

Surface Patch Reconstruction via Curve Sampling

Yan-Bin Jia

Department of Computer Science
Iowa State University
Ames, IA 50011
jia@cs.iastate.edu

Liangchuan Mi

nVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
lmi@nvidia.com

Jiang Tian

Department of Computer Science
Iowa State University
Ames, IA 50011
jiangt@iastate.edu

Abstract

This paper introduces a method that reconstructs a surface patch by sampling along three concurrent curves on the surface with a touch sensor. These data curves, each lying in a different plane, form a “skeleton” from which the patch is built in two phases. First, the Darboux frame at the curve intersection is estimated to reflect the local geometry. Second, polynomial fitting is carried out in the Darboux frame. The use of total (absolute) Gaussian curvature effectively prevents unnecessary folding of the surface normally expected to result from fitting over one-dimensional data. The reconstructed patch attains remarkable accuracy as demonstrated through experiments. This work carries a promise for in-hand manipulation. It also has potential application in building accurate models for complex curved objects which can cause occlusion to a camera or a range sensor.

1 Introduction

Objects with curved shapes are ubiquitous in our daily life, from smaller ones such as a pen, a mouse, and a teapot to bigger ones such as a chair, a bike, and a car. The differentiability of a curved shape allows smooth integration of kinematics, dynamics, and control when manipulated by the robot. Consequently, skillful maneuvers can be demonstrated [6]. Touch sensing is often the modality of choice for such purpose.

A touch sensor keeps track of the changes in the location of contact with an object, which are recorded as the tactile data. Such data in turn can be used for reconstructing the part of the object that has been touched. In this paper, we look at how to build a surface patch from data points sampled by a touch sensor along three concurrent curve segments. Each segment is generated by tracking the surface with a joystick sensor while constrained in a separate plane. From the implementation point of view, the problem of patch reconstruction is reduced to planar contour tracking [2, 14, 16].

We estimate the normal curvatures of the surface in the tangent directions of the curve segments at the point of their intersection. Here we apply a local quadratic fitting technique from [11]. The high positional accuracy ($\pm 0.02\text{mm}$) of the Adept robot allows us to obtain very accurate curvature estimates. Next, we can solve for the principal curvatures and di-

rections to set up the Darboux frame for patch fitting.¹

Within the estimated Darboux frame determined by the surface normal and the two principal directions, we intend to generate a patch description through polynomial fitting. However, the three data curves are insufficient to constrain the polynomial fit so that large variations in the geometry can appear between the curves. The total absolute Gaussian curvature is used as a measure to constrain the fitting.

The rest of this paper will be organized as follows. Section 2 describes how to construct a surface patch. Details include estimation of the Darboux frame, the fitting algorithm, and its use of total absolute Gaussian curvature. Section 3 verifies principal curvature estimation on synthetic and real data, and presents the experimental results. Finally, Section 4 gives with a summary and discusses future work.

1.1 Related Work

Methods for principal curvature estimation have been studied mostly on dense range data. They are not robust when applied on sparse tactile data. In [5], 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 [21] constructed a matrix at the point of interest. These two methods were later enhanced in [9] to better deal with noise.

Analytical methods [7, 10, 19] 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 [15] may suffer from large estimation errors due to loss of differentiability.

In computer vision, objects and shapes are often represented implicitly by polynomials, mostly for the convenience in operations such as inside-outside testing, handling of noisy data, and computation of algebraic invariants for recognition [12]. A polynomial representation of a shape is determined through least-squares fitting over its 2D image or 3D range data. Polynomials of even degrees are used because their level sets are

¹In fact, it is not difficult to derive closed forms of the principal curvatures.

bounded, which is not the case for those of odd degrees [22]. Often a subclass of quartic polynomials is chosen and constraints are incorporated into least-squares formulations [13, 20]. Multiple level sets may also be used to further constrain the fitting, as in the case of the $3L$ algorithm [3].

Fitting has also been performed over tactile data obtained with a robotic hand. Some previous work in this area involved the use of a superquadric over sparse position data [1] or a B-spline over position as well as normal data [4]. In medical imaging, a B-spline surface was fit to a set of normals for reconstructing the shape of the cornea from a videokeratograph image [8].

2 Surface Patch Reconstruction

Let p be a point of interest on a curved object. We would like to obtain some description of a region, or a “patch”, which surrounds p on the object’s surface. Throughout the paper, we make the following assumption:

A one-to-one correspondence exists between the patch to be reconstructed and its projection on the tangent plane at p .

The assumption makes sense because of the “locality” of the patch. Thus we need only consider a Monge patch $\sigma(x, y) = (x, y, z(x, y))$, which is best described in a coordinate system with p as the origin. We let the z -coordinate be represented (or approximated, more precisely speaking) by a polynomial function: $z(x, y) = \sum_{0 \leq i+j \leq d} a_{ij} x^i y^j$. 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 . The two partial derivatives of σ are tangent to the surface and therefore lie in the x - y plane:

$$\begin{aligned}\sigma_x|_{(0,0,0)} &= (1, 0, z_x)|_{(0,0,0)} = (1, 0, a_{10}); \\ \sigma_y|_{(0,0,0)} &= (0, 1, z_y)|_{(0,0,0)} = (0, 1, a_{01}).\end{aligned}$$

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. 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* \mathbf{d}_1 and \mathbf{d}_2 .

We now choose the principal directions as the x - and y -axes so they are determined by $(1, 0, 0)$ and $(0, 1, 0)$, respectively. Figure 1 shows the resulting local coordinate system referred to as the *Darboux frame*. Express the two principal directions in terms of the partial derivatives:

$$\begin{aligned}\mathbf{d}_1 &= (1, 0, 0) = 1 \cdot \sigma_x|_{(0,0,0)} + 0 \cdot \sigma_y|_{(0,0,0)}; \\ \mathbf{d}_2 &= (0, 1, 0) = 0 \cdot \sigma_x|_{(0,0,0)} + 1 \cdot \sigma_y|_{(0,0,0)}.\end{aligned}$$

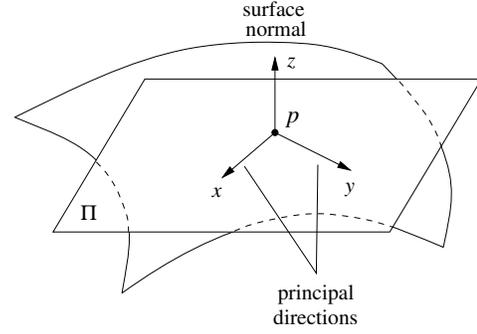


Figure 1: Local coordinate system at p with the two principal directions and the normal as the x -, y -, and z -axes.

Gather the coefficients of σ_x and σ_y in the above two equations respectively into two vectors $\mathbf{v}_1 = (1, 0)$ and $\mathbf{v}_2 = (0, 1)$.

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

$$\begin{aligned}E &= \sigma_x \cdot \sigma_x|_{(0,0,0)} = 1, \\ F &= \sigma_x \cdot \sigma_y|_{(0,0,0)} = 0, \\ G &= \sigma_y \cdot \sigma_y|_{(0,0,0)} = 1.\end{aligned}$$

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 [18, p.133] that

$$\begin{aligned}\mathbf{v}_i \begin{pmatrix} L & M \\ M & N \end{pmatrix} \mathbf{v}_j^T &= \kappa_i \mathbf{v}_i \begin{pmatrix} E & F \\ F & G \end{pmatrix} \mathbf{v}_j^T \\ &= \begin{cases} \kappa_i, & \text{if } i = j; \\ 0, & \text{if } i \neq j. \end{cases}\end{aligned}$$

The product $K = \kappa_1 \kappa_2$ is the *Gaussian curvature* at p . Substitute the expressions for E, F, G, L, M, N 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:

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

To reconstruct the patch surrounding p , we intend to fit the form (1) over data points obtained by a touch sensor. But how to estimate the principal curvatures and locate the principal axes?

2.1 Estimation of Principal Curvatures

We use a touch sensor to track the surface around the reference point p while constraining the sensor motion in a plane containing p , which we call the *sampling plane*. This is a planar contour tracking problem that has been studied by many researchers. 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. 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 2). We can fit over those data points very close to p and estimate

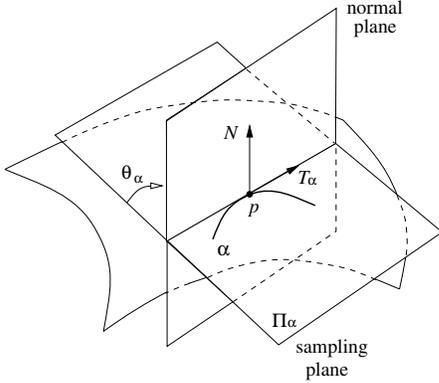


Figure 2: Data curve α lies in the sampling plane Π_α .

the tangent T_α and the curvature κ_α at p .²

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 then 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 , the tangent plane Π at p is determined. A normal plane at p is defined by N and the tangent vector T_α . Let θ_α be the angle between this normal plane and the sampling plane Π_α . By a result from differential geometry [18, 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. 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. Though θ is unknown, 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 tangent plane Π and the sampling planes Π_α, Π_β , and Π_γ . The normal curvatures are then expressed in terms of the two principal curvatures [18,

²A quadratic polynomial is used in the fitting because locally the curve resembles the osculating circle at p .

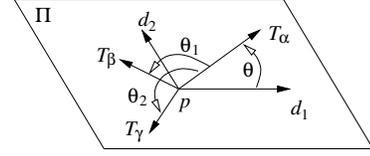


Figure 3: Tangent plane.

p. 137]:

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

There are three unknowns κ_1 , κ_2 , and θ in the three equations. Appendix A derives the closed forms of κ_1 , κ_2 , and θ . Consequently, the principal curvatures and directions are all determined. Next, we transform the coordinates of all data points to the Darboux frame at p determined by d_1 , d_2 , and N .

2.2 Constrained Surface Fitting

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 (1) over all the data points (x_i, y_i, z_i) , $1 \leq i \leq n$, sampled along α , β , and γ . The degree of the polynomial is set as $d = 4$. Write $\mathbf{a} = (a_{30}, a_{21}, \dots, a_{04})$ to include nine polynomial coefficients. They may be determined in a least-squares sense:

$$\min_{\mathbf{a}} f(\mathbf{a}) \quad \text{where} \quad f(\mathbf{a}) = \sum_{k=1}^n (z(x_k, y_k) - z_k)^2. \quad (3)$$

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 (1) defined by \mathbf{a} . The function $f(\mathbf{a})$ bounds the total squared distance from the data points to the patch.

Naturally, we would like the reconstructed patch to look “smooth”. That is, it should not have “peaks”, “valleys”, or folds unless induced by the three data curves. The reason is that the patch under construction is a rather “local” one where any drastic change of geometry between the three data curves is not expected. However, three data curves do not provide enough constraints on fitting. So the objective function for fitting needs to include a term that measures the “degree of folding” of the surface fit.

We make use of the *total Gaussian curvature* which is the integral of the Gaussian curvature over the surface patch domain D in the xy -plane³ (i.e., the tangent plane at the reference point p): $\iint_D K(x, y) dx dy$. Geometrically, every point q on the patch maps to the point on the unit sphere that represents the unit normal $N(q)$. This is called the *Gauss map* as shown in Figure 4. All the image points constitute a shaded region R

³In this work, D is the convex hull of the projection of α, β , and γ onto the xy -plane.

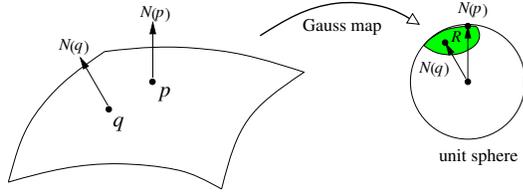


Figure 4: Gauss map from a patch to a unit sphere.

on the unit sphere, whose algebraic area is the total Gaussian curvature of the patch [17, p. 290].

If the Gaussian curvature does not change its sign over the patch, the closer the total Gaussian curvature is to zero, the flatter the patch. If the Gaussian curvature changes its sign, the Gauss map may fold the patch many times over the region R . The total Gaussian curvature thus does not tell how much the surface patch folds. To measure the folding over R , we integrate over the absolute Gaussian curvature:

$$\iint_D |K(x, y)| dx dy. \quad (4)$$

The more the patch folds, the larger the integral.

Given the coefficient vector \mathbf{a} , we can evaluate the total absolute Gaussian curvature (4) numerically. Discretize the patch domain D into a grid of m points $(u_1, v_1), \dots, (u_m, v_m)$ with uniform spacing h . The integral (4) is approximated by

$$g(\mathbf{a}) = h^2 \cdot \sum_{j=1}^m |K(u_j, v_j)|.$$

With the above change, patch reconstruction over the n data points is done through the following optimization:

$$\min_{\mathbf{a}} f(\mathbf{a}) + \lambda \cdot ng(\mathbf{a}). \quad (5)$$

This is equivalent to minimizing $\frac{1}{n}f(\mathbf{a}) + \lambda g(\mathbf{a})$ except numerically more stable. The choice of the multiplier λ is made independent of n . The first term in (5) 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.

2.3 The Optimization Algorithm

The nonlinear optimization (5) is carried out using steepest descent along the negative gradient of the objective function. Note that $g(\mathbf{a})$ is affected by the signs of the Gaussian curvatures $K(u_j, v_j)$, $1 \leq j \leq m$, which may vary from the current coefficient estimate $\mathbf{a}^{(l)}$ to the next one $\mathbf{a}^{(l+1)}$. To cope with this issue, at $\mathbf{a}^{(l)}$ we check the sign of the Gaussian curvature $K(u_j, v_j)$ at every grid point (u_j, v_j) and introduce the term

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

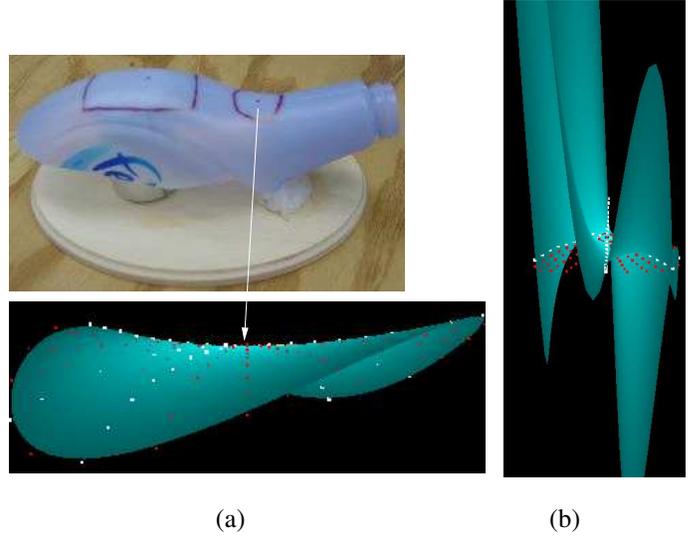


Figure 5: Reconstruction of the marked neck area on a broken plastic bottle: (a) a patch constructed according to (5), where the (white) data points are along three curves concurrent at the (marked) reference point; (b) a patch constructed according to (3) from the same data points. The red (or dark) points sampled from the same neck area are used for testing the qualities of the surface patches.

Then $g(\mathbf{a}) = h^2 \cdot \sum_{j=1}^m \delta_j K(u_j, v_j)$ in some neighborhood of $\mathbf{a}^{(l)}$ in the coefficient space. Performing steepest descent along the negative gradient of $f(\mathbf{a}) + \lambda ng(\mathbf{a})$ yields the next estimate $\mathbf{a}^{(l+1)}$. If the sign of every Gaussian curvature $K(u_j, v_j)$ does not change from $\mathbf{a}^{(l)}$ to $\mathbf{a}^{(l+1)}$, we have found a (local) minimum of (5). Otherwise, the iterations continue. The pseudocode of this optimization procedure is given below.

- 1 initialize the coefficients $\mathbf{a}^{(0)}$ of the polynomial (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 $\mathbf{v} \leftarrow$ gradient of $-f(\mathbf{a}) - \lambda nh^2 \cdot \sum_{j=1}^m \delta_j K(u_j, v_j)$
- 8 $\mathbf{a}^{(l+1)} \leftarrow$ steepest descent from $\mathbf{a}^{(l)}$ along \mathbf{v}
- 9 $l \leftarrow l + 1$
- 10 **until** for all $1 \leq j \leq m$ the sign of $K(u_j, v_j)$ under $\mathbf{a}^{(l)}$ does not change from that under $\mathbf{a}^{(l-1)}$

2.4 Results of Reconstruction

We took a broken plastic bottle shown in Figure 5 and sampled 57 points along 3 concurrent curves inside the marked neck region. The intersection point is parabolic with negative Gaussian curvature. Patch (b) in the figure was generated from fitting over positional data only according to (3). It does not nearly resemble the original region, with spikes between the curves where it is not constrained. The first data row in Table 1 shows that the average fitting error $\frac{1}{n} \sum_k |z(x_k, y_k) - z_k|$

is small (0.0459mm) over the 57 data points, but significantly larger (11.8753mm) over 171 test points sampled along extra 9 curves through the same reference point on the bottle. By incorporating the total absolute Gaussian curvature, patch (a) reduced the average fitting error over the 9 extra curves dramatically to 0.0681mm. The slight increase in the average error over the 3 original curves to 0.0646 is expected because of the extra term $\lambda n g(\mathbf{a})$ to minimize.

fitting scheme	principal curvatures (1/mm)	average fitting error (mm)	
		3 data curves (57 points)	9 verify. curves (171 points)
(3)	-0.0424, 0.0172	0.0459	11.8753
(5)	same as above	0.0646	0.0681
(5)	-0.0453, 0.019	0.0811	0.1531

Table 1: Statistics of the two reconstructed patches in Figure 5, and of a third patch over different data curves sampled from the same surface area as the second patch.

To test the robustness of the scheme (5), we also used 3 out of the 9 extra curves for patch reconstruction. The result is shown in the third row of Table 1. The last entry in the row was determined over points along the remaining 9 curves that were not used in the fitting. There is hardly any visual difference between the resulting patch and Figure 5(a). The multiplier λ in (5) was set to be 0.3 for these two reconstructions.

3 Simulation and Other Experiments

We started with testing the procedure for principal curvature estimation as described in Section 2.1. Figure 6 shows an ellipsoid $x^2/8^2 + y^2/5^2 + z^2/3^2 = 1$ intersected by the plane $y = 1$ into an ellipse $x^2/(16\sqrt{6}/5)^2 + z^2/(6\sqrt{6}/5)^2 = 1$. The synthetic data curve was chosen to be the upper left quadrant (bold line) of this ellipse from $(0, 1, 6\sqrt{6}/5)$ to $(16\sqrt{6}/5, 1, 0)$. Diagram (b) in the figure plots the two principal curvatures at points along the curve. The curvatures are evaluated analytically using the ellipse equation.

At every chosen point p , we picked three planes to intersect the ellipsoid and “sampled” nine neighboring points (with artificial noise) along each of the three intersection curves. From diagram (a) we see that the principal curvature estimates for each chosen point lie very close to the point that represents their exact values.

A joystick sensor from Interlink, Inc. is used with a pin to obtain position data on the surface. The sensor, as shown in Figure 7, has a force range of 20-70g. In reaction to the contact force between the pin and the surface, the sensor beam bends slightly upward. The contact point is located from the robot’s position reading with some compensation for the beam bending (which is assumed to be proportional to the force reading).

Experiments were also conducted on the marked convex region on the bottle in Figure 5, and on a mouse and a sphere, all

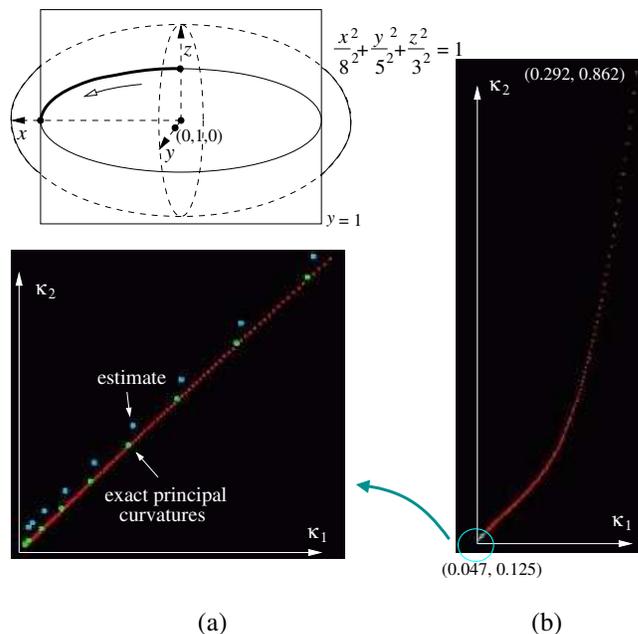


Figure 6: Principal curvatures κ_1 and κ_2 along the upper left quadrant of the intersection of an ellipsoid with a plane. Diagram (b) plots the exact values of κ_1 and κ_2 along the segment. Diagram (a) shows the enlargement of the lowest (encircled) portion of diagram (b) together with the principal curvatures estimated at nine chosen points according to the procedure in Section 2.1.

shown in Figure 7. Figure 8 displays the reconstructed patches on the bottle and the mouse. They agree well with the marked regions on these two shapes.

The estimated normal curvatures on the sphere in three chosen tangent directions were 0.0299/mm, 0.0307/mm, and 0.0305/mm, respectively. The two principal curvatures were estimated as 0.0309/mm and 0.0299/mm, very close to the value $1/(33\text{mm})$. The reconstructed patch was nevertheless not so spherical because it was more or less “flattened” due to the total absolute Gaussian curvature term in fitting.

4 Discussion and Future Work

This paper introduces a method that reconstructs a surface patch from three concurrent curves that are traced out by a touch sensor. Each curve is obtained through tracking the surface while constrained in a separate plane. The reconstruction is done through polynomial fitting in the Darboux frame at the intersection point which reflects its local geometry. Higher (than the second) order terms are included to prevent the polynomial description from being too local. The inclusion of the total absolute Gaussian curvature has effectively prevented unnecessary folding of the reconstructed patch. Tests with real shapes have demonstrated high accuracies.

The objective of this work is twofold. From basic research point of view, we would like to understand the minimum amount of tactile data sufficient for reconstructing a surface, or

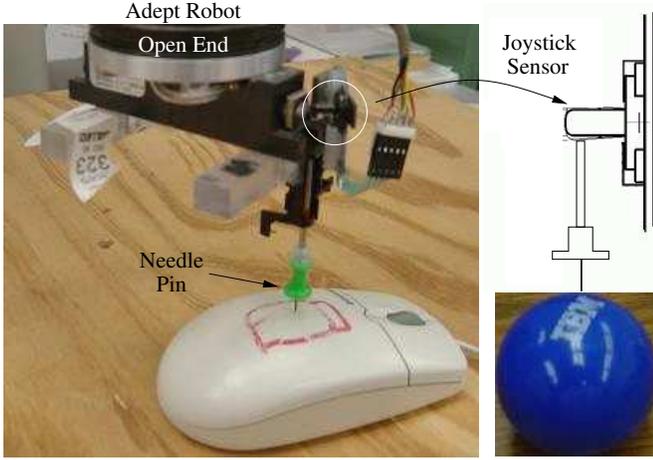


Figure 7: Experimental setup with a mouse and a sphere.

at least a patch. In practice, surfaces are often built from range data, which are subjected to camera occlusion and sometimes do not have the required precision. A situation may arise where very fine details about certain parts of an object (e.g., a bone in robot-assisted surgery) are needed. Probing or tracking with a touch sensor mounted on a high precision robot can be a good solution.

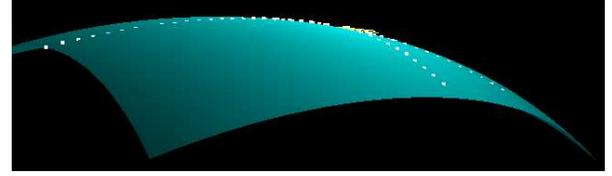
The reconstruction procedure is influenced by the accuracies of the estimated principal curvatures and directions at the reference point. Errors stem from curve sampling and normal curvature estimation, the latter of which is based on the assumption that the data curves are planar. Improvement is still needed on principal curvature estimation at parabolic points.

We have already implemented tracking in the horizontal plane. To reconstruct any part of an object, the robot must be able to track around an arbitrary point on its surface. A feedback force/position control strategy is needed to implement surface tracking constrained in an arbitrarily oriented plane.

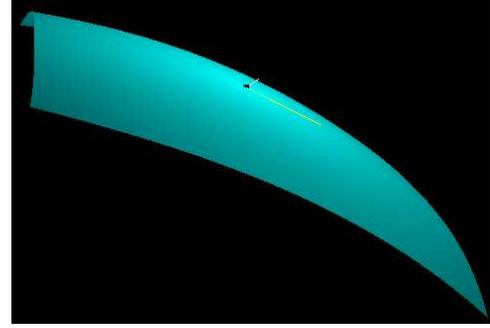
A complicated surface can be viewed as a collection of local patches. It may be represented as a graph, where every node corresponds to one patch on the surface and every edge corresponds to the boundary curve between two adjacent patches. Every patch is described in the Darboux frame at a reference point in its interior. One issue will be efficient update of the polynomial representing a patch once new tactile data are obtained. Another issue, at a higher level, will be dynamic graph maintenance. New nodes may be generated and old ones may be merged as more details of the surface are revealed during the tracking. The third issue, at the top level, will be on-line planning of the tracking trajectory in order to fully explore the surface. This, of course, will also involve adjusting the hybrid control algorithm for tracking.

A Solution of Principal Curvatures

Recall in the equations (2), κ_1 and κ_2 are the two principal curvatures at the reference point p . The normal curvatures κ'_α , κ'_β ,



(a)



(b)

Figure 8: The reconstructed patches for (a) the marked convex area on a bottle in Figure 5 and (b) the marked area on the mouse in Figure 7 with the principal axes shown.

and κ'_γ are in the directions of the tangents $T_\alpha, T_\beta, T_\gamma$ of three curves concurrent at p . Also see Figure 3. Rewrite equations (2) into the following:

$$\begin{aligned}\kappa'_\alpha &= \frac{\kappa_1 + \kappa_2}{2} + \cos(2\theta) \cdot \frac{\kappa_1 - \kappa_2}{2}, \\ \kappa'_\beta &= \frac{\kappa_1 + \kappa_2}{2} + \cos(2\theta + 2\theta_1) \cdot \frac{\kappa_1 - \kappa_2}{2}, \\ \kappa'_\gamma &= \frac{\kappa_1 + \kappa_2}{2} + \cos(2\theta + 2\theta_2) \cdot \frac{\kappa_1 - \kappa_2}{2}.\end{aligned}\quad (6)$$

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 \mathbf{d}_1 and \mathbf{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 solution of κ_1 and κ_2 exists as the three equations in (6) 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 (6) we have the following:

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

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

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

From the last equation above we obtain

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

Thus we have θ and the principal directions. Substitute θ , θ_1 , and θ_2 into the system (2) of linear equations to solve for the principal curvatures κ_1 and κ_2 .

Acknowledgment 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. Thanks to the anonymous reviewers for their valuable feedback.

References

- [1] P. K. Allen and P. Michelman. Acquisition and interpretation of 3-D sensor data from touch. *IEEE Trans. Robot. Automat.*, 6(4):397–404, 1990.
- [2] J. Baeten and J. De Schutter. Hybrid vision/force control at corners in planar robotic-contour following. *IEEE/ASME Trans. Mechatronics*, 7(2):143–151, 2002.
- [3] M. M. Blane, Z. Lei, H. Civi, and D. B. Cooper. The $3L$ algorithm for fitting implicit polynomial curves and surfaces to data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(3):298–313, 2000.
- [4] M. Charlebois, K. Gupta, and S. Payandeh. Shape description of curved surfaces from contact sensing. *Int. J. Robot. Res.*, 18(8):779–787, 1999.
- [5] X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In *Proc. European Conf. Comp. Vision*, pp. 739–743, 1992.
- [6] A. A. Cole, J. E. Hauser, and S. S. Sastry. Kinematics and control of multifingered hands with rolling contact. *IEEE Trans. Automat. Control*, 34(4):298–404, 1989.
- [7] P. J. Flynn and A. K. Jain. On reliable curvature estimation. In *Proc. IEEE Conf. Comp. Vision Pattern Recog.*, pp. 110–116, 1989.
- [8] 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 *Comp. Graphics Proc., SIGGRAPH*, pp. 335–341, 1996.
- [9] E. Hameiri and I. Shimshoni. Estimating the principal curvatures and the Darboux frame from real 3-D range data. *IEEE Trans. Sys., Man, and Cyber.*, 33(4):626–637, 2003.
- [10] R. L. Hoffman and A. K. Jain. Segmentation and classification of range images. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(5):608–620, 1987.
- [11] R. Ibrayev and Y.-B. Jia. Semi-differential invariants for tactile recognition of algebraic curves. *Int. J. Robot. Res.*, 24(11):951–969, 2005.
- [12] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(1):38–53, 1994.
- [13] D. Keren and C. Gotsman. Fitting curves and surfaces with constrained implicit polynomials. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(1):31–41, 1999.
- [14] F. Lange and G. Hirzinger. Learning force control with position controlled robots. In *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1399–1404, 1996.
- [15] C. Lin and M. J. Perry. Shape description using surface triangulation. In *Proc. IEEE Workshop Comp. Vision*, pp. 38–43, 1982.
- [16] L. Mi and Y.-B. Jia. High precision contour tracking with a joystick sensor. In *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, pp. 804–809, 2004.
- [17] B. O’Neill. *Elementary Differential Geometry*. Academic Press, Inc., 1966.
- [18] A. Pressley. *Elementary Differential Geometry*. Springer-Verlag, 2001.
- [19] E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-D objects: five practical methods. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(4):833–840, 1992.
- [20] S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3-D object modeling and recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(12):1183–1196, 1994.
- [21] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. IEEE Int. Conf. Comp. Vision*, pp. 902–907, 1995.
- [22] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(3):287–303, 1994.