for their help in completing this work.

Support for this research has been provided in part by Iowa State University, and in part by the National Science Foundation through a CAREER award IIS-0133681.