[34] Z. M. Bi, L. Wang, and S. Y. T. Lang, "Current status of reconfigurable assembly systems," *Int. J. Manuf. Res.*, vol. 2, no. 3, pp. 303–327, 2007.
[35] Exechon AB,, 2011. [Online]. Available: http://www.exechon.com/
[36] *Applications in Aerospace, Auto, and Other Industries,* Exechon World, 2009 [Online]. Available: http://exechonworld.com/document/200902/article88.htm
[37] Z. M. Bi and Y. Jin, "Kinematic modelling of parallel kinematic machine Exechon," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 186–193, 2011.
[38] O. Ibrahim and W. Khalil, "Inverse and dynamic models of hybrid robots," *Mech. Mach. Theory*, vol. 45, no. 4, pp. 627–640, 2010.
[39] Y. J. Zhang and F. Gao, "Dynamic formulation and performance evaluation of the redundant parallel manipulator," *Robot. Comput.-Integr. Manuf.*, vol. 25, no. 4, pp. 770–781, 2009.
[40] M. Pirtini and I. Lazoglu, "Forces and hole quality in drilling," *Int. J. Mach. Tools & Manuf.*, vol. 45, pp. 1271–1281, 2005.

Recognition of Curved Surfaces From "One-Dimensional" Tactile Data

Rinat Ibrayev and Yan-Bin Jia

***Abstract*—This paper presents a method for recognition of 3D objects with curved surfaces from linear tactile data. For every surface model in a given database, a lookup table is constructed to store principal curvatures precomputed at points of discretization on the surface. To recognize an object, a robot hand with touch sensing capability obtains data points on its surface along three concurrent curves. The two principal curvatures estimated at the curve intersection point are used to look up the table associated with each model to locate surface discretization points that have similar local geometries. Local searches are then performed starting at these points to register the tactile data onto the model. The model with the best registration result is recognized. The presented method can recognize closed-form surfaces as well as triangular meshes, as demonstrated through simulation and robot experiments. Potential applications include recognition of (manufactured) home items that are grasped daily, and dexterous manipulation during which recognition is simultaneously performed.**

***Note to Practitioners*—In automation, the eventual purpose of sensing is for assembly and other operations. Recognition by touch has several advantages: efficiency, robustness to disturbance, and immunity to occlusion (unlike a vision system). We demonstrate that 3D objects can be recognized with small amounts of data acquired via robot touch. In particular, recognition can be as simple as dragging one or more fingers across a part of an object's surface.**

***Index Terms*—Closed-form surfaces, curve registration, model-based 3D recognition, principal curvatures, tactile data, triangular meshes.**

I. Introduction

A HUMAN BEING can recognize many objects with his eyes. This ability depends on the presence of light and on the appearance of the object in sight. When neither condition is satisfied, the object is not visible. However, its shape can still be felt via touching by hands. The hands are complementary to the eyes in object recognition. Analogously, touch sensing should serve a complementary role to vision-based recognition.

Recognition of 3D objects has traditionally been a subject of computer vision. The availability of high accuracy laser range scanners has enabled vision researchers to develop recognition strategies for a wide range of shapes. Nevertheless, occlusion and clutter remain as obstacles. One example is in-hand manipulation in which part of an object may be occluded by the robot hand. If equipped with tactile sensors, the hand would be able to record the moving contact position on the object's surface. Such data can replace range data for recognition purpose.

The above example suggests a complementary sensing modality in tasks where the robot interacts closely with objects. Touch sensing not only allows assessment of object properties, but also guides manipulation based on contact information [12]. Generally, occlusion and clutter are much smaller issues for tactile data acquisition. This makes touch sensing a favorite modality for dexterous manipulation. Local and often sparse, tactile data do not require expensive preprocessing such as scene image segmentation.

Despite the need to decipher or process tactile information, development of recognition algorithms based on such data has lagged behind. Past work on object recognition based on tactile data has been limited to certain classes of surfaces, including polyhedra [21], [22], convex objects [3], quadrics [31], and superquadrics [1].

In this paper, we present a method that can recognize 3D objects described by closed-form functions, or represented by triangular meshes, using "one-dimensional" tactile data. To the best of our knowledge, there has been no previous work on recognizing surfaces with triangular mesh models from tactile data. Requiring only a small amount of sensor data, our method also represents a minimalist approach to recognition.

The recognition method to be proposed is potentially applicable to distinguishing many daily objects that are handy for grasping. They include tools such as screw drivers, wrenches, sockets, hammers, etc; kitchen utensils such as cups, bowls, teapots, pans, etc.; toys; and other objects like the ones shown in Fig. 6. Most of these objects are manufactured according to some models. The recognition method can significantly enlarge the range of manipulation for a robot hand equipped with touch sensors.

The model of an object to be recognized is from a finite known set. The robot hand acquires data points through tracking along three concurrent curves on the object's surface. Our idea is to find the model on which these data points can be best superposed. To achieve this, we combine the techniques of table lookup and local optimization.

In the first phase, we estimate the two principal curvatures κ_1 and κ_2 at the curve intersection point q using those data points in its small neighborhood. A table has been constructed in advance for every model in the database to store pairs of principal curvatures evaluated at points sampled over the surface. We look up the table with (κ_1, κ_2) to find a set of candidate locations on the model that have similar local geometries to that of q.

In the second phase, a local search starts at every candidate location p_i, and descends along a path to reach some neighboring minimum point p_i^* that induces locally the best superposition of all data points onto the model. The best match from all such local searches is to be compared with those yielded on other models. The model that best registers the data is then recognized.

Section II overviews the related work. Section III reviews some basic concepts from differential geometry. The recognition method is outlined in Section IV, and carried out separately on closed-form sur-

Manuscript received August 10, 2011; revised February 03, 2012; accepted April 01, 2012. Date of publication May 04, 2012; date of current version June 28, 2012. This paper was recommended for publication by Associate Editor J. Xiao and Editor V. Kumar upon evaluation of the reviewers' comments. This work was supported in part by the Iowa State University, and in part by the National Science Foundation through CAREER Award IIS-0133681.

R. Ibrayev is with Ericsson Inc., San Jose, CA 95134 USA (e-mail: rinat.ibrayev@gmail.com).

Y.-B. Jia is with the Department of Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: jia@cs.iastate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2012.2194143

faces and triangular mesh models in Sections V and VI, respectively. Section VII presents some experimental results. Section VIII concludes with some discussion.

II. Related Work

In this section, we give an overview of previous work on object recognition from 2D image contours, 3D range data, and tactile data, as well as work on estimation of surface curvatures.

A. Recognition From Image Contours and Range Data

Recognition of a 3D object from 2D images could be carried out by constructing an implicit equation for the image contours and recovering the viewing parameters through fitting [32]. To achieve efficiency on a large model database, feature-based indexing with a kd-tree was applied to select the most likely hypotheses for verification [4]. In [49], "invariant curves" were used for recognition through indexing the model database with attributes computed from a single image, with no need to recover the viewing parameters.

A more prevalent recognition scheme is to develop geometric invariants with respect to Euclidean, affine, or projective transformations, or camera-dependent parameters [38], [50], [51]. In [53], images were mapped to points in a high-dimensional manifold called the "shape space," so that recognition reduced to measuring the geodesic distances between these points on the manifold.

Representation is a key in object recognition from range data. It should possess a number of qualities: invariance to rigid transformations, abilities to represent free-form objects and to handle occlusion and clutter, robustness to noise, and computational efficiency [36], [44]. Surface representation schemes include splashes [44], spherical attribute images [24], COSMOS [13], point signatures [11], spin images [30], surface signatures [52], point fingerprints [46], regional point descriptors [19], and salient geometric features [20]. A survey of free-form object representation and recognition techniques is available in [6].

With the representation chosen, object recognition was typically performed in two steps [2]. A pointwise correspondence was first established between the model and the range data. Based on the correspondence, a geometrical transformation (i.e., rotation and translation) was then computed for the best superposition of the data onto the model [16], [26], [42], [43].

When correspondences were unknown, the iterative closest point (ICP) algorithm [5] iteratively computed an optimal transformation from sensor data to match the model. Since ICP always converged to a local minimum, correct registration between model and data was not guaranteed. The work [10] selected three dispersed data points on a sensed surface, using their estimated principal curvatures, Darboux frame, and distance constraints to locate multiple triples of points. A heuristic search was then carried out to find the optimal transformation.

Object recognition were also done by hashing certain features [33], [37] derived from all models from a database. Similar features were extracted from a scene, and used to index the hash table for casting votes on the models. The favorites were further verified via separate matches with the scene to select the best model.

B. Tactile Shape Recognition

In touch sensing, the domain of object recognition has so far been limited to convex polyhedral objects, quadrics, and superquadrics. Shape recognition has long been based on the notion of "interpretation tree," which represents correspondences between features extracted from the tactile data and those on the model [17], [21], [22]. A volumetric approximation [3] can be built over tactile data to enhance feature selection and prune incompatible models.

Path planning for a touch sensor on polygonal and polyhedral objects was addressed in [15] based on interpreting local and global geometric constraints. The exponential size of the interpretation tree made the method computationally expensive for polyhedra with very large number of facets, and inapplicable to free-form objects represented as triangular meshes.

In [3], internal and external volumetric approximations of a convex object were built over sparse tactile data. The main disadvantage of this method was its applicability—limited to convex objects only.

Allen and Roberts [1] fit a superquadric over sparse tactile data. The recovered parameters of the superquadric were matched against a set of model parameters. Keren *et al.* [31] derived differential invariants for curves lying on surfaces of spheres, cylinders, cones, and tori. Though computable from tactile data, differential invariants are usually shape specific and not applicable for recognition of general objects.

An active sensing strategy was proposed in [8] to reconstruct local shape as a second-order polynomial over dense tactile array data, and to control the robot motion based on such shape information. The work by Charlebois *et al.* [7] involved the use of a B-spline over position as well as normal data.

C. Curvature Estimation

Our proposed recognition strategy will involve estimation of principal curvatures. Rusinkiewicz [41] estimated them by averaging the accumulated curvature tensor at a vertex, which is calculated using finite differences. Taubin [48] constructed a matrix through integration at a surface point. The principal curvatures were obtained from nonzero eigenvalues of the matrix. Chen and Schmitt [9] estimated normal curvatures at a surface point in three or more tangent directions, and used them to solve for the two principal curvatures. These two methods were later modified to deal with real noisy range data [23].

In analytical approaches [14], [18], [34], [35], a local quadratic or cubic patch was fitted over a point on a surface and its geometric neighbors. The principal curvatures were then obtained from the closed-form description of the fitting patch. Two methods [34], [35] approximated the Gaussian curvature using the ratio of an "angle deficit" of all incident triangles to the sum of their areas.

Paper [47] compared several methods, including Taubin's [48], Chen and Schmitt's [9], and angle deficit [34], [35], for estimating Gaussian and mean curvatures on triangular meshes. It pointed out that paraboloidal fitting is the best method for estimating the two principal curvatures. This method will be employed by us over triangular meshes.

III. Brief Review of Surface Geometry

Let $\boldsymbol{\sigma}(u, v)$ be an orientable parametric surface. Denote by $\boldsymbol{\sigma}_u$ and $\boldsymbol{\sigma}_v$ the first-order partial derivatives of $\boldsymbol{\sigma}$ with respect to u and v, respectively, and $\boldsymbol{\sigma}_{uu}, \boldsymbol{\sigma}_{uv}, \boldsymbol{\sigma}_{vv}$ its three second-order partial derivatives. The surface normal is $\boldsymbol{n} = \boldsymbol{\sigma}_u \times \boldsymbol{\sigma}_v / \|\boldsymbol{\sigma}_u \times \boldsymbol{\sigma}_v\|$. Introduce two matrices [40, p. 132], which consist of the coefficients of the first and second fundamental forms of $\boldsymbol{\sigma}$, respectively

$$\mathcal{F}_I = \begin{pmatrix} \boldsymbol{\sigma}_u \cdot \boldsymbol{\sigma}_u & \boldsymbol{\sigma}_u \cdot \boldsymbol{\sigma}_v \\ \boldsymbol{\sigma}_u \cdot \boldsymbol{\sigma}_v & \boldsymbol{\sigma}_v \cdot \boldsymbol{\sigma}_v \end{pmatrix},$$
$$\mathcal{F}_{II} = \begin{pmatrix} \boldsymbol{\sigma}_{uu} \cdot \boldsymbol{n} & \boldsymbol{\sigma}_{uv} \cdot \boldsymbol{n} \\ \boldsymbol{\sigma}_{uv} \cdot \boldsymbol{n} & \boldsymbol{\sigma}_{vv} \cdot \boldsymbol{n} \end{pmatrix}.$$

The *principal curvatures* κ_1 and κ_2 are the eigenvalues of $\mathcal{F}_I^{-1}\mathcal{F}_{II}$. They are achieved in two orthogonal directions[1] which are referred to as the *principal vectors*. Intuitively, at every point on the surface, the geometry varies the most and the least respectively along the directions

[1]unless $\kappa_1 = \kappa_2$ in which case every direction is principal.

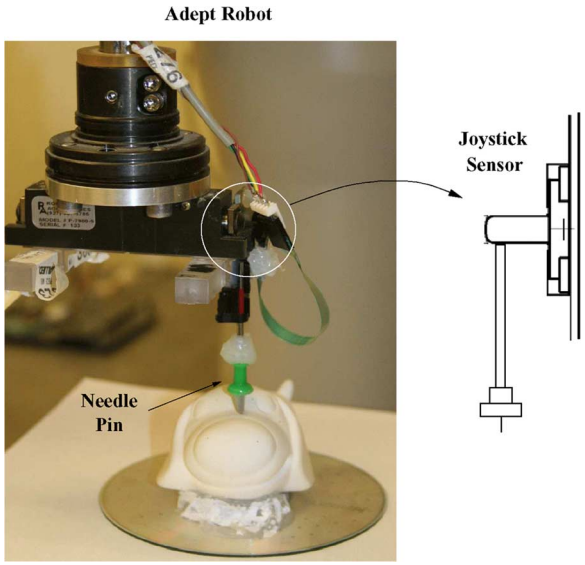


Fig. 1. Adept Cobra 600 (end-effector shown only) tracking on a dog-shaped object with a joystick sensor.

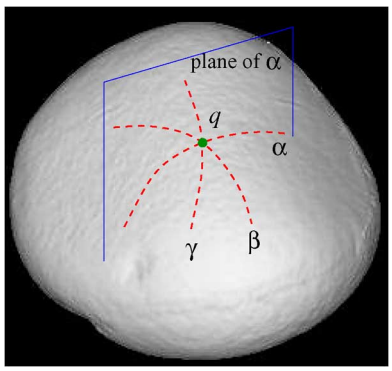


Fig. 2. Three data curves α, β, and γ intersect at one point q on a surface.

indicated by these two vectors. The rates of variation in geometry are given by the principal curvatures κ_1 and κ_2.

The *Gaussian curvature* is $\det(\mathcal{F}_I^{-1}\mathcal{F}_{II}) = \kappa_1\kappa_2$, while the *mean curvature* is $(1/2)\mathrm{trace}(\mathcal{F}_I^{-1}\mathcal{F}_{II}) = (\kappa_1 + \kappa_2)/2$.

IV. Recognition Scheme

To recognize the shape of an object, the robot acquires a number of points on its surface. These data points will be registered onto every model from a database to choose the best match. Fig. 1 shows the setup in which an Adept robot is tracking a surface with a push pin guided by a joystick sensor. Discrete contact points are recorded as the pin moves.

A. Data Configuration

To facilitate the matching process, the sampled data points are along three concurrent curves α, β, and γ on the surface.[2] The main reason is to simplify control—the pin moves on the surface while constrained sequentially in three distinct normal planes through a surface point q. The curves α, β, γ are thus the respective intersections of the surface with these planes. By a slight abuse of the terminology, the three sequences of sampled data points along these curves are referred to as "data curves," and still denoted by α, β, γ.

The tangent, normal, and curvature of each curve at q can be estimated via fitting a parabola to the corresponding data points. The surface normal N at q, referred to as the "data normal," can then be estimated in a least-squares manner from the three tangents. The *normal curvature* of the surface at q in each tangential direction is then obtained by scaling the curvature of the corresponding curve with the dot product between N and the curve normal at q. From these normal curvatures we can solve for the two principal curvatures κ_1 and κ_2 together with the principal directions [29].

We arbitrarily pick two orthogonal tangent vectors $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ at q to form a local frame with N. All data points along α, β, and γ, are then converted into local coordinates, and denoted as $q_i, 1 \le i \le n$.

B. Superposition of Data Curves

Let M be a model to be matched against the three data curves α, β, and γ. We mount the data curves onto M by coinciding their intersection point q with some point p on the model surface, and aligning the data normal N with the surface normal $\boldsymbol{n}$ at p. Suppose $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are aligned with unit tangent vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ at p in the body frame of M. The coordinates of the original data point $q_i, 1 \le i \le n$, in this body frame are then

$$q_i' = p + (\boldsymbol{d}_1\ \boldsymbol{d}_2\ \boldsymbol{n})(\boldsymbol{t}_1\ \boldsymbol{t}_2\ N)^T(q_i - q).$$

When we rotate the data curves about $\boldsymbol{n}$ through an angle $\phi \in [0, 2\pi]$, the unit vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ also rotate by the same amount to coincide with $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ in the common tangent plane. So q_i' is a function of ϕ.

Denote by $d(q_i', M)$ the distance from q_i to M after the above alignment with rotation ϕ. The *superposition error* at p minimizes over all rotations

$$E(p) = \min_{0 \le \phi \le 2\pi} \frac{1}{n} \sum_{i=1}^{n} d\left(q_i', M\right). \tag{1}$$

C. Model-Based Recognition

On the surface of every model M, we precompute the principal curvatures at a collection of points, and store them in a table T. Look up T with the pair (κ_1, κ_2) of principal curvature estimates from the data curves to find those points with similar local geometries on the model surface. Let κ_{1i} and κ_{2i} be the principal curvatures at the ith point in the table. The point is selected as a *candidate point* if $\sqrt{(\kappa_1 - \kappa_{1_i})^2 + (\kappa_2 - \kappa_{2_i})^2} < \delta$, where δ is a preset tolerance.

Let $p_1, p_2, \ldots, p_m$ be the selected points that locally resemble the intersection q of the three data curves. For $1 \le j \le m$, we superpose the data curves at p_j, and compute the error $E(p_j)$. Then, move q in the neighborhood of p_j on M until the error reaches a local minimum at some point p_j^*.

The *registration error* $\mathcal{E}(M)$ is the minimum of the superposition errors over all the points on the model (at which q can be placed). We approximate it as

$$\mathcal{E}(M) = \min\left\{E\left(p_1^*\right), E\left(p_2^*\right), \ldots, E\left(p_m^*\right)\right\}. \tag{2}$$

And the point p_k^* achieving $\mathcal{E}(M)$ is the estimated location of p on the model M.

The model in the database that yields the minimum registration error will be recognized as the shape of the object. We will describe the algorithm in details, for closed-form surfaces and triangular meshes separately in Sections V and VI. The main differences for these two types

[2]The data configuration shown in Fig. 2 is also used in [29] for surface patch reconstruction.

TABLE I
FOUR SURFACE FAMILIES USED IN THE SIMULATION

Implicit form	Parametric form
$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$ (ellipsoid)	$(a\cos u\sin v, b\sin u\sin v, c\cos v)$ $(u,v)\in[0,2\pi]\times[0,\pi]$
$\frac{x^2}{a^2}+\frac{y^2}{b^2}=z$ (elliptic paraboloid)	$(av\cos u, bv\sin u, v^2)$ $(u,v)\in[0,2\pi]\times[-1,1]$
$x^3-3xy^2=z$ (monkey saddle)	(u,v,u^3-3uv^2) $(u,v)\in[-1,1]\times[-1,1]$
$x^2y^2=z$ (crossed trough)	(u,v,u^2v^2) $(u,v)\in[-1,1]\times[-1,1]$

of surfaces lie in curvature estimation and optimization that are performed during curve registration.

V. CLOSED-FORM SURFACES

Suppose the model M has a parametric surface $\boldsymbol{\sigma}(u,v)$. We uniformly discretize the domain of $\boldsymbol{\sigma}(u,v)$. Construct the lookup table T by calculating the principal curvatures at all surface discretization points.

A. Registration Through Local Optimization

Superposition of the data curves depends on the surface parameters u and v that determine the location at which their common intersection is placed. With a slight abuse of notation, we denote the superposition error at $\boldsymbol{\sigma}(u,v)$ as $E(u,v)$.

- If the implicit form of surface $f(x,y,z)=0$ is known, we can approximate the distance $d(q_i',M)$ from the transformed data point $q_i'=(x_i',y_i',z_i')^T$ to M as follows. Apply Taylor expansion at q_i'

$$f(x,y,z)\approx f\left(x_i',y_i',z_i'\right)+\nabla f\left(x_i',y_i',z_i'\right)\cdot\left(x-x_i',y-y_i',z-z_i'\right)^T.$$

  The closest point on M to q_i' has zero f value. Letting the left-hand side of the above equation be zero, we obtain

$$d\left(q_i',M\right)=\left\|\left(x-x_i',y-y_i',z-z_i'\right)^T\right\|\approx\frac{\left|f\left(x_i',y_i',z_i'\right)\right|}{\left\|\nabla f\left(x_i',y_i',z_i'\right)\right\|}.$$

- If only the parametric form $\boldsymbol{\sigma}$ is known, the distance can be computed numerically by searching for the closest surface point.

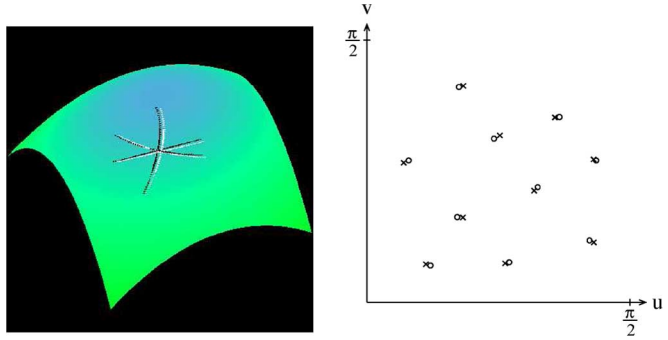
Starting at every candidate point $p_j=\boldsymbol{\sigma}(u_j,v_j)$, $1\le j\le m$, returned by the table lookup, we search for a local minimum of $E(u,v)$ using the steepest descent method [39, p. 425]. At each iterative step, it moves along the gradient $\nabla E(u,v)=(E_u,E_v)$ until the value of E no longer decreases.

The initial estimate for the search is set by letting $u_j^{(0)}=u_j$ and $v_j^{(0)}=v_j$. To find $\phi_j^{(0)}$ that minimizes the initial superposition error $E(u_j,v_j)$ according to (1), we discretize the domain $[0,2\pi)$ into intervals of length h. Within each bracket $[(i-1)h,ih,(i+1)h]$, where E attains a smaller value at $\phi=ih$ than at $(i-1)h$ and $(i+1)h$, Golden section search [39, pp. 401–403] is performed to find a local minimum. Then $\phi_j^{(0)}$ assumes the value that yields the smallest of these minima.

The angle $\phi_j^{(0)}$ may be more efficiently found by aligning the principal vectors at q estimated over the data curves and those computed at p_j on the model M. However, this is sensitive to the error in principal vector estimation. A slight rotation still needs to be performed after such alignment to minimize E locally. We have found this approach to be less robust than the domain discretization approach above.

B. Simulation

Table I lists four families of surfaces, in both implicit and parametric forms, which are used in our simulation.



(a) (b)

Fig. 3. (a) Three data curves (black) versus their registered configurations (white) on an elliptic paraboloid described by $z=(x^2/1.5^2)+(y^2/1.1^2)$. (b) Ten instances of curve registration, each shown with the real location (circular dot) of the intersection point q and its estimate (the closest cross).

TABLE II
MINIMUM, MAXIMUM, AND MEAN EUCLIDEAN DISTANCES BETWEEN THE REAL AND ESTIMATED LOCATIONS OF THE CURVE INTERSECTION CALCULATED OVER TEN REGISTRATION INSTANCES FOR EACH SURFACE

	ellipsoid	elliptic paraboloid	monkey saddle	crossed trough
min	0.0169	0.0148	0.0053	0.0171
max	0.0408	0.0448	0.0433	0.0396
mean	0.0288	0.0297	0.0234	0.0265

1) Curve Registration Results: The elliptic paraboloid from the table is displayed in Fig. 3(a). We select a point q, say, with parameter values $(u,v)=(1.21,0.43)$. Intersect the elliptic paraboloid with three arbitrary normal planes through q, and generate 61 data points along each of the three intersection curves. Random noises within the range of ±0.001 are added to these points.

The normal and the two principal curvatures at p are estimated as $\tilde{N}=(0.1611,0.5764,0.8011)$, $\tilde{\kappa}_1=0.9548$, and $\tilde{\kappa}_2=0.6370$.[3] A table lookup finds three candidate points: $p_1=(1.26,0.4)$, $p_2=(1.41,0.4)$, and $p_3=(0.63,0.5)$, all in parameter values. The superposition errors (1) at these points are, respectively, $E_1=0.000888$, $E_2=0.000961$, and $E_3=0.001654$.

Local searches starting at the candidate points yield locations $\tilde{p}_1=(1.26,0.44)$, $\tilde{p}_2=(1.27,0.44)$, and $\tilde{p}_3=(1.17,0.44)$, respectively. The corresponding superposition errors are $\tilde{E}_1=0.000662$, $\tilde{E}_2=0.000666$, and $\tilde{E}_3=0.000686$. The location of q is thus estimated to be $\tilde{p}_1=(1.26,0.44)$, which is very close to its real location (1.21, 0.43). The three curves are well registered onto the elliptic paraboloid, as shown in Fig. 3(a).

We have conducted ten registration tests on the same elliptic paraboloid, and on each of the three other shapes: an ellipsoid, the monkey saddle, and the crossed trough in Table I. Each instance is performed with a randomly generated intersection point q and three sampling planes containing it. Fig. 3(b) displays the results of ten registration instances for the elliptic paraboloid. In every instance, the real location of q is drawn as a circular dot and its estimated location q^* as the closest cross.

Table II summarizes the Euclidean distances $\|q-q^*\|$ computed over the 40 registration instances.

2) Recognition Tests: In an instance of recognizing, say, the crossed trough, we generate three data curves through a randomly selected point on the surface, and try to register them onto the four surfaces

[3]The values obtained from the surface equation are $N=(0.1612,0.5827,0.7966)$, $\kappa_1=0.9074$, and $\kappa_2=0.6519$, respectively.

TABLE III
SUMMARY OF RECOGNITION TESTS, TEN ON EACH SHAPE

successes	ellipsoid	elliptic paraboloid	monkey saddle	crossed trough
table lookup	2	1	9	0
local search	8	9	1	10

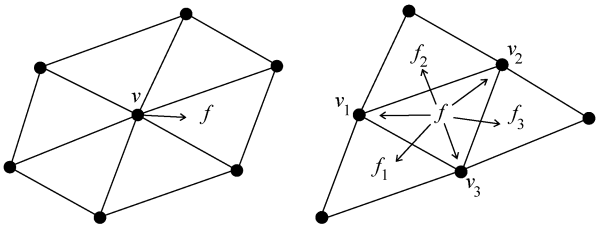


Fig. 4. Connectivity of a triangular mesh. (a) Each vertex has a pointer to one of its incident triangles. (b) Each triangle has pointers to its three vertices as well as three adjacent triangles.

in Table I. The registration error (2) on the crossed trough itself is 0.000393. Using the principal curvature estimates, we find no candidate curve intersection points on the ellipsoid and elliptic paraboloid after table lookups. The error on the monkey saddle is 0.001020, 159% higher than that on the crossed trough, which is then recognized.

For each surface, ten recognition instances with different curve intersection points have been carried out. The results are displayed in Table III. In total, 12 out of 40 tests have succeeded because table lookups yielded no candidate points on the wrong models. The other 28 successes resorted to local optimizations after table lookups. In these tests, the registration errors for the wrong models exceeded those for the right models by an average of 180%.

VI. TRIANGULAR MESH MODELS

Real objects of arbitrary topology can be well represented by triangular mesh models (or spline or subdivision surfaces spanned by them). Free-form surfaces can also be approximated by triangular meshes very well. Thus, it is very important to recognize this type of models from an application point of view.

A triangular mesh model of an object can be generated using a 3D scanner or a laser range sensor. It consists of a set of vertices, in 3D coordinates, and a set of triangular faces, each represented by the indices of its three vertices. The structure is shown in Fig. 4. A *k-ring* neighborhood of a vertex v is a set of all vertices that lie at most k edges away from v. Fig. 4(a) shows the 1-ring neighborhood. The k-ring neighborhood can be efficiently computed via a breadth-first search.

A lookup table of principal curvatures is constructed over some sampled vertices. We conduct sampling densely in high curvature regions on the mesh model and sparsely in low curvature regions. The density is adjusted according to the rate of change of the surface normal. At a vertex v, approximate the surface normal as $\boldsymbol{n} = \sum_{i=1}^{l} \boldsymbol{n}_i / \| \sum_{i=1}^{l} \boldsymbol{n}_i \|$, where l is the total number of triangles in the 1-ring neighborhood of v, and $\boldsymbol{n}_i$ is the normal of the ith such triangle. A procedure is described below to generate a list $\mathcal{L}$ of sampled vertices.

Initially, all vertices are unmarked, and $\mathcal{L}$ is set to be empty. At each step, we randomly select an unmarked vertex v and compute its 6-ring neighborhood $\mathcal{N}$. Next, the angles between the normal at v and the normals of all the triangles in $\mathcal{N}$ are calculated. If any of these angles is greater than some specified value θ, we reduce the ring size of $\mathcal{N}$ by one. This is repeated until all angles are less than θ, or $\mathcal{N}$ becomes a 2-ring neighborhood of v. Mark all the vertices in $\mathcal{N}$, and add v to the list $\mathcal{L}$.

The procedure terminates when half of the vertices are marked. At this point, the model surface presumably is well covered by $\mathcal{L}$. Fig. 5 shows the model of a cat object (in front and rear views) with 64821 vertices and 129638 faces, generated by NextEngine's desktop 3D scanner. The average distance between two adjacent vertices on the mesh is 0.493 mm. The sampling procedure selects 550 vertices (displayed as black dots).

The principal curvatures on a triangular mesh do not have closed forms. But they can be easily solved from the Gaussian and mean curvatures. A number of methods for estimating the latter pair of curvatures on triangular meshes were compared in [47], where paraboloidal fitting was found to be the best one. Here, at every vertex v from $\mathcal{L}$, we fit a paraboloid

$$z = ax^2 + bxy + cy^2 + dx + ey + f \tag{3}$$

repeatedly over five local neighborhoods consisting of the 30, 40, 50, 60, and 70 closest vertices to v (by the Euclidean distance). Each resulting paraboloid yields a pair of principal curvatures via closed-form

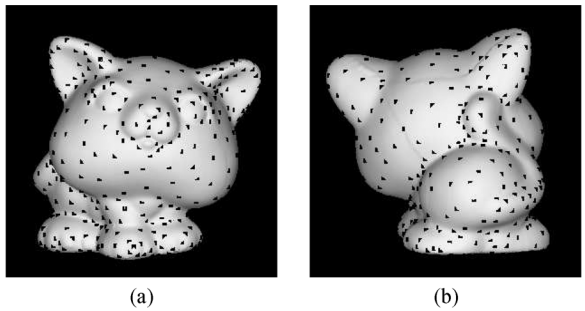


Fig. 5. Vertex sampling on a cat model: (a) front and (b) back views of the model with 550 sampled vertices (black dots) out of 64821.

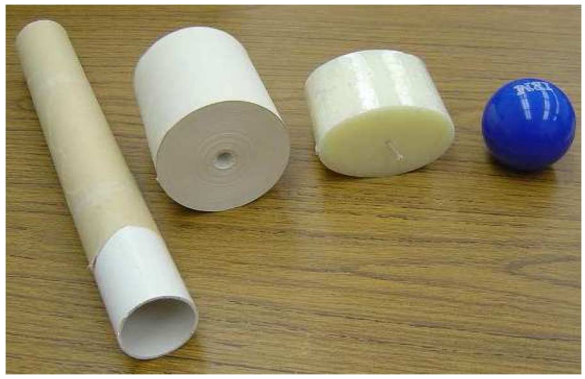


Fig. 6. Objects with algebraic surfaces in an experiment: two regular cylinders with diameters 50.4 and 94 mm, respectively, an elliptic cylinder with semimajor axis 50.8 mm and semiminor axis 31.75 mm, and a sphere with radius 33 mm.

TABLE IV
REGISTRATION ERRORS (MM) FOR DATA ACQUIRED FROM THE FOUR OBJECTS SHOWN IN FIG. 6 ONTO THEIR MODELS

model \ object	cylin. 1	cylin. 2	ell. cylin.	sphere
cylinder 1	**0.033**	0.155	0.446	0.296
cylinder 2	0.182	**0.072**	0.195	1.056
elliptic cylinder	0.230	0.224	**0.156**	0.254
sphere	0.271	0.306	0.929	**0.055**

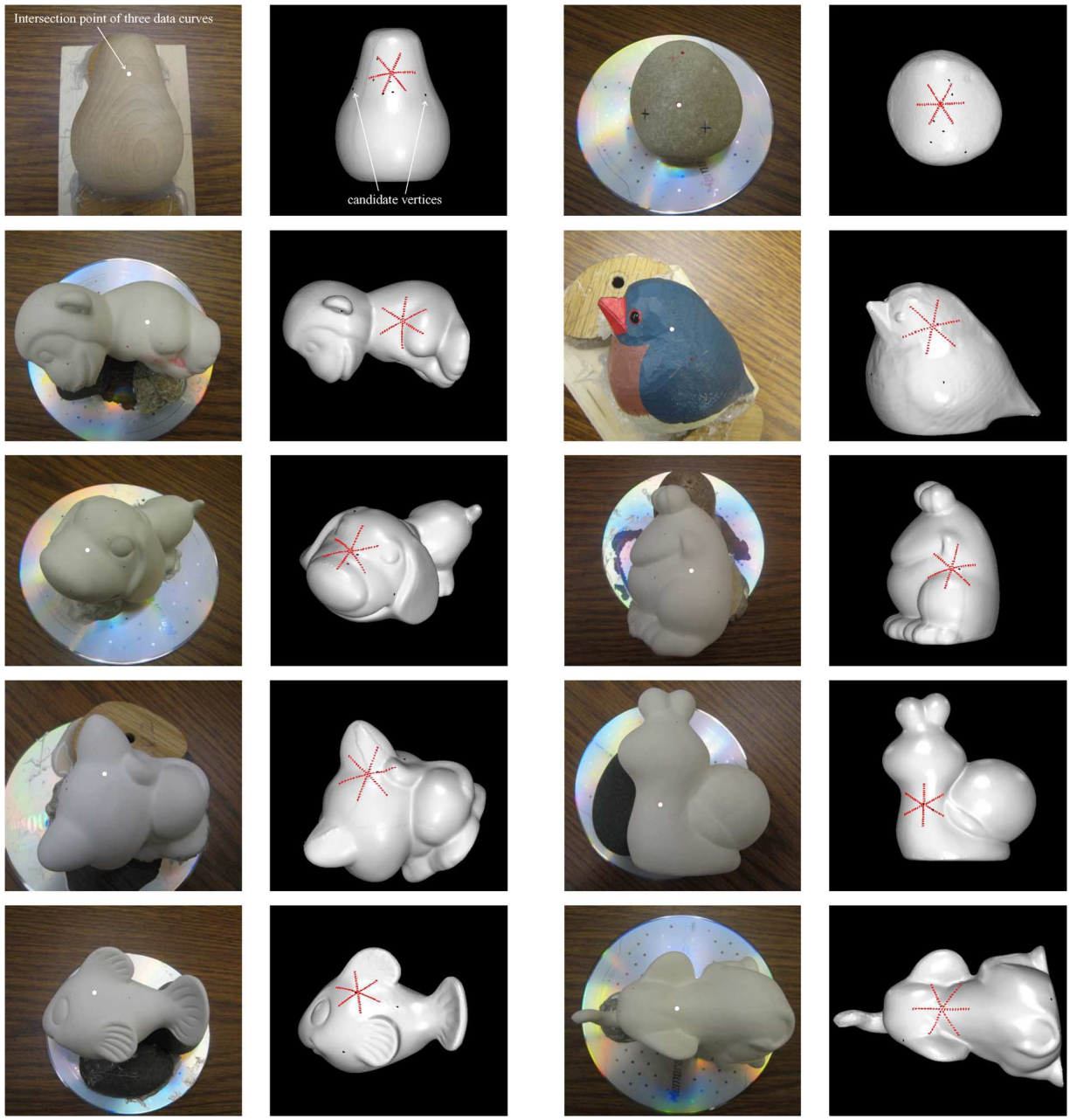


Fig. 7. Curve registrations on ten objects. Each model is displayed next to the original object, where the white dot on the object indicates the reference point through which the robot sampled along three concurrent curves, and the black dots on the model are the candidate vertices found after a table lookup. All data curves, displayed in red color, were successfully registered on all models.

calculation. The median values are selected as the estimates for the principal curvatures at v.[4]

A. *Registration Onto a Triangular Mesh*

To superpose three concurrent data curves α, β, and γ onto a triangular mesh M at a candidate vertex, say, v_j, we place the curve intersection point q at v_j, aligning the data normal N with the mesh normal $\boldsymbol{n}$ at v_j. This eliminates all degrees of freedom except the rotation ϕ of α, β, and γ about $\boldsymbol{n}$. To estimate the superposition error E at v_j, we minimize over discretized values of ϕ: $0, \pi/18, \pi/9, \ldots, 35\pi/18$.

[4]The selected values are independent of each other in the sense that they might have been obtained from different paraboloids.

For each value of ϕ, we sample points along the intersections of M with the three planes containing α, β, γ, maintaining the same separations between the points as on α, β, and γ, respectively. Note that the points on M corresponding to the data points are not necessarily vertices—they may be on the boundaries or in the interiors of triangles.

The superposition error E is further minimized in the neighborhood of v_j using the following *greedy* approach. We calculate E at each *unvisited* vertex of a 1-ring neighborhood $\mathcal{N}$ of v_j. Then, we move from v_j to a vertex in $\mathcal{N}$ with the smallest value of E and mark all unvisited vertices in $\mathcal{N}$ as *visited*. This movement is repeated until E achieves a local minimum E_j^* at some vertex v_j^*.

Finally, the registration error $\mathcal{E}$ is set as the minimum of all E_j^*s. The vertex v_k^* which yields the value of $\mathcal{E}$ is the estimated location of the intersection point q of α, β, and γ on M.

TABLE V
DESCRIPTION OF TEN MODELS

Model	Number of vertices	Number of triangles	Bounding box (mm)	Surface area (mm^2)	Sample vertices	Sampling time (sec)	Lookup table construction (sec)	Avg. distance (mm) between adj. vertices
Pear	84656	169308	$83.9 \times 62.6 \times 62.4$	15169	400	4.857	5.017	0.469
Stone	42777	85550	$62.7 \times 56.3 \times 43.5$	9131	250	4.446	3.195	0.516
Monkey	75443	150882	$80.0 \times 71.9 \times 51.5$	14488	500	2.654	6.389	0.488
Bird	63323	126642	$78.6 \times 61.4 \times 55.9$	12258	400	9.234	5.077	0.490
Dog	67101	134198	$90.3 \times 62.5 \times 43.2$	13654	450	1.593	5.748	0.504
Frog	91970	183936	$79.4 \times 78.9 \times 58.5$	17023	600	4.657	7.511	0.480
Cat	64821	129638	$71.9 \times 67.2 \times 53.5$	12681	550	3.415	7.030	0.493
Snail	101268	202532	$94.8 \times 89.6 \times 52.2$	18257	650	5.709	8.271	0.474
Fish	71490	142976	$90.5 \times 71.5 \times 61.4$	14103	570	4.216	7.341	0.495
Elephant	72665	145326	$92.0 \times 62.1 \times 61.5$	13308	650	5.398	8.462	0.477

TABLE VI
REGISTRATION ERRORS (MM) INVOLVING TEN OBJECTS. THE LAST ROW SHOWS THE TOTAL TIME (SECONDS) OF REGISTRATION ONTO ALL TEN MODELS

		Objects									
		Pear	Stone	Monkey	Bird	Dog	Frog	Cat	Snail	Fish	Elephant
Models	Pear	**0.0724**	0.1314	0.8347	0.1411	NC	NC	0.3803	0.4146	0.1607	0.6271
	Stone	0.2523	**0.0741**	0.5507	0.4308	NC	NC	NC	0.5157	0.2306	NC
	Monkey	0.4520	0.1027	**0.1304**	0.2228	0.3403	0.6033	0.3908	0.2057	0.2816	0.7358
	Bird	0.2856	0.1430	0.1661	**0.0933**	0.4341	0.6639	0.5049	0.4089	0.2617	0.9158
	Dog	0.2940	0.1340	0.3131	0.2250	**0.1420**	0.3675	0.2131	0.3328	0.2283	0.8359
	Frog	0.2026	0.1273	0.4004	0.1575	0.3011	**0.2494**	0.4869	1.2953	0.1766	0.8584
	Cat	0.3429	0.1863	0.5486	0.3981	0.3741	0.5384	**0.1569**	0.4772	0.2640	1.0529
	Snail	0.1907	0.1557	0.4128	0.2308	0.2552	0.6508	0.4575	**0.1312**	0.2162	NC
	Fish	0.2355	0.1083	0.3370	0.2562	0.3386	0.6460	0.4270	0.2692	**0.0762**	0.3796
	Elephant	0.3875	0.2998	0.4680	0.2754	0.7291	0.7552	0.4427	0.3568	0.2475	**0.1844**
	Recognition time (s)	*11.118*	*17.914*	*4.567*	*11.486*	*5.459*	*3.435*	*6.118*	*8.821*	*19.308*	*6.091*

VII. EXPERIMENTS

In this section, we present experimental results on several objects bounded by algebraic and arbitrary surfaces. Data points were obtained through tracking with a push pin mounted on an Adept robot, which was guided by a joystick sensor as shown in Fig. 1. A detailed description of the tracking device is given in [28]. The robot has positioning accuracies 0.02 mm in the x- and y-directions and 0.01 mm in the z-direction. It employed a tracking procedure from [35], which could reliably generate 2D contours with an accuracy of ± 0.1 mm according to our measurements. Force control used a simple feedback loop to avoid unnecessary oscillations, while position control updated the tracking direction based a quadratic fit over local turning of the contour. Contact positions were read by the Adept with an offset due to the push pin.

Robot movements were constrained in three vertical planes containing a common line. The sensor sampled points along their intersection curves with an object's surface. The angle between any two sampling planes was set to be $\pi/3$, so that the data curves are spread out the most.[5]

A. Objects With Closed-Form Surfaces

Fig. 6 displays four objects bounded by (piecewise) algebraic surfaces used in one experiment. A total of 41 points were acquired along each data curve with a spacing of 0.5 mm between the points.

Due to symmetry, curve registration results on the sphere and two cylinders were meaningless. The registration error, however, was still useful for the recognition purpose. In Table IV, every column lists the errors (2) for registering three data curves acquired from one object onto all four models. The diagonal cells correspond to registrations onto the correct models, while the non-diagonal cells correspond to those onto incorrect models. It is clear that every diagonal cell has the smallest value in its column. As a result, all four objects were correctly recognized.

[5]Preliminary tests showed that the success rate of recognition tended to increase with the angle up until $\pi/3$.

B. Objects With Arbitrary Topology

Fig. 7 shows the results from another experiment over ten objects. The statistics about their mesh models and hash tables are given in Table V. The objects are displayed in the first and third columns in Fig. 7. On each object, we selected a point (white dot) and let the robot sample three data curves passing through it.

The number of points per data curve was 21 and the spacing among the points was 1.2 mm. The tolerance δ for comparisons between principal curvatures in a table lookup was set to be 0.01 mm^{-1}. The registered data curves (red) are displayed on the model surfaces (white) in the second and fourth columns in Fig. 7. The found intersection points lie very close to the original locations.

Table VI shows the results of recognition with the tactile data curves in Fig. 7. Each cell a_{ij} in the table records the registration error $\mathcal{E}$ for data obtained on the object in the jth column onto the model in the ith row. An NC entry in the table means that a table lookup operation found *no candidate* vertices on the mesh model. In such a case, the mesh model was immediately rejected. Within each column, the minimum registration error appears in a diagonal cell. As a result, all ten objects were correctly identified.

VIII. DISCUSSION

We have presented an approach to recognition of a 3D object with curved surface out of a finite set of models using data points sampled by a touch sensor on the surface along three curves through a common point q. The problem turns into finding the model that yields the best superposition of these "data curves." The special configuration of the three data curves achieves efficiency computationally and mechanically.

- It allows us to estimate the principal curvatures at q reliably, and to use a lookup table to find a small number of surface locations with similar local geometry. This dramatically reduces the search space. Next, the span of the data curves acts as a "global" constraint to eliminate those locations whose neighborhoods do not match.
- It also ensures the data points to be acquired efficiently through tracking. Ideally, the robot hand need only drag three of its fingers across a surface region on the object. A grid of points generated by mechanical probing may be more robust for the recognition purpose. However, probing repeatedly is much slower than tracking since most of the time would be spent on the up-and-down movements between the probes.

Geometric hashing over the principal curvatures can be used to cut down the time spent on table lookups. The accuracy of data collection was 0.1 mm, which is quite achievable in contour tracking.

To focus on the recognition scheme, we have used the setup in Fig. 1 to simplify tracking control for data acquisition. The principle of recognition with "one-dimensional" data points is natural for in-hand dexterous manipulation in the following scenario. The robot hand first picks up an object by caging it, for instance, and holds the object steady while tracking its surface using one or more extra fingers. Applying the scheme to process the acquired data, the hand recognizes the object and determines its relative pose before a manipulation.

One issue is to avoid interferences between the tracking fingers and the (still) grasping fingers. The tracking fingers can also be switched via design of finger gaits [25] for stability and acquisition of more distinguishable features. For efficiency, tracking may be performed during a dexterous manipulation while tactile data are being gathered. Design of a reliable dynamic control strategy that integrates tracking is a challenging research topic and worthy of exploration.

Tracking can continue with registration of new data curves until the object is recognized. This addresses a situation where the tactile data obtained from a surface region are very similar to those on a different shape, or where a high curvature region does not yield reliable data for curvature estimation due to sensor vibrations.

Although the recognition method is presented using tactile data, it is not limited to one sensing modality. The method can be used with any type of sensor that generates contact data in a similar or denser configuration. Due to its requirement of local data, the method is more robust to occlusion than recognition based on range sensing.

Table lookup by principal curvatures renders the proposed method inapplicable to polyhedral objects. Nevertheless, most objects in reality can be well approximated as triangular meshes within the range of the introduced method, as demonstrated in Sections VI and VII.

ACKNOWLEDGMENT

The authors thank J. Tian for generating the tactile data used in our experiments. Section V of this paper was revised from an IROS paper [27], on which the review feedback is appreciated. They are grateful to the anonymous TASE reviewers for their valuable comments. Last but not least, we would like to acknowledge the review feedback over an earlier (unsuccessful) regular submission to a different journal.

REFERENCES

[1] P. K. Allen and K. S. Roberts, "Haptic object recognition using a multifingered dexterous hand," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1989, pp. 342–347.

[2] F. Arman and J. K. Aggarwal, "Model-based object recognition in denserange images—A review," *ACM Comput. Surveys*, vol. 25, no. 1, pp. 5–43, 1993.

[3] G. Beccari, S. Caselli, and F. Zanichelli, "Pose-independent recognition of convex objects from sparse tactile data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1997, pp. 3397–3402.

[4] J. S. Beis and D. G. Lowe, "Indexing without invariants in 3D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 1000–1014, Oct. 1999.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[6] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Comp. Vision Image Understanding*, vol. 81, pp. 166–210, 2001.

[7] M. Charlebois, K. Gupta, and S. Payandeh, "Shape description of curved surfaces from contact sensing using surface normals," *Int. J. Robot. Res.*, vol. 18, pp. 779–787, 1999.

[8] N. Chen, R. Rink, and H. Zhang, "Local object shape from tactile sensing," in *Proc. IEEE Int. Conf. Robotics Autom.*, 1996, pp. 3496–3501.

[9] X. Chen and F. Schmitt, "Intrinsic surface properties from surface triangulation," in *Proc. Eur. Conf. Comp. Vis.*, 1992, pp. 739–743.

[10] C. S. Chua and R. Jarvis, "3D free-form surface registration and object recognition," *Int. J. Comp. Vision*, vol. 17, pp. 77–99, 1996.

[11] C. S. Chua and R. Jarvis, "Point signatures: A new representation for 3D object recognition," *Int. J. Comp. Vision*, vol. 25, no. 1, pp. 63–85, 1997.

[12] R. S. Dahiya, G. Metta, G. Cannata, and M. Valle, "Guest editorial: Special issue on robotic sense of touch," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 385–388, 2011.

[13] C. Dorai and A. K. Jain, "COSMOS—A representation scheme for 3D free-form objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 10, pp. 1115–1130, Oct. 1997.

[14] I. Douros and B. F. Buxton, "Three-dimensional surface curvature estimation using quadric surface patches," in *Proc. Scanning*, 2002.

[15] R. E. Ellis, "Planning tactile recognition paths in two and three dimensions," *Int. J. Robot. Res.*, vol. 11, no. 2, pp. 87–111, 1992.

[16] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 27–52, 1986.

[17] R. S. Fearing, "Tactile sensing for shape interpretation," in *Dextrous Robot Hands*, S. T. Venkataraman and T. Iberall, Eds. Berlin, Germany: Springer-Verlag, 1990, pp. 209–238.

[18] P. J. Flynn and A. K. Jain, "On reliable curvature estimation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recog.*, 1989, pp. 110–116.

[19] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Eur. Conf. Comp. Vis.*, 2004, pp. 224–237.

[20] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Trans. Graphics*, vol. 25, no. 1, pp. 130–150, 2006.

[21] P. C. Gaston and T. Lozano-Pérez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 3, pp. 257–266, May 1984.

[22] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robot. Res.*, vol. 3, no. 3, pp. 3–35, 1984.

[23] E. Hameiri and I. Shimshoni, "Estimating the principal curvatures and the Darboux frame from real 3-D range data," *IEEE Trans. Syst., Man, Cybern.*, vol. 33, no. 4, pp. 626–637, Aug. 2003.

[24] M. Hebert, K. Ikeuchi, and H. Delingette, "A spherical representation for recognition of free-form surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 681–690, Jul. 1995.

[25] J. Hong, G. Lafferriere, B. Mishra, and X. Tan, "Fine manipulation with multifinger hands," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 1568–1573.

[26] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A*, vol. 4, no. 4, pp. 629–642, 1987.

[27] R. Ibrayev and Y.-B. Jia, "Surface recognition by registering data curves from touch," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2006, pp. 55–60.

[28] Y.-B. Jia, L. Mi, and J. Tian, "Surface patch reconstruction via curve sampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 1371–1377.

[29] Y.-B. Jia and J. Tian, "Surface patch reconstruction from 'onedimensional' tactile data," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 2, pp. 400–407, Apr. 2010.

[30] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
[31] D. Keren, E. Rivlin, I. Shimshoni, and I. Weiss, "Recognizing 3D objects using tactile sensing and curve invariants," *J. Math. Imaging Vision*, vol. 12, no. 1, pp. 5–23, 2000.
[32] D. J. Kriegman and J. Ponce, "On recognizing and positioning curved 3-D objects from image contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 12, pp. 1127–1137, Dec. 1990.
[33] Y. Lamdan and H. J. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," in *Proc. 2nd Int. Conf. Comp. Vision*, 1988, pp. 238–249.
[34] D. S. Meek and D. J. Walton, "On surface normal and Gaussian curvature approximations given data sampled from a smooth surface," *Comput. Aided Geometric Design*, vol. 17, no. 6, pp. 521–543, 2000.
[35] L. Mi and Y.-B. Jia, "High precision contour tracking with a joystick sensor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots. Syst.*, 2004, pp. 804–809.
[36] A. S. Mian, M. Bennamoun, and R. Owens, "Automated 3D model-based free-form object recognition," *Sensor Rev.*, vol. 24, no. 2, pp. 206–215, 2004.
[37] A. S. Mian, M. Bennamoun, and R. A. Owens, "A novel algorithm for automatic 3D model-based free-form object recognition," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2004, pp. 6348–6353.
[38] J. L. Mundy and A. Zisserman, "Introduction—Towards a new framework for vision," in *Geometric Invariance in Comput. Vision*, J. L. Mundy and A. Zisserman, Eds. Cambridge, MA: The MIT Press, 1992, pp. 1–39.
[39] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[40] A. Pressley, *Elementary Differential Geometry*. London, U.K.: Springer-Verlag, 2001.
[41] S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," in *Proc. Symp. 3D Data Processing, Visualization, and Transmission.*, 2004, pp. 486–493.
[42] J. T. Schwartz and M. Sharir, "Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 29–44, 1987.
[43] Q. Shi, N. Xi, Y. Chen, and W. Sheng, "Registration of point clouds for 3D shape inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2006, pp. 235–240.
[44] F. Stein and G. Medioni, "Structural indexing: Efficient 3-D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 125–145, Feb. 1992.
[45] E. M. Stokely and S. Y. Wu, "Surface parametrization and curvature measurement of arbitrary 3-D objects: Five practical methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 833–840, Aug. 1992.
[46] Y. Sun, J. Paik, A. Koschan, D. L. Page, and M. A. Abidi, "Point fingerprint: A new 3D object representation scheme," *IEEE Trans. Sys., Man, Cybern.*, vol. 33, no. 4, pp. 712–717, 2003.
[47] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin, "A comparison of Gaussian and mean curvatures estimation methods on triangular meshes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, pp. 1021–1026.
[48] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," in *Proc. Int. Conf. Comp. Vis.*, 1995, pp. 902–907.
[49] B. Vijayakumar, D. Kriegman, and J. Ponce, "Invariant-based recognition of complex curved 3D objects from image contours," *Comp. Vision and Image Understanding*, vol. 72, no. 3, pp. 287–303, 1998.
[50] I. Weiss, "Geometric invariants and object recognition," *Int. J. Comput. Vision*, vol. 10, no. 3, pp. 207–231, 1993.
[51] I. Weiss, "Model-based recognition of 3D objects from single images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 116–128, Feb. 2001.
[52] S. M. Yamany and A. A. Farag, "Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1105–1120, Aug. 2002.
[53] J. Zhang, X. Zhang, H. Krim, and G. G. Walter, "Object representation and recognition in shape spaces," *Pattern Recognit.*, vol. 36, pp. 1143–1154, 2003.

Kalman Filter-Based Coarse-to-Fine Control for Display Visual Alignment Systems

SangJoo Kwon*, *Member, IEEE*, Haemin Jeong, and Jaewoong Hwang

***Abstract*—A coarse-to-fine two-stage control method is investigated for the display visual alignment systems. The proposed visual servo is with hierarchical loops, where the original fine but slow vision loop is necessary for the exact localization of alignment marks while the coarse but fast vision loop of exploiting pruned image data is to compensate for the mask-panel misalignment. The degraded resolution of the reduced images is recovered in terms of the Kalman filter which tracks the mark centroids in near real-time. In order to construct the recursive estimation algorithm, the motion model for the moving alignment marks is determined by solving the forward kinematics of positioning mechanism and the measurements from vision sensors are given by means of the geometric template matching (Kwon and Hwang, "Kinematics, pattern recognition, and motion control of mask-panel alignment system," *Control Eng. Practice*, vol. 19, pp. 883–892, 2011). Compared with the conventional alignment methods, this approach enables a fast and fine alignment control. Experimental results are followed to validate the proposed control framework.**

***Note to Practitioners*—In order to successfully apply the developed alignment control to any display manufacturing equipment, it is necessary to well understand the principle of the geometric template matching (GTM) as an alignment mark specific fast algorithm, the details on which can be consulted in our preceding works (Kwon and Hwang, "Kinematics, pattern recognition, and motion control of mask-panel alignment system," *Control Eng. Practice*, vol. 19, pp. 883–892, 2011). The new approach has the goal of updating the pose of an alignment mark as fast as the capturing rate of a frame grabber by utilizing pruned image data but recovering the lost resolution in terms of the Kalman filter. For example, in using a common 30 fps grabber, the reduced image of 320 $\times$ 240 pixels is a proper choice to finish the image processing and Kalman filtering within 30 ms under GTM. The proposed algorithm can be implemented in the current industrial display aligners by modifying the control software so that the reference inputs for the distributed joint servos follow the error compensation trajectory in Fig. 4 with the Kalman filter estimates.**

***Index Terms*—Flat panel display, image processing, Kalman filter, observer-based control, visual alignment.**

I. Introduction

In THE flat panel display industry, the mask-panel visual alignment system is operating to correct the misalignment between mask and panel prior to performing a specific manufacturing process. The efficiency of the autonomous visual alignment system is a critical problem to determine the productivity in any manufacturing equipment which includes the mask-panel alignment process as well as in

Manuscript received November 30, 2010; revised November 21, 2011; accepted April 18, 2012. Date of publication May 15, 2012; date of current version June 28, 2012. This paper was recommended for publication by Associate Editor C.-F. Chien and Editor K. Goldberg upon evaluation of the reviewers' comments. This work was supported by the Korea National Research Foundation under contract number 2009-0077203. This was presented in part at the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Singapore, July 14–17, 2009. *Asterisk indicates corresponding author.*

*S. J. Kwon is with the School of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang 412-791, Korea (e-mail: sjkwon@kau.ac.kr).

H. Jeong and J. Hwang are with the School of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang 412-791, Korea (e-mail: {seamanjhm,bluelead}@kau.ac.kr, +82-2-300-0178).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2012.2196693