

Curvature estimation for meshes via algebraic quadric fitting

Marcel Makovník*
 Pavel Chalmovianský

Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava

April 2023

Abstract

We introduce the novel method for estimation of mean and Gaussian curvature and several related quantities for polygonal meshes. The algebraic quadric fitting curvature (AQFC) is based on local approximation of the mesh vertices and associated normals by a quadratic surface. The quadric is computed as an implicit surface, so it minimizes algebraic distances and normal deviations from the approximated point-normal neighbourhood of the processed vertex. Its mean and Gaussian curvature estimate is then obtained as the respective curvature of its orthogonal projection onto the fitted quadratic surface. Experimental results for both sampled parametric surfaces and arbitrary meshes are provided. The proposed method AQFC approaches the true curvatures of the reference smooth surfaces with increasing density of sampling, regardless of its regularity. It is resilient to irregular sampling of the mesh, compared to the contemporary curvature estimators. In the case of arbitrary meshes, obtained from scanning, AQFC provides robust curvature estimation.

1 Introduction

Characterization of the shape is well described for smooth curves and surfaces. The basic notion of curvature is subject to differential geometry, whose rigorous review may be found in e. g. [DC16] or [Spi99]. With the development of computer graphics, the approximation of surfaces by piecewise linear surfaces – polygon meshes – became predominant. Hence, the natural question arises about how the geometric properties known for such surfaces (especially smooth) may be described for their piecewise linear approximations. The curvature estimation is widely used in areas like surface segmentation [LKK15, BB20], mesh simplification [YHXL15], remeshing [SLC⁺19] or feature line extraction [HPW05] and even in architecture [PLW⁺07].

Approaches for the estimation of curvature may be divided into several categories. One of the most frequently used is the construction of discrete curvature operators based solely on the geometry of the mesh. The second approach estimates curvature by taking its value from the underlying approximating surface.

We employ the approach, where we fit an implicit quadric to a small part of the mesh. In contrary to the majority of methods, our contribution is, that we use also vertex normals in the fitting and the quadric is fitted in algebraic sense. This means, we do not minimize Euclidean distance, which usually leads to solving least squares. Instead, we focus on algebraic distance which approximates the Euclidean distance well in the vicinity of the fitted quadric, see [Tau94]. Moreover, we minimize also the deviation between the normals associated to vertices and the gradient of the fitted quadric.

The paper is organized as follows. In Section 2, we mention several frequently used approaches for approximation of curvature for meshes. Then, we provide the description of our approach for computation of curvatures using quadric fitting in Section 3. The last part of our contribution, Section 5, is dedicated to experimental results, comparing our approach to frequently used methods on publicly available data. The quality of the approximating mesh and its approximation of the curvature values are examined. The summary of our work is found in Section 6.

*Corresponding author: marcel.makovnik@fmph.uniba.sk

2 Related works

The first works on approximation of curvatures on meshes were motivated not only by needs of geometric modelling, but also by many algorithms used in computer vision. One of the early approaches used construction of parametric surface, which is a graph of a function given by a bivariate polynomial. For each vertex of the triangle mesh, this polynomial approximated the distances between the vertices of one-ring and the tangent plane passing through that vertex [Ham93]. One-ring neighbourhood of the inspected vertex was used also by Taubin in [Tau95] to estimate the curvature tensor. The proposed algorithm works for triangle meshes, since the necessary vertex normals are computed from face normals. The discretization of the curvature theory using mesh parallelity was explored in [BPW10].

One of the most frequently used approximations was introduced in [MDSB03]. The popularity of this method for meshes is caused by its simple implementation, since the mean and Gaussian curvatures are computed using the areas and angles of triangles adjacent to the explored vertex. In [YLH⁺06] the extraction of principle curvatures was achieved using principal component analysis of local neighbourhoods.

Another class of the estimation methods is to choose the curvature of the underlying surface. Semenova estimated curvatures using quadric, which were represented as functions in Monge form [SSH04]. The local fitting of a biquadratic Bézier patch, whose control net was obtained by least squares fitting, was utilized in [RB05]. Another method uses algebraic fit of spheres for either point sets or point-normal sets, which immediately gives the value of mean curvature by using the radius of the sphere [GG07]. Another way of describing the underlying surface was presented in [ÖGG09], where moving least squares are utilized. One of the main advantages of using the underlying surface is, that it is independent on the graph structure of a polygon mesh, i. e. the curvature may be described also for point clouds. Estimation of the curvature per triangle rather than vertex was suggested in [ZGYL11]. Each triangle, whose each corner was given by a vertex and an associated normal, was interpolated by triangular Bézier patch and the first and the second fundamental form was obtained from its parametrization.

A robust statistical approach for estimation of the curvature tensor was introduced in [KSNS07]. The provided algorithm is capable of correcting the noise in the meshes. Its extension for processing of point clouds was introduced in [KNSS09]. Principled construction of discrete differential operators may be used for computation of curvatures of arbitrary polygon meshes, see [DGBD20]. Several analyses and empirical comparisons of curvature estimation methods are available in [GG04, GG06, DM14, VVP⁺16] or [KWH18].

3 Curvature for implicit surfaces

Since in our research we are interested in algebraic fitting of the surfaces, we enlist basic notions concerning implicit surfaces which are used throughout our work. Subsequently, we include the curvature related notions in terms of implicit surfaces.

Let $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a differentiable real valued function of three variables. Then, an *implicit surface* $\mathcal{Z}(g)$ is the set of all zeros of g , i. e.

$$\mathcal{Z}(g) = \{(x, y, z)^\top \in \mathbb{R}^3 \mid g(x, y, z) = 0\}. \quad (1)$$

The gradient of the implicit function $\nabla g(x, y, z)$ represents the normal vector at the point $(x, y, z)^\top \in \mathcal{Z}(g)$. If the function $q : \mathbb{R}^3 \rightarrow \mathbb{R}$ is given by

$$\begin{aligned} q(x, y, z) = & a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + a_{12}xy + a_{13}xz + a_{23}yz \\ & + a_{14}x + a_{24}y + a_{34}z + a_{44}, \end{aligned} \quad (2)$$

with $a_{11}, \dots, a_{44} \in \mathbb{R}$ and at least one of the numbers a_{11}, \dots, a_{23} being non-zero, we obtain the surface $\mathcal{Z}(q)$ known as a *quadratic surface* or simply *quadric*.

Consider a regular surface $\mathcal{S} \subset \mathbb{R}^3$. For each point $P \in \mathcal{S}$, we may determine the values κ_1 and κ_2 , called *principal curvatures*, that are the maximum and minimum of normal curvatures at that point. For parametrized surfaces, the principal curvatures κ_1 and κ_2 are easily computed using the coefficients of the *first* and *second fundamental forms* [Spi99]. Subsequently, we may define the two well-known measures of curvature – *Gaussian curvature* K and *mean curvature* H given by

$$K = \kappa_1 \kappa_2, \quad \text{and} \quad H = \frac{\kappa_1 + \kappa_2}{2}. \quad (3)$$

For our needs, it is suitable to use curvature formulas, obtained from the function $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ defining the implicit surface $\mathcal{Z}(g)$. More precisely, these formulas rely on the gradient ∇g and Hessian \mathbf{H}_g of the function g . These formulas (and their proofs) for both implicit planar curves and implicit surfaces may be found in [Gol05]. The principal curvatures are derived from the Gaussian and mean curvatures, as follows.

Let $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ and let V be the regular point of the corresponding surface $\mathcal{Z}(g)$. The Gaussian curvature K at the point V is then given by

$$K = \frac{\nabla g(V)^\top \text{adj}(\mathbf{H}_g(V)) \nabla g(V)}{\|\nabla g(V)\|^4}, \quad (4)$$

and the mean curvature H at V is given by

$$H = \frac{\nabla g(V)^\top \mathbf{H}_g(V) \nabla g(V) - \|\nabla g(V)\|^2 \text{tr}(\mathbf{H}_g(V))}{2\|\nabla g(V)\|^3}, \quad (5)$$

where $\nabla g(V)$ and $\mathbf{H}_g(V)$ denote the value of the gradient and the Hessian of the function g at the point $V = (x_V, y_V, z_V) \in \mathcal{Z}(g)$, respectively. For completeness, knowing the values K and H we can now easily compute the principal curvatures as

$$\kappa_{1,2} = H \pm \sqrt{H^2 - K}. \quad (6)$$

As introduced in [Koe90], one may define the measure of the size and shape using the polar coordinates in the (κ_1, κ_2) -plane. Let κ_1 and κ_2 be the principal curvatures of a point V of the regular surface $\mathcal{S} \subset \mathbb{R}^3$. Then, the number

$$R = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \quad (7)$$

is called *curvedness* and the number

$$S = -\frac{2}{\pi} \arctan \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2} \quad (8)$$

is called *shape index* at the point V .

The interpretation of curvedness and shape index might be more intuitive in comparison to the traditional measures – the mean and Gaussian curvature. Clearly, $R \geq 0$ and $-1 \leq S \leq 1$. On contrary, the bounds of H and K vary from surface to surface. The curvedness describes, what is the difference between the surface and a plane in the vicinity of the point $V \in \mathcal{S}$. The values of R approaching zero signify more gently curved points, while $R = 0$ implies that the point is planar (i. e. $\kappa_1 = \kappa_2 = 0$). The values of S determine the shape of a local patch at the point V . This allows us to classify points – whether they are elliptic, parabolic, hyperbolic or umbilic. Note, that the shape index is indeterminate for planar points.

To compute curvedness and shape index for implicit surfaces, it is more efficient to omit the computation of principal curvatures and use the Gaussian and mean curvature instead. Then, the curvedness R and shape index S at the point V are given by

$$R = \sqrt{2H^2 - K}, \quad (9)$$

and

$$S = -\frac{2}{\pi} \arctan 2\sqrt{H^2 - K}. \quad (10)$$

4 Quadric fitting curvature

In this section we propose the novel method, referred to as Algebraic quadric fitting curvature (AQFC), to determine mean and Gaussian curvature at vertex of a polygon mesh. The method AQFC is based on finding an underlying surface to a certain area of the mesh, similarly as [GG07] and [ÖGG09]. More precisely, the surface is a quadric fitted to a certain set of points and their associated normals.

4.1 Input data and quadric computation

The proposed method may process any type of polygonal mesh $\mathcal{M} = \{\mathcal{V}_N, \mathcal{E}, \mathcal{F}\}$, where \mathcal{V}_N is the set of vertex-normals, i. e. each vertex is associated with a normal vector, and \mathcal{E} and \mathcal{F} are respectively the set of edges and faces. The vertex normals may be prescribed or we may use any of the methods of the computation listed in [JLW05], e. g. simple averaging of adjacent face normals or averaging weighted by vertex angles.

For each processed vertex-normal $(V, \mathbf{n}) \in \mathcal{V}_N$, we assign a neighbourhood \mathcal{H} by finding the smallest possible vertex ring (including (V, \mathbf{n})), containing fixed number of vertex-normals. Subsequently, we determine a quadric $\mathcal{Z}(q)$ which is the best approximation of \mathcal{H} as follows.

Let $\mathcal{H} = \{(V_i, \mathbf{n}_i)\}, i = 1, \dots, m$. We define an objective function $f : \mathbb{R}^{10} \rightarrow \mathbb{R}$ as

$$f(a_{11}, \dots, a_{44}) = \sum_{i=1}^m (\omega_{V_i} q(V_i)^2 + \omega_{\mathbf{n}_i} \|\mathbf{n}_i - \nabla q(V_i)\|^2), \quad (11)$$

where $\omega_{V_i}, \omega_{\mathbf{n}_i} > 0$ are fixed weights assigned to the vertices V_i and normals \mathbf{n}_i , respectively. The arguments of the function f are the coefficients of the unknown quadric q and the function accumulates the weighted squared algebraic distances between the vertices and the surface $\mathcal{Z}(q)$ and deviations between gradients of the quadric and input normal vectors of \mathcal{H} .

The coefficients of the fitted quadric are obtained by finding such a_{11}, \dots, a_{44} that the function f attains a minimal value. Since f is a quadratic function, its global minimum is located at a critical point of f . Assuming the Hessian of f is positive definite, the coefficients are obtained by solving the system of linear equations

$$\frac{\partial f}{\partial a_{ij}}(a_{11}, \dots, a_{44}) = 0, \quad 1 \leq i \leq j \leq 4. \quad (12)$$

If we used unit weights for both vertices and normals, the quadric would fit all the point-normals of the neighbourhood \mathcal{H} with the same importance. This may result in unexpected visual result, since we naturally expect, that the quadric should fit the vertices in the first place. Also, the quadric is assigned to the processed vertex-normal (V, \mathbf{n}) . Hence, we may modify the quadric in such way, that the point-normals closer to the processed vertex-normal are approximated better, i. e. we use the weights

$$\omega_{V_i} = \frac{1}{e^{\|V-V_i\|^2}}, \quad \omega_{\mathbf{n}_i} = \frac{10^{-4}}{e^{\|\frac{\mathbf{n}}{\|\mathbf{n}\|} - \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|}\|^2}}, \quad i = 1, \dots, m. \quad (13)$$

These weights reflect the Euclidean distance between V_i and V and deviation between the normalized associated normal \mathbf{n}_i and the normalized processed normal \mathbf{n} .

4.2 Estimation of mean and Gaussian curvature

Let $p(V)$ be the orthogonal projection of V onto the quadric $\mathcal{Z}(q)$ given by

$$p(V) = \arg \min_{W \in \mathcal{Z}(q)} \|V - W\|. \quad (14)$$

Obviously, for the point of the quadratic surface $V \in \mathcal{Z}(q)$, we have $p(V) = V$. Geometrically, the point $p(V)$ is the footpoint of the point V onto the tangent plane of $\mathcal{Z}(q)$ at the point $p(V)$. In other words, the vector $V - p(V)$ and the gradient $\nabla q(p(V))$ are linearly dependent, as seen in Figure 1.

Also, let $s(\mathbf{n}, V)$ denote the gradient with corrected length given by

$$s(\mathbf{n}, V) = \|\mathbf{n}\| \frac{\nabla q(p(V))}{\|\nabla q(p(V))\|}. \quad (15)$$

The estimate of the mean curvature \tilde{H} and Gaussian curvature \tilde{K} assigned to the vertex-normal (V, \mathbf{n}) are computed as

$$\tilde{K} = \frac{s(\mathbf{n}, V)^\top \text{adj}(\mathbf{H}_q(p(V))) s(\mathbf{n}, V)}{\|s(\mathbf{n}, V)\|^4} \quad (16)$$

and

$$\tilde{H} = \frac{s(\mathbf{n}, V)^\top \mathbf{H}_q(p(V)) s(\mathbf{n}, V) - \|s(\mathbf{n}, V)\|^2 \text{tr}(\mathbf{H}_q(p(V)))}{2\|s(\mathbf{n}, V)\|^3}. \quad (17)$$

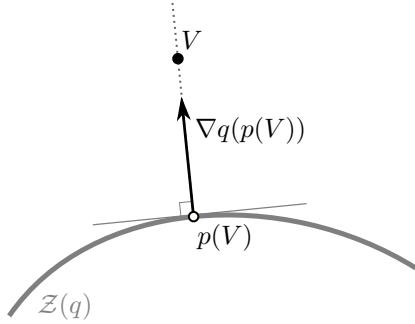


Figure 1: Projection of the point V onto the quadratic surface $\mathcal{Z}(q)$.

Note, that we intentionally compute the curvatures at the point $p(V)$ instead of the point V . The reason is, that the point $p(V)$ is the point of the quadric $\mathcal{Z}(q)$, hence the respective gradient is not distorted by the displacement of the point V . The possible error caused by the computation of the gradient at the point V would be reflected also in its norm. The displacement of V does not affect the Hessian, since q is given by quadratic polynomial. However, the error would be magnified in the computation of the mean and Gaussian curvatures, since the third and fourth power of the gradient norm is present in the denominators of the respective formulas. For the same reason, we rescale the original gradient $\nabla q(p(V))$ by the length of the vector \mathbf{n} in the estimations.

All remaining notions related to curvature – principal curvatures, curvedness and shape index – may be computed using the respective formulas (6), (9) and (10).

5 Experimental results

In this part, we explore the suitability of AQFC for estimating curvatures on polygon meshes. Firstly, we describe the dataset, on which the experiments are made. For each mesh of the dataset we emphasize the features which may affect the precision of the curvature estimations. Then, we describe the measures used for the comparison. Finally, we provide the illustrations and measured values for the experiments and discuss the pros and cons of our method with respect to the selected competitors.

5.1 Meshes for testing

We compare our method on several categories of polygonal meshes, whose description is provided below. The selected meshes are representative for the purposes of the error measurement, described in Subsection 5.2.

5.1.1 Regularly sampled surfaces

The regular sampling of a surface is obtained from its parametric equations given by the function $\mathbf{s} : \mathcal{D} \rightarrow \mathbb{R}^3$, where $\mathcal{D} \subset \mathbb{R}^2$.

For testing, we consider the torus $\mathbf{s}_{\mathbb{T}^2}$ with the minor radius equal to 1 and the major radius being 3. Its parametrization is given by

$$\mathbf{s}_{\mathbb{T}^2}(\theta, \varphi) = \begin{pmatrix} (3 + \cos \theta) \cos \varphi \\ (3 + \cos \theta) \sin \varphi \\ \sin \theta \end{pmatrix}, \quad (\theta, \varphi) \in [0, 2\pi) \times [0, 2\pi). \quad (18)$$

The mean and Gaussian curvature is computed directly from the parametrization as

$$H_{\mathbb{T}^2}(\theta, \varphi) = -\frac{3 + 2 \cos \theta}{2(3 + \cos \theta)}, \quad (19)$$

and

$$K_{\mathbb{T}^2}(\theta, \varphi) = \frac{\cos \theta}{3 + \cos \theta}, \quad (20)$$

respectively. By computing extrema of functions $H_{\mathbb{T}^2}(\theta, \varphi)$ and $K_{\mathbb{T}^2}(\theta, \varphi)$, we get the bounds of the mean and Gaussian curvatures, which are

$$\begin{aligned} -0.65 &\leq H_{\mathbb{T}^2}(\theta, \varphi) \leq -0.25 \\ -0.5 &\leq K_{\mathbb{T}^2}(\theta, \varphi) \leq 0.25. \end{aligned} \quad (21)$$

By uniform sampling of the domain for parameters θ and φ , we constructed the quad meshes with 400, 3,600, 10,000, 360,000 and 1,000,000 vertices. Our aim is to inspect the behaviour of curvature estimation for increasing level of detail of the respective meshes.

5.1.2 Surfaces with irregular sampling

Having a parametrization $\mathbf{s}(\theta, \varphi)$ of the surface, we randomly choose the fixed number m of distinct pairs $(\theta_i, \varphi_i) \in \mathcal{D}$, $i = 1, \dots, m$. The samples are then obtained by plugging these random values of the parameters into the parametrization surface, i. e. the vertices $V_i \in \mathcal{V}$ of the mesh are given by points $\mathbf{s}(\theta_i, \varphi_i)$ for $i = 1, \dots, m$.

The faces of the mesh are given by the triangulation of the set of vertices \mathcal{V} obtained from Ball pivoting algorithm [BMR⁺99]. The input parameters for pivoting were manually adjusted to obtain the mesh with as few holes as possible. The remaining holes were enclosed, so the resulting mesh has no boundary. For testing, we use the irregular sampling of the torus $\mathbf{s}_{\mathbb{T}^2}(\theta, \varphi)$ (18) with 10,000 vertices.

The mesh obtained by this procedure is irregular in terms of edge lengths, the ratio of obtuse and acute triangles, their areas and valence of the vertices. Our aim is to test the robustness of the curvature estimation with respect to such irregularities.

Another instance of the irregular sampling is the one obtained from a unit sphere as follows. In the first step, we constructed the regular sampling of the sphere from its parametrization, similarly to the case of torus. Then, we split each quad face into two triangles by choosing a random diagonal of that quadrilateral. The multiple vertices at the poles of the sphere were merged to avoid degenerate faces. The resulting sphere has 482 vertices. The main feature of the unit sphere is, that its mean and Gaussian curvatures are equal and constant, that is

$$H_{\mathbb{S}^2} = K_{\mathbb{S}^2} = 1. \quad (22)$$

This sampling of the unit sphere is not as irregular as the sampling of the torus. Our goal is to check if even slight changes in the regularity of the sampling affect the precision of the estimation of the curvature.

5.1.3 Meshes obtained from scanning

We provide comparison also on the meshes – Retheur and Damme Assise – which are obtained by the surface reconstruction of the point cloud from the 3D scanning. These models are publicly available in the test dataset used in [YWH⁺19]. The meshes have approximately 250,000 vertices.

5.2 Tested methods and error measurement

For comparison, we selected the curvature estimation by two commonly used state-of-the-art methods. The first one is DDGO [MDSB03], which estimates curvature directly from the geometry of the input mesh. The second competitor is APSS [GG07], which is a method based on describing the underlying surface, similarly as our method AQFC. We used the Python library *PyMeshLab* [MC21] for the computation of curvatures using DDGO and APSS.

For fairness, we consider all tested meshes without prescribed normals at vertices, even if they may be obtained from the parametrization or implicit equation of the sampled surface. For computation of vertex normals necessary for AQFC, we used the simple averaging of adjacent face normals.

For each method, we inspect separately on Gaussian curvature and mean curvature. Consider an input mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$. For any vertex $V \in \mathcal{V}$, let us denote by \tilde{K} the estimation of Gaussian curvature obtained by either AQFC, DDGO or APSS. By K we denote the Gaussian curvature of the underlying

surface. Then, we propose the following three measures $K_{avg}, K_{min}, K_{max} \in \mathbb{R}$ given by

$$\begin{aligned} K_{avg} &= \frac{1}{|\mathcal{V}|} \sum_{V \in \mathcal{V}} |\tilde{K} - K|, \\ K_{min} &= \min_{V \in \mathcal{V}} \tilde{K}, \\ K_{max} &= \max_{V \in \mathcal{V}} \tilde{K}. \end{aligned} \tag{23}$$

The meshes in the dataset are obtained by sampling of a surface, where the curvature may be computed for each sample (vertex of the mesh). Hence, we may measure the absolute value of the difference between the estimated and true curvature at any mesh vertex. The value K_{avg} is a mean value of such differences and it provides us with the information how precisely each method approximates the curvatures of the underlying surface. Also, we inspect on the minimal and maximal values K_{min} and K_{max} to see, how well are the bounds of the curvature preserved. The difference between the bounds of estimated and true curvatures also shows the robustness of the estimates to outliers, caused by the features of the sampled mesh.

Analogously, we define the measures $H_{avg}, H_{min}, H_{max} \in \mathbb{R}$ for mean curvature. These are dependent on the estimated values \tilde{H} and the true values H of mean curvature for any $V \in \mathcal{V}$. Hence,

$$\begin{aligned} H_{avg} &= \frac{1}{|\mathcal{V}|} \sum_{V \in \mathcal{V}} |\tilde{H} - H|, \\ H_{min} &= \min_{V \in \mathcal{V}} \tilde{H}, \\ H_{max} &= \max_{V \in \mathcal{V}} \tilde{H}. \end{aligned} \tag{24}$$

If the computed values of the proposed error measures for Gaussian and mean curvature are sufficiently small for certain method, we may assume that the method correctly estimates the principal curvatures, curvedness and shape index as well. Hence, the latter are not subject of our interest for the included experiments on regular and irregular sampling of the surfaces.

In the case of the meshes from 3D scans, we inspect on curvedness rather than Gaussian and mean curvature, since their bounds are unknown. We visualize the distribution of the curvedness across the tested meshes. We are interested in the presence of noise in the distribution. In this case, the noise is perceived as the difference in curvedness within small homogeneous region of the mesh. For instance, in the areas with small curvedness (almost planar areas), we expect no vertices with high curvedness.

5.3 Comparison with other methods and discussion

In this subsection, we provide the measurements of curvature on the meshes from Subsection 5.1, using the proposed method AQFC and methods DDGO and APSS. The figures of the meshes and their colourings were created using *MeshLab* [CCC⁺08].

5.3.1 Estimation precision with respect to level of detail

For all methods, we measured minimal, maximal and average mean and Gaussian curvatures (see (23) and (24)) of the regularly sampled torus at various level of density, as described in Subsection 5.1. The values of the measurements are provided in Figure 2(i) and 2(ii), respectively.

As seen in these charts, the best average deviation values of both mean and Gaussian curvature were obtained by DDGO for the sampling of 400 vertices. The average deviations H_{avg} and K_{avg} produced by the proposed method AQFC for 3,600 and 10,000 were comparable to the ones given by DDGO. Nevertheless, we consider these values to be satisfactory since the approximation of lower bounds H_{min} , K_{min} and upper bounds H_{max} , K_{max} were similar to DDGO and both of them approximate the true bounds of that torus (21) fair well. The method APSS was not able to approximate upper bound of mean curvature H_{max} and the lower bound of Gaussian curvature K_{min} sufficiently. For illustration, we depict the distribution of curvatures on the torus with 10,000 vertices in Figure 4(i). The lower and upper bounds are cropped to the values of true bounds. Visually, all three methods depict curvatures correctly,

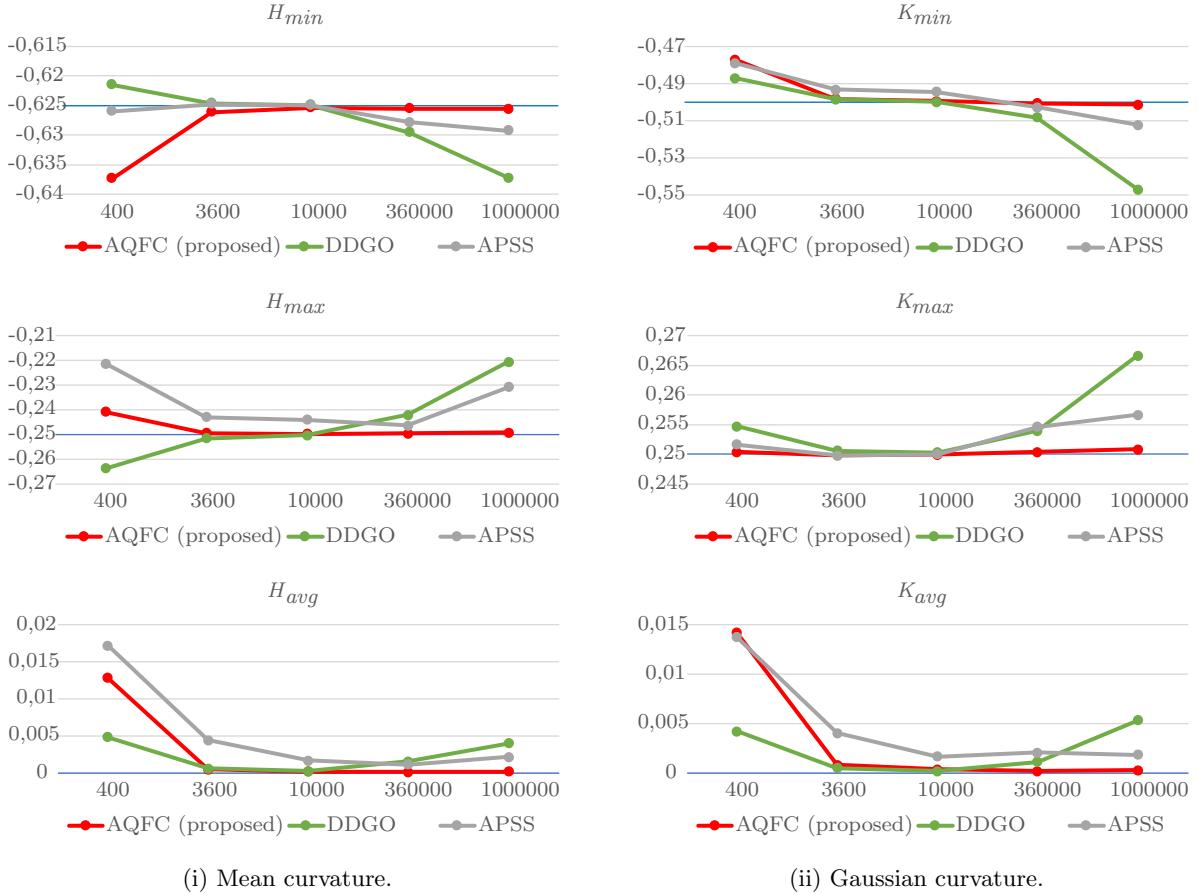


Figure 2: The comparison of different methods of the minimal, maximal and average values for estimation of mean and Gaussian curvature on the regularly sampled torus at various levels of detail. The values on the horizontal axis denote the number of vertices of the sampled meshes.

except for the mean curvature from APSS (row (c) left), where the graduation from the areas with low mean curvature to the areas with higher is sharper.

If the torus is finely sampled, that is $|\mathcal{V}| = 360,000, 1,000,000$, the method AQFC outperformed both DDGO and APSS. The average deviations are significantly smaller for both mean and Gaussian curvature. Also the upper and lower bounds were best preserved by our proposed method.

5.3.2 Influence of sampling regularity on estimation

For the irregular sampling of torus and sphere, we calculate the same measures as in the case of regular sampling. The measured values for both meshes are present in Table 1.

As indicated by average deviations in this table, the method AQFC is remarkably better in handling of irregular sampling. The approximation of the bounds is worse than in the case of regular sampling of the torus. However, the irregularities cause major issues for estimation of the bounds for both DDGO and APSS. The distributions of curvatures across the torus are shown in Figure 4(ii). Note, that the cropping of the color scale to the bounds of true values affects the colouring of the meshes. Still, we are able to notice for how many vertices the curvatures were estimated fair well. As seen in the row (b) of Figure 4(ii), the method AQFC produces only minor disruptions in the vertex curvatures compared to the estimation on the regular sampling (see Figure 4(i) (b)).

Now, we discuss the case of the sphere, whose irregularities are caused only by flipping some of the diagonals of the regularly sampled quadrilaterals. The depiction of curvatures for the sphere are provided in Figure 3. The color scale is adjusted, so it covers the majority of deviations produced by the tested methods. As seen in the row (d), the problems with estimation by APSS were mainly at the poles of the

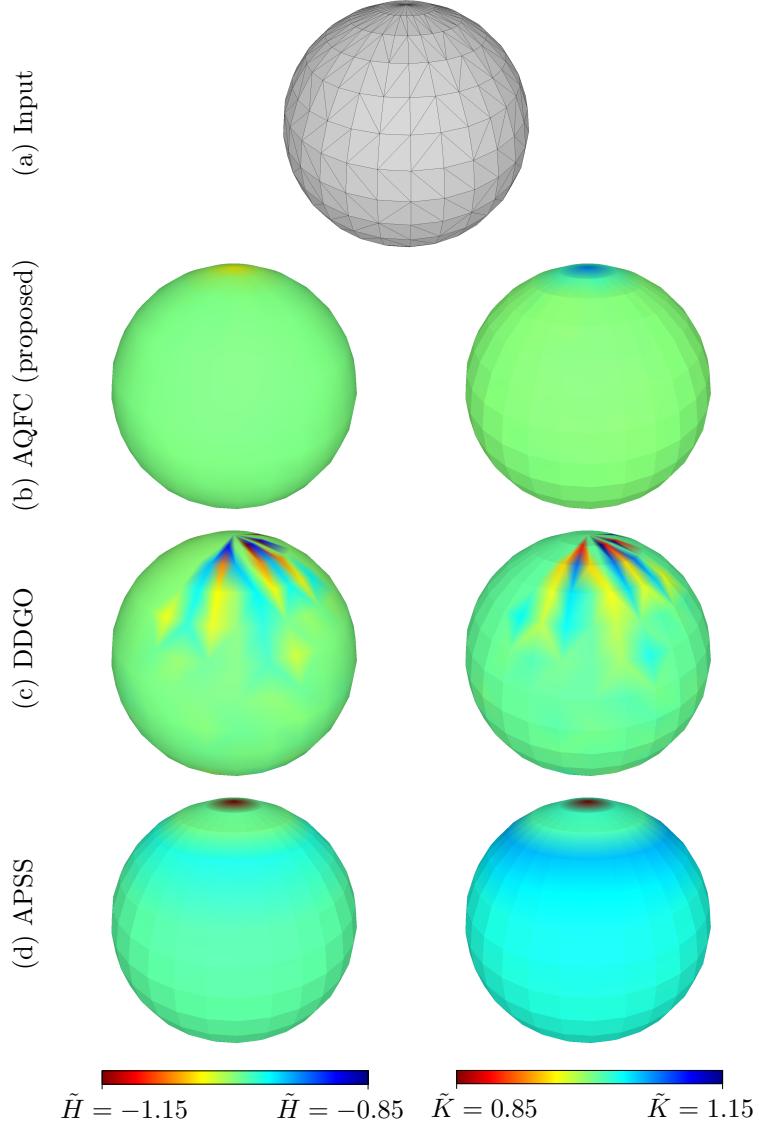


Figure 3: The estimation of mean (left column) and Gaussian (right column) curvatures on the irregularly sampled sphere with 482 vertices.

spheres, which were vertices with high valence. The method DDGO (row (c)) produced bad estimates in the areas where the regularity of the sampling was violated by flipped edges and these deviations are magnified nearer to the poles. The estimation by our method (row (b)) is not visibly affected by the slight irregularities in sampling.

5.3.3 Curvedness distribution over meshes

Finally, we compared the tested method on real data meshes obtained from scanning. The comparison of curvedness distribution for Retheus is depicted in Figure 5(i) and for Dame Assise in Figure 5(ii). Since the curvedness generally has no upper bound, we set the colour scale to the interval $[0, 200]$, so all features of the mesh are visible.

In the case of Retheus, the noise in the curvedness distribution is present for both DDGO and APSS (Figure 5(i) (c) and (d)). The most notable disruption is in the areas of the forehead and the left part of the chest, which may be perceived as homogeneous areas in terms of the curvedness. On contrary, the proposed AQFC does not produce any significant noise and the main features, characterized by high curvedness (blue color in the figure), are easily recognizable, e. g. curls on the beard, creases on the toga

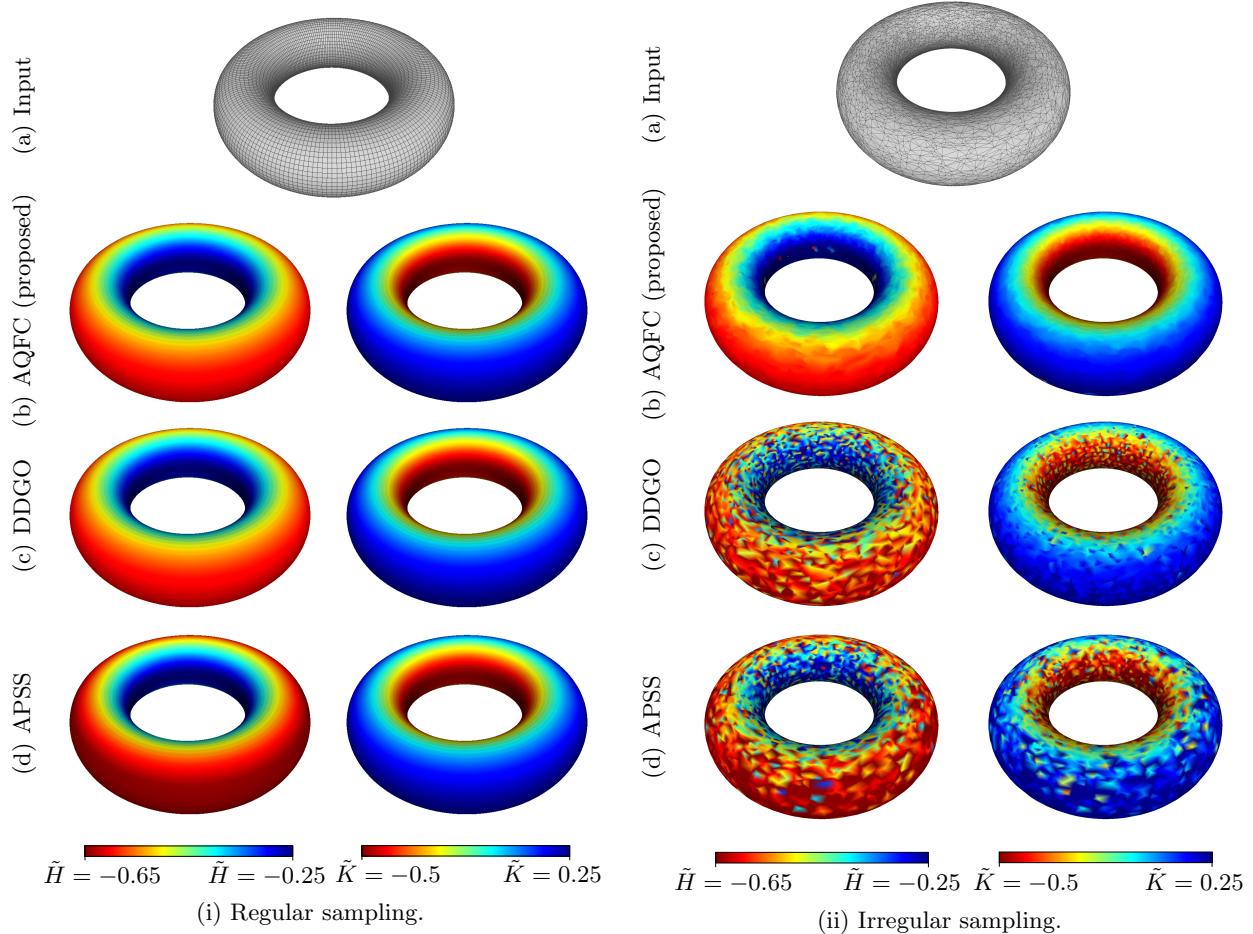


Figure 4: The estimation of mean (left column) and Gaussian (right column) curvatures on the sampled torus with 10,000 vertices with different sampling.

or the sharp edges of the pedestal. The slight deviations of curvatures are present in the smooth areas, e.g. chest, however these are caused by the material of the scanned statue and are considered the feature of the mesh as well.

The effect of imprecise curvedness estimation by DDGO and APSS is even more visible in the case of Dame Assise, see Figure 5(ii) (c) and (d). The features like creases on the dress or parts of the face are hardly distinguishable. However, AQFC captures even the fine details as waviness of the bottom part of the dress (Figure 5(ii) (b)).

6 Conclusion

In our work, we introduced mean and Gaussian curvature estimation method AQFC, which is based on local algebraic fitting of quadratic surfaces. AQFC is capable to process arbitrary polygon meshes, whose vertices are equipped with associated normal vectors.

The experimental results indicate, that AQFC is feasible for estimation of the mean and Gaussian curvature and their derivatives. We consider our method to be more universal albeit the well established method DDGO provides more precise results for the regular sampling of the surfaces. However, DDGO fails to provide satisfactory results for meshes with irregular sampling. Even if APSS is less prone to irregularities compared to DDFO, AQFC provides more precise estimate of curvatures.

The proposed method AQFC is also robust at the computation of curvedness for meshes, where the prescription of the underlying surface is unknown. Thus, it has potential to be used in various application as denoising or smoothing of the meshes.

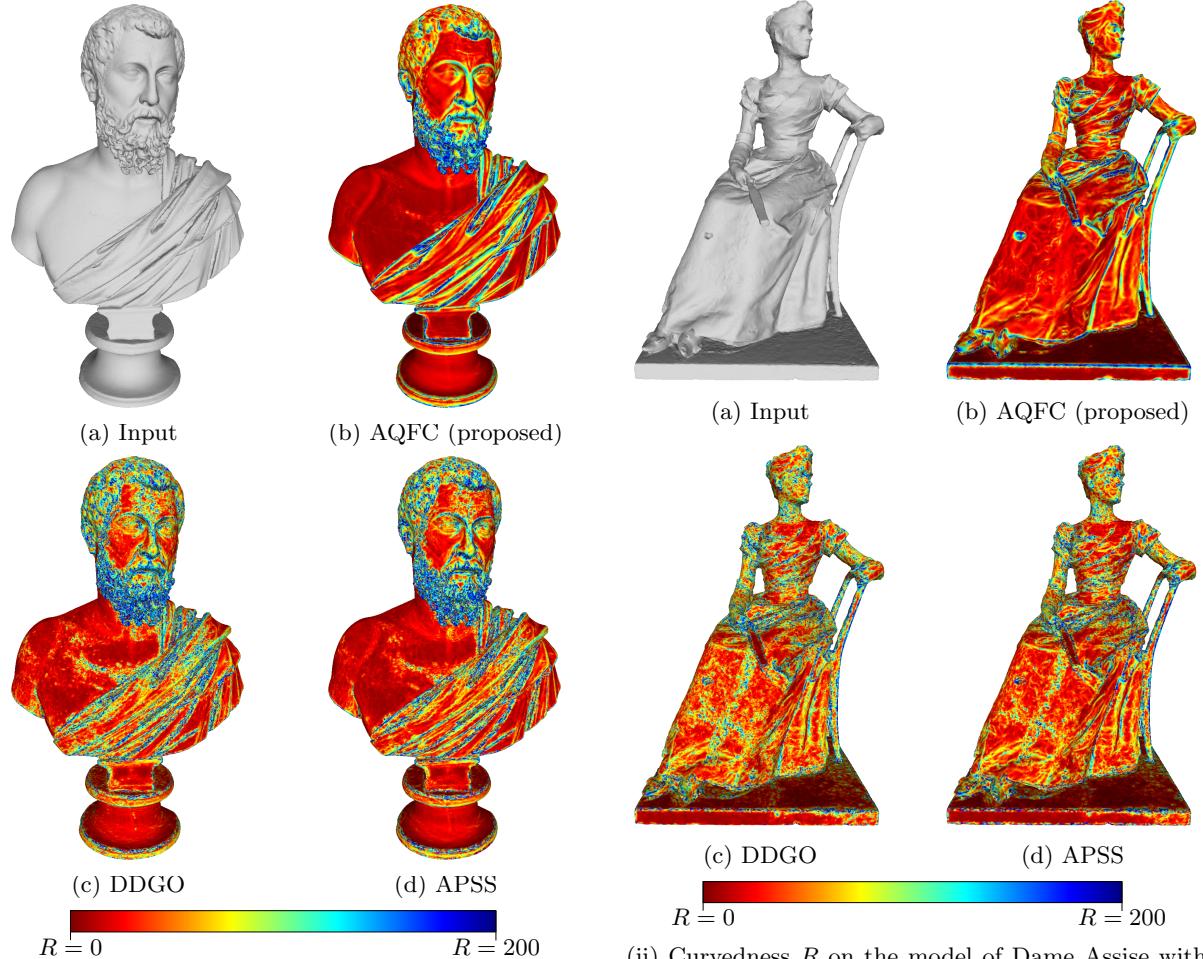
	Torus		
	AQFC (proposed)	DDGO	APSS
H_{min}	-0.627818	-30.7685	-0.57315
H_{max}	-0.244277	21.7019	0.0500141
H_{avg}	0.00651111	0.0874593	0.220647
K_{min}	-0.503117	-318.85	-1.29606
K_{max}	0.251109	3232.91	0.925429
K_{avg}	0.00687775	0.909176	0.0962692
	Sphere		
	AQFC (proposed)	DDGO	APSS
H_{min}	-1.03639	-1.14966	-1.07738
H_{max}	-0.997533	-0.890629	0.0058888
H_{avg}	0.00191217	0.0113342	0.0214852
K_{min}	0.995068	0.894789	-0.00106665
K_{max}	1.0741	1.15526	1.16044
K_{avg}	0.00383101	0.0163219	0.0405045

Table 1: The comparison of different methods of the minimal, maximal and average values for estimation of mean and Gaussian curvature on the irregularly sampled torus with 10,000 vertices and sphere with 482 vertices. The lowest average of deviations are highlighted by bold. Note, that the best approximation of minimal and maximal values of curvatures is achieved by using AQFC.

In future work we plan on to extend this approach for processing of point clouds. The modifications that need to be made is the way of obtaining neighbourhood since edges and faces are not present. This may be solved by using the feasible space partitioning structure as octrees or k -d trees.

References

- [BB20] Mohamed Yaghmorasan Benzian and Nacéra Benamrane. 3d mesh segmentation by region growing based on discrete curvature. In *2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, pages 271–276. IEEE, 2020.
- [BMR⁺99] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
- [BPW10] Alexander I Bobenko, Helmut Pottmann, and Johannes Wallner. A curvature theory for discrete surfaces based on mesh parallelity. *Mathematische annalen*, 348:1–24, 2010.
- [CCC⁺08] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.



(i) Curvedness R on the model of Rethour with 250,562 vertices (a). Curvature estimation by the proposed method AQFC (b) generates less noise than the methods DDGO (c) and APSS (d).

(ii) Curvedness R on the model of Dame Assise with 257,818 vertices (a). Curvature estimation by the proposed method AQFC (b) highlights the features (wrinkles, sharp edges) without significant disruption compared to DDGO (c) and APSS (d).

Figure 5: Curvedness distribution over meshes.

- [DC16] Manfredo P. Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.
- [DGBD20] Fernando De Goes, Andrew Butts, and Mathieu Desbrun. Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics (TOG)*, 39(4):110–1, 2020.
- [DM14] Julie Digne and Jean-Michel Morel. Numerical analysis of differential operators on raw point clouds. *Numerische Mathematik*, 127:255–289, 2014.
- [GG04] Tim Gatzke and Cindy Grimm. Improved curvature estimation on triangular meshes. 2004.
- [GG06] Timothy D Gatzke and Cindy M Grimm. Estimating curvature on triangular meshes. *International journal of shape modeling*, 12(01):1–28, 2006.
- [GG07] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM Transactions on Graphics*, pages 23–es. 2007.
- [Gol05] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632–658, 2005.

- [Ham93] Bernd Hamann. Curvature approximation for triangulated surfaces. In *Geometric Modelling*, pages 139–153. Springer, 1993.
- [HPW05] Klaus Hildebrandt, Konrad Polthier, and Max Wardetzky. Smooth feature lines on surface meshes. In *Symposium on geometry processing*, pages 85–90, 2005.
- [JLW05] Shuangshuang Jin, Robert R. Lewis, and David West. A comparison of algorithms for vertex normal computation. *The Visual Computer*, 21(1):71–82, 2005.
- [KNSS09] Evangelos Kalogerakis, Derek Nowrouzezahrai, Patricio Simari, and Karan Singh. Extracting lines of curvature from noisy point clouds. *Computer-Aided Design*, 41(4):282–292, 2009.
- [Koe90] Jan J. Koenderink. *Solid shape*, volume 2. MIT press Cambridge, MA, 1990.
- [KSNS07] Evangelos Kalogerakis, Patricio Simari, Derek Nowrouzezahrai, and Karan Singh. Robust statistical estimation of curvature on discretized surfaces. In *Symposium on Geometry Processing*, volume 13, pages 110–114, 2007.
- [KWH18] Markus Kronenberger, Oliver Wirjadi, and Hans Hagen. Empirical comparison of curvature estimators on volume images and triangle meshes. *IEEE transactions on visualization and computer graphics*, 25(10):3032–3041, 2018.
- [LKK15] Jung Lee, Seokhun Kim, and Sun-Jeong Kim. Mesh segmentation based on curvatures using the gpu. *Multimedia Tools and Applications*, 74:3401–3412, 2015.
- [MC21] Alessandro Muntoni and Paolo Cignoni. PyMeshLab, January 2021.
- [MDSB03] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, pages 35–57. Springer, 2003.
- [ÖGG09] Cengiz Öztireli, Gaël Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, volume 28, pages 493–501. Wiley Online Library, 2009.
- [PLW⁺07] Helmut Pottmann, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang. Geometry of multi-layer freeform structures for architecture. In *ACM SIGGRAPH 2007 papers*, pages 65–es. 2007.
- [RB05] Anshuman Razdan and MyungSoo Bae. Curvature estimation scheme for triangle meshes using biquadratic bézier patches. *Computer-Aided Design*, 37(14):1481–1491, 2005.
- [SLC⁺19] Kehua Su, Na Lei, Wei Chen, Li Cui, Hang Si, Shikui Chen, and Xianfeng Gu. Curvature adaptive surface remeshing by sampling normal cycle. *Computer-Aided Design*, 111:1–12, 2019.
- [Spi99] Michael Spivak. *A comprehensive introduction to differential geometry*, volume 2. Publish or Perish, Incorporated, 1999.
- [SSH04] Irina Semenova, Vladimir Savchenko, and Ichiro Hagiwara. Curvature based mesh improvement. *WSEAS Transactions on Systems*, 3(4):1346–1354, 2004.
- [Tau94] Gabriel Taubin. Distance approximations for rasterizing implicit curves. *ACM Transactions on Graphics (TOG)*, 13(1):3–42, 1994.
- [Tau95] Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 902–907. IEEE, 1995.
- [VVP⁺16] Libor Váša, Petr Vaněček, Martin Prantl, Vera Skorkovská, Petr Martínek, and Ivana Kolingerová. Mesh statistics for robust curvature estimation. In *Computer Graphics Forum*, volume 35, pages 271–280. Wiley Online Library, 2016.

- [YHXL15] Li Yao, Shihui Huang, Hui Xu, and Peilin Li. Quadratic error metric mesh simplification algorithm based on discrete curvature. *Mathematical Problems in Engineering*, 2015, 2015.
- [YLH⁺06] Yong-Liang Yang, Yu-Kun Lai, Shi-Min Hu, Helmut Pottmann, et al. Robust principal curvatures on multiple scales. In *Symposium on Geometry Processing*, pages 223–226, 2006.
- [YWH⁺19] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3D point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019.
- [ZGYL11] Mao Zhihong, Cao Guo, Ma Yanzhao, and Kunwoo Lee. Curvature estimation for meshes based on vertex normal triangles. *Computer-Aided Design*, 43(12):1561–1566, 2011.