Estimating the Principal Curvatures and the Darboux Frame from Real 3D Range Data

Eyal Hameiri, Ilan Shimshoni

Abstract— **As products of second-order computations, estimations of principal curvatures are highly sensitive to noise. Due to the availability of more accurate 3D range imaging equipment, evaluation of existing algorithms for the extraction of these invariants and other useful features from discrete 3D data, is now relevant. The work presented here, makes some subtle but very important modifications to two such algorithms, originally suggested by Taubin [14] and Chen and Schmitt [2]. The algorithms have been adjusted to deal with real discrete noisy range data. The results of this implementation were evaluated in a series of tests on synthetic and real input yielding reliable estimations. Our conclusion is that with current scanning technology and the algorithms presented here, reliable estimates of the principal curvatures and the Darboux frame can be extracted from real data and used in a variety of more comprehensive tasks.**

I. Introduction and Previous Related work

The importance of principal curvatures as invariant features of free-form rigid surfaces has been widely recognized. In addition, the local coordinate system consisting of the normal to the surface at that point and the two principal directions (the *Darboux frame*) can also be exploited for a vast variety of tasks. The Darboux frame is not invariant to rigid transformations but it can be used to establish a local coordinate frame which is attached to the object at each of its sampled points.

These two types of features are locally associated with the surface and therefore can be used in a local manner. For example, the principal curvatures can be used to find corresponding points between a free-form scene object and a library object. Then, the local Darboux frame on the two objects can be used to register them and test the validity of the matched points. But, the features can also be used in a global manner. For example, in the case of geometric primitives, we have pre-knowledge of the different characteristics of the two features when they are collected from the entire surface of the primitive. Then, by gathering the feature values from all points of a range scene, evidence for the presence or absence of geometric primitives will emerge. Furthermore, when a primitive is identified in the range image, its accurate dimensions and orientation can be recovered from the values of the features. In both types of uses, global as well as local, it is therefore necessary to accurately estimate the principal curvatures and the Darboux frame from range data.

Eyal Hameiri is with the Dept. of Computer Science, The Technion - Israel Inst. of Technology, Haifa, Israel. E-mail: eyl1@cs.technion.ac.il.

Ilan Shimshoni is with the Dept. of Industrial Engineering and Management, The Technion - Israel Inst. of Technology, Haifa, Israel. E-mail: ilans@ie.technion.ac.il.

While these features are continuous properties of a regular surface, range data is discrete in nature. This is where the challenge of how to extract these continuous features from a discrete approximation of a surface, emerges. Principal curvatures are second-order derivatives of a surface which makes them very sensitive to noise. Thus, the main goal of the work presented here is to suggest algorithms for principal curvatures and Darboux frame estimations from real noisy data and to test their validity.

The sensitivity of the features to noise imposed limitations on some early works, restricting their use to only the sign of the mean and Gaussian curvatures for segmentation tasks [1], [7], [13], [15], [16]. In the recognition system presented in [12], only the normal to the surface which requires only a first derivative of the surface (without the principal directions), is used to recognize 3D objects. In recent years, following the availability of more accurate 3D sensing equipment, as well as the use of polyhedral meshes to approximate free-form surfaces, more and more works involve full principal curvature estimation from discrete data in their suggested algorithms [14], [2], [8], [3], [4].

Two main approaches are being taken by all previous works. The more popular one, until recent years, is the analytical approach. In this approach, for each 3D point within the input data, a local patch of surface is fitted to the point and its geometric neighbors. The fitted patch is formulated as an explicit function $f(u, v)$. Then, the first and second partial derivatives of $f(u, v)$ are analytically obtained and used within a general and closed analytic form to compute the principal curvatures and directions or the mean and Gaussian curvatures. One analytic method differs from another by the technique in which the surface patch is being fitted. In [3], the patch is a general bi-quadratic patch. In [8], an analytic approach which uses a cubic b-spline fitting technique presented in [5], is tested. In [1], the patch is also formulated as a bi-quadratic formula but the basis functions consist of a different triplet of discrete orthogonal polynomials and not of the canonical polynomials.

The second approach extracts the features directly (or almost directly) from the discrete data. For example, in [4], a Gaussian curvatures operator was developed based on a known expression for the Gaussian curvature (see [11] for example) as the limit of the ratio of a local surface area and of its corresponding area on the Gauss map. From a similar limit of ratio of areas, an operator for the mean curvatures was developed directly from discrete data after defining special local areas. In [14], the algorithm is based on an integral formula which defines at each sampled point a matrix whose eigenvectors and eigenvalues produce the

principal curvatures and the Darboux frame. The matrix is based on discrete data except for a continuous approximations of normal sections (curves). The algorithm in [2] is based on solving a least-squares problem at each vertex. The least-squares problem is formulated directly from the discrete data except for the approximation of cross sections (curves) as circles. We based our two algorithms on the last two non-analytic methods and therefore our suggested algorithms belong to the non-analytic approach.

Most of the curvature estimation algorithms suggested so far, had presented their results only on none-noisy synthetic data. That is the case in [14] and [2] for example where satisfactory results have been achieved for synthetic objects. Few others, like the work in [8], were also tested on noisy synthetic data and on real data. But as far as real data is concerned , it concluded that curvature estimation should be dealt with caution. In [4], full and accurate estimation of principal curvatures and Darboux frame is presented while the input is a mesh, approximating an artificial spatial surface, and which is arbitrarily triangulated. There is no reference to the algorithm's performance in case of real data or of synthetic noisy data.

Two algorithms, presented in [14], [2], for the estimation of principal curvatures and Darboux frame from range images, have been modified in this work. We will refer to these two algorithm as to the "T algorithm" and "C algorithm" respectively. The T algorithm, which deals correctly with continuous surfaces has been modified to deal with real discrete surfaces (open or close) represented by a cloud of noisy 3D points. In the C algorithm the modification improved the results for both synthetic and real noisy data and also reduced the complexity of the algorithm. We have tested the performance of the modified algorithms on synthetic noisy range data and on real data obtained from a Cyberware 3D scanner. Cases of noiseless synthetic data were also tested, especially for comparative reasons. For each surface point, the features are estimated and a reliability factor is applied, to indicate how accurate the estimations are. Wherever the ground-truth is available, statistics of the errors, resulting from the comparison, were studied.

The paper continues as follows. In Section II we explain terms related to curvatures. In Section III the two original algorithms are presented in detail followed by descriptions of the modifications we have made to them. Section IV is dedicated to our implementation of the suggested algorithms. In Section V, representative examples of tests we have conducted are presented. This section also includes conclusions about the conditions under which the implementation should be used, its limitations and the reliability of its results. We conclude in Section VI.

II. On Curve and Surface Curvatures

Let $C(s)$ denote an arc-length parameterized curve passing through a point P where $C(0) = P$. In this parameterization the derivative, $C'(0)$, produces the tangent to the curve at P, denoted by T. The second degree derivative, $C''(0)$, gives us the curvature vector denoted by $\kappa \cdot n$ where κ is a nonnegative scalar and n denotes the normal to the curve at P. The reader can find more details in [6].

Now, consider S to be a regular surface illustrated in Figure 1, P a point lying on S and $C \subset S$ a regular curve passing through P. Let N and n be the unit length normals at P, of S and C respectively, and T the unit length tangent of C at P. If $\vec{\kappa} = \kappa n$ is the curvature of C at P and ϕ is the angle between N and n, then the quantity $\kappa_P(T) = \kappa \cos\phi$ is defined as the normal curvature of C at P. Note that for any normal section of S, incident on P, the curvature of this normal section at P, equals to the normal curvature at that point.

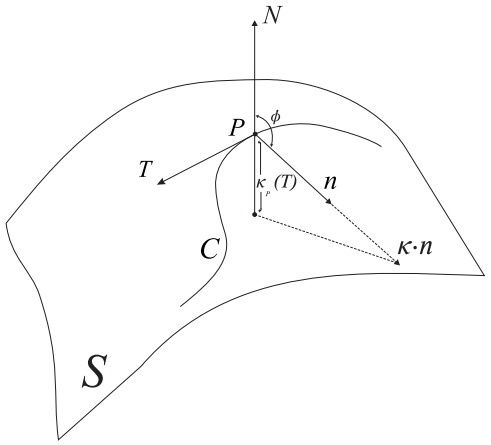


Fig. 1. The local differential properties of a curve and a free-form surface on which it lies.

The *Meusnier theorem* [6] states that all curves in S, intersecting P and sharing at that point the same unit tangent vector, also share the same normal curvature. Therefore, the quantity $\kappa_P(T)$ has been established as the *directional curvature* of S at P along the tangent direction T. Note that the sign of the directional curvature is determined by the orientation of N. In all our discussions from now on, N is pointing out in case of a closed surface or has a nonnegative component in the direction of the viewer in case of an open one.

The maximal and minimal values of $\kappa_P(T)$ along "all" possible tangent directions to S at P, are referred to as the principal curvatures, and their two associated tangents as the principal directions of S at P. If the two principal curvatures share the same value, then the principal directions are undefined. It can be shown [6] that when defined, the principal directions are always orthogonal and thus forming together with N an orthogonal triplet known as the *Darboux frame*. While the principal curvatures are invariant to rigid transformations the Darboux frame is not, but it still can be used to establish a local coordinate frame which is attached to the object regardless of its orientation.

Let κ_P^1 and κ_P^2 denote the two principal curvatures of S at P and T_1, T_2 their two associated unit length principal directions. For every tangent direction T_θ, at P, where θ is the angle between T_θ and T_1 the *Euler formula* [6] states that:

$$\kappa_P(T_\theta) = \kappa_P^1 \cdot \cos^2\theta + \kappa_P^2 \cdot \sin^2\theta \qquad (1)$$

III. The algorithms and their modifications

In this section we will describe each one of the original algorithms in [14], [2] followed by their modifications to deal with real noisy range images. We shall refer to the algorithm in [14] as the "T algorithm" and to the one in [2] as the "C algorithm".

A. The T Algorithm

The T algorithm is based upon the construction of a matrix, expressed as an integral, at each 3D point where the principal curvatures and Darboux frame have to be estimated. As illustrated in Figure 2, let us arbitrarily choose T - a tangent to S at P. For $-\pi \leq \theta \leq \pi$ let T_θ denote the tangent to S at P in the direction which creates an angle θ with T. The matrix M_P is then defined as follows:

$$M_P = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \kappa_P(T_\theta) T_\theta T_\theta^T d\theta \tag{2}$$

The integrand in (2) is a matrix which is the product of a tangent at P and its transpose. Therefore, N, the normal to the surface at P, is an eigenvector of M_P associated with a 0 eigenvalue.

Let T_1 and T_2 denote the principal directions at P. If $\hat{\theta}$ is the angle between T and T_1 then

$$T_\theta = T_1 \cdot \cos(\theta + \hat{\theta}) + T_2 \cdot \sin(\theta + \hat{\theta}) \tag{3}$$

Thus, by using (1) and (3) it can be shown, for any selection of T, that the two principal directions at P are the other two eigenvectors of M_P [9]. Furthermore, if e_1, e_2 denote the eigenvalues of M_P, associated with T_1, T_2 respectively, we can obtain that

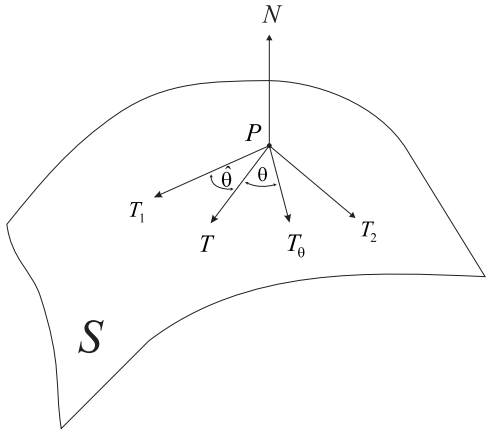


Fig. 2. The T algorithm - tangent directions at P.

$$e_1 = \frac{3}{8}\kappa_P^1 + \frac{1}{8}\kappa_P^2 \qquad e_2 = \frac{1}{8}\kappa_P^1 + \frac{3}{8}\kappa_P^2 \tag{4}$$

where κ_P^1, κ_P^2 are the principal curvature at P. Therefore, in the continuous case, the principal curvatures can be extracted by solving the linear equations (4) where the coefficients are constant for all points on the surface.

In order to estimate the directional curvature, $\kappa_P(T)$, used in (2), let us consider $C(s)$ as the arc-length parameterized normal section, at P, along the direction T. According to the parameterization $C(0) = P$, $C'(0) = T$ and $C''(0) = \kappa_P(T)N$. By expanding $C(s)$ to a Taylor series around $s = 0$ we obtain

$$C(s) = C(0) + C'(0)s + \frac{1}{2}C''(0)s^2 + O(s^3)$$

or

$$C(s) - P = Ts + \frac{1}{2}\kappa_P(T)Ns^2 + O(s^3) \tag{5}$$

By taking the length of the vectors on both sides we get

$$\|C(s) - P\|^2 = s^2 + O(s^4)$$

Dividing equation (5) multiplied by $2N^T$, by the last equation yields

$$\frac{2N^T(C(s) - P)}{\|C(s) - P\|^2} = \kappa_P(T) + O(s)$$

Thus, if a point Q lies on S, very close to P and T denotes the projection of $\overrightarrow{PQ}$ on the tangent plane to S, at P, then $\kappa_P(T)$ can be approximated as

$$\kappa_P(T) \approx \frac{2N^T(Q - P)}{\|Q - P\|^2} \tag{6}$$

B. T algorithm modifications

A surface S is given as a set of nodes V, where each one of them represents a 3D point lying on S. By connecting geometrically neighboring nodes into polygons (usually but not necessarily the polygons are triangles) one can approximate the original continuous surface S as a *polyhedral mesh*.

Let G be the set of surface nodes containing all the geometric neighbors of a node v. At that node, let M_v be an approximation of the continuous case equivalent matrix - M_P. We also will refer by T_i to the tangent to S at v in the direction connecting v to $v_i \in G$ (i.e. T_i is the projection of the vector connecting these two nodes on the tangent plane to S at v). If κ_i is the directional curvature associated with T_i then

$$M_v = \sum_{v_i \in G} \omega_i \kappa_i T_i T_i^T$$

where ω_i is proportional to the inverse of the geometric distance from v to v_i and the sum of all weights, ω_i, equals 1. The use of the weights is not needed in case of a symmetric polyhedral mesh where neighbors of a node are scattered in equal distances, more or less, around it. It is needed in cases of non-symmetrical meshes, especially when G is defined as the "extended neighborhood", a term that will be explained later in this section.

Like in the continuous case, the normal to S at v is an eigenvector of M_v. As for the principal directions at v, let T denote an arbitrary tangent to S at v, T_1 and T_2

the principal directions at v, κ_v^1 and κ_v^2 their associated principal curvatures and let $\hat{\theta}$ denote the angle between T and T_1. If θ_i is the angle between the tangent direction T_i and T then by using (3), (1) and the orthogonality of T_1 and T_2 we obtain that

$$\begin{aligned} M_v T_1 &= \sum_{v_i \in G} \omega_i \kappa_i T_i T_i^T T_1 \\ &= T_1 \sum_{v_i \in G} \omega_i \left[\kappa_v^1 \cos^4(\theta_i+\hat{\theta}) + \kappa_v^2 \sin^2(\theta_i+\hat{\theta}) \cos^2(\theta_i+\hat{\theta}) \right] \\ &+ T_2 \sum_{v_i \in G} \omega_i \left[\kappa_v^1 \sin(\theta_i+\hat{\theta}) \cos^3(\theta_i+\hat{\theta}) + \kappa_v^2 \sin^3(\theta_i+\hat{\theta}) \cos(\theta_i+\hat{\theta}) \right] \end{aligned}$$

The principal direction T_1 is an eigenvector of the matrix M_v only if specific selections, of tangent directions and the weights given to these directions, are made. Practically, since T_1 is unknown at this stage, it is impossible to control these selections. Furthermore, the tangent directions are restricted by the sampling, in case of real range data, or by the triangulation, in case of an artificially triangulated mesh. Real range images are rather dense and mostly acquired in fixed horizontal, vertical or angular sampling intervals. In such cases this obstacle does not emerge, thanks to the homogenous scattering of neighboring nodes around each analyzed node. To deal with arbitrarily triangulated (but still not sparse) meshes, we addressed this discrete case problem by taking homogeneously spread tangent directions, while using not only the nearest neighbors of the analyzed node, but further neighbors as well. In our tests, this procedure has proven to be effective except for extreme cases. This extended neighborhood of the node is also used for indirect pre-smoothing, to overcome noise interference, as will be discussed in Section IV-A. From now on we will refer to nodes contained in the extended neighborhood of v as its neighbors. By the same arguments and in the same conditions T_2 is also an eigenvector of M_v.

Now that we have the principal directions at v, T_1 and T_2, we also have θ_i - the angle between any tangent direction T_i and the principal direction T_1. As in the continuous case we will use (1) and (3) to obtain

$$\begin{aligned} e_1 &= T_1^T M_v T_1 = \sum_{v_i \in G} \omega_i \kappa_i T_1^T T_i T_i^T T_1 \\ &= \sum_{v_i \in G} \omega_i \left[\kappa_v^1 \cos^4 \theta_i + \kappa_v^2 \sin^2 \theta_i \cos^2 \theta_i \right] \\ &= A \cdot \kappa_v^1 + B \cdot \kappa_v^2 \end{aligned} \quad (7)$$

$$\begin{aligned} e_2 &= T_2^T M_v T_2 = \sum_{v_i \in G} \omega_i \kappa_i T_2^T T_i T_i^T T_2 \\ &= \sum_{v_i \in G} \omega_i \left[\kappa_v^1 \sin^2 \theta_i \cos^2 \theta_i + \kappa_v^2 \sin^4 \theta_i \right] \\ &= B \cdot \kappa_v^1 + C \cdot \kappa_v^2 \end{aligned} \quad (8)$$

Where

$$\begin{aligned} A &= \sum_{v_i \in G} \omega_i \cos^4 \theta_i \\ B &= \sum_{v_i \in G} \omega_i \sin^2 \theta_i \cos^2 \theta_i \\ C &= \sum_{v_i \in G} \omega_i \sin^4 \theta_i \end{aligned}$$

In contrast to the continuous case, where at all nodes, the same set of linear equations in κ_v^1, κ_v^2 is needed to be solved, at each node of the discrete case, a different set of equations has to be established by calculating A, B and C. The principal curvatures are extracted by solving (7) and (8) at each node v.

C. The C Algorithm

The algorithm presented by X. Chen and F. Schmitt in [2] is based upon a construction of a least-squares problem at each vertex to be analyzed for its principal curvatures and directions.

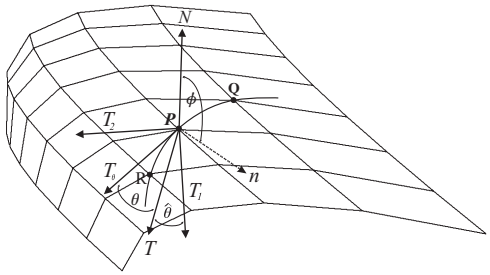


Fig. 3. The C Algorithm illustrated on a polyhedral mesh

Consider Figure 3 where P, Q and R denote nodes on a polyhedral approximation of the surface S. Let T denote an arbitrary tangent to S at P and T_θ denote the tangent which creates an angle θ with T. According to equation (1) (the *Euler formula*) the directional curvature of S at P along the tangent direction T_θ is given by

$$\kappa_P(T_\theta) = \kappa_P^1 \cdot \cos^2(\theta + \hat{\theta}) + \kappa_P^2 \cdot \sin^2(\theta + \hat{\theta}) \quad (9)$$

Using trigonometric identities in 9 we get

$$\kappa_P(T_\theta) = A \cdot \cos^2\theta - 2B \cdot \cos\theta \sin\theta + C \cdot \sin^2\theta \quad (10)$$

Where

$$\begin{aligned} A &= \kappa_P^1 \cos^2\hat{\theta} + \kappa_P^2 \sin^2\hat{\theta} \\ B &= (\kappa_P^1 - \kappa_P^2) \cos\hat{\theta} \sin\hat{\theta} \\ C &= \kappa_P^1 \sin^2\hat{\theta} + \kappa_P^2 \cos^2\hat{\theta} \end{aligned}$$

Thus, by building 3 or more instances of the linear equation (10), the unknowns A, B and C can be estimated using a direct solution or by solving a least-squares problem when more than 3 equations exist. The principal curvatures are given by:

$$\frac{1}{2}\left(A + C \pm \sqrt{(A-C)^2 + 4B^2}\right) \quad (11)$$

The principal directions are obtained by rotating T by the angle $\hat{\theta}$:

$$\hat{\theta} = \frac{1}{2} \arctan\left(\frac{2B}{A-C}\right) \quad (12)$$

In order to build the set of equations, the directional curvatures at P have to be estimated. The authors of [2] suggested to estimate these curvatures by approximating a curve which lies on the surface S while passing through P and two of its neighbors. According to the *Meusnier theorem*, presented in Section II, all curves lying on the surface S and share the same tangent at P also share the same normal curvature at that point which is also the surface's directional curvature at P along this tangent. As shown in Figure 3 the node P and two of its geometric neighbors Q and R are used to find a spatial circle passing through all three of them. κ_P, the curvature of this curve at P, can be easily obtained. If n denotes the curve normal at P, ϕ the angle between n and the estimated surface normal N and T_θ is the tangent to the curve at P then from the *Meusnier theorem* we know that

$$\kappa_P(T_\theta) = \kappa_P \cos\phi \tag{13}$$

Thus, from every different pair of P's neighbors, together with P, an instance of (10) can be generated. Another consideration one should take is how to choose the triplets. In order to minimize noise effects it is better to choose P's neighbors in such a way that the angle ϕ between the curve and surface normals at P will be small as possible. This strategy is implemented by picking pairs of nodes which are as close as possible to lie on opposite geometric sides of P. This implementation tries to guarantee that equation (13) is used within an interval where the cosine function slope is moderate and therefore reduces the effect of possible noisy data on the estimation of the directional curvatures.

D. C algorithm modifications

Several problems and disadvantages emerge from the technique used in [2] to estimate the directional curvatures.

First, the approximation of a curve which lies on the surface and passes through the analyzed node, as a circular curve, is simply not accurate enough, resulting in unreliable directional curvature estimations whenever the normal section passing through the analyzed node differs from a circle. This inaccuracy is even more crucial when the pair of neighbors of the analyzed node are not within its immediate neighborhood. Another problem is regarding the attempt to approximate curves which are close to normal sections by choosing pairs of nodes which lie on opposite sides (geometrically) of the analyzed node. It is very possible that a pair of nodes, which lie on opposite sides of the analyzed point, do not lie on the normal section. Furthermore, since the locations of the nodes are fixed by the sampling device, in many cases, even the best selection of a triplet of nodes, using the strategy presented by the authors of [2], is doomed to generate curve whose normal creates a large angle with the surface normal. Thus, in many cases the attempt to minimize the effects of noise through the cosine function in equation (13) fails. Finally, the need to check every two neighbors of an analyzed node, in order to find the best pairs in the sense of lying on opposite directions of the node, yields for the algorithm in [2], a quadratic time and space complexity.

Our suggestion is to determine the directional curvatures by using the estimation in (6):

$$\kappa_P(T) \approx \frac{2N^T(Q-P)}{\|Q-P\|^2}$$

By this expression we estimate the normal section directly as a general quadratic curve and do not limit it to a circle, yielding better results for all types of objects except for spheres where all cross sections are indeed circles. At the same time we are improving the complexity of the algorithm to a linear time algorithm.

IV. The Implementation

A. Noise Interference

In spite of improvements in 3D sampling equipment, noise in the acquired range image, still has a crucial effect on the estimation of second-order features such as principal curvatures. One way to overcome noise is pre-smoothing of the surface. But, an excessive direct smoothing, might change the local properties of the surface, resulting in distorted values for the estimated curvatures. We therefore apply only minor and very local pre-smoothing. Another, indirect smoothing, is applied by turning not only to the nearest neighbors of a node during the directional curvature estimation stage. This approach has also has a distorting effect but it is very minor for limited extensions of a node surroundings, as was studied in our tests. We define a topological extended neighborhood of a node by two parameters. For each node v (see Figure 4) we define a "ring" of initial radius r and width w, measured in topological neighbors, that surround v. All nodes contained in such an extended neighborhood are used to build M_v in the T algorithm. In the C algorithm, they are used to generate more instances of equation (10). The values of r and w

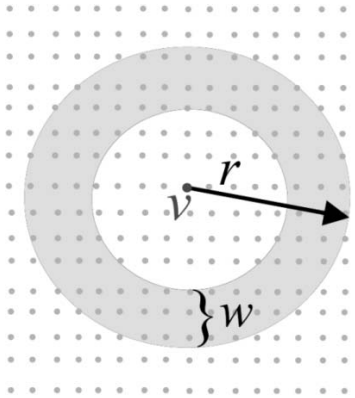


Fig. 4. The extended neighborhood of a node v. Nodes within the gray area are considered as v's neighbors.

($w \leq r$) have to be adjusted for optimal results according to the level of noise.

Let us remember that the extended neighborhood also guarantee the correctness of the T algorithm in the general discrete case. Therefore, regardless of noise in data, the extension is needed at areas where sampling of the surface is inhomogeneous.

B. Surface Normal Estimations

There are several known techniques to estimate the surface normals at discrete surface points. All methods perform some kind of averaging or compromising process between different evaluations of the surface normal at the analyzed node. We have found that for our purposes (real range images), where the data is dense and homogeneously sampled, there isn't much difference between the methods. We preferred to take the normal as the weighted average of all the normals of the planes, which contain the node and two of its adjacent neighbors. We use the angle created at the node, when connecting it with two of its adjacent neighbors as weight.

C. The Reliability Estimator

Since noise in the sampled points is not homogenous, the accuracy of the feature estimates is not homogenous either. Therefore, it will be very helpful for future applications to assign to each analyzed point an estimator to the reliability of its extracted features. We use the estimated principal curvatures and Darboux frame at an analyzed point to approximate the surface locally, and then compare it to the locations of the points in the local neighborhood, captured by the sampling device. Smaller differences between the two, suggest greater reliability of the extracted features.

We wish to evaluate the reliability of the features k_1, k_2, T_1, T_2 and N as were estimated at point P. Let Q be a neighbor of P and T the projection of $\overrightarrow{PQ}$ on the tangent plane to the surface at P. An approximation of the normal section passing through P and in the direction of Q can be obtained from equation (5)

$$C(s) = P + Ts + \frac{1}{2}\kappa_P(T)Ns^2.$$

$\kappa_P(T)$ is computed by using the extracted principal curvature in equation (1) and s is the curve parameter. Now, we examine how close is Q to the approximate normal section. Let $C(s_0)$ denote the closest point on C to Q. To find s_0 we minimize the Euclidean distance between Q and $C(s)$. The square of the error is given by:

$$[Q - C(s)]^2 = \left[Q - P - Ts - \frac{1}{2}\kappa_P(T)Ns^2\right]^2$$

The optimal s is found by equating the derivative of this expression to zero. Using the orthogonality of the Darboux frame we obtain the condition for an extremum in the squared error expression:

$$\frac{1}{2}[\kappa_P(T)]^2 s^3 + [1 - (Q-P)\cdot N\kappa_P(T)]\, s - (Q-P)\cdot T = 0$$

For each real root of the last polynomial we can calculate the distance of $C(s)$ from Q and take the one which minimizes it to be s_0. We repeat this procedure for each Q_i in G - the immediate neighborhood of P. The reliability is proportional to the inverse of the reliability factor as defined in:

$$\sqrt{\frac{1}{|G|}\sum_{Q_i \in G}[Q_i - C_i(s_{0_i})]^2}$$

D. Final implementation

Both algorithms have been implemented and found to produce almost identical performance when synthetic surfaces, with or without noise, were analyzed. The T algorithm yields better results in cases of real noisy range images. The reason is that the C algorithm is based on solving a least-squares problem which makes it quite sensitive to noise. When the noise can not be neutralized effectively enough by pre-smoothing, like in the case of a real noisy data where statistics of the noise are unavailable, it leads to less accurate estimations. On the other hand, the C algorithm can deal with any non-symmetric sampling or with any arbitrary triangulation in case of artificial data.

V. Experimental Results

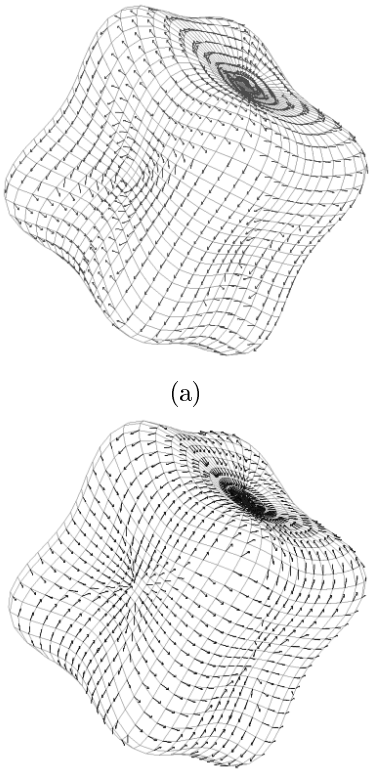
Of all previous related works, only the one in [8] included evaluation of the algorithms performance with real noisy range data. In [8], five different methods to estimate principal curvatures were tested and none of them proved to yield sufficient reliability with real data. Other previous works that we have examined, had only been tested on non-noisy simulated data, probably due to the sensitivity of the features to noise. Our suggested improvements yielded better results, in the general case, with simulated data as well. As for the principal directions, probably because of difficulty in obtaining the ground truth, non of the previous works had presented results in regard to principal directions.

Where it is possible to obtain the ground truth we studied the statistics of the error between estimated values and real values. In other cases we examine the results qualitatively from illustrations of the features, histograms and by locally examining the feature values at key points on the surface. An example for some of these illustrations can be found in Figures 5, 6 and 7. One of our main tools is the *2D* histogram in which the values of the principal curvatures are used as the histogram entries. We term this histogram as the "principal curvatures histogram".

A. Examining the Effects of Sampling Density and Smoothing

Using synthetic data, we studied the effects of the sampling density, the distorting effects of the pre-smoothing of data and of the extended neighborhood analysis in order to establish an optimal platform for real data analysis. At the same time we objectively tested the algorithms on noisy synthetic geometric primitives where the ground truth can be obtained. We added Gaussian noise in all coordinates with s.d. which is 10 to 20 percents of the sampling interval in our real data. We checked the statistics of the error between the estimated values and the values which were extracted analytically. In principal curvatures the units are $1/cm$. In the vectors of the Darboux frame the error is the angle between two vectors and therefore units are radians. μ denotes the mean difference between real and estimated values and σ denotes the standard deviation of this difference.

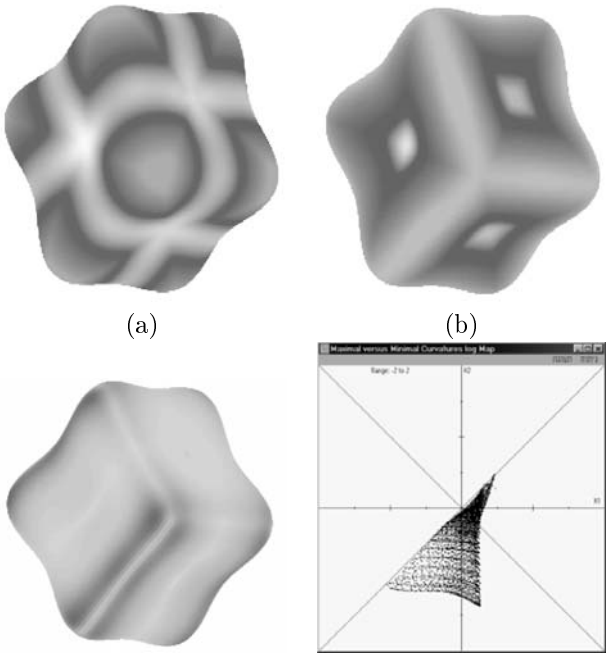
In Figure 8, typical results of a test which examine the



(a)

(b)

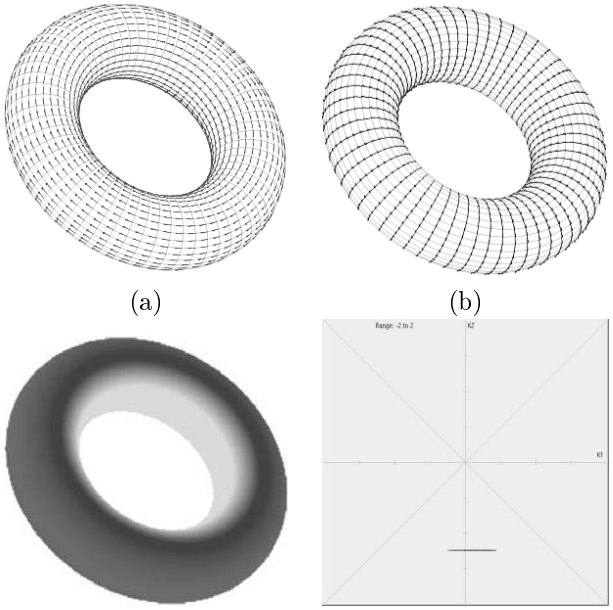
Fig. 5. Examining tools for Darboux frames demonstrated on a deformed cube: (a) Maximal principal directions; (b) Minimal principal directions.



(a) (b)

(c) (d)

Fig. 6. Examining tools demonstrated on a deformed cube: (a) Maximal principal curvatures; (b) Minimal principal curvatures; In (a) and (b) brighter gray indicates greater absolute value; (c) Reliability factor, brighter areas indicate higher reliability; (d) Principal curvatures histogram.

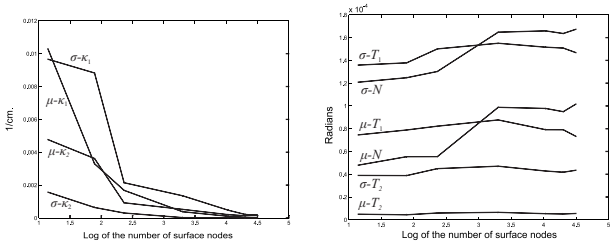


(a) (b)

(c) (d)

Fig. 7. Synthetic torus: (a) Maximal principal directions; (b) Minimal principal directions; (c) Maximal principal curvatures, brighter means greater absolute value; (d) Principal curvatures histogram.

effect of sampling density is presented. We can see that the



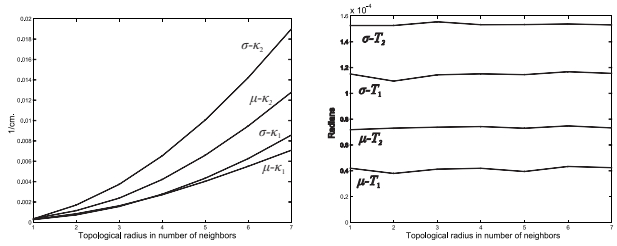
(a) (b)

Fig. 8. Synthetic cone in different densities - mean and s.d. of differences: (a) In principal curvatures; (b) In Darboux frame.

denser the sampling is the smaller the standard deviations are, although both mean and s.d. of both curvatures are stabilizing very rapidly for very low values that represent even sparser sampling rates then those of a typical range image. We conclude, at least for geometric primitives, that the algorithms yield very accurate results.

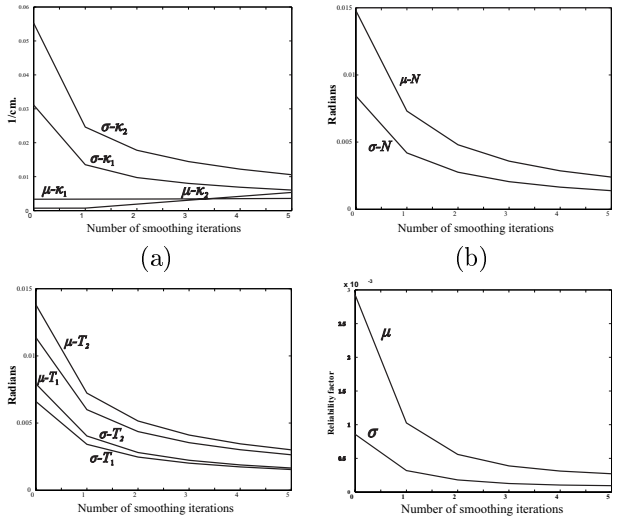
In Figure 9, we present statistics of typical test which examined the distorting effect of the extended neighborhood. It was conducted at a sampling density equal to the density in our real tests which is about 140 to 570 samples per square cm. We can see that errors in the principal directions are low and stable for all the checked neighborhoods. The principal curvatures become quite distorted when we use neighborhoods with a radius of more than 4 to 6 neighbors.

In another series of tests we checked what results are obtained with noisy primitives when combining presmoothing with an extended neighborhood. Typical results are demonstrated on a synthetic noisy torus in Figure 10. By examining the statistics of the errors in all features we



(a) (b)

Fig. 9. Synthetic torus in different neighborhood extensions - mean and S.D. of differences: (a) In principal curvatures; (b) In principal directions.



(a) (b)

(c) (d)

Fig. 10. Synthetic noisy torus, analyzed with a neighborhood radius of 4, in different number of pre-smoothing iterations - mean and s.d. of differences: (a) In principal curvatures; (b) In normals; (c) In principal directions. (d) Reliability factor, μ - mean, σ - s.d.

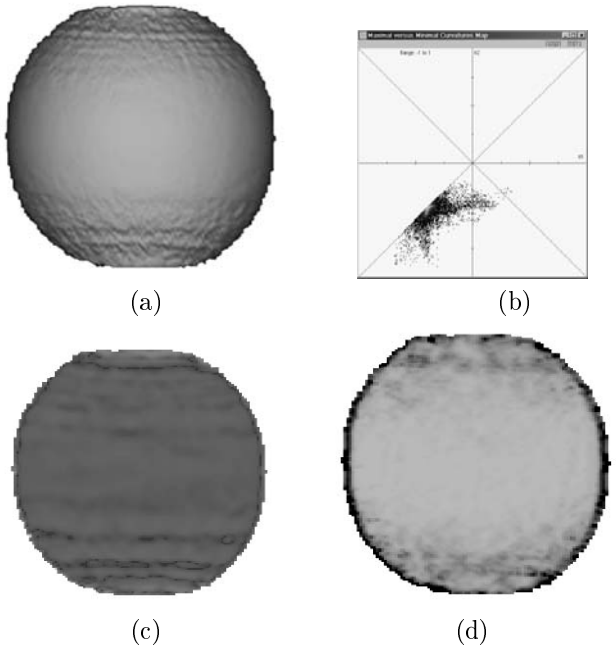
can see the tradeoff between the effect of the reduction of noise and the surface deforming effect caused by smoothing.

From tests such as ones described in this section we conclude that by combining extended neighborhoods with minor pre-smoothing of data, the algorithms extract reliable estimations of the features under typical noise conditions. Another benefit from these tests is that now we can establish an optimal platform to analyze real data whose sampling density and noise levels were simulated in these tests. As a conclusion, in our tests on real data we combine pre-smoothing of no more than 3 iterations of Gaussian filter of size 3×3 and an extended neighborhood of radius that is no more than 4 to 6 neighbors.

B. Real data

Even though, most of the ground truth in real data can not be obtained, in order to have some quantitative estimates of the results, we tested the algorithms on simple geometric primitives (spheres and cylinders). In these primitives, part of the ground truth can be obtained, in a limited level of accuracy, by measuring the physical dimensions of the primitives. For a qualitative examination of the algorithms performance, we also tested them on different free-form surfaces.

The Cyberware Laser Scanner, Model 3030 (1993 model) served us to acquire all the 3D images used to perform the real data tests. All quantitative tests yielded similar results. Here we present results of only two such tests. A sphere with a measured radius of 2.55 cm. was scanned. The surface was analyzed after three iterations of smoothing by a Gaussian filter with window size of 3×3, and with ring parameters of $r = 4$ and $w = 4$. The real value of both curvatures should have been -0.392. With the C algorithm we got a mean of -0.395 and standard deviation of 0.114. With the T algorithm we got a mean of -0.393 and standard deviation of 0.089. We also compared the statistics of the reliability factors to those obtained in synthetic noisy data. These values were a little higher than those of the synthetic noisy data indicating lower reliability but still the values were very low. The range image and some qualitative results are shown in Figure 11.



(a) (b)

(c) (d)

Fig. 11. Real sphere: (a) Range image; (b) Principal curvatures histogram; (c) Principal curvatures, brighter level indicates greater absolute value; (d) Reliability factor, brighter gray indicates better reliability.

A cylinder, slightly tilted in the $X \circ Y$ plane, with a measured radius of 1.39 cm was sampled. The cylinder was also analyzed using the same parameters as the sphere. With the T algorithm we got a mean of 0.005 (should have been 0) and s.d. of 0.074 for the maximal principal curvature. For the minimal one we got mean of -0.719, which is exactly the real value, and s.d. of 0.097. With the C algorithm we obtained the following means and standard deviations: 0.007 and 0.09 for the maximal principal curvature, -0.687 and 0.099 for the minimal one. The results obtained for real cylinders were a little better than those obtained for real spheres. This is due to the fact that the spheres have more boundaries which are almost parallel to the sampling

direction, and therefore more noise in their range images. Illustrations of some qualitative results are shown in Figure 12.

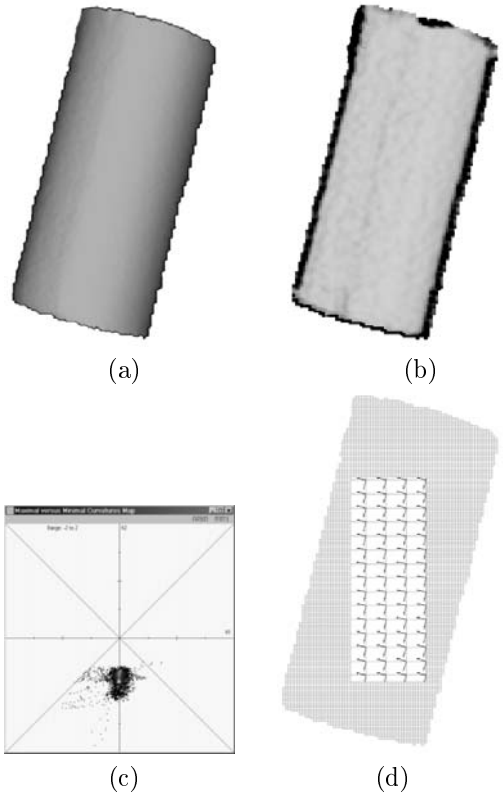


Fig. 12. Real cylinder: (a) Range image; (b) Reliability factor, brighter gray indicates better reliability; (c) Principal curvatures histogram; (d) Principal directions within an enlargement of portion of the real cylinder mesh.

From the quantitative results we can see that the accuracy of the features estimation is not as good as in the synthetic noisy data case but still, the means of the principal curvatures are quite accurate. The problem is the large standard deviations. Similar scattering around means of the principal curvature values were obtained in all similar tests with primitives. The cause is that noise in real data does not have a Gaussian distribution and therefore is not being smoothed effectively enough. In a general free-form object, where values of principal curvatures can not be estimated by their mean value, we can avoid being misled by the less accurate estimations, by using the reliability factor which is assigned to each estimated set of features. If we look at the reliability illustrations we can notice that the problematic areas were identified and located by the reliability factor (the darker areas).

Other series of tests were conducted with general free-form surfaces. Qualitative results of a representative example of such tests are illustrated in Figures 13 and 14.

By examining the illustration and the features quantitative values at key points, we can estimate the results. We found the results to be reliable except at points which lies along boundaries that separate between two scene areas,

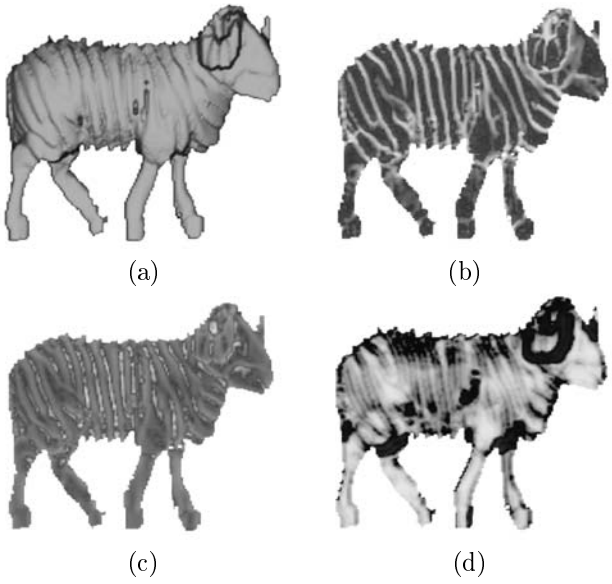


Fig. 13. A toy model of a sheep: (a) Range image; (b) Maximal principal curvatures; (c) Minimal principal curvatures; In b and c brighter gray indicates greater absolute value; (d) Reliability factor, brighter gray indicates better reliability.

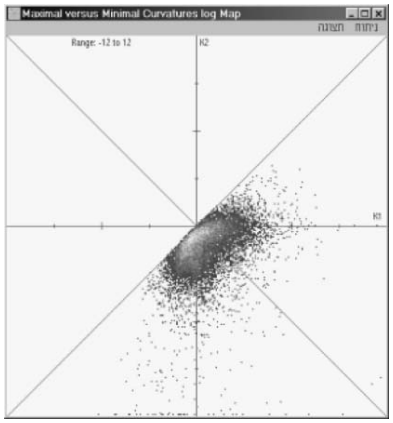


Fig. 14. A toy model of a sheep - Principal curvatures histogram.

of totally different depth. These less accurate areas can be clearly seen in the illustration of the reliability factor. Other problematic areas can be seen wherever a higher level of noise is noticed in the range image illustrations.

The only quantitative estimators that we still have in the case of general free-form surfaces, are the means and standard deviations of the reliability factors. These statistic values were higher than in the case of noisy synthetic primitives but still low enough to indicate good accuracy in the over all estimations of the extracted features.

One additional aspect that emerged here is the difference between the curvatures histograms of two or more, similar in general shape and size, objects. The issue of "almost uniqueness" of a principal curvatures histogram can be exploited in a recognition process of general free-form objects which is based on global analysis of the features. This is due to the fact that as long as the same areas of an objects are scanned, its curvatures histogram remains the same, regardless of its orientation or position. Full

or patrial comparison between a curvatures histogram of a library object and the histogram of a sampled scene or object can be the basis of such a recognition process. We did not however explore this aspect in this work.

VI. Discussion

In this work we have presented two, previously published algorithms, aimed to extract the Darboux frame and principal curvatures, at each point on a free-form surface. We have suggested subtle modifications for these methods, improving their results considerably when dealing with real range images and making them more generally applicable for practical use.

Different aspects of the modified algorithms, their limitations and ways to overcome noise in the data, were presented and discussed. An estimator for a local reliability of the extracted local features was developed.

We have tested the suggested algorithms on synthetic noiseless and noisy data to estimate its correctness and accuracy and to establish an optimal analyzing platform for the analysis of real noisy data. We then made a series of tests to estimate the accuracy of the estimations with real range images. Our tests show that both algorithms produce very accurate estimations when synthetic noiseless or noisy range images were analyzed. The results with real data were less accurate when compared to the synthetic noisy cases but they still were very reliable and accurate enough for practical usage. This was demonstrated in another work of ours [10] in which we used the estimated features to recover 3D geometric primitives from real range images.

We conclude that it is now possible to use the principal curvatures and the Darboux frame as a stage within a vast variety of more complex applications. But, when using these estimated values, it is extremely important to take into account that the accuracy of the results may vary considerably. Therefore, they should be incorporated with some local estimation for their reliability, as was done with our reliability factor.

References

[1] P. J. Besl and R. C. Jain. Invariant surface characteristics for 3D objects recognition in range images. *Comp. Vis. Graph. Im. Proc.*, 33(1):33–80, January 1986.

[2] X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In *Proc. European Conf. Comp. Vision*, pages 739–743, 1992.

[3] C. S. Chua and R. Jarvis. 3-D free form surface registration and object recognition. *Int. J. of Comp. Vision*, 17(1):77–99, January 1996.

[4] M. Desbrun, M. Meyer, P. Schroder, and A.H. Barr. Discrete differential-geometry operators in nD. July 2000.

[5] P. Dierckx. An Algorithm for least-squares fitting of cubic spline surfaces to functions on a rectilinear mesh over a rectangle. *J. Comput. Appl. Math.*, 3(2):113–129, 1977.

[6] M.P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, NJ, 1976.

[7] T.J. Fan, G. Médioni, and R. Nevatia. Description of surfaces from range data using curvature properties. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 88–91, Miami Beach, Fla., June 1986.

[8] P. J. Flynn and A. K. Jain. On reliable curvatures estimation. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 110–116, 1989.

[9] E. Hameiri. Estimating the principal curvatures and the darboux frame from real 3D range data and its application to the recovery of primitives. Master's thesis, The Technion, 2001.

[10] E. Hameiri and I. Shimshoni. Using principal curvatures and darboux frame to recover 3D geometric primitives from range images. In *Symp. on 3D Data Processing Visualization Transmission*, 2002.

[11] B.K.P. Horn. *Computer Vision.* MIT Press, Cambridge, Mass., 1986.

[12] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 21(5):433–449, May 1999.

[13] P. T. Sander and S. W. Zucker. Stable surface estimation. In *Proceedings, Eighth International Conference on Pattern Recognition (Paris, France, October 27–31, 1986)*, IEEE Publ. 86CH2342-4, pages 1165–1167, 1986.

[14] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. Int. Conf. Comp. Vision*, pages 902–907, 1995.

[15] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal. Curvature-Based Representation of Objects from Range Data. *Image and Vision Computing*, 4(2):107–114, May 1986.

[16] N. Yokoya and M. D. Levine. A hybrid approach to range image segmentation. In *Ninth International Conference on Pattern Recognition (Rome, Italy, November 14–17, 1988)*, pages 1–5, Washington, DC, 1988. Computer Society Press.