ELSEVIER

A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data

Evgeni Magid, Octavian Soldea*, Ehud Rivlin

Computer Science Faculty, The Technion, Israel Institute of Technology, Haifa 32000, Israel

Abstract

Estimating intrinsic geometric properties of a surface from a polygonal mesh obtained from range data is an important stage of numerous algorithms in computer and robot vision, computer graphics, geometric modeling, and industrial and biomedical engineering. This work considers different computational schemes for local estimation of intrinsic curvature geometric properties. Four different algorithms and their modifications were tested on triangular meshes that represent tessellations of synthetic geometric models. The results were compared with the analytically computed values of the Gaussian and mean curvatures of the non-uniform rational B-spline (NURBS) surfaces from which these meshes originated. The algorithms were also tested on range images of geometric objects. The results were compared with the analytic values of the Gaussian and mean curvatures of the scanned geometric objects. This work manifests the best algorithms suited for Gaussian and mean curvature estimation, and shows that different algorithms should be employed to compute the Gaussian and mean curvatures.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Geometric modeling; Principal curvatures; Gaussian curvature; Mean curvature; Polygonal mesh; Triangular mesh; Range data

1. Introduction

A number of approaches have been proposed to represent a 3D object for the purposes of reconstruction, recognition, and identification. The approaches are generally classified into two groups: volumetric- and boundary-based methods. A volumetric description utilizes global characteristics of a 3D object: principal axes, inertia matrix [25], tensor-based moment functions [13], differential characteristics of iso-level surfaces [35,50], etc. Boundary-based methods describe an object based on distinct local properties of its boundary and their relationships. This method is well suited for recognition purposes because local properties are still available when only a partial view of an object is acquired.

Differential invariant properties such as Gaussian and mean curvatures are one of the most essential features in boundary-based methods, extensively used for segmentation, recognition, and registration algorithms [2,57]. Unfortunately, these significant geometric quantities are defined only for twice differentiable (C^2) surfaces. In contrast, geometric data sets are frequently available as polygonal, piecewise linear approximations, typically as triangular meshes. Such data sets are common output of, for examples, 3D scanners.

A plethora of work [1,5,11,20,24,28,30,31,33,34,42, 44,49,54] describing algorithms for curvature estimation from polygonal surfaces exists. Great effort has been invested in designing methods of computing curvature that target real range image data [3,52]. The existence of such a large number of curvature computation methods made it necessary to find some way of comparing them [15,22,27,44,46,51]. Unfortunately, although error analysis methods for several different algorithms exist [33,34], the known comparison results are insufficient and provide only

---

* Corresponding author.
  E-mail address: octavian@cs.technion.ac.il (O. Soldea).

a partial image. The reason for the absence of such work may lie in the large amount of factors that has to be analyzed when comparing curvature. A few examples of such factors are finding interesting, representative, and fair surfaces for computing curvatures, developing a reliable way of evaluating the accuracy of computations, and, nevertheless, the time and memory requirements of different methods.

Starting in the 1980s, the curvature computation field made great strides. Probably the first comparison work was published in 1989, [15]. The authors' conclusion was that the algorithms existing at that time were appropriate for computing the signs of the curvatures, while the values of the curvatures were extremely sensitive to quantization noise. In this context, the authors in [52] reached the same conclusion six years later, in 1995.

Recently, methods providing the ability to compute curvature values are reported in literature. They can be grouped into several main approaches. A good and updated characterization of the existing literature can be found in [51]. Following [51], we describe the most common curvature computation methods.

1.1. Methods employing local analytic surface approximations

Curvature computation essentially means evaluations of second order derivatives [7]. This process is known to be sensitive to quantization noise. Probably the most popular approach for trying to cope with this phenomenon computes an approximation of the vicinity of a node with an analytic surface. In general, quadratic or cubic surfaces are involved. To the best of our knowledge, the most popular approach is paraboloid fitting together with its variants [20,27,28,42,44].

A lot of effort has been invested in finding good local approximation surfaces. A comparison of local surface geometry estimation methods in terms of accuracy computation of curvatures can be found in [32].

Several extensions of the paraboloid fitting methods were also proposed in [14,17,29,40]. We mention that there are methods that employ local analytic surface approximations via computing nearly isometric parameterizations for each vertex [41] and via a spline surface fitting stage [36].

1.2. Methods employing discrete approximation formulas

Discrete approximation formulas employ the 3D information existent in a node and its neighbors in a direct formula. In general, the formulas used are relatively short, a fact that provides some gain in computation time at the cost of the attainable accuracy. The authors in [24,27,33] considered such methods based on the Gauss–Bonnet [9,45] theorem. We describe such a method, that is based on the Gauss–Bonnet theorem, in detail in Section 3.2.

In this context, the Gauss–Bonnet theorem can be used on simplex meshes [8], which are representations considered to be topologically dual to triangulations. Moreover, applications of angle excess in the context of minimal surfaces and straight geodesics can be found in [38,39].

1.3. Methods employing Euler and Meusnier theorems

In numerous works, curvature computation is based on evaluating the curvature of curves that cross through a vertex and its neighbors. Each neighbor provides a directional curvature value. These values are further merged employing Euler and Meusnier theorems (see [9,45]). In many other works, curvature computation is based on approximating tangent circles to surfaces. In this context, in [5,31], circular cross sections, near the examined vertex, are fitted to the surface. Then, the principal curvatures are computed using Meusnier and Euler theorems (see [9,45]). An interesting application of the Euler theorem toward curvature computation can be found in [54]. We describe this method in detail in Section 3.3.

1.4. Methods employing tensor evaluations

The tensor curvature is a map that associates to each surface tangent direction (at a surface point) the corresponding directional curvature (see [49] for details). Probably the most popular method for curvature computations based on tensor curvature evaluation is [49]. Note that an extension of the Taubin scheme [49] appears in [19]. Gopi et al. [19] employs extended regions on neighbors and uses a different weighting scheme.

1.5. Methods employing voting mechanisms

Besides extended regions of neighbors, and closely related to, voting mechanisms are often employed towards algorithms accuracy improvement. The authors feel that most of the voting mechanisms existent in the literature, targeted the tensor computation accuracy improvement. In this context, Taubin approach [49] is at center stage and several very interesting improvements were reported as follows.

A method considered to be robust to noise when evaluating the signs of curvatures [51] can be found in [47]. Following [47] and [48], the authors in [37] proposed an improvement over Taubin's method [49] that, in addition, detects crease values over large meshes. Another improvement in the quality of curvature computation was recently reported in [51].

1.6. Methods employing multi-scaling and total and global techniques

Relative recent works investigated the developing of local surface descriptors that employ smoothing and filtering at different scales toward reliable curvature computation. Probably the most representative method in the multi-scaling methods category is [34]. The results achieved in [34] are dependent on a 2D Gaussian filtering iteratively

applied on the input free-form. An estimation of the error is also provided [34]. Note that the advantage of multi-scale techniques is that they provide solutions for multi-target problems, such as interpolation, smoothing, and segmentation simultaneously [10].

The authors in [37] identified a category of curvature computation methods that provides the values of curvatures at each point at the end of the computations. In this approach, intermediate computation stages have to be finished over the entire input in order to be able to evaluate the curvature values at any point of interest. This approach is used in [56], where the authors proposed a method for computing the electrical charge distributions of a field targeting segmentation. The electrical charge distribution is, in fact, a differential characteristic that is proven to have similar properties to the curvature, especially for segmentation tasks.

1.7. Choosing methods for comparison

Following [46], we attempt to quantitatively compare four methods for estimating the Gaussian and mean curvatures of triangular meshes. We tested these methods on triangular meshes that represent tessellations of four synthetically generated objects: a cylinder, a cone, a sphere, and a plane, as well as on non-uniform rational B-spline (NURBS) surfaces. The results are compared to the analytic evaluation of these curvature properties on the surfaces. Moreover, we present the results of tests of the performance of the different methods on range images of geometric objects scanned using a 3D Cyberware scanner [23]. The results are compared to the analytic values of the Gaussian and mean curvatures of the scanned geometric objects. The analytic values of the Gaussian and mean curvatures were inferred from the dimensions of the real objects. The dimensions of the objects were measured using a micrometer.

In our experiments, we employed four synthetically generated objects and their real objects counterparts, which have similar dimensions and tessellations. This fact represents a key factor in our comparisons.

Our main goal is to obtain a high level of understanding of the main current approaches. In order to obtain such insight, we selected four specific methods, each one representative of a different approach. The chosen methods are: paraboloid fitting (Section 3.1), Gauss–Bonnet (Section 3.2), Watanabe and Belayev (Section 3.3), and Taubin (Section 3.4).

1.7.1. Methods employing local analytic surface approximations: paraboloid fitting

We feel that paraboloid fitting is an appropriate representative method, it being by far the most popular method of its class. In [27] and [44] the paraboloid fitting method is compared to other ones. The authors in [27] provide us a general overview of three algorithms on four types of primitive surfaces: spheres, planes, cylinders, and trigonometric surfaces. No general surfaces are considered, however. In [44], five methods are compared on a cube, a sphere, a noisy sphere, and an approximated image of a ventricle.

1.7.2. Methods employing discrete approximations formulas: the Gauss–Bonnet scheme

In [24,27,33], an algorithm based on the Gauss–Bonnet theorem [9,45] is described. We refer to it as the Gauss–Bonnet scheme. The Gauss–Bonnet scheme is also known as the *angle deficit* method [33]. The authors in [33] provide an error analysis for this method. The reasons for choosing the Gauss–Bonnet scheme is that it is extremely popular for computing the Gaussian curvature and both the Gaussian and mean curvatures involve a convenient, elegant, and fast evaluation of a formula based on turning the angles of its neighbors.

1.7.3. Methods employing Euler and Meusnier theorems: the Watanabe and Belyaev approach

Numerous methods of curvature computation employ Euler and Meusnier theorems [9,45]. We have chosen the Watanabe and Belyaev scheme [54] as representative of this group. The authors in [54] focused on mesh decimation, and provide a comparison of mesh decimation methods. Unfortunately, there is no mention of the accuracy of the curvature computation the authors proposed. The reason for choosing the Watanabe and Belyaev scheme is that, in our opinion, its performance primarily as regards curvature computation accuracy is unknown. Specifically, [31] concluded that the paraboloid fitting provides better results than a method that involves Euler and Meusnier theorems, called the circular cross sections method, on noisy data. We mention that the interest in analyzing the Watanabe and Belyaev scheme comes also from its behavior on noisy data captured in range images of real 3D objects as we show in Section 4.2.

1.7.4. Methods employing tensor evaluations and voting mechanisms: the Taubin approach

Our belief is that the method that best fits the tensor evaluating methodology is Taubin's method. Moreover, the vast majority of algorithms that attempt improvements of curvature computation accuracy by voting target tensor evaluation and, therefore, the Taubin method. Another benefit of Taubin's method is that it provides very good results and was tested on real objects; see [21].

The paper is organized as follows: a short review of differential geometry of surfaces is provided in Section 2. In Section 3, a concise description of the four algorithms we considered is given. Then, results of our comparison are presented in Section 4. Finally, we conclude in Section 5.

2. Differential geometry of surfaces

Let $\vec{S}(x, y)$ be a regular C^2 continuous freeform parametric surface in $\mathbb{R}^3$. The unit normal vector field of $\vec{S}(x, y)$ is defined by

$$\vec{N}(x, y) = \frac{\frac{\partial}{\partial x}\vec{S}(x, y) \times \frac{\partial}{\partial y}\vec{S}(x, y)}{\left\|\frac{\partial}{\partial x}\vec{S}(x, y) \times \frac{\partial}{\partial y}\vec{S}(x, y)\right\|}.$$

The curvature κ of curve $\vec{C}(t) : [a, b] \rightarrow S$ passing through the point $\vec{S}(x_0, y_0) = \vec{C}(t_0)$ is defined by

$$\kappa = \frac{|\vec{C}'(t_0) \times \vec{C}''(t_0)|}{\langle \vec{C}'(t_0), \vec{C}'(t_0) \rangle^{3/2}},$$

where $a, b \in R$ and $\langle \cdot, \cdot \rangle$ is the scalar product of vectors.

The *normal curvature* κ_n of curve $\vec{C} \subset \vec{S}$ passing through the point $\vec{S}(r_0, t_0)$ is defined by the following relation, known as

Theorem 2.1 (Meusnier's theorem).

$$\kappa_n = \kappa \cos \varphi, \tag{2.1}$$

where κ is the curvature of $\vec{C}$ at $\vec{S}(r_0, t_0)$ and φ is the angle between the curve's normal n and the normal $\vec{N}(r_0, t_0)$ of $\vec{S}$.

A graphical representation of Meusnier theorem can be seen in Fig. 1.

The principal curvatures, $\kappa_1(r_0, t_0)$ and $\kappa_2(r_0, t_0)$, of $\vec{S}$ at $\vec{S}(r_0, t_0)$ are defined as the maximum and minimum normal curvatures at $\vec{S}(r_0, t_0)$, respectively. The directions for which these values are attained are called the *principal directions* (see [9,45]). We denote the principal directions with P_1 and P_2, respectively. Bearing in mind that Meusnier's theorem associates to each direction a normal curvature at each surface point $\vec{S}(r_0, t_0)$, the definitions of principal curvatures and directions are consistent. The normal curvature κ_n of surface $\vec{S}(r, t)$ in tangent direction $\vec{T}$ is equal to:

Theorem 2.2 (Euler's theorem).

$$\kappa_n = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta, \tag{2.2}$$

where θ is the angle between the first principal direction and $\vec{T}$.
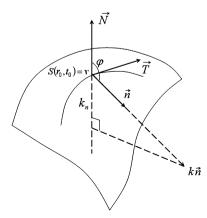

Fig. 1. Curvature and normal curvature at a point on a curve on a surface. k is the curvature of C at v, $\vec{n}$ is the curvature of C at v, $\vec{N}$ is the normal to the surface S at v, k_n is the normal curvature at v, and $k\vec{n}$ is a vector oriented in the direction of $\vec{n}$ with length $\frac{1}{k}$.

The Gaussian and mean curvatures, $K(r,t)$ and $H(r,t)$, are uniquely defined by the principal curvatures of the surface:

$$K(r, t) = \kappa_1(r, t) \cdot \kappa_2(r, t), \tag{2.3}$$

$$H(r, t) = \frac{\kappa_1(r, t) + \kappa_2(r, t)}{2}. \tag{2.4}$$

3. Algorithms for curvature estimation

In this work, we consider four methods for the estimation of the Gaussian and mean curvatures, for triangular meshes. We assume that each given triangular mesh approximates a smooth, at least twice differentiable, surface.

3.1. Paraboloid fitting

The paraboloid fitting method, as well as several variants of it, were described in [20,27,28,42,44]. In [20,27,28,42,44], the principal curvatures and principal directions of a triangulated surface are estimated at each vertex by a least squares fitting of an osculating paraboloid to the vertex and its neighbors. These references use linear approximation methods where the approximated surface is obtained by solving an over-determined system of linear equations. More recently, in [33], the authors provided an asymptotic analysis of the paraboloid fitting scheme adapted to an interpolation case.

Interestingly enough, the paraboloid fitting method was considered a good estimator for differential parameter estimation in iterative processes. For example, in [42], the authors presented a nonlinear functional minimization algorithm that is implemented as an iterative constraint satisfaction procedure based on local surface smoothness properties.

The paraboloid fitting algorithm approximates a small neighborhood of the mesh around a vertex v by an osculating paraboloid. The principal curvatures of the surface are considered to be identical to the principal curvatures of the paraboloid (see [20,27,28,42,44]).

Vertex v_i is considered an immediate neighbor of vertex v if edge $e_i = \overline{vv_i}$ belongs to the mesh. Denote the set of immediate neighboring vertices of v by $\{v_i\}_{i=0}^{n-1}$ and the set of the triangles containing the vertex v by $\{\Delta_i^v\}_{i=0}^{n-1}$,

$$\Delta_i^v = \triangle(v_i \, v \, v_{(i+1) \bmod n}), \quad 0 \leqslant i \leqslant n - 1. \tag{3.1}$$

Fig. 2 shows the notations used for vertices, edges, and triangles.

Let $\vec{N}_v$ be the normal of surface $\vec{S}$ at vertex v. Normals are, in many cases, provided with the mesh in order to enable Gouraud and/or Phong shading [16]. Otherwise, let

$$\vec{N}_i^v = \frac{(v_i - v) \times (v_{(i+1) \bmod n} - v)}{\|(v_i - v) \times (v_{(i+1) \bmod n} - v)\|}, \tag{3.2}$$

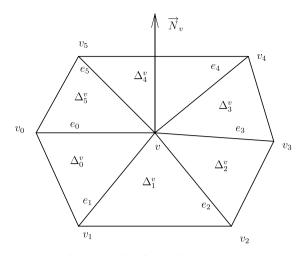be the unit normal of triangle Δ_i^v. Then, $\vec{N}_v$ could be estimated as an average of normals $\vec{N}_i^v$:

Fig. 2. Notation for vertices and edges.

Fig. 4. Internal and external angles in the Gauss–Bonnet method, when the analyzed vertex and its neighbors are planar.
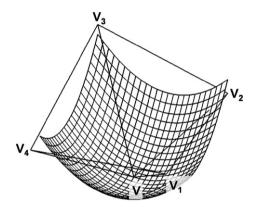

Fig. 3. A visual representation of the paraboloid fitting method.

$$\overline{N}_v = \frac{1}{n} \sum_{i=0}^{n-1} \vec{N}_i^v; \quad \vec{N}_v = \frac{\overline{N}_v}{\|\overline{N}_v\|}, \tag{3.3}$$

where v is now transformed along with its immediate neighboring vertices, $\{v_i\}_{i=0}^{n-1}$, to the origin such that $\vec{N}_v$ coalesces with the z axis. Assume an arbitrary direction x (and $y = z \times x$). Then, the osculating paraboloid of this canonical form equals,

$$z = ax^2 + bxy + cy^2. \tag{3.4}$$

The coefficients a, b, and c are found by solving a least squares fit to v and the neighboring vertices $\{v_i\}_{i=0}^{n-1}$. Then, the Gaussian and mean curvatures are computed as

$$K = 4ac - b^2, \quad H = a + c. \tag{3.5}$$

Fig. 3 shows a visual representation of the paraboloid fitting method.

3.2. The Gauss–Bonnet scheme

Consider again vertex v and its immediate neighborhood $\{v_i\}_{i=0}^{n-1}$. Then, for $i = 0, \ldots, n - 1$, let $\alpha_i = \angle(v_i, v, v_{(i+1) \mod n})$ be the angle at v between two successive edges $e_i = \overline{vv_i}$. Further, let $\gamma_{i+1} = \angle(v_i, v_{(i+1) \mod n}, v_{(i+2) \mod n})$ be the outer
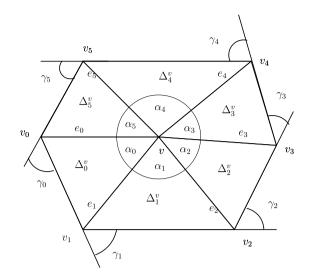
angle between two successive edges of neighboring vertices of v. Then, simple trigonometry can show that

$$\sum_{i=0}^{n-1} \alpha_i = \sum_{i=0}^{n-1} \gamma_i. \tag{3.6}$$

The vertices and the edges are indexed as shown in Fig. 2. Fig. 4 shows specific notations used in the Gauss–Bonnet method.

The Gauss–Bonnet [9,45] theorem reduces, in the polygonal case, to

$$\int\int_A K \, dA = 2\pi - \sum_{i=0}^{n-1} \gamma_i, \tag{3.7}$$

which, according to Eq. (3.6) equals,

$$\int\int_A K \, dA = 2\pi - \sum_{i=0}^{n-1} \alpha_i, \tag{3.8}$$

where A is the accumulated areas of triangles Δ_i^v (Eq. (3.1)) around v.

Assuming K is constant in the local neighborhood, Eq. (3.8) can be rewritten as [24]

$$K = \frac{2\pi - \sum_{i=0}^{n-1} \alpha_i}{\frac{1}{3}A}. \tag{3.9}$$

This approach for estimating K is used, for example, by [1,11,24,33,44]. In [11,24] a similar integral approach to the computation of the mean curvature is proposed as

$$H = \frac{\frac{1}{4} \sum_{i=0}^{n-1} \|e_i\| \beta_i}{\frac{1}{3}A}, \tag{3.10}$$

where $\|e_i\|$ denotes the magnitude of e_i and β_i measures normal deviations $\beta_i = \angle(N_i^v, N_{(i+1) \mod n}^v)$ (see Eq. (3.2)).

3.3. The Watanabe and Belyaev approach

A simple method for estimating the principal curvatures of a surface that is approximated by a triangular mesh was proposed in [54].

Consider an oriented surface $\vec{S}$. Let $\vec{T}$ be a tangent vector and $\vec{N}$ be the unit normal at a surface point v. A normal section curve $\vec{r}(s)$ associated with $\vec{T}$ at v is defined as the intersection between the surface and the plane through v that is spanned by $\vec{T}$ and $\vec{N}$ (see Fig. 5). Let $\vec{P}_1$ and $\vec{P}_2$ be the principal directions at v associated with the principal curvatures κ_1 and κ_2, respectively. $\kappa_n(\varphi)$ denotes the normal curvature of the normal section curve, where φ is the angle between $\vec{T}$ and $\vec{P}_1$. Using Euler's theorem (see Theorem 2.2), integral formulas of $\kappa_n(\varphi)$ and its square are derived [54]:

$$\frac{1}{2\pi}\int_0^{2\pi} k_n(\varphi)\,\mathrm{d}\varphi = H; \qquad \frac{1}{2\pi}\int_0^{2\pi} k_n(\varphi)^2\,\mathrm{d}\varphi = \frac{3}{2}H^2 - \frac{1}{2}K. \tag{3.11}$$

In order to estimate the integrals of Eq. (3.11), one needs to estimate the normal curvature around v, in all possible tangent directions.

Consider v being a mesh vertex and recall its normal $\vec{N}_v$ (Eq. (3.3)). Here, the average of the normals of the faces adjacent to vertex v takes into account the relative areas of the different faces.

v is now transformed along with its immediate neighboring vertices, $\{v_i\}_{i=0}^{n-1}$, to the origin such that $\vec{N}_v$ coalesces with the z axes. Consider the intersection curve $\vec{r} = \vec{r}(s)$ of the surface by a plane through v that is spanned by N_v (the z axis in our canonical form) and edge $e_i = \overline{vv_i}$. A Taylor series expansion of $\vec{r}(s)$ gives

$$\vec{r}(s) = \vec{r}(0) + s\vec{r}'(0) + \frac{s^2}{2}\vec{r}''(0) + \cdots$$

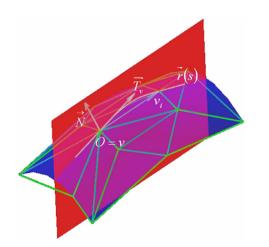$$= \vec{r}(0) + s\vec{T}_r + \frac{s^2}{2}\kappa_n\vec{N}_r + \cdots, \tag{3.12}$$

Fig. 5. Specific notations in the Watanabe and Belyaev method.

where T_r and N_r are the unit tangent and normal of $r(s)$. Recall that $v = r(0)$ and that $v_i = r(s)$. The arclength s could be approximated by the length of edge $e_i = \overline{vv_i}$, or $s \approx \|\overline{vv_i}\|$. Multiplying Eq. (3.12) by $\vec{N}_v = \vec{N}_r$ yields,

$$\vec{N}_v \cdot \overline{vv_i} \approx \kappa_n \frac{\|\overline{vv_i}\|^2}{2}, \qquad \kappa_n \approx \frac{2\vec{N}_v \cdot \overline{vv_i}}{\|\overline{vv_i}\|^2}. \tag{3.13}$$

The vertices and the edges are indexed as shown in Fig. 2. Fig. 5 shows specific notations used in the Watanabe and Belyaev method.

The trapezoid approximation of Eq. (3.11) leads to

$$2\pi H \approx \sum_{i=0}^{n-1} \kappa_n^i \left(\frac{\alpha_{(i-1)\bmod n} + \alpha_i}{2} \right) \tag{3.14}$$

and

$$2\pi\left(\frac{3}{2}H^2 - \frac{1}{2}K\right) \approx \sum_{i=0}^{n-1} \kappa_n^{i\,2}\left(\frac{\alpha_{(i-1)\bmod n} + \alpha_i}{2}\right). \tag{3.15}$$

3.4. The Taubin approach

Let $\vec{P}_1$ and $\vec{P}_2$ be the two principal directions at point v of surface $\vec{S}$ and let $\vec{T}_\theta = \vec{P}_1\cos(\theta) + \vec{P}_2\sin(\theta)$ be some unit length tangent vector at v. Taubin, in [49], defines the symmetric matrix $\mathbf{M}_v$ by the integral formula of

$$\mathbf{M}_v = \frac{1}{2\pi}\int_{-\pi}^{+\pi} \kappa_n^v(\vec{T}_\theta)\vec{T}_\theta\vec{T}_\theta^{\mathrm{T}}\,\mathrm{d}\theta, \tag{3.16}$$

where $\kappa_n^v(\vec{T}_\theta)$ is the normal curvature of S at v in the direction $\vec{T}_\theta$.

Since the unit length normal vector $\vec{N}$ to $\vec{S}$ at v is an eigenvector of $\mathbf{M}_v$ associated with the eigenvalue zero, it follows that $\mathbf{M}_v$ can be factorized as follows

$$\mathbf{M_v} = \mathbf{M_{12}^{\mathrm{T}}}\begin{pmatrix} m_v^{11} & m_v^{12} \\ m_v^{21} & m_v^{22} \end{pmatrix}\mathbf{M_{12}}, \tag{3.17}$$

where $\mathbf{M}_{12} = [\vec{P}_1, \vec{P}_2]$ is the 3×2 matrix constructed by concatenating the column vectors $\vec{P}_1$ and $\vec{P}_2$. Note that $m_v^{ij} = \mathbf{M}_{ij}^{\mathrm{T}}\mathbf{M}_v\mathbf{M}_{ij}$ for any $i, j \in \{1,2\}$. The principal curvatures can then be obtained as functions of the nonzero eigenvalues of $\mathbf{M}_v$ [49]:

$$k_1 = 3m_v^{11} - m_v^{22}, \qquad k_2 = 3m_v^{22} - m_v^{11}. \tag{3.18}$$

The first step of the implementation estimates the normal vector $\vec{N}_v$ at each vertex v of the surface with the help of Eq. (3.2). Then, for each vertex v, matrix $\mathbf{M}_v$ is approximated with a weighted sum over the neighbor vertices v_i:

$$\tilde{M}_v = \sum_{i=0}^{n-1} w_i\kappa_n(\vec{T}_i)\vec{T}_i\vec{T}_i^{\mathrm{T}}, \tag{3.19}$$

where

$$\vec{T}_i = \frac{(I - \vec{N}_v\vec{N}_v^{\mathrm{T}})(v - v_i)}{\|(I - \vec{N}_v\vec{N}_v^T)(v - v_i)\|} \tag{3.20}$$

is the unit length normalized projection of vector $v_i - v$ onto the tangent plane $\langle \vec{N}_v \rangle^{\perp}$. The normal curvature in direction $\vec{T}_i$ is approximated with the help of Eq. (3.13) as $\kappa_n(\vec{T}_i) = \frac{2\vec{N}_v^T(v_i - v)}{\|v_i - v\|^2}$. The vertices and the edges are indexed as shown in Fig. 2.

The weights w_i are selected to be proportional to the sum of the surface areas of the triangles incident to both vertices v and v_i (two triangles if the surface is closed, and one triangle if both vertices belong to the boundary of S).

By construction, the normal vector $\vec{N}_v$ is an eigenvector of the matrix $\tilde{M}_v$ associated with the eigenvalue zero. Then, $\tilde{M}_v$ is restricted to the tangent plane $\langle \vec{N}_v \rangle^{\perp}$ and, using a Householder transformation [18] and a Givens rotation [18], the remaining eigenvectors $\vec{P}_1$ and $\vec{P}_2$ of $\tilde{M}_v$ (i.e., the principal directions of the surface at v) are computed. Finally, the principal curvatures are obtained from the two corresponding eigenvalues of $\tilde{M}_v$ using Eq. (3.18).

3.5. Modifications

We suggest several modifications for the paraboloid fitting [20,27,28,42,44], Watanabe and Belyaev [54], and Taubin [49] methods. We now describe our proposed modifications.

We employed the paraboloid fitting method [20,44,42,28,27] on the rings of immediate neighbors (see [12] for a free implementation). Moreover, we extended these rings to neighbors that are not immediate. We refer to the paraboloid fitting n method when rings from 1 up to n were involved in computations.

We considered two modifications to the algorithm of Watanabe and Belyaev [54]. In the following two modified algorithms, we do not change the first step of the Watanabe and Belyaev method. We employ the tangent directions $\vec{T}_i$ at vertex v produced by Eq. (3.20). Having two vertices (v and v_i), tangent direction $\vec{T}_i$ and the normal in v, we compute the radius of the fitted circle and from that derive $\kappa_n(\varphi_i)$, the normal curvature in the specified direction $\vec{T}_i$.

- *Watanabe A:* Having the normal curvatures, we apply Eqs. (3.14) and (3.15).
- *Watanabe B:* From the set of the normal curvatures of each vertex v, $\{\kappa_n^i\}_{i=0}^{n-1}$, we select the maximal (k_1) and the minimal (k_2) normal curvature values and apply the classic Eqs. (2.3) and (2.4).

We also considered two modifications of Taubin's algorithm [49]:

- *Taubin A (Constant integration):* In Eq. (3.19), the weights w_i are selected to be proportional to angles $\angle(v_i, v, v_{i+1})$ instead of the surface areas.
- *Taubin B (Smoothing with a trapezoidal rule):* The directional curvature $\kappa_n(\vec{T}_i)$ in Eq. (3.19) is selected as an average of values $\kappa_n(\vec{T}_{(i-1)\bmod n})$ and $\kappa_n(\vec{T}_i)$.

4. Experimental results

We differentiate between two categories of data: synthetic and real range. While the interest for the synthetic data is generated from the fact that it is accurate and allows for ground truth to be produced at any point, the interest in range data is motivated by the fact that in most cases it is noisy, with direct influence on the accuracy and stability of the algorithms.

Denote by K_i and H_i the values of Gaussian and mean curvatures computed by one of the methods from the triangular mesh data in vertex v_i, while $\hat{K}_i$ and $\hat{H}_i$ are the exact (analytically computed) values of the Gaussian and mean curvatures, at the same surface location $v_i = \vec{S}(r_i, t_i)$ on the corresponding surface. We considered the following error values:

(1) Average of the absolute error value of the Gaussian curvature K

$$\frac{1}{m} \sum_{i=1}^m |K_i - \hat{K}_i|;$$

(2) Average of the absolute error value of the square of the mean curvature H^2

$$\frac{1}{m} \sum_{i=1}^m |H_i^2 - \hat{H}_i^2|.$$

4.1. Comparison using synthetic data

We tested all the algorithms described in Section 3 on a set of synthetic models that represent the tessellations of four objects: a cylinder, a conus, a sphere, and a plane. Moreover, we tested all the algorithms described in Section 3 on a set of synthetic models that represent the tessellations of four NURBS surfaces: a surface of revolution generated by a non-circular arc, the body and the spout of the infamous Utah teapot model, and an ellipsoid (see Fig. 7)[1].

We built a library of triangular meshes that represent approximations (with different resolutions) of the synthetic models. For each synthetic surface, we have produced several polyhedral approximations with a varying number of triangles. The cylinder, the cone, the sphere, and the plane were created artificially and ray-traced as if they were scanned (see Fig. 6).

Consider the way in which the Cyberware range scanner captures 3D objects. This device registers the distances of 3D points to the sensors. The sensors are situated at fixed points in a vertical line, relative to the ground. The captured object is located at a platform that moves linearly in the front of the sensors. The sensors are activated at

---

[1] The synthetic models along with their dimensions and curvature values for each vertex (where is relevant) are also available in http://www.cs.technion.ac.il/~octavian/poly_crvtrs/poly_synthetic_data.
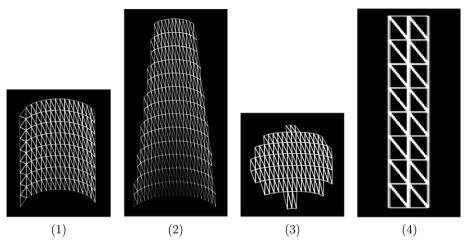
Fig. 6. The tessellations types of the synthetic counterparts of the real objects shown in Fig. 17, at relative low resolutions. The captured interior points are grouped in square-like neighborhoods. The tessellation enriches these grouping with diagonal lines.

Table 1
Dimensions of the cylinder, the cone, the sphere, and the plane in millimeters and their implicit analytic curvature values used in comparisons

				K	H^2
Cylinder	Radius = 33.25			0	0.015037594
Cone	Small radius = 24	Big radius = 50	Height = 150	0	
Sphere	Radius = 19.8			0.00255076	0.00255076
Plane	Width	Length		0	0

constant intervals of times, therefore, range images are parameterizable 3D sets of points. The directions of parameterizations are two: the first is defined by the locations where the sensors are activated, while the second is defined by the density of the sensors on the vertical line they are located at. The vast majority of interior points are captured in square-like neighborhoods. In Fig. 6, we show the square-like neighborhood simulation results together with diagonals added for triangulations building purposes.

The dimensions and the analytic curvature values for the cylinder, the cone, the sphere, and the plane are specified in millimeters in Table 1. The tessellations for the NURBS surfaces was performed using samplings in the parametric domain followed by evaluations of the 3D values on surfaces. The library files contain, for each NURBS surface and for each vertex v_i, its 3D coordinates and analytically precomputed values of the Gaussian curvature $\hat{K}_i$ and the squared value of the mean curvature $\hat{H}_i^2$. The Gaussian and mean curvature values are computed from the original NURBS surfaces.

The tessellations of each model were produced for several different resolutions: from about one hundred triangles to several thousand triangles for the finest resolution. The different tessellations of the spout surface are shown in Fig. 8. These different resolutions helped us gain some insight into the convergence rates of the tested algorithms as the accuracy of the tessellation improves.

In the vast majority of the previous results, only primitives such as cones and spheres were examined for the accuracy of these curvature approximation algorithms. The output of the tests of four different schemes on seven models (see Figs. 6 and 7) shows that while the best algorithm for the estimation of the Gaussian curvature is the *Gauss–Bonnet scheme,* the best method for the estimation of the mean curvature is the *paraboloid fitting method.*
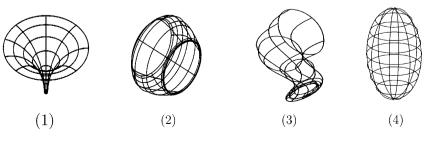

Fig. 7. The NURBS surfaces that were used for curvature estimation tests.

Bearing in mind that each mesh of the objects shown in Fig. 6 is obtained from an orthographic ray-tracing, we can attach a parametrization of two perpendicular axes to the mesh. Moreover, we create different resolutions of meshes by decreasing the resolutions, that is we purge each second column and row of samplings. In all figures in this section, the horizontal axis is used to mark the resolutions of the tessellations of the analyzed surfaces, where the origin means the coarsest resolution. The different resolutions are labelled according to their number of triangles as well as with a relative resolution indicator of form $n \times n$. This indicator shows the relative resolution of the mesh in the two directions of the attached parametrization.

Figs. 9 and 10 show the results of the tests for the tessellations of the sphere (see Fig. 6(3)). Figs. 11–16, show the results of the tests for the tessellations of the surface of revolution (see Fig. 7(1)), the spout of the Utah teapot (see Figs. 7(3) and 8) and the ellipsoid (see Fig. 7(4)).

These graphs show a partial set of the examples of the results we got throughout our tests. The Gauss–Bonnet scheme shines when K is computed and the parabolic fitting scheme works better for H, as compared to the Gauss–Bonnet scheme as well as all other schemes. Hence, the *optimal approximation scheme for triangular meshes should be based on a synergy of the two schemes.*

In the case of the sphere and the surface of revolution, (see Fig. 6(3) and Fig. 7) one can see that the accuracy in the mean curvature computation provided by the paraboloid fitting is slightly outperformed by Watanabe and Taubin's variants. However, on the free form surfaces, the paraboloid fitting is close to the Gauss–Bonnet scheme, in the case of ellipsoid being the best and in the case of the Utah spout being the second one.

Another significant result that can be drawn from these graphs is that this synergetic scheme does not always converge as the fineness of the mesh is improved, that is the higher the resolution of the mesh, the closer the values of the curvatures computed at the mesh points are to the exact (analytically computed) values. This convergence was not witnessed in all schemes, yet the parabolic fitting scheme for H always converged in the case of the free-form surfaces. In this context, the Gauss–Bonnet scheme also converges, except in the case of the surface of revolution when computing H.

The authors feel that the convergence of the paraboloid fitting method relies on the approximation provided by the paraboloid that locally approximates each point of interest. This method is designed to work well on free-form surfaces. However, bearing in mind that the paraboloid is essentially not a sphere, we think that methods that locally approximate surfaces by spheres could have advantages on spherical surfaces.

The authors consider the results shown in Figs. 13 and 14 for the spout of the Utah teapot (see Fig. 7(3) and Fig. 8), which is a free-form surface, the most relevant. The experiments on the ellipsoid free-form surface (see Figs. 15 and 16) strengthen this conclusion.

4.2. Using real range data

We tested all the algorithms described in Section 3 on triangular meshes that represent tessellations of a cylinder, a cone, a sphere, and a plane. These objects were scanned using a 3D Cyberware scanner [23]. The images of the tested objects are shown in Fig. 17. The results were compared with the analytic values of the Gaussian and mean curvatures of the scanned geometric objects, see Table 1.[2] (In the case of the cone, only the Gaussian curvature was computed.)

We built a library of triangular meshes that represent approximations (with different resolutions) of the four real objects that were scanned. For each surface, we produced several polyhedral approximations with a varying number of triangles. The library files contain the 3D coordinates of each vertex v_i.

The tessellations of each model were produced for several different resolutions: from about one hundred triangles to several thousand triangles for the finest resolution. The

Fig. 8. The tessellations of the spout surface were produced for the following resolutions: (1) 128 triangles, (2) 288 triangles, (3) 512 triangles, (4) 1152 triangles, (5) 2048 triangles, and (6) 5000 triangles.

---

[2] These models, which resulted from range image data, along with the curvature values are also available at http://www.cs.technion.ac.il/~octavian/poly_crvtrs/poly_range_data.
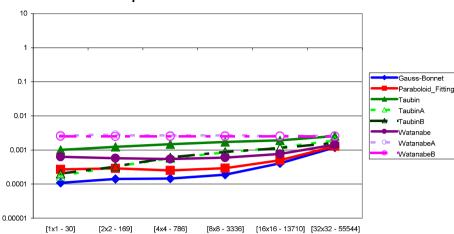
Sphere Absolute Error K

Fig. 9. Average of the absolute error for the value of the Gaussian curvature for the tessellations of the sphere (see Fig. 6(3)). The Gauss–Bonnet scheme is the most accurate being closely followed by the paraboloid fitting method.
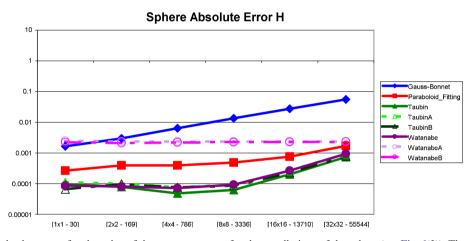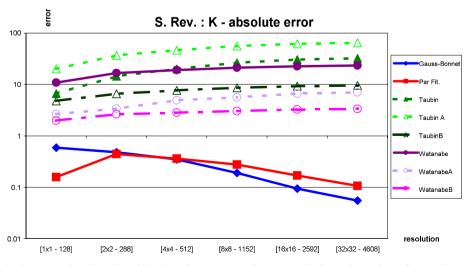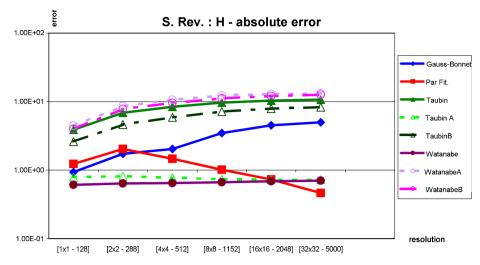
Sphere Absolute Error H

Fig. 10. Average of the absolute error for the value of the mean curvature for the tessellations of the sphere (see Fig. 6(3)). The paraboloid fitting method is slightly outperformed by Taubin and its variants, due to the fact that the normal curvature in Taubin's method is based on tangent circle approximations.

S. Rev. : K - absolute error

Fig. 11. Average of the absolute error for the value of the Gaussian curvature for the tessellations of a surface of revolution (see Fig. 7(1)). The Gauss–Bonnet and the paraboloid fitting schemes provide the best accuracy, they having decreasing errors as the resolution increases.

Fig. 12. Average of the absolute error for the value of the mean curvature for the tessellations of a surface of revolution (see Fig. 7(1)). The paraboloid fitting scheme presents increasing accuracy as the resolution increases. It clearly outperforms the Gauss–Bonnet scheme although it is slightly outperformed by Taubin A and Watanabe, which do not have increasing accuracies.
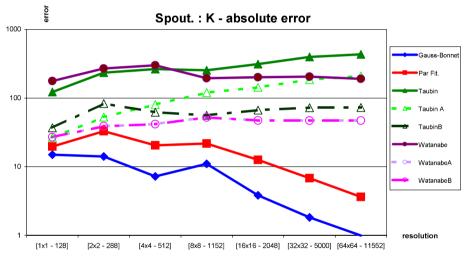

Fig. 13. Average of the absolute error for the value of the Gaussian curvature for the tessellations of the Utah teapot's spout (see Fig. 7(3)). The Gauss–Bonnet scheme provides the best accuracy.

sizes of the objects being known from a-priori measurements, their intrinsic analytic values of Gaussian and mean curvatures are available for error evaluation purposes. These four objects have the same dimensions as their synthetic counterparts (see Section 4.1). The same software that created the tessellations in the synthetic case (see Section 4.1) was used here, therefore the tessellations in the synthetic and real cases are similar (see Fig. 6).

4.2.1. Comparing methods on subsequent refined meshes

We tested all the algorithms described in Section 3 on triangular meshes that represent refined tessellations of range data images representing a cylinder, a cone, a sphere, and a plane. We used the same way of representing the results as in Section 4.1.

In Figs. 18–21, we show the results for the sphere and the plane in Fig. 17(3) and (4). At high resolutions,

high errors in computing the Gaussian and mean curvatures were detected, due to the fact that the relative distances among the points are comparable to the scanning errors. Fig. 22 illustrates this problem. Note that when filtering is used, this problem is alleviated (see Section 4.2.3).

Two different tessellated surfaces of a scanned ping-pong ball (the sphere. in Fig. 17(3)) are shown in Fig. 22. When the scanning resolution is low, the relative distances among the points are higher than the scanning errors and the graphs (Figs. 18 and 19) are consistent with the observation presented in Section 4.1. As can be seen, at high resolutions, the Watanabe A and B methods provide the best results, although, at lower resolutions, the Gauss–Bonnet and the paraboloid fitting methods are preferable. We computed the corresponding graphs for the cylinder and the cone, and obtained similar results.
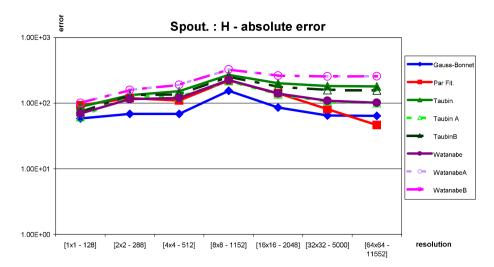
Fig. 14. Average of the absolute error for the value of the mean curvature for the tessellations of the Utah teapot's spout (see Fig. 7(3)). The paraboloid fitting is outperformed by the Gauss–Bonnet scheme only.
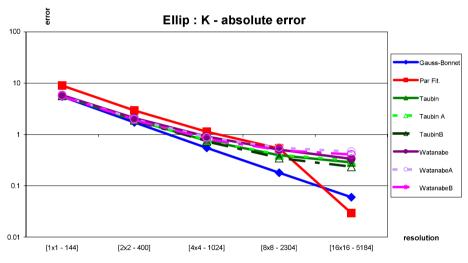

Fig. 15. Average of the absolute error for the value of the Gaussian curvature for the tessellations of an ellipsoid (see Fig. 7(4)). The Gauss–Bonnet scheme provides the best accuracy.
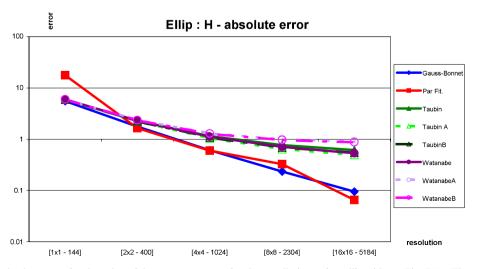

Fig. 16. Average of the absolute error for the value of the mean curvature for the tessellations of an ellipsoid (see Fig. 7(4)). The paraboloid fitting and the Gauss–Bonnet scheme provide the best accuracy.
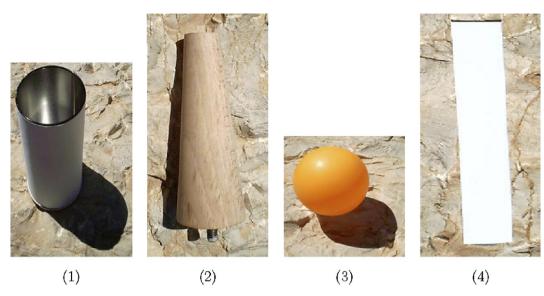
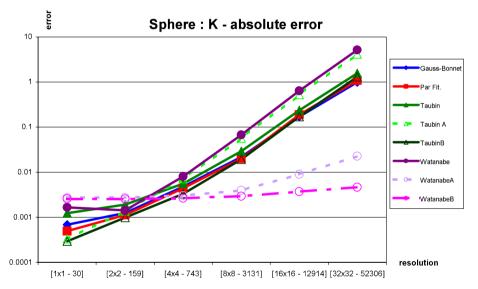Fig. 17. Real objects used in experiments.

Fig. 18. Average of the absolute error for the value of the Gaussian curvature for the tessellations of the sphere, which is the ping-pong ball (see Figure 17(3)). Although at high resolution, Watanabe A and B have higher accuracies, the Gauss–Bonnet and paraboloid fitting schemes present more accurate computations at lower resolution meshes.

A common feature of all the graphs for the cylinder, the cone, the sphere, and the plane is that at very high level of noise, the Watanabe's A and B method present improved accuracies. However, at low level resolutions of the meshes, all the methods give very small errors in curvature accuracy computation.

As a common characteristic of all the graphs in this section and their counterparts in Section 4.1, we observe that the errors detected at higher resolutions of the meshes are higher than the ones computed at lower resolutions. All the graphs, except Gauss–Bonnet and paraboloid fitting in the free-form cases, have an ascending tendency.

4.2.2. Comparing paraboloid fitting multi-ring methods

Contemporary 3D acquisition devices are able to provide very dense clouds of points. However, the accuracy of these 3D points is not satisfactory. One way to cope with the inaccuracy is to use extended regions of neighborhoods [19,21].

The authors believe that the best accuracy in curvature computation can be achieved when one employs multi-ring methods. We ran the paraboloid fitting method using a different number of rings on the same image. For example, Fig. 23 shows the results of running the paraboloid fitting method on the cylinder (with maximum resolution). We computed similar graphs for the cone, the sphere, and the
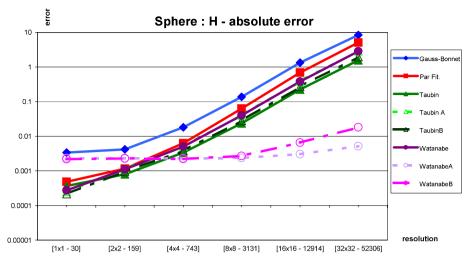
Fig. 19. Average of the absolute error for the value of the mean curvature for the tessellations of the sphere, which is the ping-pong ball (see Fig. 17(3)). Although at high resolution, Watanabe A and B have higher accuracies, the paraboloid fitting scheme presents more accurate computations at lower resolution meshes.
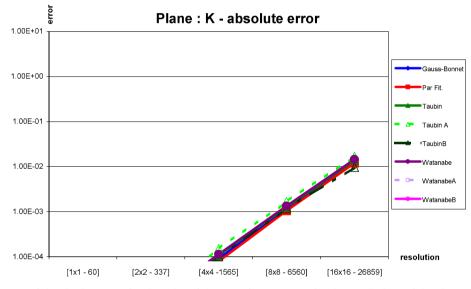

Fig. 20. Average of the absolute error for the value of the Gaussian curvature for the tessellations of the plane (see Fig. 17(4)).

plane. In all these graphs the objects selected have the maximal attainable resolution. The number of the rings varies from one to four, on the horizontal axis. Fig. 23 as well as graphs that reflect results of running the multi-ring paraboloid fitting method on the cone, the sphere, and the plane show that methods of curvature computation that use more rings provide better approximations to curvature values.

All these graphs are characterized by convergence to the exact results when the first four rings were considered. In this context, an interesting and non-negligible issue when working with multi-ring methods is their computation time consumption (see Section 4.3).

The multi-ring version of the paraboloid fitting method behaves as a low pass filter. At the highest attainable resolution, the distances between adjacent points of the meshes is approximatively 0.5 mm on average and the errors in the measurements are approximatively 0.1 mm, which amounts to approximatively twenty percent. At these resolutions, low pass filtering is the key for better approximations, and the graphs show that four rings still do not introduce an error higher than the relative error in the positions of the captured points. The limits at which a low pass filter, or equivalently, the multi-rings method provides better results are shown in Figs. 24 and 25.

Figs. 24 and 25 show the results of running the multi-ring versions of the paraboloid fitting method on the same object with, however, different resolutions. We gradually increased the resolution of the meshes by four gradually at each abscissa value. The graphs show that more rings improve the approximations as long as the error in the
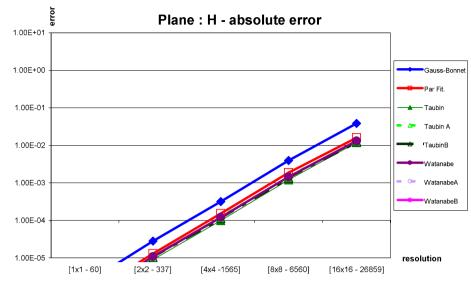
Fig. 21. Average of the absolute error for the value of the mean curvature for the tessellations of the plane (see Fig. 17(4)).
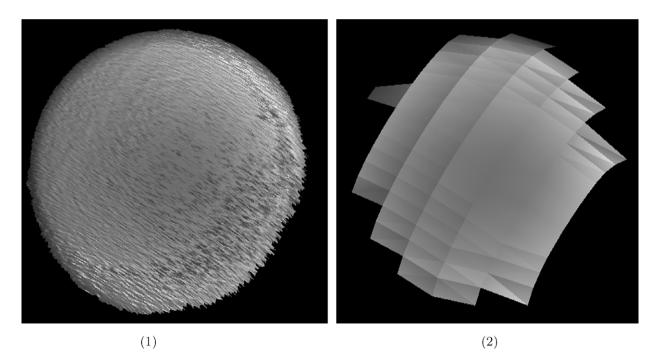

Fig. 22. The surface of the ping-pong ball as it was scanned: 52,306 triangles (1) and the most coarse approximation: 159 triangles (2) (the sphere in Fig. 17(3)).

information provided by the rings does not exceed the error in the information at the exact location of the analyzed point. Practically, if methods that employ more rings are available, they should be preferred when working on high resolution meshes.

Consider, for example, Fig. 24. At the highest attainable resolution, the multi-ring paraboloid fitting with 40 rings provides the best estimation for curvature approximation. The more rings are used, the better the result is at this resolution. The differentiation is the same even at lower resolutions, 4×4 times coarser than the highest one considered

here. However, when the meshes are sparser, and the points are farther from each other, we do not attain more accurate results by applying low pass filters. The average distance between adjacent points is $0.5 \times 4 = 2$ mm (at 2×2 coarser resolution-marked 4×4 in Figs. 24 and 25) whereas the error remains the same 0.1 mm. In conclusion, multi-ring methods provide better performance than one-ring methods when applied on meshes with errors in point locations of upto 20 percent, relative to distances between adjacent points. Multi-rings methods are able to overcome errors in scanning, however, only up to the point where they
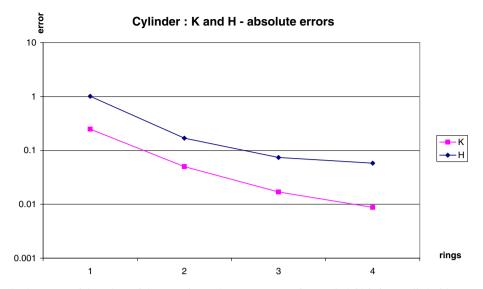
Fig. 23. Average of the absolute error of the values of the Gaussian and mean curvature for paraboloid fitting applied with one to four rings. The method receives as input the cylinder with maximal resolution.
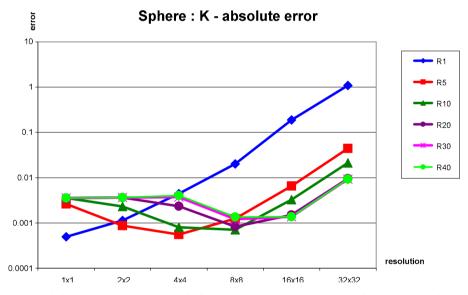

Fig. 24. Results of running the multi-ring versions of the paraboloid fitting method on a sphere provided in several resolutions, for the computation of the Gaussian curvature. The amount of rings is indicated in the right box. The resolutions are represented on the horizontal axis. Note that R20, R30, and R40 have almost equal values at the maximum resolution.

modify the locally approximated surface of the captured objects. The use of any more rings introduces errors in computing curvatures values that are greater than the ones resulting from the point capturing processes.

Fig. 25, shows the same behavior as Fig. 24. We computed the corresponding graphs for the cylinder, the cone, and the plane, and obtained similar behavior.

4.2.3. Comparing methods on filtered range images

We tested all the algorithms described in Section 3 on triangular meshes that represent tessellations of Gaussian filtered range data images representing a cylinder, a cone, a sphere, and a plane. The Gaussian filter was applied to

the depth component of the scanned points that form the meshes of the four objects. In all these graphs we represent on the abscissa the α factor used in the Gaussian filter,

$$h(t) = \frac{\sqrt{\pi}}{\alpha} \exp\left(\frac{-\pi^2 t^2}{\alpha^2}\right). \tag{4.1}$$

We show comparison results for the cylinder and the sphere in Fig. 17(1) and (3). The results as graphs are shown in Figs. 26–29. We filtered the meshes representing the cylinder and the sphere employing a Gaussian low pass filter Eq. (4.1). The values of the graphs represent the detected average errors in computing the Gaussian and
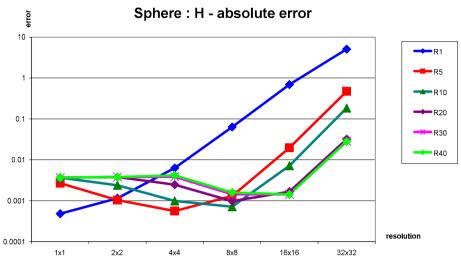
Fig. 25. Results of running the multi-ring versions of the paraboloid fitting method on a sphere provided in several resolutions, for the computation of the mean curvature. The amount of rings is indicated in the right box. The resolutions are represented on the horizontal axis. Note that R20, R30, and R40 have almost equal values at the maximum resolution.
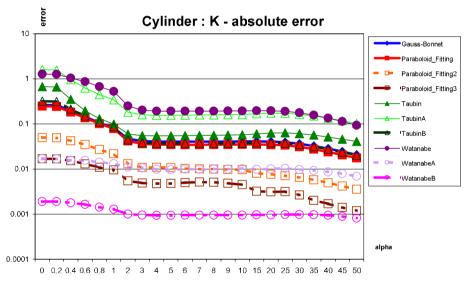

Fig. 26. Results of running all the methods on the cylinder, for the computation of the Gaussian curvature. The cylinder model was filtered with a Gaussian filter. The horizontal axis represents the radius of the filter whereas the vertical one represents the error. Note that the Gauss–Bonnet and the paraboloid fitting methods have very close values.

the mean curvatures. Similar graphs were obtained for the cone and the plane.

Figs. 26 and 27 show that all the graphs are monotonically decreasing for $\alpha \in [0..4]$. For values of $\alpha \geqslant 4$ the graphs are non-monotonically decreasing. Moreover, they are even increasing due to the fact that the Gaussian filter modifies the objects and thus the values of the Gaussian and the mean curvatures at any point on the meshes. Similar behavior can be seen in all the graphs; see the additional examples in Figs. 28 and 29.

The best method for computing the Gaussian curvature when Gaussian filtering is used is Watanabe B. For the mean curvature the best method is Watanabe A. In this context, we mention that the paraboloid fitting

method is one of the best methods for mean curvature estimation. The four objects analyzed in this section are particular geometric objects and the authors feel that the paraboloid fitting method is very appropriate, especially on free-form surfaces such as the surface of revolution (see Fig. 7(1)) and the spout of the Utah teapot (see Figs. 7(3) and 8). A study on such surfaces requires solving registration problems and is not in the scope of the current work.

Note that in the way in which we applied the Gaussian low pass filter, the graphs showed convergence of the methods up to point ($\alpha \approx 4$), where the surfaces begin to change. Taking into account that Figs. 26 and 27 have logarithmic scale representations, we conclude that all the
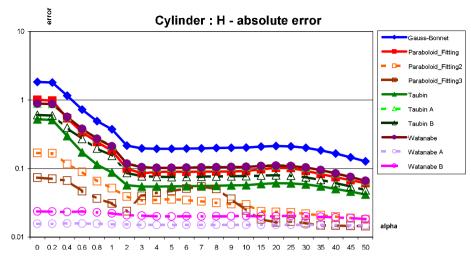
Fig. 27. Results of running all the methods on the cylinder, for the computation of the mean curvature. The cylinder model was filtered with a Gaussian filter. The horizontal axis represents the radius of the filter whereas the vertical one represents the error.
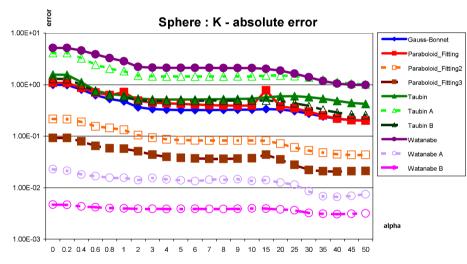

Fig. 28. Results of running all the methods on the sphere, for the computation of the Gaussian curvature. The sphere model was filtered with a Gaussian filter. The horizontal axis represents the radius of the filter whereas the vertical one represents the error. Note that the Gauss–Bonnet, the paraboloid fitting, the Taubin, and the Taubin B methods have very close values.

methods received improved input by filtering up to ($\alpha \approx 4$). By applying more specialized filters, one can recover the geometry of scanned objects better. In this case, the Gauss–Bonnet scheme remains the best choice for the Gaussian curvature computation and the paraboloid fitting method is the best for the mean curvature computation.

Note that especially for the case of the Gaussian curvature, the comparison between error results is difficult and perhaps irrelevant when all the methods report very low values. This fact is dictated by numerical reasons such as the condition numbers of the implied formulas.

4.3. Computation time requirements

We compared the running times of all the algorithms described in Section 3. Table 2 represents a comparison

of times required for computing the curvatures on the highest resolution available tessellations for a cylinder, a conus, a sphere, and a plane. The most interesting result relates to the paraboloid fitting 2 and 3 methods. We measured computation times on a personal computer equipped with two Pentium IV hyper-threading 2.4 GHz processors and 1 Gb of memory.

5. Conclusions and future work

In this work, we provided a comparison of four different approaches for curvature estimation of triangular meshes. For each approach, we selected a representative algorithm. The input data comprised synthetic geometric objects as well as range data obtained from scanning real 3D objects.

In the case of synthetic models, the Gauss–Bonnet scheme excels when K is computed and the parabolic fitting
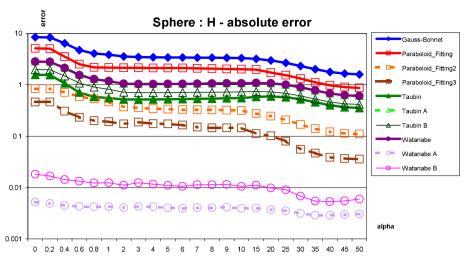
Fig. 29. Results of running all the methods on the sphere, for the computation of the mean curvature. The sphere model was filtered with a Gaussian filter. The horizontal axis represents the radius of the filter whereas the vertical one represents the error. Note that the Taubin A and the Watanabe methods have very close values.

Table 2
Computation time required by each of the methods for a cylinder, a conus, a sphere, and a plane, in seconds

	Cylinder	Conus	Sphere	Plane
Gauss–Bonnet	1	1	1	1
Watanabe	2	1	<1	1
Watanabe A	3	3	1	1
Watanabe B	3	3	1	1
Taubin	2	2	<1	1
Taubin A	2	2	1	1
Taubin B	2	2	<1	1
Paraboloid fitting	51	15	4	6
Paraboloid fitting 2	72	21	6	8
Paraboloid fitting 3	85	36	9	15

scheme works best for *H*. Hence, the *optimal approximation scheme for triangular meshes should be based on a synergy of the two schemes*. Moreover, for *K*, the parabolic fitting scheme was second in many cases, especially on synthetic surfaces, and followed the Gauss–Bonnet scheme closely. If one must select only one method of choice, the most stable method that always has a good convergence is the osculating paraboloid fitting scheme. Another significant result is that both the Gauss–Bonnet and the paraboloid fitting methods converge as the fineness of the mesh is improved on synthetic surfaces. This convergence was not witnessed in all the schemes. We mention that in [33] the authors proved that the paraboloid fitting method, in a particular case of interpolation, has a quadratic error bound (using asymptotic analysis) even for non-uniform meshes (that is usually the case in practice), while the Gauss–Bonnet scheme has a quadratic error bound in the case of uniform mesh and linear error bound.

In the case of real range image data, the output of the tests of four different schemes on four models (Fig. 17) shows that the conclusions obtained on synthetic data are valid over the set of real range images data. However, when the resolution is very high, the relative error of the scanning process perturbs the accuracy of the Gaussian and the mean curvatures values. In this case, the most stable method is Watanabe B for Gaussian curvature and Watanabe A for the mean curvature.

Developing analytic expressions for the error of the approximations is of high interest. One of the future directions of our research is to choose several basic primitive objects, such as a sphere, an ellipsoid, or a paraboloid, and so on, and to compute analytic evaluations for the errors of the different methods.

An interesting aspect is hidden in the triangulations that one uses. Different methods provide different accuracies of the Gaussian and mean curvatures values on different styles of meshes. This problem is partially analyzed in [6]. We consider this issue as a future research direction that has a high potential and one that could shed new light on the quality of curvature computations.

An aspect to which curvature computation comparison work should relate is finding the best parameters for each method that participates in the comparison [43]. In this context, the use of adapting filters [53,55] on the input data should be more thoroughly considered. In the future, it will be interesting to do further research into a comparison of methods with parameters tuned to their highest capacity. In this context, the authors believe that studying the generalizations of the methods of curvature computation to ones that use extended neighborhoods is the key to fully exploiting the information provided by 3D scanners.

In [4], the authors proved that there are differential characteristics other than the Gaussian and the mean curvatures (see [26]) that provide better accuracy for description of objects. The knowledge of the accuracy provided by other differential characteristics accompanied by their memory and time requirements is an important future research theme.

Acknowledgments

The authors thank Dr. Tatiana Surazhsky and Prof. Gershon Elber for their advice during this research.

References

[1] L. Alboul, R. van Damme, Polyhedral metrics in surface reconstruction: Tight triangulations, in: University of Twente, Department of Applied Mathematics, Technical Report, Memorandum No. 1275 (1995) pp. 309–336.

[2] R. Bergevin, A. Bubel, Detection and characterization of junctions in a 2d image, Computer Vision and Image Understanding 93 (2004) 288–309.

[3] P.J. Besl, R.C. Jain, Invariant surface characteristics for 3d object recognition in range images, Computer Vision Graphics and Image Processing 33 (1986) 33–80.

[4] H. Cantzler, R.B. Fisher, Surface shape and curvature scales, in: The Third International Conference on 3-D Digital Imaging and Modeling, 2001, pp. 285–291.

[5] X. Chen, F. Schmitt, Intrinsic surface properties from surface triangulation, in: The Second European Conference on Computer Vision, May 1992, pp. 739–743.

[6] D. Cohen-Steiner, J.-M. Morvan, Restricted delaunay triangulations and normal cycle, in: Proceedings of the Nineteenth Annual Symposium on Computational Geometry, 2003, pp. 237–246.

[7] G. Dahlquist, Å. Björck, Numerical methods, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[8] H. Delingette, General object reconstruction based on simplex meshes, International Journal of Computer Vision 32 (2) (1999) 111–146.

[9] M. DoCarmo, Differential Geometry of Curves and Surfaces, Prentice-Hall, NJ, 1976.

[10] G. Dudek, J.K. Tsotsos, Shape representation and recognition from multiscale curvature, Computer Vision and Image Understanding 68 (2) (1997) 170–189.

[11] N. Dyn, K. Hormann, S. Kim, D. Levin, Optimizing 3d triangulations using discrete curvature analysis, in: T. Lyche, L.L. Schumaker (Eds.), Mathematical Methods in CAGD: Oslo 2000, Vanderbilt University Press, Nashville, TN, 2001, pp. 135–146.

[12] G. Elber. www.cs.technion.ac.il/~irit/.

[13] T.L. Faber, E.M. Stokely, Orientation of 3d structures in medical images, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (1988) 626–633.

[14] F.P. Ferrie, M.D. Levine, Deriving coarse 3d models of objects, in: Conference on Computer Vision and Pattern Recognition, June 1988, pp. 345–353.

[15] P.J. Flynn, A.K. Jain, On reliable curvature estimation, in: Conference on Computing Vision and Pattern Recognition, June 1989, pp. 110–116.

[16] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, in: Computer graphics, principles and practice in C, second ed., Addison-Wesley Publishing Company, MA, 1996.

[17] D.B. Goldgof, T.S. Huang, H. Lee, A curvature-based approach to terrain recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (11) (1989) 1213–1217.

[18] G.H. Golub, C.F.V. Loan, in: Matrix computations, third ed., The Johns Hopkins University Press, 1996.

[19] M. Gopi, S. Krishnan, C.T. Silva, Surface reconstruction based on lower dimensional localized delaunay triangulation. in: M. Gross, F.R.A. Hopgood (Eds.) Computer Graphics Forum, Eurographics, vol. 19, 3, 2000.

[20] B. Hamann, Curvature approximation for triangulated surfaces, Computing Supplements 8 (1993) 139–153.

[21] E. Hameiri, I. Shimshoni, Estimating the principal curvatures and the darboux frame from real 3-d range data, IEEE Transactions on Systems, Man and Cybernetics, Part B, Issue 4 33 (2003) 626–637.

[22] A. Hilton, J. Illigworth, T. Windeatt, Statistics of surface curvature estimates, Pattern Recognition Letters 28 (8) (1995).

[23] http://www.cyberware.com.

[24] S.J. Kim, C.-H. Kim, D. Levin, Surface simplification using discrete curvature norm, in: The Third Israel-Korea Binational Conference on Geometric Modeling and Computer Graphics, Seoul, Korea, October 2001.

[25] Y. Kim, R.C. Luo, Validation of 3d curved objects: cad model and fabricated workpiece, IEEE Transactions on Industrial Electronics 41 (1) (1994) 125–131.

[26] J. Koenderink, A. van Doorn, Surface shape and curvature scales, Image and Vision Computing 10 (8) (1992) 557–565.

[27] P. Krsek, C. Lukács, R.R. Martin, Algorithms for computing curvatures from range data, in: The Mathematics of Surfaces VIII, Information Geometers, in: A. Ball et al. (Eds.), 1998, pp. 1–16.

[28] P. Krsek, T. Pajdla, V. Hlaváč, Estimation of differential parameters on triangulated surface, in: The 21st Workshop of the Austrian Association for Pattern Recognition, May 1997.

[29] C.-K. Lee, R.M. Haralick, K. Deguchi, Estimation of curvature from sampled noisy data, in: Conference on Computer Vision and Pattern Recognition, 1993, pp. 536–541.

[30] C. Lin, M.J. Perry, Shape description using surface triangulation, in: Proceedings of the IEEE Workshop on Computer Vision: Representation and Control, 1982, pp. 38–43.

[31] R.R. Martin, Estimation of principal curvatures from range data, International Journal of Shape Modeling 4 (1998) 99–111.

[32] A.M. McIvor, J. Valkenburg, A comparison of local surface geometry estimation methods, Machine Vision and Applications 10 (1997) 17–26.

[33] D.S. Meek, D.J. Walton, On surface normal and gaussian curvature approximations given data sampled from a smooth surface, Computer Aided Geometric Design 17 (6) (2000) 521–543.

[34] F. Mokhtarian, N. Khalili, P. Yuen, Estimation of error in curvature computation on multi-scale free-form surfaces, IEEE Journal of Computer Vision 48 (2) (2002) 131–149.

[35] O. Monga, S. Benayoun, Using partial derivates of 3d images to extract typical surface features, Computer Vision and Image Understanding 61 (2) (1995) 171–189.

[36] S.M. Naik, R.C. Jain, Spline-based surface fitting on range images for cad applications, in: Conference on Computer Vision and Pattern Recognition, June 1988, pp. 249–253.

[37] D.L. Page, Y. Sun, J. Paik, M.A. Abidi, Normal vector voting: Crease detection and curvature estimation on large, noisy meshes, Graphical Models 64 (2002) 199–229.

[38] U. Pinkall, K. Polthier, Computing discrete minimal surfaces and their conjugates, Experimental Mathematics 2 (1) (1993) 15–36.

[39] K. Polthier, M. Schmies, Straightest geodesics on polyhedral surfaces, in: K. Polthier, H.C. Hege (Eds.), Mathematical Visualization, Springer-Verlag, Berlin/New York, 1998, pp. 391–409.

[40] F.K.H. Quek, R.W.I. Yarger, C. Kirbas, Surface parametrization in volumetric images for curvature-based feature classification, IEEE Transactions on Systems, Man, and Cybernetics 33 (5) (2003) 758–765.

[41] C. Rösl, L. Kobbelt, H. Seidel, Extraction of feature lines on triangulated surfaces using morphological operators, in: Smart Graphics, AAAI Symposium, 2000, pp. 71–75.

[42] P.T. Sanders, S.W. Zucker, Inferring surface trace and differential structure from 3d images, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (9) (1990) 833–854.

[43] A.J. Stoddart, J. Illingworth, T. Windeatt, Optimal parameter selection for derivative estimation from range images, Image and Vision computing 13 (8) (1995) 629–635.

[44] E.M. Stokely, S.Y. Wu, Surface parameterization and curvature measurement of arbitrary 3d-objects: five practical methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (8) (1992) 833–840.

[45] D. Struik, Lectures on Classical Differential Geometry, Addison-Wesley Series in Mathematics (1961).

[46] T. Surazhsky, E. Magid, O. Soldea, G. Elber, E. Rivlin. A comparison of gaussian and mean curvatures estimation methods on triangular

meshes, in: International Conference on Robotics and Automation (ICRA2003), Taipei, Taiwan, 14–19 September 2003, pp. 1021–1026.

[47] C.-K. Tang, G. Medioni, Curvature-augmented tensor voting for shape inference from noisy 3d data, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (6) (2002) 858–864.

[48] C.-K. Tang, G. Medioni, M.-S. Lee, N-dimensional tensor voting and application to epipolar geometry estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8) (2001) 829–844.

[49] G. Taubin, Estimating the tensor of curvature of a surface from a polyhedral approximation, in: The Fifth International Conference on Computer Vision, 1995, pp. 902–907.

[50] J.P. Thirion, A. Gourdon, Computing the differential characteristics of isointensity surfaces, Computer Vision and Image Understanding 61 (2) (1995) 190–202.

[51] W.S. Tong, C.-K. Tang, Robust estimation of adaptive tensors of curvature by tensor voting, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 434–449.

[52] E. Trucco, R.B. Fisher, Experiments in curvature-based segmentation of range data, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (2) (1995) 177–182.

[53] L.J. van Vliet, I.T. Young, P.W. Verbeek, Recursive gaussian derivative filters. in: Proceedings of the 14th International Conference on Pattern Recognition, vol. 1, August 1998, pp. 509–514.

[54] K. Watanabe, A.G. Belyaev, Detection of salient curvature features on polygonal surfaces, in: Eurographics 2001, vol. 20, No. 3, 2001.

[55] I. Weiss, High-order differentiation filters that work, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (7) (1994) 734–739.

[56] K. Wu, M.D. Levine, 3d part segmentation using simulated electrical charge distributions, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (11) (1997) 1223–1235.

[57] S.M. Yamany, A.A. Farag, Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002).