Robust Estimation of Curvature Information from Noisy 3D Data for Shape Description*

Chi-Keung Tang and Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089-0273
{chitang,medioni}@iris.usc.edu

Abstract

We describe an effective and novel approach to infer sign and direction of principal curvatures at each input site from noisy 3D data. Unlike most previous approaches, no local surface fitting, partial derivative computation of any kind, nor oriented *normal vector recovery is performed in our method. These approaches are noise-sensitive since accurate, local, partial derivative information is often required, which is usually unavailable from real data because of the unavoidable outlier noise inherent in many measurement phases. Also, we can handle points with zero Gaussian curvature* uniformly *(i.e., without the need to localize and handle them first as a separate process). Our approach is based on* **Tensor Voting**, *a unified, salient structure inference process. Both the sign and the direction of principal curvatures are inferred* directly *from the input. Each input is first transformed into a synthetic tensor. A novel and robust approach based on tensor voting is proposed for curvature information estimation. With faithfully inferred curvature information, each input ellipsoid is aligned with curvature-based dense tensor kernels to produce a dense tensor field. Surfaces and crease curves are extracted from this dense field, by using an extremal feature extraction process. The computation is non-iterative, does not require initialization, and robust to considerable amounts of outlier noise as its effect is reduced by collecting a large number of tensor votes. Qualitative and quantitative results on synthetic as well as real and complex data are presented.*

1 Introduction

Curvature information gives a unique, viewpoint independent description for local shape [1]. In differential geometry, it is well known that a surface can be reconstructed up to second order (except for a constant term) if the two principal curvatures at each point is known, by using the first and second fundamental forms [4]. Therefore, curvature information provides a useful shape descriptor for various tasks in computer vision, ranging from image segmentation and feature extraction, to scene analysis and object recognition. A previous effort [10] proposes a procedure for integrating smooth surfaces and junction curves. While good results are obtained, without the use of explicit curvature information, a rather complex coordination process among detected features needs to be implemented. Here, the contribution of this paper is twofold:

- A robust method is proposed for the accurate estimation of sign and direction of principal curvatures. Experiments are performed on synthetic and real data with large amount of outlier noise (as much as 500% noise, i.e., one out of six points is good).
- By using explicit curvature information, the process of integrating smooth surfaces and junction curves can be simplified (c.f. [10]), since we know which side w.r.t. each estimated surface normal (oriented or not) the surface to be inferred should locally curve to. This allows tensor votes to propagate in the proper and preferred direction.

When considered alone, our curvature estimation can be regarded as a plug-in to other applications, which labels each data point as locally planar, elliptic, parabolic, or hyperbolic, at a discontinuity or an outlier. It runs fast, in $O(nk)$ time where n is input size and k is neighborhood size (a detailed complexity analysis is found in [10], which also applies here since our curvature estimation is also implemented as tensor voting). When used as described in this paper, it provides a stronger mathematical basis to integrate smooth surfaces and junction curves, or surface orientation discontinuities.

1.1 Previous Work

A detailed treatment of curvature can be found in a classical differential geometry text by do Carmo [3]. Despite the extensive study on recovery of curvature information from range data and other data sources, results are still not satisfactory. One technique involves fitting a local surface patch, and computing partial second order derivatives from it [2, 8, 9, 14]. Derivative computation is unstable in real data, and the estimated curvature is thus very noise-sensitive. Another approach recovers principal curvatures and direction from range data, by collecting four directional curvatures at 45° apart [4]. Unfortunately, these directional curvatures also rely on accurate, local first and second order partial derivatives, which are often unavailable in real data.

Another methodology involves recovery of surface normal vectors from the data, and then usually followed by a local surface patch fitting. With normal information, a better fit, and thus a better curvature estimate, may be obtained. Shi et al. [9] diagonalize a scatter matrix for normal estimation from a set of sampled surface points. A least square process for fitting

[*] This research is supported by the National Science Foundation under grant no. 9811883.

a local quadric patch is then followed. Unfortunately, parameterizing such local patch requires that the orientation of the estimated normals be consistent throughout the whole surface, which is either unavailable, or has to be estimated separately.

Rather than numerically computing curvature information, another approach involves estimation of the sign of Gaussian curvature. For example, Angelopoulou and Wolff [1] compute the sign of Gaussian curvature, without surface fitting, local derivative computation, nor normal recovery. Sign of Gaussian curvature is determined by checking the relative orientation of two simple, local closed curves (one from the surface and one from its corresponding curve on the Gaussian sphere) is preserving or otherwise reversing. However, zero Gaussian curvature areas need to be first located in a separate process.

Empirical analyses on curvature estimation are reported in literature. The classical paper by Flynn and Jain [5] evaluates five methods of curvature estimation. The conclusion of the experiments performed by Trucco and Fisher [13] agree with [5]: qualitative curvature properties (e.g. sign of Gaussian curvature) can be more reliably estimated than quantitative ones (e.g. curvature magnitude). Our method agrees with their conclusion, in addition we show (by example) that principal directions can also be estimated fairly robustly by our method, while this estimation is not addressed in [5, 13].

1.2 Overview of Our Approach

Our method robustly recovers the sign and direction of principal curvatures for surface reconstruction *directly* from 3D data, without surface fitting nor partial derivative computation of any kind. Zero curvature areas are detected and handled *uniformly*, by using homogeneous coordinates (section 3.1). The basis of our method is grounded on two elements: local structures are uniformly represented by a second order symmetric **tensor**, which effectively encodes preferred direction, while avoiding early decision on normal orientations and maintenance of global orientation consistency. Data communication is accomplished by a linear **voting** process, which simultaneously ignores outlier noise, corrects erroneous orientation (if given), and detects surface orientation discontinuities. While approaches at one extreme completely trust the estimated normals (e.g. [9]), and methods at the other extreme completely bypass the normal recovery process (e.g. [1]), our method is more flexible: it makes use of reliably inferred surface orientation information, corrects erroneous normals, and ignores inconsistent votes.

The use of a linear voting process for feature inference from *sparse* and noisy data was first introduced by Guy and Medioni [6], formalized into a unified tensor framework in [7], and later modified for feature integration in [10]. The methodology is non-iterative and robust to considerable amount of outlier noise. The only free parameter is the scale of analysis, which is indeed a property of human perception. Input data is made to align (votes) with precomputed dense tensor kernels, by propagating preferred information in a neighborhood. This preferred information includes proximity, smoothness, continuity, preferred surface orientation, and *low and constancy of curvature*. We shall detail these issues in section 2.

Many affordable laser range finders can now produce *dense* and *mostly* accurate information. However, outlier noise is still unavoidable in the measurement phase (Figures 14 and 16). With dense, mostly accurate but imperfect data, if we can *robustly estimate curvature information, and use it to tune the dense voting kernels accordingly in voting*, more reliable surface reconstruction should result. To elaborate, suppose that we vote near the endpoints of a circular arc in 2D (Figure 1) to fill the gap. Without the use of curvature information, using Guy and Medioni's method, a circle will not be produced since straight connection (zero curvature) is preferred.

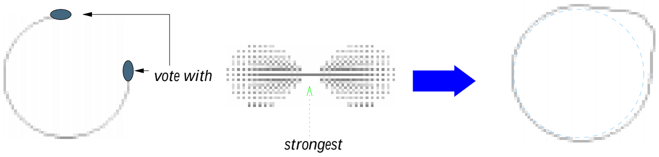


Figure 1 *Voting without curvature information.*

We summarize the tensor voting formalism in this section, and refer interested readers to [6, 7] for details. Figure 2 depicts the overall strategy. The input can be points, edgels, or normals, or any combination of them, which will be first quantized and then *"tensorized"* to produce a discrete tensor field. This tensor field will then be enhanced or *"densified"* for subsequent surface and curve extraction. A new, voting-based processing for curvature estimation is added between tensorization and densification, which will be described in the next section. Tensor voting is grounded on tensors for representation, and voting for data communication.

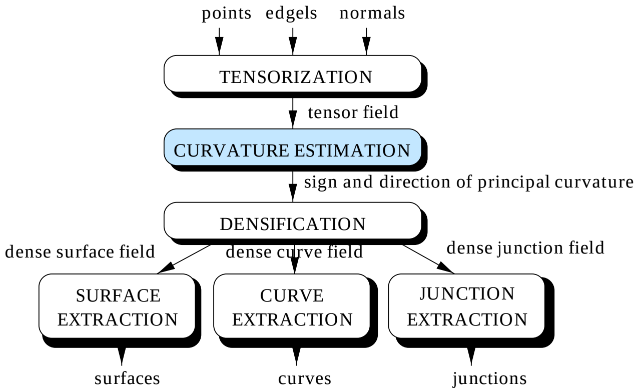


Figure 2 *Overall approach.*

2 The Tensor Voting Formalism

Tensor representation A point in the 3D space can assume either one of the three roles: surface patch, discontinuity (curve or point junctions), or outlier. Consider the two extremes, in which a point on a smooth surface is very certain about its surface (normal) orientation, whereas a point on a (curve or point) junction has absolute orientation uncertainty. This whole continuum can be abstracted as a second order symmetric 3D *tensor*, which can be visualized geometrically as an **ellipsoid**. Such an ellipsoid can be fully described by the corresponding eigensystem with its three unit eigenvectors $\hat{V}_{max}$, $\hat{V}_{mid}$, and $\hat{V}_{min}$ and the three corresponding eigenvalues $\lambda_{max} \geq \lambda_{mid} \geq \lambda_{min}$. Rearranging the eigensystem, the 3D ellipsoid is given by: $(\lambda_{max} - \lambda_{mid})\mathbf{S} + (\lambda_{mid} - \lambda_{min})\mathbf{P} + \lambda_{min}\mathbf{B}$, where $\mathbf{S} = \hat{V}_{max}\hat{V}_{max}^T$ defines a *stick tensor*, $\mathbf{P} = \hat{V}_{max}\hat{V}_{max}^T + \hat{V}_{mid}\hat{V}_{mid}^T$ defines a *plate tensor*, and $\mathbf{B} = \hat{V}_{max}\hat{V}_{max}^T + \hat{V}_{mid}\hat{V}_{mid}^T + \hat{V}_{min}\hat{V}_{min}^T$ gives a *ball tensor*. These tensors define the three *basis* tensors for any 3D ellipsoid. See Figure 3.

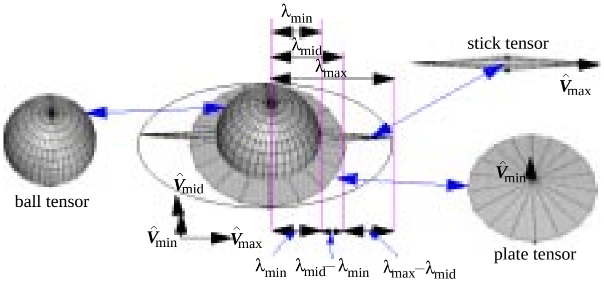


Figure 3 *A second order symmetric 3D tensor.*

Geometric Interpretation The *eigenvectors* effectively encode *orientation (un)certainties*: *surface* orientation (normal) is described by the *stick* tensor, which indicates *certainty* along a single direction (a line). *Uncertainties* are abstracted by two other tensors: *curve* junction is resulted from two intersecting surfaces, where the uncertainty in orientation only spans a single *plane* perpendicular to the tangent of the junction curve, and is thus described by a *plate* tensor. At *point* junctions where more than two intersecting surfaces are present, a *ball* tensor is used since there is no preferred orientation. The *eigenvalues*, on the other hand, effectively encode the *magnitudes* of orientation (un)certainties, since they indicate the size of the corresponding ellipsoid.

Therefore, after the eigensystem analysis, three *dense vector maps* are defined. Each voxel of these maps has a 2-tuple $(s, \overline{v})$, where s is a scalar indicating feature saliency or strength, and $\overline{v}$ is a unit vector indicating direction[1]:

- *Surface map* (*SMap*): $s = \lambda_{max} - \lambda_{mid}$, and $\overline{v} = \hat{V}_{max}$ indicates the normal direction.
- *Curve map* (*CMap*): $s = \lambda_{mid} - \lambda_{min}$, and $\overline{v} = \hat{V}_{min}$ indicates the tangent direction.
- *Junction map* (*JMap*): $s = \lambda_{min}$, $\overline{v}$ is arbitrary.

Densification by voting Having defined the tensor formalism, we are now ready to describe the *voting* algorithm for obtaining the tensor representation at *each* voxel that gives the above dense vector maps, thus achieving densification. *Suppose the data has been tensorized* (tensorization will be described shortly), in which stick components indicate preferred normal directions. These input tensors **votes**, or are made to align (by translation and rotation), with predefined, discrete versions of the three basis tensors (we shall call them voting *kernels*) in a convolution-like way.

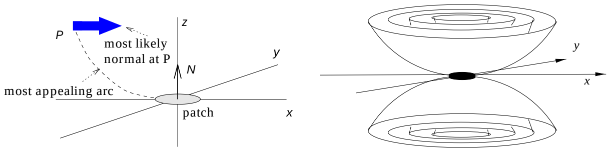


Figure 4 *The design of the stick kernel.*

Here, we explain the design of the stick kernel, which is a key component in curvature estimation. Suppose we have a normal N at the origin O and a point P (Figure 4). A spherical surface is chosen as the "most likely" continuation between O and P, since curvature along the circular arc connecting O and P is kept constant (this implicitly encodes the smoothness constraint as well). We say that O with normal N has cast a *vote* at P. This vote consists of a stick tensor (a unit vector) $\overline{v}$ indicating the preferred *direction*, and a scalar indicating its *strength*, or surface *saliency*. The direction of $\overline{v}$ is given by the normal of the hypothetical sphere at P (shown as the thick arrow in Figure 4). (Note that the vote $\overline{v}$ at P and N at O lie on the *same* plane.) The strength, or saliency, is attenuated by the following decay:

$$DF(P) = e^{-(\frac{s^2+c\rho^2}{\sigma^2})} \tag{1}$$

where s is the arc length connecting O and P (therefore, s encodes proximity), ρ is the curvature, c is a constant (chosen *a priori*, *not* a parameter) that controls the amount of attenuation with higher curvature, and σ is the scale of analysis (the only parameter). The set of all such votes in the 3D space is discretized and collected as the stick kernel. (Figure 4 also shows the general shape of the stick kernel, depicting a family of hypothetical spherical surfaces voting for smooth connection. Figure 5 shows one slice of the stick kernel at $y = 0$ plane.)

The *plate* and the *ball* kernels are dense isotropic tensor fields. Please refer to [7] for details. Figure 5 also depicts one slice of the plate kernel (cut along the $x = 0$ plane). An input tensor is said to *cast its stick votes*, or *votes with the stick kernel*, when its stick component is aligned with the stick kernel.

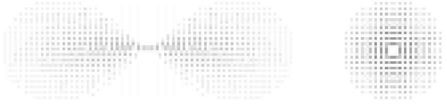


Figure 5 *One slice of stick (on the left) and plate kernel.*

When each input tensor has cast their stick, plate, and ball votes to neighboring voxels by aligning with the respective *dense* basis kernels, each voxel in the volume receives a set of directed votes. These directed votes are collected, using tensor addition, as a 3 × 3 covariance matrix. We diagonalize this matrix into the corresponding eigensystem. Then, the tensor formalism is applied.

Tensorization by Voting In practice, we may have points, edgels, normals, or any combination of them, as input. First, the input is *unified* by encoding it as a tensor field. For scalar input, each data point is encoded as a ball tensor since initially there is no preferred orientation. For edgels, they are encoded as plate tensors. For normals, they are represented by stick tensors, which are actually very elongated ellipsoids. Then, these input tensors will vote *exactly* as described above, except that votes are only collected at the input tensors (but not everywhere in its neighborhood as in densification). After the voting step, the original input is replaced by true ellipsoids, which encode preferred surface normal and curve tangent information.

[1] Note that since a second order symmetric tensor is used, the resultant eigenvector only indicates the direction along $\overline{v}$, hence there are still two choices of orientation.

Erroneous input normals are thus discarded and replaced with these inferred ellipsoids.

Extremal Surface and Curve Extraction After dense feature maps have been obtained, we extract features in terms of coherent surfaces and 3D space curves from them. In essence, we detect zero crossings using an elaborate process defined by *saliency extrema*, and group these zero crossings into surface patches or curve segments, by using a modified marching process. This marching process differs from the Marching Cubes algorithm since the orientation of the normals in SMap $\{(s,\overline{v})\}$ need not be known in advance or consistently oriented. For this reason, our curvature estimation does not distinguish if a point is convex or concave, if its estimated curvature label (next section) is parabolic or elliptic. We refer readers to [10, 11] for more detail on extremal surface and curve algorithms.

3 Surface Curvature by Tensor Voting

When the input has been tensorized, each input site (point, edgel, normal) is replaced by a set of true ellipsoids encoding preferred surface normal and curve tangent information. Each input tensor will vote again for estimating sign and direction of principal curvatures (the shaded process in Figure 2).

3.1 Sign of Principal Curvature Estimation

After tensorization, we first analyze each input ellipsoid, by tensor voting, to label it as locally (a) *planar*, (b) *elliptic*, (c) *parabolic*, or (d) *hyperbolic, an outlier, or a discontinuity*. Except for the last case, when this process is done, each input site will locally know which *side* w.r.t. its stick component the surface should locally curve to. Therefore, curvature-based stick kernels will be generated for densification that followed. Thus, we do not need to vote at both sides of stick as shown in Figure 4, in which no explicit curvature information is used.

Representation of sign of curvature For each input tensor, we arbitrarily pick an orientation out of the two choices of its stick component $\hat{V}_{max}$ as reference. Then, we align with it a local coordinate system. The sign of curvature is indicated by the side w.r.t. the oriented $\hat{V}_{max}$ the surface should locally curve to. *Homogeneous coordinates* are used to represent the sign of curvature so that zero curvature can be handled *uniformly*.

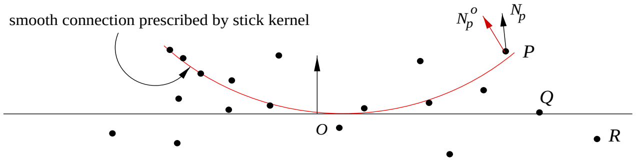


Figure 6 *Estimating sign of curvature by tensor voting.*

To elaborate, refer to Figure 6. Let O be the input site with arbitrarily oriented stick $\hat{V}_{max}$. W.l.o.g., consider P, Q, and R in the neighborhood of O, $nbhd(O)$, as specified by $DF(\cdot)$. Let N_p (resp. N_q and N_r) be the stick component of the input ellipsoid at P (resp. Q and R) obtained after tensorization, and *suppose* that O casts a stick vote with direction N_p^o (resp. N_q^o and N_r^o) which is received at P (resp. Q and R), by alignment with the stick kernel. We say that O *induces* sign of curvature votes at P, Q, R, which are collected at O. The direction and strength of these votes are defined as tabulated as follows:

Sign of curvature vote collected at O

	P (above)	Q (on)	R (below)
direction	$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$	$\begin{bmatrix}0\\1\end{bmatrix}$	$\frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}$
strength	$\frac{\lvert\frac{N_p^o}{\lVert N_p^o\rVert}\cdot\frac{N_p}{\lVert N_p\rVert}\rvert}{dist(O,P)}$	$\frac{\lvert\frac{N_q^o}{\lVert N_q^o\rVert}\cdot\frac{N_q}{\lVert N_q\rVert}\rvert}{dist(O,Q)}$	$\frac{\lvert\frac{N_r^o}{\lVert N_r^o\rVert}\cdot\frac{N_r}{\lVert N_r\rVert}\rvert}{dist(O,R)}$

where $dist(\cdot)$ is the Euclidean distance. The dot product in the vote strength definition above indicates *vote consistency*, i.e., if the magnitude of the dot product is close to zero, it means that the stick vote N_p^o (resp. N_q^o and N_r^o) supposedly cast by O is not consistent with N_p (resp. N_q and N_r). One or both of the followings occur:

- the ellipsoid (or its stick component) inferred during tensorization at O or at the point is not accurate
- the smoothness constraint prescribed by the stick kernel is not satisfied

These situations may arise for severely corrupted data or at a discontinuity. In either case, the vote is unreliable and should be ignored by the dot product.

Vote Collection O is a vote collector, aggregating sign of curvature vote (with its direction and strength defined above) cast by a point P in a neighborhood. Denote such vote by $\overline{v_p}$, where $||\overline{v_p}||$ defines the vote strength as above. The voting process is exactly the same as described in section 2, but the interpretation different, as explained in the following. We compute the (sample) mean M and covariance matrix $\mathbf{S}$ of the vote distribution. First, we let

$$M = \begin{bmatrix} M_x \\ M_y \end{bmatrix} = \frac{1}{n}\sum_{P\in nbhd(O)} \overline{v_p} \quad (2)$$

$$\mu = \frac{M_x}{M_y} \qquad \text{(note that } M_y > 0\text{)} \quad (3)$$

where $n = \#P, P \in nbhd(O)$. For $p = 1, 2, \cdots, n$, define

$$\overline{v'_p} = \overline{v_p} - M \quad (4)$$

$$\mathbf{B} = \left[\ \overline{v'_1} \quad \overline{v'_2} \quad \cdots \quad \overline{v'_n}\ \right] \quad (5)$$

Therefore, $\overline{v'_p}$ denotes the deviation of $\overline{v_p}$ from the “weighted-averaged sign” M. Using $\mathbf{B}$, we compute the covariance matrix $\mathbf{S}$ as the 2×2 positive semidefinite matrix, and the total variance $\sum$ of $\mathbf{S}$ is the trace of $\mathbf{S}$, as follows:

$$\mathbf{S} = \frac{1}{n-1}\mathbf{B}\mathbf{B}^T \quad (6)$$

$$\sum = trace(\mathbf{S}) \quad (7)$$

Geometric interpretation μ and $\sum$ together indicate which side w.r.t. to the stick at O the (recovered) surface should locally curve to. We have the following cases (Figure 7):

- $|\mu| \approx 0, \sum \approx 0$. Curvature is zero (i.e. within an empirical tolerance) in $nbhd(O)$. It indicates that O is locally *planar* (Figure 7(a)).
- $|\mu| \not\approx 0, \sum \approx 0$. Points in $nbhd(O)$ prefers *smooth* connection ($\sum \approx 0$), on only *one side* of the stick at O ($|\mu| \not\approx 0$). It indicates that O is locally *elliptic* (Figure 7(b)).

•$|\mu| \approx 0, \sum \not\approx 0$. Curvature votes cancel out each other, as indicated by non-zero $\sum$. Points in $nbhd(O)$ do not prefer either side of the stick, which indicates that O is locally *hyperbolic*, or an outlier, or a discontinuity (Figure 7(c)).

•$|\mu| \not\approx 0, \sum \not\approx 0$. Curvature votes indicate that one side of the stick is preferred. Yet, there exists votes with zero curvature (implied by non-zero $\sum$). This indicates that O is locally *parabolic* (Figure 7(d)).

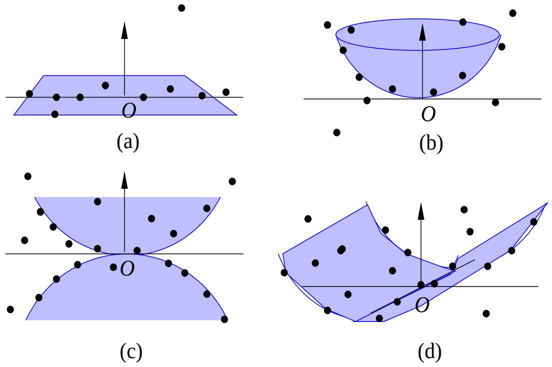


Figure 7 *Geometric interpretation of vote collection.*

If O is inferred to be locally hyperbolic, an outlier, or a discontinuity, then we use the original dense stick kernel defined in section 2 for densification, because of the inconclusive vote. Note, despite that the sign estimates are unreliable at regions of surface orientation discontinuities, these *singularities* are detected as point and curve junctions, which are characterized by a high disagreement of oriented votes collected at such sites. No surface patch is produced in these regions of low surface saliencies where surface saliency extremum does not exist.

If O is labeled as locally planar, we use the original kernel, but redefine DF to impose more decay with high curvature.

3.2 Estimation of Principal Directions

Vote Definition We define the vote for principal directions as follows. Refer to Figure 8. Let N_p be the stick component of the ellipsoid inferred at P after tensorization, and that it casts a vote (using the stick kernel) which is received at Q. Let N be the direction of this stick vote. Then, by the definition of the stick kernel, N at Q, N_p at P lie on the same plane. We denote this plane by Π_{PQ}.

Let N_q be the stick component of the ellipsoid inferred at Q after tensorization. Let T_q be the tangent plane at Q, by *assuming* that N_q is the direction of surface normal. Then, $N_q \perp T_q$ (Figure 8). The reliability of this assumption is indicated by vote inconsistency.

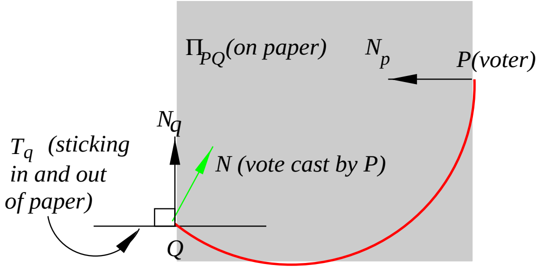


Figure 8 *Vote for principal directions using the stick kernel.*

If $T_q \cap \Pi_{PQ} \neq \emptyset$ (otherwise we simply skip the following), this intersection gives a line. We define the direction and strength of the vote as follows:

Direction. We put a vote $\overline{v}$ along the line $T_q \cap \Pi_{PQ}$, because P "curves to" Q along this direction (Figure 9).

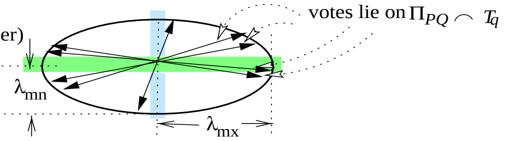


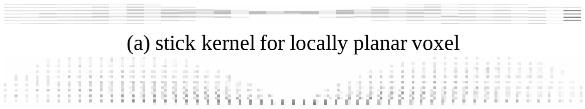
Figure 9 *Principal direction vote on T_q.*

Strength. Scale and project the curvature along the circular arc (denote it by ρ) onto N_q by $\frac{|\rho|}{dist(P,Q)}|\frac{N}{||N||} \cdot \frac{N_q}{||N_q||}|$, which prefers directions on the tangent plane T_q which curve *more* (as indicated by $|\rho|$), giving more weight to such directions. In other words, we vote for *maximum direction* (as the minimum direction is merely 90° off). If $\frac{N}{||N||} \cdot \frac{N_q}{||N_q||} \approx 0$, it means that the stick vote N cast by P is not consistent with N_q (this unreliable vote will thus be ignored).

Vote Collection The votes $\overline{v}$'s on the tangent plane T_q are collected as a second order symmetric tensor, or equivalently as an *ellipse*. This is analogous to the 3D tensor voting formalism (Figure 3). This topological ellipse (Figure 9) describes the equivalent eigensystem with its two unit eigenvectors $\hat{V}_{mx}$ and $\hat{V}_{mn}$ and the two corresponding eigenvalues $\lambda_{mx} \geq \lambda_{mn}$. Rearranging the eigensystem, the ellipse is given by: $(\lambda_{mx} - \lambda_{mn})\mathbf{S} + \lambda_{mn}\mathbf{P}$, where $\mathbf{S} = \hat{V}_{mx}\hat{V}_{mx}^T$ defines a *stick tensor*, and $\mathbf{P} = \hat{V}_{mx}\hat{V}_{mx}^T + \hat{V}_{mn}\hat{V}_{mn}^T$ defines a *plate tensor*, in 2D.

Geometric Interpretation Here, the *eigenvectors* denote the *principal directions*: $\hat{V}_{mx}$ (resp. $\hat{V}_{mn}$) gives the maximum (resp. minimum) direction. The eigenvalues, however, do *not* indicate the magnitude of principal curvatures, because the value of curvature at a point in direction ϕ is given by [4]: $k_{max}\cos^2(\phi - \alpha) + k_{min}\sin^2(\phi - \alpha)$ (α is the principal curvature directions) which does *not* follow an elliptic distribution, and thus cannot be described by a second order symmetric 2D tensor (ellipse). In fact, the general shape of the distribution resembles that of a peanut [4], not an ellipse, and that our experiments indicate an over-estimation (resp. under-estimation) of maximum (resp. minimum) curvature if we assume elliptic distribution. However, given that the strength of votes is properly defined, this elliptic distribution assumption should still valid for approximating principal directions, if only the direction of the resultant eigensystem is considered.

3.3 Curvature-based Stick Kernels



(a) stick kernel for locally planar voxel

(b) stick kernel for locally parabolic or elliptic voxel

Figure 10 *Curvature-based stick kernels.*

Since we can only estimate sign and direction of principal curvatures reliably, but not the magnitude, we only use the *sign* information to derive the curvature-based stick kernel for densification and coherent surface and curve extraction. Magnitude of curvature is known to be hard to estimate, because of the instability of second order estimates. Tensor voting does not involve such unstable, partial derivative computation. Also, knowing the principal directions should help recovering the magnitude, which is the subject of future research. According to the label inferred for each input ellipsoid, we have the following cases:

- *Hyperbolic region, outlier or discontinuity.* Voxels labeled as hyperbolic, outlier or discontinuity are characterized by inconclusive curvature votes. We use the original stick kernel as defined in section 2 for densification.

- *Planar.* If voxels are labeled as locally planar, we use the same stick kernel, but impose more decay with high curvature, i.e. $DF(P) = e^{-(\frac{s^2+Cp^2}{\sigma^2})}, C \gg c$ where c is the attenuation factor in Equation (1). Figure 10(a) shows one slice (at $y = 0$) of this stick kernel. Note the "thinness" of the kernel, as opposed to the kernel in Figure 5. This kernel prefers planar connection.

- *Parabolic or elliptic.* If voxels are labeled as locally parabolic or elliptic, we only consider the set of directions and strengths of the stick kernel for which $sgn(\mu)sgn(\rho) > 0$, i.e., one side of the stick kernel, upon kernel alignment.

4 Robustness of Sign Estimation

We perform experiments on synthetic data to evaluate the accuracy and robustness of our tensor voting approach to curvature estimation.

Accuracy of Labeling Point samples are collected from a spherical, a parabolic, and a hyperbolic surface. We add a large amount of random noise to the bounding box of each shape. The following table summarizes the result. Note that over 95% of accuracy is still reported with as much as 200% of outliers, i.e., only one out of three points is correct (Underlined (resp. plain) text on the leftmost column indicates the percentage of correct (resp. incorrect) labeling.)

	SPHERE (489 POINTS)			
	50% noise	*100% noise*	*150% noise*	*200% noise*
elliptic	100.00%	100.00%	99.40%	95.88%
parabolic	0.00%	0.00%	0.00%	4.12%
planar	0.00%	0.00%	0.60%	0.00%
hyperbolic	0.00%	0.00%	0.00%	0.00%
	CYLINDER (3844 POINTS)			
	50% noise	*100% noise*	*150% noise*	*200% noise*
elliptic	0.00%	0.00%	0.00%	0.00%
parabolic	97.89%	97.41%	99.60%	96.30%
planar	0.17%	1.85%	0.17%	2.72%
hyperbolic	1.94%	0.74%	0.23%	0.99%
	SADDLE (605 POINTS)			
	50% noise	*100% noise*	*150% noise*	*200% noise*
elliptic	0.00%	0.33%	0.67%	0.50%
parabolic	0.17%	0.00%	0.00%	0.00%
planar	0.00%	0.00%	0.00%	0.00%
hyperbolic	99.83%	99.67%	99.33%	99.50%

Grouping by Curvature Here, we demonstrate the use of faithfully inferred curvature information for performing scene segmentation in more complex synthetic scenario. By making use of curvature, we can perform the integrated surface inference in a less complex way than [10].

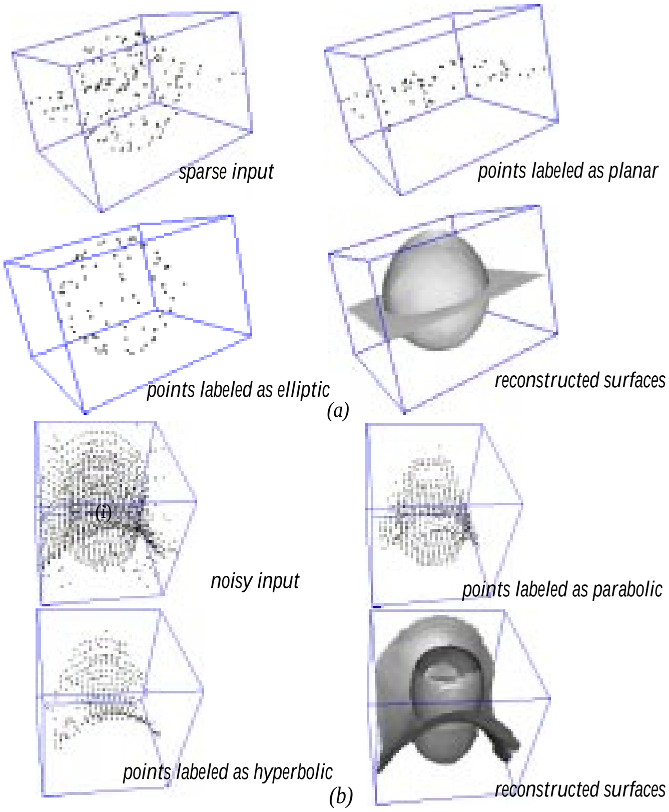


Figure 11 *Results on synthetic scene segmentation and surface reconstruction using estimated curvature information for (a) sparse plane-sphere, and (b) noisy saddle-cylinder. Using curvature information, we obtain comparable results with [10] using a less complex method.*

(A) SPARSE PLANE-SPHERE. A sparse set of points is sampled from a sphere intersecting with a plane. Having accurate curvature estimation, we can segment the scene into the corresponding spherical and planar components, and then reconstruct the corresponding surfaces (Figure 11(a)). A more elaborate and complex process is needed in [10] to obtain comparable result.

(B) NOISY SADDLE-CYLINDER. We sample data from a hyperbolic surface intersecting with a cylindrical surface, and randomly add a considerable amount of outliers in the volume. We can recover curvature information, segment the input features, and derive the underlying surface descriptions (Figure 11(b)).

5 Robustness of Estimation of Principal Directions

A total of 3969 point samples are obtained from a *toroidal* surface, a genus-one object. We add a large amount of (up to 1200%) outlier noise in the corresponding bounding box. Figure 12 is a histogram showing the accuracy of the estimated principal directions at different noise levels. Figure 13 shows the noisy input and the reconstructed surface. Our method degrades gracefully with increasing amount of outliers. Note that the reconstructed surface can be extracted even up to 400% noise, and most part of it can still be extracted in the presence of as much as 1200% noise.

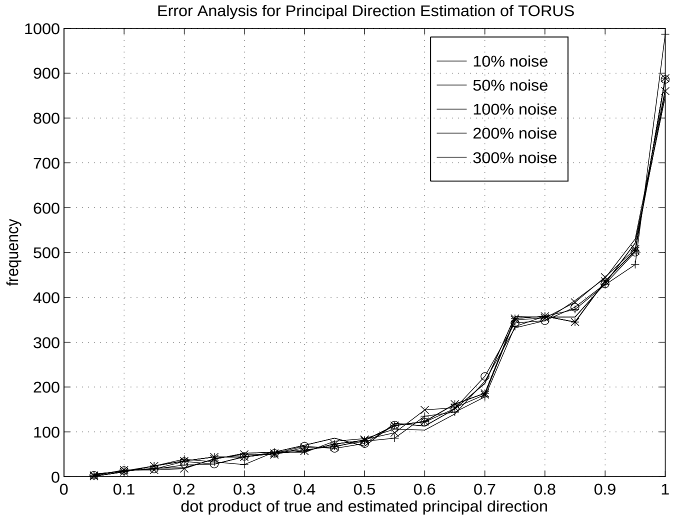
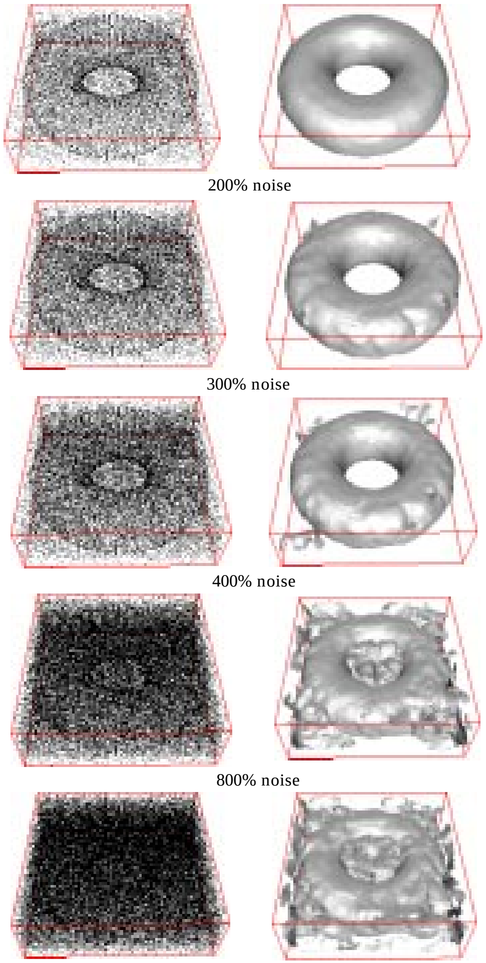


Figure 12 *Graceful degradation of the estimated curvature direction for* torus. *The estimated direction is not adversely affected by outlier noise.*


200% noise

300% noise

400% noise

800% noise

1200% noise

Figure 13 *Noisy input and the corresponding reconstructed surface for* torus, *showing the graceful degradation of our method.*

6 More Results on Real Data

A good choice of *real* data for evaluating our system is data for dental CAD/CAM. A patient-specific dental restoration involves complex, *distinct* but close-by surfaces. For example, a *crown* restoration consists of an upper surface (on which food is chewed); and a lower surface to which the patient's tooth is fused, along a very precise preparation line (which is only implicit in the input point set) that should *not be smoothed out* during the inference of the surface model.

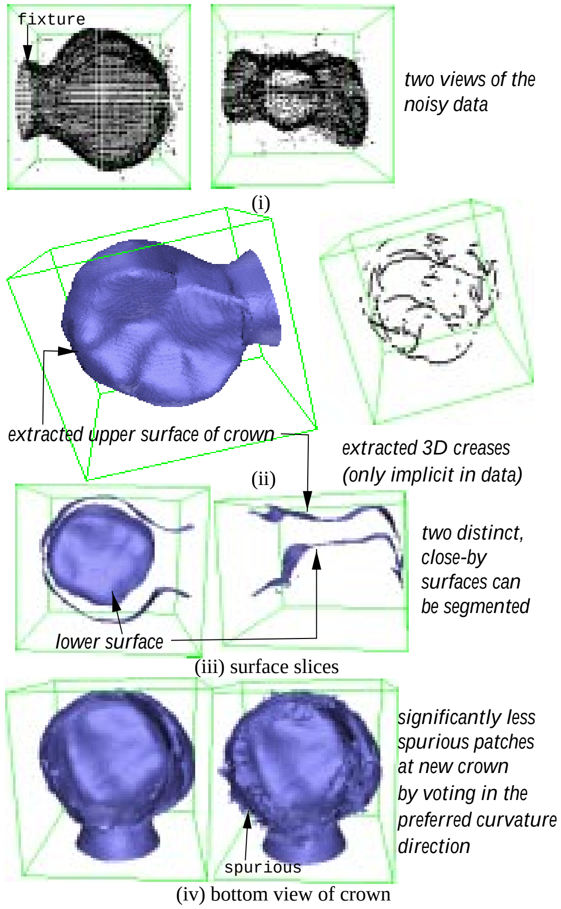

fixture
two views of the noisy data
(i)
extracted upper surface of crown
extracted 3D creases (only implicit in data)
(ii)
two distinct, close-by surfaces can be segmented
lower surface
(iii) surface slices
significantly less spurious patches at new crown by voting in the preferred curvature direction
spurious
(iv) bottom view of crown

Figure 14 *Results on surface and curve inference from noisy* crown *data. (please see movie clip at* http://iris.usc.edu/∼chitang/research.html*)*

(A) CROWN. A set of 9401 points is sampled from a *crown* restoration. Figure 14 shows the noisy input data, the extracted surfaces and 3D crease curves. By using curvature information, we can segment the two distinct but very close-by surfaces, as shown in the two slices. Also, since we generate considerably less irrelevant votes by voting only at one side of the input stick in the preferred curvature direction, we made an improvement over a previous work [12] on the same data. Significantly less spurious surface patches are produced at the "crown": a curve junction (low surface but high curve saliencies) corresponding to surface orientation discontinuity. In [10], a rather elaborate and complex process is described for feature integration that does not take curvature information into consideration, accounting for the use over 60000 points for comparable result.

(B) MOD. A set of 4454 points is sampled from a *mod* restoration, which is to be put into a patient's sliced tooth. In the movie clip, we shows the input data, the extracted surface model, and the inferred creases and other anatomical lines which are only implicit in the data. Due to the space limit, please refer to the movie clip at
http://iris.usc.edu/∼chitang/research.html.

(C) FEMUR. A set of 18224 points is sampled from *femur* (courtesy of INRIA), to which we add 400 outliers. A femur is the proximal bone of the lower limb. We infer the surface description from the noisy data, and label the regions detected as negative Gaussian curvature (saddle) as red, Figure 15.

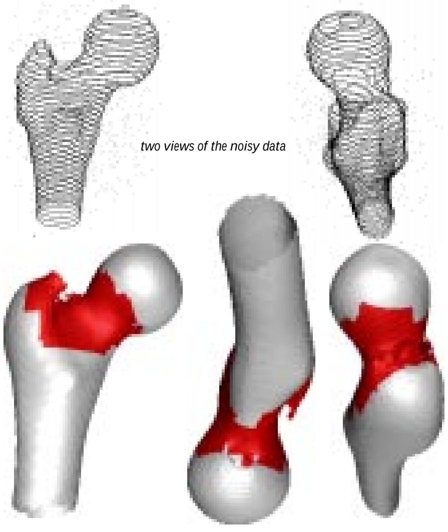

two views of the noisy data
three views of the reconstructed surface

Figure 15 *Results on surface reconstruction for* femur *data. Regions of negative Gaussian curvature are marked in red.*

(D) BUST. A set of 57712 points as obtained from dense stereo, *bust*, a bald human head model. This is a noisy data set, with very few data points on the top of the head (Figure 16). Here, interesting regions of negative Gaussian curvature are labeled, such as the eyes, the area between the nose and cheek, the back of neck, and regions close to both ears. Figure 16 also shows the faithful reconstructed surface.

7 Conclusion and Future Work

We have described an approach that infers sign and direction of principal curvatures *directly* from the input, and uses this information for coherent surface and curve extraction. No partial derivative computation or local surface fitting is performed, which are very unstable in real data because outlier noise is not uncommon. Also, zero curvature is handled uniformly. Curvature information is incorporated into an existing computation framework, and we have shown results on real, noisy and complex data. The future work on this research focuses on more comparative study, a more quantitative error analysis of our method, investigation of the effect of quantization and multi-resolution.

References

[1] E. Angelopoulou and L.B. Wolff, "Sign of Gaussian Curvature From Curve Orientation in Photometric Space," *IEEE Trans. Patt. Analy. Machine Intell.*, vol. 20, no. 11, Oct 1998.

[2] P.J. Besl and R. Jain, "Segmentation Through Variable-order Surface Fitting," in *Computer Vision, Graphics and Image Process.*, vol. 33, pp. 33–80, 1986.

[3] M.P.D. Carmo, "Differential Geometry of Curves and Surfaces," Prentice Hall, 1976.

[4] T.-J. Fan, G. Medioni, and R. Nevatia, "Description of Surfaces From Range Data Using Curvature Properties," in *Proc. Comput. Vision Patt. Recogn.*, pp. 86–91, 1986.

[5] P.J. Flynn and A.K. Jain, "On Reliable Curvature Estimation", in *IEEE Conf. Comput. Vision Patt. Recogn.*, pp. 110–116, 1989.

[6] G. Guy and G. Medioni, "Inference of Surfaces, 3-D Curves and Junctions from Sparse, Noisy 3-D Data," *IEEE Trans. Patt. Analy. Machine Intell.*, vol. 19, no. 11, pp. 1265–1277, 1997.

[7] M.-S. Lee, "Tensor Voting for Salient Feature Inference in Computer Vision", Ph.D. Thesis, University of Southern California, 1998.

[8] P.T. Sander and S.W. Zucker, "Stable Surface Estimation," in *Proc. Intl Conf. Patt. Recogn.*, vol. 1, p. 1165–1167, Oct 1986.

[9] P. Shi, G. Robinson and J. Duncan, "Myocardial Motion and Function Assessment Using 4D Images," in *Proc. IEEE Conf. Vis. Biomedical Comput.*, Rochester, MN, Oct 1994.

[10] C.-K. Tang and G. Medioni, "Inference of Integrated Surface, Curve, and Junction Descriptions from Sparse 3-D Data," in *Proc. IEEE Trans. Patt. Analy. Machine Intell.*, vol. 20, no. 11, pp. 1206–1223, 1998.

[11] C.-K. Tang and G. Medioni, "Extremal Feature Extraction from 3D Vector and Noisy Scalar Fields," in *Proc. IEEE Visualization '98*, pp. 95–102, 1998.

[12] C.-K. Tang, G. Medioni, and F. Duret, "Automatic, Accurate Surface Model Inference for Dental CAD/CAM," in *Proc. Intl Conf. Med. Image Comput. & Comput. Assist. Inter.*, Cambridge, MA, pp. 732–742, 1998.

[13] E. Trucco and R.B. Fisher, "Experiments in Curvature-Based Segmentation of Range Data," in *IEEE Trans. Patt. Analy. Machine Intell.*, vol. 17, no. 2, pp. 177–182, 1995.

[14] B.C. Vemuri, A. Mitiche, and J. Aggarwal, "Curvature-Based Representation of Objects from Range Data," *Image and Vision Computing*, vol. 4, no. 2, pp. 107–114, May 1986.

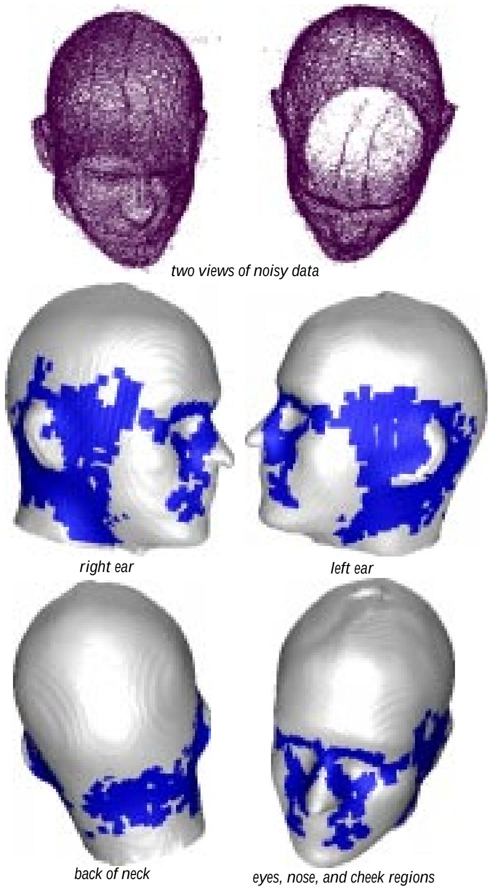


Figure 16 *Results on surface reconstruction for* bust *data. Detected negative Gaussian curvature regions are marked in blue.*