

Robust statistical estimation of curvature on discretized surfaces

Evangelos Kalogerakis, Patricio Simari, Derek Nowrouzezahrai and Karan Singh

Dynamic Graphics Project, Computer Science Department, University of Toronto

Abstract

A robust statistics approach to curvature estimation on discretely sampled surfaces, namely polygon meshes and point clouds, is presented. The method exhibits accuracy, stability and consistency even for noisy, non-uniformly sampled surfaces with irregular configurations. Within an M-estimation framework, the algorithm is able to reject noise and structured outliers by sampling normal variations in an adaptively reweighted neighborhood around each point. The algorithm can be used to reliably derive higher order differential attributes and even correct noisy surface normals while preserving the fine features of the normal and curvature field. The approach is compared with state-of-the-art curvature estimation methods and shown to improve accuracy by up to an order of magnitude across ground truth test surfaces under varying tessellation densities and types as well as increasing degrees of noise. Finally, the benefits of a robust statistical estimation of curvature are illustrated by applying it to the popular applications of mesh segmentation and suggestive contour rendering.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems; curve, surface, solid, and object representations.

1. Introduction

Surface derivatives such as normal and curvature are as important as surface position for the perception and understanding of shape. Aggregates of these attributes are referred to in design as *surface features*. Thus, the computation and processing of these features is crucial in geometric modeling. While this problem is straightforward for analytic surface representations, normal and curvature estimation for discretely sampled surfaces such as polygon meshes is an area of ongoing research [DKT06]. The importance of curvature as a fundamental descriptor for shape analysis and understanding is highlighted by its critical role in numerous applications such as surface segmentation [LPRM02, LDB05], surface alignment [RL01], mesh simplification [HG99, She01, SS05], remeshing [ACSD*03], denoising [MDSB02, LP05], 3D puzzle assembly [HFG*06], symmetry detection [MGP06], partial shape matching [GG06], non-photorealistic rendering [Rus04], feature line extraction [HPW05] and image synthesis [RMB07], to name a few.

Motivation: As pointed out by Meyer *et al.* [MDSB02], there is no consensus on the most appropriate approach to normal and curvature estimation on a discretely sampled sur-

face, largely because the discrete sampling could be taken from any of a continuum of (piecewise) smooth surfaces. The problem is further exacerbated in practice by intended surface derivative discontinuities (*i.e.* edges) as well as signal noise that is inherent to real surface scans. The actual mesh connectivity in many meshes is often just an artifact of the scanning methodology or the mesh reconstruction algorithms for CAD or implicit surface models (*marching cubes*, see figure 3). Interestingly, irregularities and noise in sampling are amplified in surface derivative calculations while many surface denoising and reconstruction methods as well as normal estimation algorithms rely upon their estimation [CDR00, MDSB02, MNG04, DTB06]. Surface smoothing and restructuring are also undesirable in some geometric processing pipelines such as those pertaining to precision engineering and analysis.

A critical issue at the heart of the problem is determining the appropriate shape and size of the neighborhood around a point for the curvature operator. Most commonly used one-ring neighborhood operators produce accurate results for well conditioned meshes but are not robust to noise and mesh irregularities (see figure 2). Operators with a larger region of support are more stable but tend to have reduced accuracy,

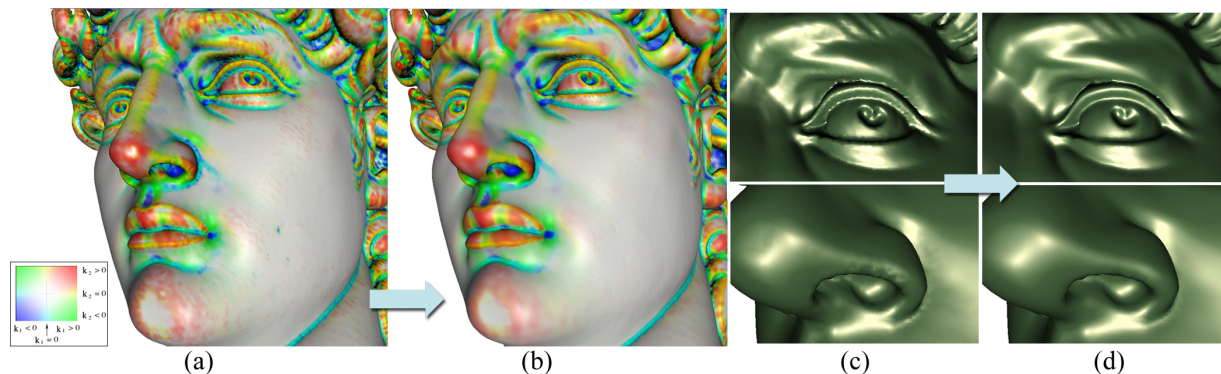


Figure 1: *a) Initial curvature estimate; b) Curvature values estimated at the final iteration; c) example renderings of the David's left eye and part of the nose based on the initial normals; d) same renderings based on the normals corrected by our algorithm. Note that our method respects feature edges.*

often propagating curvature extrema to irrelevant points past feature boundaries and may over-smooth the curvature field. Similar problems occur if a-posteriori smoothing of the field is performed [GI04]. Alternatively, a-priori smoothing of the mesh is a common approach to dealing with noise. However, indiscriminate smoothing depreciates surface detail if not applied judiciously [GG06].

The motivation of this paper is thus to develop robust normal and curvature estimation that is automatic and operates on any discretely sampled surface in the presence of surface derivative discontinuities, sample noise, non-uniform sampling and irregularities.

Approach and contributions: The input to our approach is a discretized surface including normal estimates. The outputs are principal curvature directions and values for the point samples, as well as re-estimated surface normals and derivatives of curvature, if desired. We address the problem of adapting the shape and size of the point neighborhood for our curvature operator using an M-estimation approach (see section 2). The algorithm iteratively refines the shape and size of this neighborhood by weighting samples appropriately based on fitting error. It automatically adapts to small neighborhoods for well conditioned surfaces, and bigger, possibly anisotropic neighborhoods in the presence of noise, irregularities and feature boundaries (see figures 6 and 7.) In section 3, we show how our results can be used to correct noisy surface normals (see figures 1 and 9). We have exhaustively compared our approach with three representative curvature estimators [CSM03, GI04, Rus04]. Section 4 illustrates how our approach compares favorably over a wide range of models from a variety of sources, with varying degrees of noise and other irregularities (see figures 2, 3, 10 and 11.) We also illustrate how our curvature estimates can improve the performance of existing mesh segmentation [LDB05] and suggestive contour rendering [DFRS03] applications (see figures 12 and 13.)

A considerable number of papers exist in the computer

graphics, vision and engineering literature concerning differential operators on discrete surfaces, especially polygon meshes (see [GG06] for a recent survey). To the best of our knowledge our method is the only one to yield a maximum likelihood estimate of the curvature tensor at a surface point according to the locally observed normal variation. This estimate accounts for varying noise and non-uniform sampling (in both polygon meshes and point clouds) and irregular tessellations (in meshes) without any preprocessing of the input. Through iterative reweighting, the optimal size and shape of the neighborhood is adaptively determined for each point separately, resulting in a highly accurate approximation of the curvature tensor (see figures 10 and 11). In the following, we will briefly present a categorization of the existing methods.

1.1. Related work

Local curve and surface fitting approaches: Hamann [Ham93] fits quadratic functions locally to the surface based on the projection of one-ring neighborhood points onto the tangent plane of the point of interest. The curvature tensor is then analytically determined from the derived surface. Quadratic patch fitting is also used in computer vision approaches, usually on $N \times N$ pixel windows in range images [JF89]. Stokely and Wu [SW92] and Gatzke and Grimm [GG06] suggest building a natural local parameterization of the surface to fit the patch. Chen and Schmitt [CS92] locally fit circular curves, given the center point and two of its neighbors. Hameiri and Shimsoni [HS02] instead suggest fitting quadratic curves on the two immediate neighbor points of the center point or all the neighbor points inside a given radius across the normal sections to improve stability. Cazals and Pouget [CP03] show that in the case of a general smooth surface, the fitting of an osculating n -jet results in convergence of the estimated curvatures to the true ones as the degree of the jet and the sampling density increase. Goldfeather and Interrante [GI04] expanded the quadratic patch fitting method by using cubic fits from the points and their

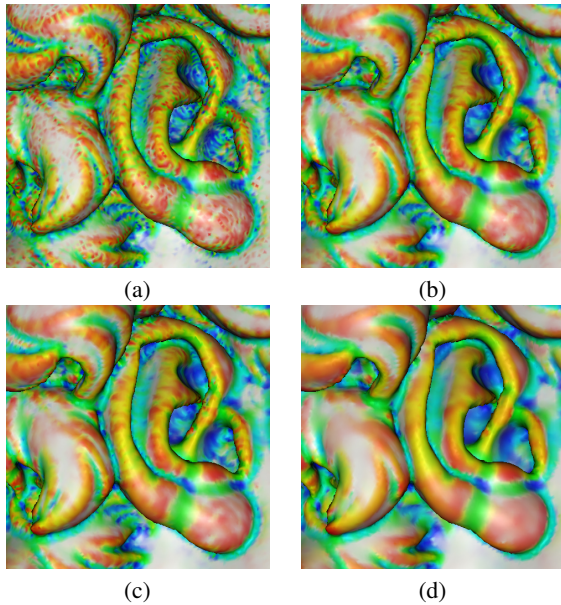


Figure 2: Comparative illustration of the curvature values produced by different methods on the original high-resolution scan of the David's right ear: (a) normal cycle method on a one-ring neighborhood; (b) cubic patch fitting to points and normals in a two-ring neighborhood as suggested by Goldfeather and Interrante; (c) per-triangle computation and per-vertex averaging using Rusinkiewicz's approach; (d) our results. Note how our method produces smooth results while respecting feature edges.

normals in a one-ring or two-ring neighborhood, avoiding the degenerate cases of quadratic fitting and improving accuracy.

All existing patch fitting methods use a notion of arbitrarily fixed locality in their approximation such as a one-ring, two-ring or N -ring neighborhood of a vertex in a mesh or the k -nearest neighbors to each point of interest in a point cloud. In the presence of noise in vertex positions and associated normals, non-uniform tessellations, or feature boundaries that represent inherent discontinuities in the curvature field it is not clear how this neighborhood should be chosen to ensure stability. In such cases higher-order local surface approximations can easily result in overfitting. The accuracy in the curvature estimation always depends on the parametric model that is being fit. Lipman *et al.* derive an interesting parametrization-free projection operator for robust MLS smooth surface approximation, but deriving the size of its influence radius from the surface data itself is still an open question [LCOLTE07].

Discrete vertex-ring methods: Taubin [Tau95] introduced a 3D curvature tensor having the principal curvature directions as two of the three eigenvectors and the principal curvatures as linear combinations of two of the three eigenvalues.

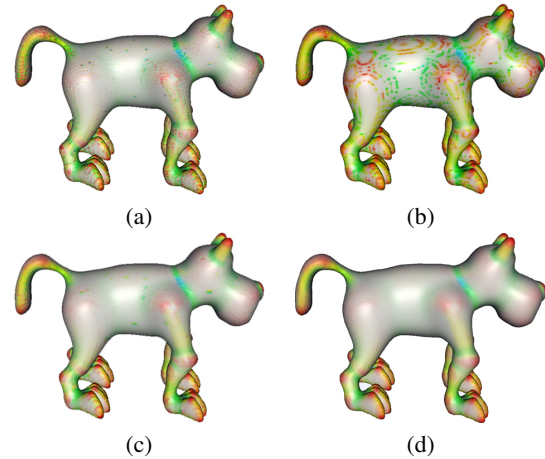


Figure 3: Comparative illustration of the curvature values produced by different methods on a smooth implicit surface model tessellated using marching cubes: (a) normal cycle method on a one-ring neighborhood; (b) cubic patch fitting to points and normals in a two-ring neighborhood as suggested by Goldfeather and Interrante; (c) per-triangle computation and per-vertex averaging using Rusinkiewicz's approach; (d) our results.

Langer *et al.* [LBS07] derived novel weights to obtain exact quadratures of Taubin's integral representation of the curvature tensor in the case of non-noisy polygon meshes, but results are still not competitive with those of the cubic patch fitting method. Hameiri and Shimsoni [HS02] and Page *et al.* [PSK*02] propose modifications of Taubin's method by including more neighbor points inside a geodesic neighborhood of user-defined size and suggest a weighting based on the inverse of their geodesic distance from the center point. Stokely and Wu [SW92] estimate the Gaussian curvature based on the Gauss-Bonnet theorem, considering a closed path around each point of interest and using the whole area enclosed in this path for their approximation. Meyer *et al.* [MDSB02] suggest the approximation of discrete mean and Gaussian curvature based on the Euler-Lagrange equation and Gauss-Bonnet theorem respectively, using the notion of a mixed area around each vertex of interest to address varying face proportions. Cohen-Steiner and Morvan [CSM03] also give discrete definitions for the mean and Gaussian curvature and the curvature tensor based on the normal cycle theory, and prove that the estimated curvature tensors converge to the true ones of the smooth surface under specific sampling conditions.

Despite their elegance, simplicity and relatively low computational cost, the above methods are sensitive to irregular tessellations and noise. In order to account for noise, it is suggested that one take into account a larger region for the operator, usually a geodesic disc [ACSD*03]. Liu *et al.* [LPW*06] and Yang *et al.* [YLHP06] perform PCA also inside spherical kernels of fixed radius centered at each

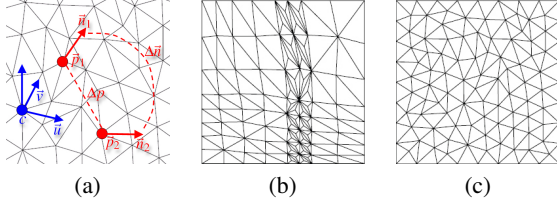


Figure 4: (a) Example of a normal variation sample; (b) example of mixed irregular tessellation, including non-uniform sampling, polar distortion and bad aspect ratio triangles [GGRZ06]; (c) randomly irregular tessellation

vertex. However, it is unclear how to choose the neighborhood to compensate for noise, non-uniform sampling and irregular tessellation. Moreover, large neighborhoods arbitrarily defined by users may propagate curvature extrema from feature edges and destroy fine details in the curvature field. Tong and Tang [TT05] proposed a tensor voting technique to robustly estimate curvature tensors, which still relies on fixed preprocessing passes to reject outliers that are decoupled from curvature estimation. Maximum likelihood estimation of curvature tensors is not guaranteed. The optimal radius of their operator is globally chosen for the input surface by exhaustively searching an input range of values, preferring the one that results in the highest peaks among all the acquired curvature histograms.

Per triangle curvature tensor estimation: Theisel *et al.* [TRZS04] and Rusinkiewicz [Rus04] suggest estimating the curvature tensor per triangle based on vertex normals. Rusinkiewicz suggests a least-squares fitting of the curvature tensor to the normal variations across the edges of a given triangle. After solving for a tensor at every incident triangle, the tensor at the vertex is obtained by rotating the face tensors so that their coordinate systems align with the coordinate system of the vertex and then taking a weighted average based on the corresponding vertex’s associated area (as defined by Meyer *et al.* [MDSB02]) for each such triangle. In essence a linear model is locally fit to each minimal surface element and then an average model is taken in a weighted fashion. While this tends to produce smooth results because of this local averaging, it does not necessarily correspond to the maximum likelihood model given the observed normal variation in the point’s vicinity and noise and outliers can still affect this estimate.

2. Robust statistical estimation of curvature

Our approach is based on curvature tensor fitting performed on appropriate neighborhoods which adapt to noise, irregularities and non-uniform sampling. Constraining the curvature tensor fitting to consider all normal variation samples inside the chosen neighborhood (as opposed to partially constraining it per surface element and then averaging) yields a maximum likelihood estimate of the curvature tensor under Gaussian noise on surface points and normals. Moreover,

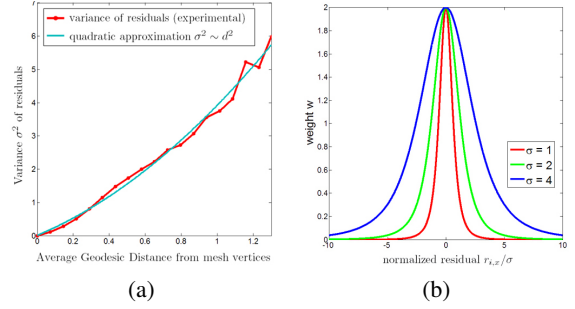


Figure 5: (a) Variance of the residuals of the normal variation samples in a three-ring neighborhood around each vertex versus their average geodesic distance to the neighborhood center on a monkey saddle surface (red) and the approximating quadratic curve (blue); (b) The Geman-McLure weight as a function of the normalized residual.

this formulation allows us to employ M-estimation in order to guarantee a more robust approximation of the tensor by rapidly decreasing the influence of noise and outliers.

M-estimation [HRRS86, FP02] consists of robustly fitting a model by minimizing an objective function of the residuals of the samples with an Iteratively Reweighted Least Squares (IRLS) scheme. It has been used in various computer vision applications (see [Ste99] for a survey) and recently in computer graphics for robust symmetry detection [SKS06].

In the following, we explain our normal variation sampling scheme. We then proceed to formulate the problem of curvature tensor fitting using these samples and describe a weighting scheme that takes into account relevance based on distance to the point of interest as well as the residual from said fitting.

2.1. Curvature tensor fitting

Our goal is to estimate the curvature tensor from the second fundamental form of surfaces. This is expressed in terms of the shape operator based on the covariant derivatives of the given normal vector field \vec{N} within a local neighborhood around each point of interest:

$$\begin{bmatrix} \nabla_{\vec{u}} \vec{N} \cdot \vec{u} & \nabla_{\vec{v}} \vec{N} \cdot \vec{u} \\ \nabla_{\vec{u}} \vec{N} \cdot \vec{v} & \nabla_{\vec{v}} \vec{N} \cdot \vec{v} \end{bmatrix} \quad (1)$$

Given a point of interest c at which the surface has orthogonal tangential directions \vec{u}, \vec{v} , for every pair of points p_1, p_2 with corresponding normals \vec{n}_1, \vec{n}_2 inside a local neighborhood of c , we can constrain the curvature tensor according to the following equation:

$$\underbrace{\begin{bmatrix} \nabla_{\vec{u}} \vec{N} \cdot \vec{u} & \nabla_{\vec{v}} \vec{N} \cdot \vec{u} \\ \nabla_{\vec{u}} \vec{N} \cdot \vec{v} & \nabla_{\vec{v}} \vec{N} \cdot \vec{v} \end{bmatrix}}_{\text{unknowns}} \cdot \begin{bmatrix} \vec{\Delta p} \cdot \vec{u} \\ \vec{\Delta p} \cdot \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{\Delta n} \cdot \vec{u} \\ \vec{\Delta n} \cdot \vec{v} \end{bmatrix} \quad (2)$$

where $\vec{\Delta p} = p_2 - p_1$ and $\vec{\Delta n} = \vec{n}_2 - \vec{n}_1$. This arrangement

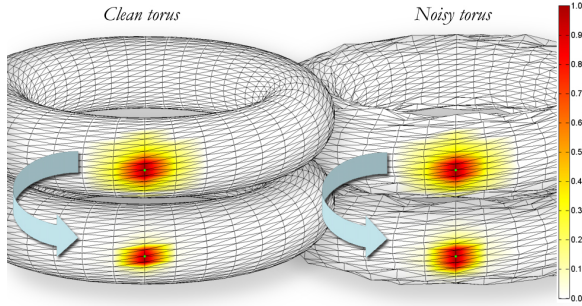


Figure 6: Visualization of estimation weights on a torus mesh. Vertices are colored according to the average weight of all variation pairs that include it. **Top:** weighting on initial iteration; **bottom:** weighting on final iteration; **left:** noiseless mesh; **right:** noisy mesh. Notice how in the noiseless case the method automatically focuses in more tightly on the vertex, while in the noisy case the method adapts and considers a wider area.

is illustrated in figure 4a. A system of such equations needs at least three different point pairs with their normal differences in order to produce an overdetermined solution for the curvature tensor contained in the fitted model. Similarly to Rusinkiewicz [Rus04], we also estimate the $2 \times 2 \times 2$ derivative of curvature tensor by fitting it to the obtained curvature tensor variation samples.

2.2. Normal variation sampling

In order to robustly estimate curvature at each point of a sampled surface we consider normal variation samples obtained from all pairs of surface samples and their associated normals initially within an operating region. This dense sampling, in general, proves to be much more stable and accurate than taking into account only the associated normal variations across tessellation edges in the case of a polygon mesh (see figure 10.) Of course, the price to pay is the increased computational cost as the overdetermined linear system of equations (2) includes many more samples. In regular meshes with near-equidistant vertices, the sampling based on edge pairs alone returns very similar results with a dense all-pairs sampling and can thus be used instead. In our tests, presented in the results section, we show error curves based on both the edge sampling mode of our algorithm and the dense sampling mode (see figure 10.) Finally, for efficiency reasons and to quickly reject obvious outliers, we initially prune any samples whose point normals have a larger than $\pi/2$ angle with the normal of the center point of interest. For polygon meshes we also prune any samples whose points have one of their incident face normals creating a larger than $\pi/2$ angle with the center point normal.

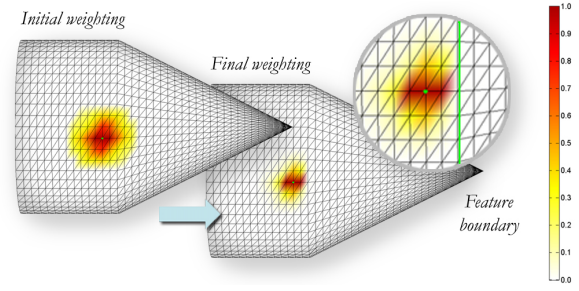


Figure 7: Visualization of estimation weights on a joined cylinder/cone mesh. Vertices are colored according to the average weight of all variation pairs that include it. **Left:** weighting on initial iteration; **right:** weighting on final iteration. Notice how, by the final iteration, the method has discarded the structured outlier pairs crossing the feature edge into the conical region.

2.3. Distance weighting

The IRLS process of the M-estimation iteratively reweights the samples accordingly in order to minimize an objective function of their residual error. However, the M-estimation weights should be combined (multiplied) with prior weights that also capture the relative spatial relevance of the samples. The statistics literature [CR88] suggests weighting samples according to the variances of the residuals grouped by their distance; *i.e.* for samples in a distance class with variance $\text{var}(r_i)$, their weight should be $1/\text{var}(r_i)$ so that the response variances are mapped to a constant value. Unfortunately, these optimal weights are never known. A strategy to estimate them is to consider a function that approximates the variance as a function of sample distance. Figure 5a illustrates that a quadratic function is a good approximation by plotting the ground truth variance of the residuals of the samples by distance in comparison to the average geodesic distance from the center point for a monkey saddle test surface. We found that this behavior holds for all our C^2 continuous test surfaces. In essence, this weighting scheme will correspond to a prior which weights samples according to the inverse of the average squared distance of their associated points from the center point. This coincides with the intuition that variation samples should be less relevant when they are further from the point of interest. Similar weightings have been successfully used [HS02] and this proves to be a good approximation. In polygon meshes, a geodesic distance can be approximated by a shortest path along edges or more accurate approaches [SSK*05], while in point clouds the Euclidean distance between points can be used instead.

2.4. Cost estimation and reweighting

In the M-estimation formulation, the goal is to minimize the sum cost of residuals:

$$\min_x \sum_{s_i \in S} \rho(r_{i,x}/\sigma) \quad (3)$$

where x is the model with the unknown parameters containing the curvature tensor, s_i is the i th variation sample, S is the sample set, $r_{i,x}$ is the residual of sample s_i with respect to model x , σ is a scale factor that is automatically estimated (see below) and ρ is a robust cost function. In simple least squares the cost function is quadratic. However, such a cost function is very sensitive to outliers. For a robust estimation, we use the Geman-McLure cost function (GM), which is a proven choice in computer vision [FP02] and graphics applications [SKS06] as it exhibits very good behavior in quickly rejecting structured outliers (see figure 7.)

The minimization of sum cost is achieved by solving:

$$\sum_{s_i \in S} \Psi(r_{i,x}/\sigma) \frac{dr_{i,x}}{dx} \frac{1}{\sigma} = 0$$

where $\Psi(r_{i,x}/\sigma) = \rho'(r_{i,x}/\sigma)$. The above equation can be solved using an IRLS process with a weight function expressed for each sample as $w(r_{i,x}/\sigma) = \frac{\Psi(r_{i,x}/\sigma)}{r_{i,x}/\sigma}$. The cost function and the weights are given as:

$$\rho(r_{i,x}/\sigma) = \frac{(r_{i,x}/\sigma)^2}{1 + (r_{i,x}/\sigma)^2} \quad w(r_{i,x}/\sigma) = \frac{2}{(1 + (r_{i,x}/\sigma)^2)^2}$$

This weight function is illustrated in figure 5 for varying values of σ . The value of this parameter, called the scale estimate, determines the rate with which cost rises and weight drops with respect to growing residual values. By assuming that the residuals are modeled with a contaminated Gaussian distribution [FP02], which is also reasonable in our case, σ can be automatically estimated as:

$$\sigma = 1.4826 \cdot \text{median} |r_{i,x}|$$

This is derived from the fact that the median absolute value of a large enough sample of unit variance normal-distributed 1D values is $1/1.4826 = 0.6745$. This scale estimation causes the estimator to tolerate almost 50% of outliers [SAG95]. We also do not let σ fall below its first estimate as given by the initial guess tensor in order to ensure stability. In order to control leverage, we set to 0 the weights of the non one-ring neighborhood samples with residuals more than 2σ [FP02].

2.5. Operating region

The operating region contains all the points whose normal differences will be taken into account for the initial solution of the system of equations (2). The operating region bounds the normal variation sampling and is fixed for efficiency reasons. The M-estimation algorithm will reweight the samples accordingly inside this initial region in order to converge to an anisotropic support area during the IRLS process (see figures 6, 7 and 8). This operating region should be large enough to guarantee a stable fitting of the curvature tensor. On the other hand, considering a very large number of samples would slow down the solving of the system. In our case,

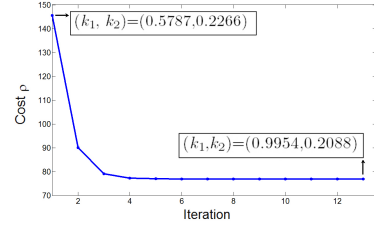


Figure 8: Iteration number versus cost function during the IRLS process on the noisy torus of figure 6 (20% of the median edge length). The M-estimation process minimizes the cost function after 13 iterations. The ground truth curvature values for this vertex on the torus were $(k_1, k_2) = (1.0, 0.2)$. Note that the initial guess was highly inaccurate and yet the final estimated values are very close to ground truth.

we determine the operating region from a geodesic ball, centered on the point of interest with a radius equal to 3 times the average distance to the 6-nearest neighbor points. The factor of 3 was determined to be a good heuristic experimentally. The M-estimation process tends to reject the increasing number of outliers even in larger operating regions.

One of the advantages of our approach is its ability to automatically adapt to local surface features. It is conceivable, however, that this may not be desirable for a specific application [LZH*07]. Should a user indeed have reliable information about the neighborhood size for each point of interest, it is trivial to modify our algorithm by limiting the operating region to this neighborhood and clamping sample weights from below, not allowing them to drop beneath any particular desired value.

2.6. Initial model guess

In order to initialize the M-estimation algorithm, an initial guess of the model, representing the curvature tensor, should be made. A reasonable initial guess is the estimation using only the normal variation samples of the one-ring neighborhood edges in polygon meshes, or the normal variations between the 6-nearest neighbor points in point clouds.

In polygon meshes we also take into account surface areas associated with each input edge with respect to the vertex of interest. This area is computed as the average of the mixed areas [MDSB02] associated with the center vertex in the triangles incident to the edge. This initial guess produces results very similar to Rusinkiewicz's [Rus04] method.

2.7. Convergence

There are two issues of convergence. On the one hand, there is the convergence of the IRLS approach to a final solution. Secondly, there is the issue of convergence of the method overall to true curvature values as tessellation (sample density) increases.

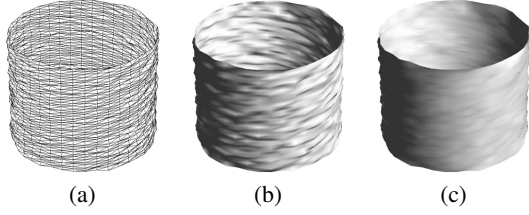


Figure 9: Re-estimation of normals from final weights on normal pairs. (a): mesh with 10% noise of median edge length added to vertices; (b): estimated vertex normals using [Max99] weighted face normal averaging. (c): Re-estimated normals using weights obtained from our algorithm resulting in a 66% error reduction.

Regarding IRLS convergence, in general, such a method is guaranteed to converge since each iteration decreases the fitting error and this error is bounded from below by zero [Hub81]. Note that we use a support region to control leverage. This is a standard approach and, in practice, we observe no convergence problems in any of our experiments. Figure 8 is indicative of the algorithm’s behavior. In practice, we typically achieve convergence in 10 to 20 iterations, even under high noise.

Regarding convergence to true values with increased sampling, maximum likelihood estimates (such as those obtained by M-estimation) converge with increased sample density (maximum likelihood bias tends to zero as the number of samples tends to infinity [Kay93].) We also show this holds in practice in figures 10b through 10d. Not only does the approach converge, it can be seen to converge with a higher rate than the others [CSM03, GI04, Rus04]. In particular, figure 10d addresses the non-noisy but irregular tessellation case.

3. Robust normal recomputation

The derived curvature tensors and M-estimation weights from the IRLS process can be used to recompute the normal vector field on the surface in order to improve noisy normals. The normal vector field recomputation is feature preserving as the M-estimation weights reveal which normal variation samples are inliers or outliers for the estimation of the differential properties at each point of interest.

In order to recompute the normals, we derive a slightly different expression of equation (4) by including all the normal variation components in the 3D orthonormal basis formed by the orthogonal tangent vectors \vec{u} , \vec{v} and the point’s normal \vec{w} :

$$\underbrace{\begin{bmatrix} \nabla_{\vec{u}} \vec{N} \cdot \vec{u} & \nabla_{\vec{v}} \vec{N} \cdot \vec{u} \\ \nabla_{\vec{u}} \vec{N} \cdot \vec{v} & \nabla_{\vec{v}} \vec{N} \cdot \vec{v} \\ \nabla_{\vec{u}} \vec{N} \cdot \vec{w} & \nabla_{\vec{v}} \vec{N} \cdot \vec{w} \end{bmatrix}}_{\text{unknowns}} \cdot \begin{bmatrix} \vec{\Delta p} \cdot \vec{u} \\ \vec{\Delta p} \cdot \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{\Delta n} \cdot \vec{u} \\ \vec{\Delta n} \cdot \vec{v} \\ \vec{\Delta n} \cdot \vec{w} \end{bmatrix} \quad (4)$$

After the final M-estimation iteration for a given vertex c ,

we recompute the normal differences between c and every point p on the surface in the operating region using the final M-estimation weights. Then the point’s new normal is computed as the normalized weighted sum of the normals of its neighbor points in the operating region plus the derived $\vec{\Delta n}$ between them transformed back from the $\vec{u}, \vec{v}, \vec{w}$ frame:

$$\vec{n}_c \leftarrow \text{unit} \left(\sum_p w_{pc} (\vec{n}_p + \vec{\Delta n}_{pc}) \right)$$

The results obtained from this method are illustrated in figures 1c and d, and 9.

4. Implementation and results

The algorithm for the robust statistical estimation of curvature begins by gathering the normal variation samples in the operating region of each point of interest, and computes an initial estimate of the curvature tensor. It then proceeds to iterate between computing the residuals, estimating cost and new weights, and computing a new tensor estimate based on these values.

In our experiments we ran this IRLS process for no more than 50 iterations or until convergence. The resulting M-estimation weights are then also used to obtain our curvature derivative estimates.

If there is considerable noise inside a support region, a higher median residual (and thus larger σ) will imply a slower-decreasing weight function which gives more weight to points further from the center point. Conversely, with lower noise, the weight function becomes more tightly focused, resulting in a curvature tensor estimate based on the closest neighbors of each point of interest. This is illustrated in figures 6 and 7.

Due to the iterative local optimization nature of the M-estimation algorithm, we acknowledge that the approach becomes slower than fixed-ring neighborhood approaches. Note, however, that its overall complexity remains linear in the number of points or vertices of the discrete surface. We also notice that curvature estimation is an off-line or non real-time dependent step for many applications, like non-photorealistic rendering and shape analysis for non-deforming models.

Indicatively, our current CGAL implementation needs about 20 seconds for a polygon mesh of 10K vertices, while for a mesh of 100K vertices it may need approximately 2 minutes on a single core Pentium IV 3.0GHz. For 1M vertices, the running time can be approximately 20 minutes.

Figure 10 summarizes the thorough empirical validation of our method. This figure presents tests performed on ground truth surfaces with increasing noise as well as tessellation density. Furthermore, the tests include various types of tessellations, both regular and irregular (see figure 4b and c). In all cases our method is compared against the state of

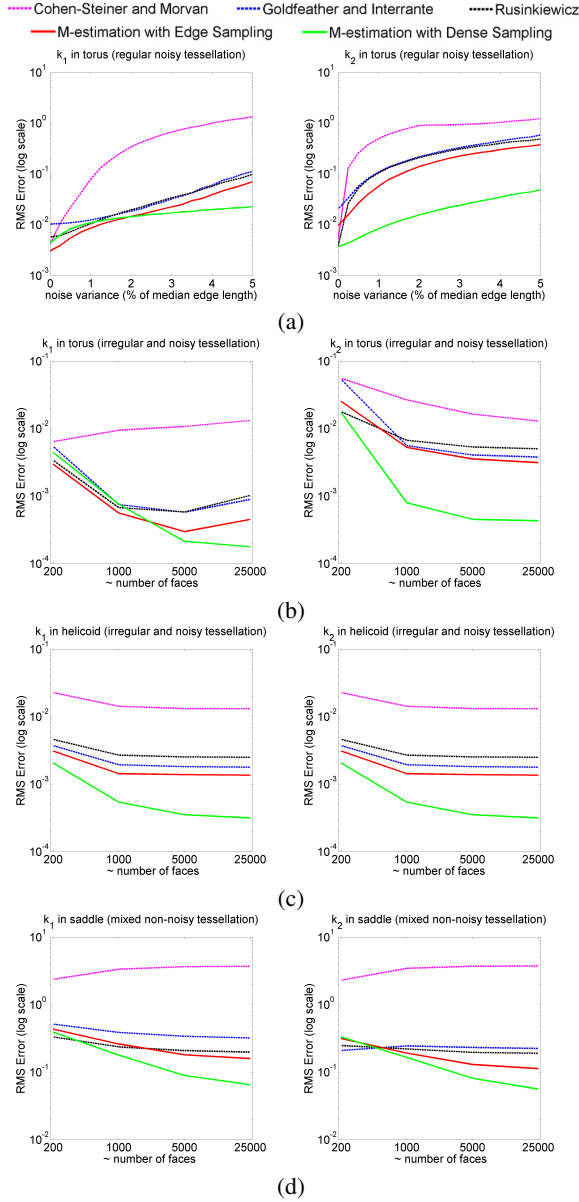


Figure 10: Comparison results of our method with state-of-the-art approaches. The same weighting scheme for normal estimation [Max99] is used for the cubic patch fitting method, Rusinkiewicz’s operator and our algorithm. All error values are plotted on a logarithmic scale. (a) RMS error of k_1 and k_2 in a regularly tessellated torus mesh versus increasing random normal and tangential noise on its vertices; (b) RMS error of k_1 and k_2 in an irregularly tessellated (see figure 4c) torus mesh with 2% added random noise versus increasing mesh resolution (c) RMS error of k_1 and k_2 in an irregularly tessellated helicoid with 2% added random noise versus increasing mesh resolution; (d) RMS error of k_1 and k_2 in a monkey saddle with non-noisy mixed tessellation versus increasing mesh resolution.

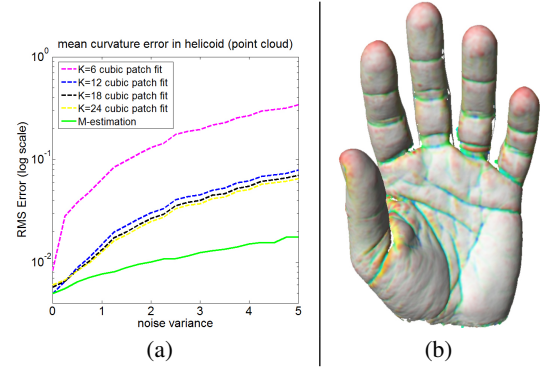


Figure 11: Robust statistical estimation of curvature applied to point clouds (a) Error of our mean curvature values on a helicoid point cloud compared to a cubic patch fitting approach on k -nearest neighbors (b) PointShop3D rendering of principal curvature values obtained by our method for a scanned point cloud model of a hand. Normals are estimated by locally fitting planes [HDD*92]

the art and shown to improve on the estimates by up to an order of magnitude (note the log scale.) Figure 2 illustrates an indicative comparison of results on a scanned mesh, while figure 3 does so for a marching cubes tessellation of an implicit surface model. Figure 11 shows the comparison of our method to cubic patch fitting on point clouds. Finally, figure 12 illustrates the impact of applying our robust estimate of curvature to mesh segmentation using the curvature-based method of Lavoue *et al.* [LDB05] and figure 13 does so for suggestive contour rendering.

5. Conclusion and future work

In this paper we presented a robust statistical approach for the estimation of curvature in discretized surfaces. We showed an algorithm which is capable of achieving an estimate of the curvature tensor by robustly fitting a linear model to the normal variation samples in appropriately varying regions around each point of interest. Furthermore, the method automatically converges to an anisotropic area for an accurate and stable curvature estimation with no user intervention or preprocessing. Our method was shown to outperform current techniques in accuracy, sometimes by an order of magnitude.

An interesting topic of future research is how robust statistics approaches can be applied to other problems that require robust estimates of properties in discretized surfaces.

Acknowledgements: We thank Eitan Grinspun for providing us with tessellations for our test surfaces, Guillaume Lavoué for the executable of his method and Ryan Schmidt for the implicit surface dog model. We acknowledge the AIM@SHAPE repository for the pulley and hand models and the Stanford repository for the David model. This project is partially funded by MITACS.

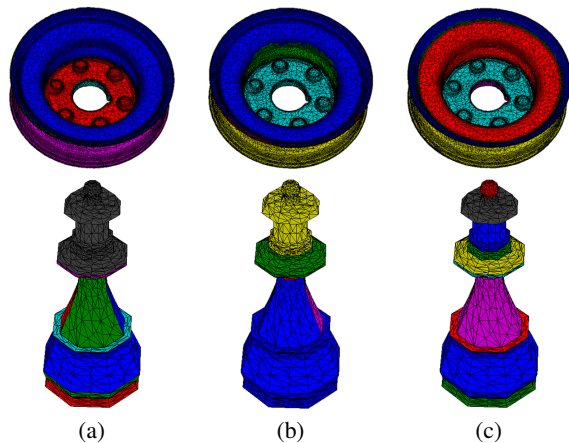


Figure 12: Lavoue et al.'s curvature clustering segmentation [LDB05] comparison of results for a scanned pulley and a slightly noisy queen model. **Column (a):** using the normal cycle method in a one ring neighborhood; **column (b):** best result obtained from applying the normal cycle method on a larger neighborhood to improve estimate; **column (c):** using our curvature results. The same number of clusters and requested regions were used in all three cases. Note how our method produces results which respect the feature edges of the models.

References

- [ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. *ACM Transactions on Graphics* 22, 3 (2003), 485–493.
- [CDR00] CLARENZ U., DIEWALD U., RUMPF M.: Anisotropic geometric diffusion in surface processing. In *VIS '00: Proceedings of the conference on Visualization '00* (2000), IEEE Computer Society Press, pp. 397–405.
- [CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *Eurographics Symposium on Geometry processing* (2003), Eurographics Association, pp. 177–187.
- [CR88] CARROLL R., RUPPERT D.: *Transformation and Weighting in Regression*. Chapman and Hall, 1988.
- [CS92] CHEN X., SCHMITT F.: Intrinsic surface properties from surface triangulation. In *European Conference on Computer Vision* (1992), Springer-Verlag, pp. 739–743.
- [CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted delaunay triangulations and normal cycle. In *Symposium on Computational geometry* (2003), pp. 312–321.
- [DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. In *ACM SIGGRAPH* (2003), ACM Press, pp. 848–855.
- [DKT06] DESBRUN M., KANSO E., TONG Y.: Discrete differential forms for computational modeling. In *ACM SIGGRAPH Courses* (2006), ACM Press, pp. 39–54.
- [DTB06] DIEBEL J. R., THRUN S., BRUNIG M.: A bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics* 25, 1 (2006), 39–59.

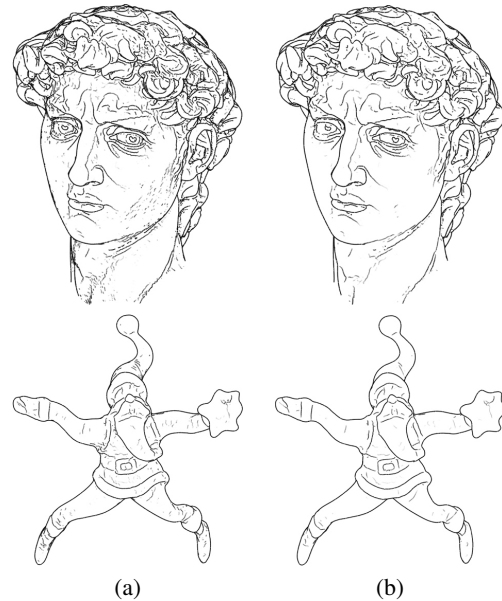


Figure 13: Comparative suggestive contour renderings. **Column (a):** using the curvature field obtained with Rusinkiewicz's approach; **column (b):** using the curvature field obtained with our algorithm. Notice that in both cases no a-posteriori smoothing or thresholding of the values was performed. Although a result qualitatively similar to ours could be obtained by such means, it requires time consuming and interactive user manipulation to obtain the proper parameters while our results were fully automated.

- [FP02] FORSYTH D. A., PONCE J.: *Computer Vision: A Modern Approach*, first ed. Prentice Hall, 2002.
- [GG06] GATZKE T., GRIMM C.: Estimating curvature on triangular meshes. *International Journal of Shape Modeling* 12, 1 (2006), 1–29.
- [GGRZ06] GRINSPUN E., GINGOLD Y., REISMAN J., ZORIN D.: Computing discrete shape operators on general meshes. *Eurographics (Computer Graphics Forum)* 25, 3 (2006).
- [GI04] GOLDFEATHER J., INTERRANTE V.: A novel cubic-order algorithm for approximating principal direction vectors. *ACM Transactions on Graphics* 23, 1 (2004), 45–63.
- [Ham93] HAMANN B.: Curvature approximation for triangulated surfaces. *Geometric modelling* (1993), 139–153.
- [HDD*92] HOPPE H., DEROSE T., DUCHAMP T., McDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *ACM SIGGRAPH* (1992), ACM Press, pp. 71–78.
- [HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics* 25, 3 (2006), 569–578.
- [HG99] HECKBERT P. S., GARLAND M.: Optimal triangulation and quadric-based surface simplification. *Comput. Geom. Theory Appl.* 14, 1-3 (1999), 49–65.
- [HPW05] HILDEBRANDT K., POLTHIER K., WARDETZKY M.:

- Smooth feature lines on surface meshes. In *Eurographics Symposium on Geometry processing* (2005), pp. 85–90.
- [HRRS86] HAMPÉL F. R., RONCHETTI E. M., ROUSSEEUW P. J., STAHEL W. A.: *Robust Statistics: The Approach Based on Influence Functions*. Wiley-Interscience, 1986.
- [HS02] HAMEIRI E., SHIMSHONI I.: Estimating the principal curvatures and the darbox frame from real 3d range data. *3dpvt 00* (2002), 258–266.
- [Hub81] HUBER P.: *Robust Statistics*. Wiley-Interscience, 1981.
- [JF89] JAIN A., FLYNN P.: Error bounds and optimal neighborhoods for mls approximation. In *Computer Vision and Pattern Recognition* (1989), pp. 110–116.
- [Kay93] KAY S. M.: *Fundamentals of Statistical Signal Processing*. Prentice Hall PTR, 1993.
- [LBS07] LANGER T., BELYAEV A., SEIDEL H.-P.: Exact and interpolatory quadratures for curvature tensor estimation. *Computer Aided Geometric Design Accepted* (2007).
- [LCOLTE07] LIPMAN Y., COHEN-OR D., LEVIN D., TALEZER H.: Parameterization-free projection for geometry reconstruction. In *ACM SIGGRAPH, to appear* (2007), ACM Press.
- [LDB05] LAVOUE G., DUPONT F., BASKURT A.: A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design (CAD)* 37, 10 (2005), 975–987.
- [LP05] LANGE C., POLTHIER K.: Anisotropic smoothing of point sets. *Computer Aided Geometric Design* 22, 7 (2005), 680–692.
- [LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. In *ACM SIGGRAPH* (2002), ACM Press, pp. 362–371.
- [LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics* 25, 3 (2006), 681–689.
- [LZH*07] LAI Y.-K., ZHOU Q.-Y., HU S.-M., WALLNER J., POTTMANN H.: Robust feature classification and editing. *IEEE Transactions on Visualization and Computer Graphics* 13, 1 (2007), 34–45.
- [Max99] MAX N.: Weights for computing vertex normals from facet normals. *Journal of Graphics Tools* 4, 2 (1999), 1–6.
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer-Verlag, 2002, pp. 35–57.
- [MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3d geometry. *ACM SIGGRAPH* 25, 3 (2006), 560–568.
- [MNG04] MITRA N. J., NGUYEN A., GUIBAS L.: Estimating surface normals in noisy point cloud data. In *Special issue of International Journal of Computational Geometry and Applications* (2004), vol. 14, pp. 261–276.
- [PSK*02] PAGE D. L., SUN Y., KOSCHAN A. F., PAIK J., ABIDI M. A.: Normal vector voting: crease detection and curvature estimation on large, noisy meshes. *Graph. Models* 64, 3/4 (2002), 199–229.
- [RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling* (2001), pp. 145–152.
- [RMB07] RAMAMOORTHI R., MAHAJAN D., BELHUMEUR P.: A first-order analysis of lighting, shading, and shadows. *ACM Transactions on Graphics* 26, 1 (2007), 2.
- [Rus04] RUSINKIEWICZ S.: Estimating curvatures and their derivatives on triangle meshes. *3DPVT* (2004), 486–493.
- [SAG95] SAWHNEY H. S., AYER S., GORKANI M.: Model-based 2d&3d dominant motion estimation for mosaicing and video representation. In *ICCV* (1995), IEEE Computer Society, pp. 583–590.
- [She01] SHEFFER A.: Model simplification for meshing using face clustering. *Computer Aided Design (CAD)*, 33 (2001), 925–934.
- [SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Symposium on Geometry Processing* (2006), pp. 111–119.
- [SS05] SIMARI P. D., SINGH K.: Extraction and remeshing of ellipsoidal representations from mesh data. In *Graphics Interface* (2005), pp. 161–168.
- [SSK*05] SURAZHISKY V., SURAZHISKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Transactions on Graphics* 24, 3 (2005), 553–560.
- [Ste99] STEWART C. V.: Robust parameter estimation in computer vision. *SIAM Rev.* 41, 3 (1999), 513–537.
- [SW92] STOKELY E., WU S. Y.: Surface parametrization and curvature measurement of arbitrary 3-d objects: Five practical methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 8 (1992), 833–840.
- [Tau95] TAUBIN G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *ICCV* (1995), IEEE Computer Society, pp. 902–909.
- [TRZS04] THEISEL H., ROSSL C., ZAYER R., SEIDEL H.-P.: Normal based estimation of the curvature tensor for triangular meshes. In *Computer Graphics and Applications* (2004), IEEE Computer Society, pp. 288–297.
- [TT05] TONG W.-S., TANG C.-K.: Robust estimation of adaptive tensors of curvature by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 3 (2005), 434–449.
- [YLHP06] YANG Y.-L., LAI Y.-K., HU S.-M., POTTMANN H.: Robust principal curvatures on multiple scans. In *Eurographics Symposium on Geometry Processing* (2006), pp. 223–226.