



Denoising of point cloud data for computer-aided design, engineering, and manufacturing

Hao Chen¹ · Jie Shen^{1,2}

Received: 8 September 2016 / Accepted: 14 November 2017
© Springer-Verlag London Ltd., part of Springer Nature 2017

Abstract

In this review paper, we first introduce different sensors used in geometric sensing for obtaining the geometric information of various objects or parts. Next, two types of measurement errors are defined. Then, we discuss the existing methods for removing or correcting short-range and long-range measurement errors as well as noise at geometric discontinuity. Finally, some conclusions are drawn and future research directions are provided.

Keywords Measurement noise · Geometric noise · Noise reduction · Geometric discontinuity · Laser scanning · Geometric information rectification

1 Introduction

Geometric sensing (GS) is a key procedure to generate digital models by acquiring the geometric information of existing objects via sensor technology. The geometric information obtained from GS can be used to reconstruct digital models in various disciplines such as measurement, computer-aided design/analysis/manufacturing, control, and gaming. A laser scanner is one type of geometric sensor and produces digital data of object surface on the basis of triangulation principle (Fig. 1a), in which the laser emitter, camera, and each surface point form a measurement triangle. The edge length between the laser emitter and camera is known, and two angles at the laser emitter corner and the camera corner can be determined for each surface point. Thus, the location of the surface point (i.e., one end node of the triangle) is accurately computed based on trigonometry for objects at a short range of distance (less than several meters).

Three-dimensional flash LIDAR (LIght Detection And Ranging) is a 3D camera technique based on laser–radar paradigm. It captures both 3D point cloud and intensity (albedo) data. No extra camera is needed in this case for obtaining intensity information, as illustrated in Fig. 1d. LIDAR is based on the time-of-flight principle in which the distance of each surface point is determined by timing the round-trip time of a pulse of laser light. Since the speed of light is known, the travel distance of light pulse from each pixel of the sensor to the corresponding surface point is easily computed. Each pixel of the sensor acts like a “3D smart” pixel which records both time of flight and intensity of surface point. Another type of sensor based on the time-of-flight principle is 3D laser rangefinder, which detects the distance of only one point in its direction view. The rangefinder needs to rotate itself or to use a system of rotating mirrors for detecting the range along different directions, as illustrated in Fig. 1c. A typical laser rangefinder can measure 10,000 points per second for objects at a long-distance range (up to the order of kilometers) with a relatively low accuracy (the order of millimeters), compared with the accuracy of a triangulation laser scanner (the order of tens of micrometers).

The time-of-flight rangefinder performs poorly at the edge or corner of an object, because the light scattering due to geometric discontinuity at the edge may send back false information about two different locations for one laser pulse or because laser light pulses seldom hit the exact edge and corner. Similarly, the triangulation scanner loses its measurement accuracy at the edge or corner of an object because

✉ Jie Shen
shen@umich.edu

Hao Chen
pschenhao@sues.edu.cn

¹ School of Automotive Engineering, Shanghai
University of Engineering Science, Shanghai 201620,
People's Republic of China

² Department of Computer and Information Science,
University of Michigan-Dearborn, Dearborn, MI 48128, USA

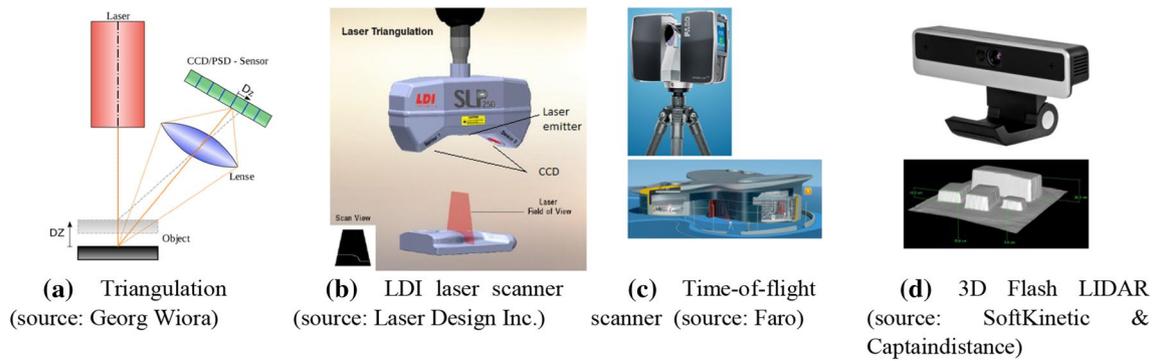


Fig. 1 Examples of geometric sensing principles and sensors

of edge flare caused by geometric discontinuity. Even an averaging operation on multiple measurement points still generates a wrong estimation at the location of the edge or corner point. Laser triangulation sensors, one type of the most popular non-contact optical sensors for 2-m close range measurement [1–3], have a very low measurement accuracy because of coherent or speckle noise [4] and varying surface reflectance. This hinders their applications in tolerance-sensitive metrology and reverse engineering. This paper will present the efforts and advances in existing studies for tackling the aforementioned problems.

The rest of this paper is organized as follows. In Sect. 2, three major problems of measurement errors are introduced. Long-range measurement noise and its rectification are discussed in Sect. 3, followed by the correction of short-range measurement noise in Sect. 4. Section 5 is dedicated to the measurement errors at geometric discontinuity, and in Sect. 6, computational efficiency is discussed. Some concluding remarks and future research directions are provided in Sect. 7. Because of a broad scope covered by this paper, it is impossible for the authors to mention all the published papers in this field. Our main goal is to present a relatively complete high-level picture about the state-of-the-art rather than giving an exhaustive list of existing studies.

2 Errors caused by measurement noise

Among all the types of geometric sensing techniques (laser scanning, time-of-flight range finding, structural lighting, stereo vision, etc.), there are three major unsolved problems, as illustrated in Fig. 2. The ghost points in Fig. 2a lead to a number of measurement outliers. These outliers can be considered as long-range measurement error. Depending upon the correlation among the outliers, we have isolated outliers and non-isolated clusters. The former is easy to be handled, while the latter poses a technical challenge especially when a cluster of outliers is non-isolated, i.e., the cluster is closely attached to the main surface of the scanned part.

The second major problem is that measurement errors tend to be much greater at C^1 discontinuity, where sharp edges or corners are located, as illustrated in Fig. 2b. This is caused by light scattering at a sharp edge or corner. In the case of rangefinder, a laser beam will normally miss its collision with a point that is exactly on a sharp edge. Figure 3 illustrates two possible types of errors caused by independent piecewise fitting.

The third problem in Fig. 2c would result in some missing data. Semantic feature extrapolation and geometric

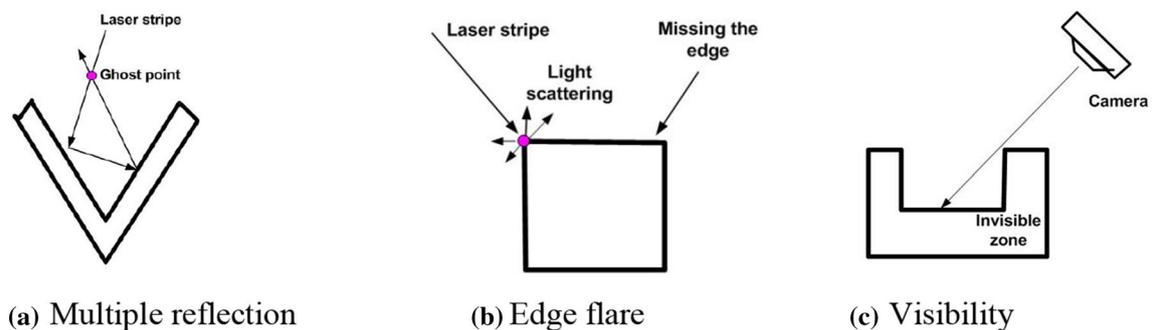


Fig. 2 Major problems with geometric sensing [4]

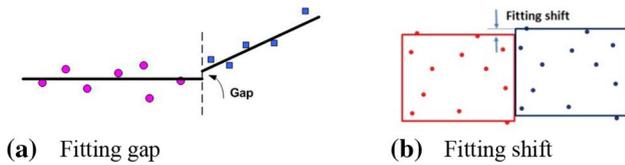


Fig. 3 Potential problems caused by piecewise fitting

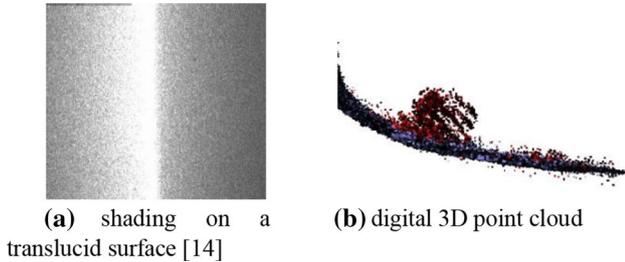


Fig. 4 Outlier cluster on a translucent surface

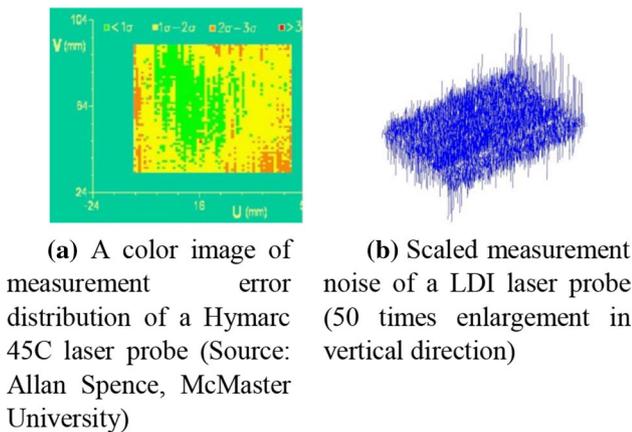


Fig. 5 Measurement error distribution

discontinuity knowledge database are two possible ways to tackle it. This problem is out of the scope of this paper. In addition, Fig. 4 shows a serious threat to geometric sensing caused by the multiple reflections on shiny metallic surfaces of mechanical parts and the scattering of light inside translucent materials. Besides the outliers, a large quantity of small-range measurement noise may be generated. In the next section, the existing efforts in removing or correcting long-range measurement outliers are reviewed.

There were very few investigations on the distribution of measurement errors over a surface of a scanned part. Chan [5] produced a color error map of a laser probe, Hymarc 45C in Fig. 5a, which demonstrates the spatial distribution of measurement error on a rectangular face of a calibration target. Shen [6] demonstrated measurement error of an LDI

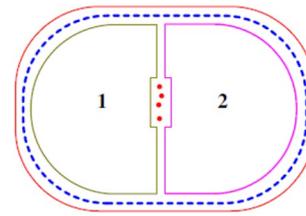


Fig. 6 Outlier cluster at the middle of two mono-oriented groups. (Source: Fig. 10b in [15])

laser scanner in Fig. 5b in which the error was enlarged by 50 times in vertical direction. The measurement error of the laser probe is at least 10–20 times larger than that of a touch probe. Therefore, the accurate correction of laser sensor measurement is crucial for metrology and quality control. Several researchers investigated the effect of different operation parameters on the digitizing errors of laser scanners [7–11]. Some simulation models for laser probes were established [10, 12, 13]. Measurement error of touch probes or laser sensors may be caused by multiple factors, including operator, material, surface property, and sensor itself.

3 Methodologies for handling long-range measurement error—outliers

Many previous studies have been dedicated to reducing or removing measurement outliers. In this paper, we categorize these studies into the two following groups.

3.1 Approaches with an assumption of rare outliers

A mono-oriented grouping method was proposed by Xie et al. [15] for identifying long-range outliers. The “mono-oriented” group was defined as all the grid points of a volumetric grid in the group with the same orientation, as illustrated in Fig. 6. Discrete data points were first organized into an octree via the volumetric grid. Next, the grid points were clustered into mono-oriented groups via an active contour method. By assuming that outliers exist only in rare cases, each outlier cluster can be identified by two neighboring mono-oriented groups with the same orientation. The same orientation infers that the middle part must be an outlier cluster.

3.2 Approaches with an assumption of discrete outliers

A novel spectral surface reconstruction method was designed by Kolluri et al. [16] for handling noisy point clouds, as shown in Fig. 7. Their key idea was the use of spectral graph partitioning and Delaunay triangulation. First, a set



Fig. 7 Spectral partition for removing discrete outliers. (Source: Fig. 1 in [16])

of Delaunay tetrahedrons is formed by a set of data points and eight bounding box vertices, and the data points also serve as dual Voronoi vertices. Next, a subset of the Voronoi vertices is selected as poles from which a pole graph, G , is generated and represented by a pole matrix, L . The graph is cut into two parts (inside and outside) by the eigenvector corresponding to the smallest eigenvalue of L . Consequently, each pole in the inside and outside part is labeled as “inside” and “outside,” respectively. Then, all the remaining Voronoi vertices (i.e., non-pole vertices) form another graph, H . The partitioning of H is based on a different goal to make a relatively smooth surface with low genus. The Voronoi vertices of H are also labeled as either inside or outside. Finally, a surface reconstruction proceeds along the interface between inside and outside tetrahedrons. If all neighboring tetrahedrons of a data point belong to outside (or inside, but not both), this point is removed as an outlier. Unfortunately, only discrete point outliers were suited to this method (Fig. 8).

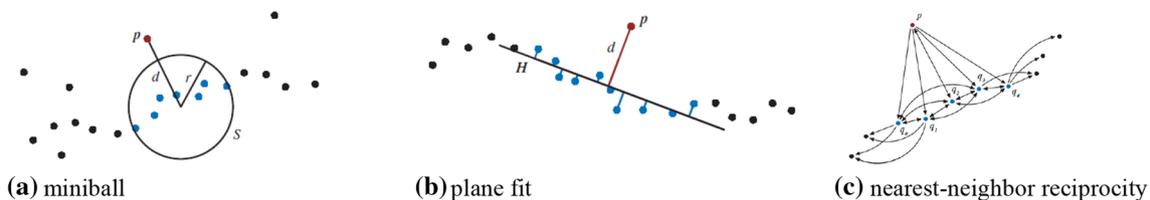


Fig. 8 Three criteria proposed by Weyrich et al. [17]

3.3 Technical challenges

One very difficult challenge is non-isolated outlier clusters, as shown in Fig. 9a, where outliers are clustered and not separated from the main surface of a scanned part. In other words, these outliers are connected to each other as well as the main surface of physical objects. This type of outliers frequently exists in real-world laser scanning of mechanical components and cannot be processed by most of existing approaches. For instance, this non-isolated property is a big challenge to distance-based criteria [17], because distance between outliers in a cluster could be very small and the distance between the cluster and the main surface can be small too. As a result, distance becomes a less effective metric to spate outliers from regular data points. In another example, the non-isolated connection also causes a trouble to the projection method [18] where a moving least square fitting plane or patch could be easily misaligned by non-isolated outlier clusters. Although the method in [18] can handle isolated outlier clusters, it is not suited to dealing with the non-isolated outlier clusters.

Shen et al. [19] were the scholars who first proposed a solution to tackle this type of outliers. There are three technical components in their approach:

A minimum variance principle is used to make a preliminary label on each data point. The minimum variance principle implies that a larger data variance likely leads to a statistically dispersed point cloud, which probably belongs to outliers. The smallest eigenvalue (λ_3) of a covariance matrix in a local point neighborhood is used as the minimum variance.

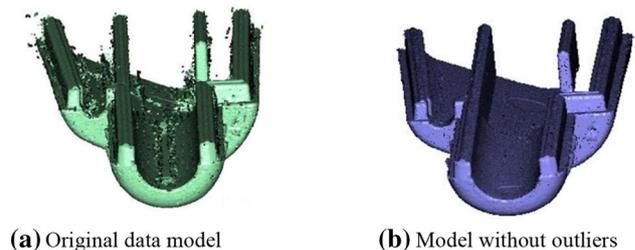


Fig. 9 Spectral moving removal of non-isolated outlier clusters [19]

A bi-means clustering of a normalized histogram is utilized to separate all the data points into two groups : potential outliers and potential data points. The task of identifying outliers is transformed to a data-clustering problem with two target categories: regular data points and outliers. This bi-means clustering can be carried out by a number of existing algorithms such as Lloyd’s algorithm [20].

A surface propagation for a geometric coherence check is carried out to find all the estimated data points as well as estimated outliers. The propagation starts from a data point with low variance.

Figure 9 demonstrates the effectiveness of the spectral moving removal method for non-isolated outlier clusters. The original data model in the left sub-figure contains 1,919,242 data points, and 11,921 outliers are removed in Fig. 9b. The execution time of the method was 2.3 min, which is faster than the method in [16].

Wang and Feng further advanced the method in [19] by proposing a majority voting method [21]. The concept of majority voting has been used in shape recognition and scene classification. In [21], regular data points serve as voters on classifying irregular points. With respect to each irregular point, only its neighboring regular points are selected as voters, while the regular points far away are not used. Based on the voting result, the current point is classified as a good data point if the majority of voters give a favorable vote. Otherwise, it is considered as an outlier. Intuitively, this approach is more robust, because it is based on probability to decide the classification. One precondition to use this algorithm is that a certain portion of original data points can be identified as regular data points before the processing of irregular data points.

In summary, the research in the area of outlier removal tends to be matured in the past decade. The best algorithms have achieved a reasonably good accuracy of removal. One remaining challenge is the slow execution speed associated with those high-accuracy methods. This is crucial to the real-time applications, where time is a crucial factor.

4 Methodologies for handling short-range measurement error—noise

For short-range measurement errors (i.e., local noise), many studies were conducted in the past. Different methods are grouped into five subsections below. Many approaches require a surface mesh as a precondition for handling this type of measurement errors. If an input is a set of discrete data points, an extra step is needed to convert the discrete data points into a surface mesh [22–24] for those methods that require a surface mesh.

4.1 Signal processing

In this group of approaches, signal-processing techniques were used to handle the short-range measurement errors. Typical methods include Laplacian [25–27], bilaplacian [28], mean filter [29], median filter [30, 31], alpha-trimming mean-median filter [32], Gaussian filter [33, 34], Weiner filter [35–37], band-pass filtering [37, 38], and Fourier transform [39, 40].

Laplacian [25–27] is one of the earliest methods in this field. In a discrete case in Fig. 10a, the discrete Laplacian can be expressed by

$$\Delta f = V_L = \sum_{i=0}^4 [V_i - C_V], \tag{1a}$$

$$C_V = C_V + \lambda V_L, \tag{1b}$$

where C_V is the current vertex, and V_i ($i=0,4$) refers to neighboring vertices with respect to C_V . The basic idea of Laplacian smoothing is to use V_L as a correction vector to smooth out the position of C_V . In other words, C_V is updated by V_L during a Laplacian smoothing process. To avoid oversmoothing, a small-value (less than 1 and greater than 0) coefficient, λ , is used, such that λV_L will be the correction vector. Note that the Laplacian smoothing does not converge no matter how small λ is. The benefit of using a small value λ is that users can then use the number of iterations as a parameter to control the smoothing process.

The consequence of oversmoothing is volume shrinkage of the digital model. This can be eliminated via a volume-preserving Laplacian smoothing proposed by Taubin [25] and Vollmer [27]. The basic idea of Vollmer’s approach is to push the vertices of a smoothed mesh back towards their previous location, as illustrated in Fig. 10b. For the sake of clarity, a two-dimensional mesh is drawn in this figure, in which Q1 through Q5 are mesh vertices at their original positions. V2, V3, and V4 are the correction vectors of the conventional Laplacian smoothing for vertices Q2, Q3, and Q4, respectively. To achieve a volume-preserving goal, vector V for each vertex is further modified by a vector B (Fig. 10b) as follows:

$$B_i = - \frac{\mu}{|\text{Neighbor}_{Q_i}|} \sum_{j \in \text{Neighbor}_{Q_i}} V_j, \tag{2}$$

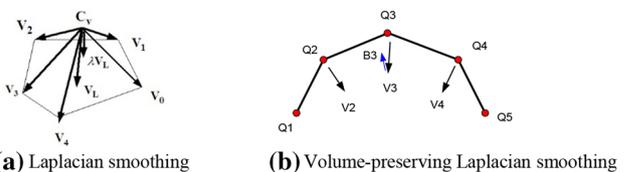


Fig. 10 Laplacian filter on a surface mesh

where Neighbor_{Q_i} refers to a set of neighboring vertices of vertex Q_i . $|\text{Neighbor}_{Q_i}|$ means the number of neighboring vertices around Q_i . μ is another small-value constant, and its absolute value should be smaller than λ .

One way to implement a mean filter in surface meshes [30] is to utilize the information of surface normal for each triangle, as shown in Fig. 11a. The mean of surface normal vectors of neighboring triangles is defined as a new vector \mathbf{M} for each current triangle :

$$\mathbf{M}_{T_i} = \frac{1}{\sum_{j \in \text{Neighbor}(T_i)} A(T_j)} \sum_{j \in \text{Neighbor}(T_i)} A(T_j) \mathbf{N}(T_j), \quad (3)$$

where T_i and T_j represent two triangles. $A(T_j)$ and $\mathbf{N}(T_j)$ are the area and surface normal of triangle T_j , respectively. \mathbf{M}_{T_i} is the mean surface normal of triangle T_i , and is used to construct a correction vector for vertex i :

$$\mathbf{V}_P = \frac{1}{\sum_{j \in \text{Neighbor}(P)} A(T_j)} \sum_{j \in \text{Neighbor}(P)} A(T_j) \mathbf{V}(T_j), \quad (4a)$$

$$\mathbf{V}(T_j) = (\tilde{\mathbf{M}}_{T_j} \cdot \mathbf{PC}_{T_j}) \tilde{\mathbf{M}}_{T_j}, \quad (4b)$$

$$\tilde{\mathbf{M}}_{T_j} = \frac{\mathbf{M}_{T_j}}{|\mathbf{M}_{T_j}|}, \quad (4c)$$

where $\tilde{\mathbf{M}}_{T_j}$ is the normalized vector of \mathbf{M}_{T_j} , and $\mathbf{V}(T_j)$ is the contribution of triangle T_j to the overall correction vector \mathbf{V}_P at vertex P . $\text{Neighbor}(P)$ refers to a set of all neighboring triangles with respect to vertex P , as illustrated in Fig. 11b. \mathbf{C}_{T_j} is the centroid of triangle T_j . \mathbf{PC}_{T_j} is a vector from vertex P to centroid \mathbf{C}_{T_j} , and \cdot refers to a vector dot product. Equation (4a) is the main formula for the mean filter of surface meshes [30].

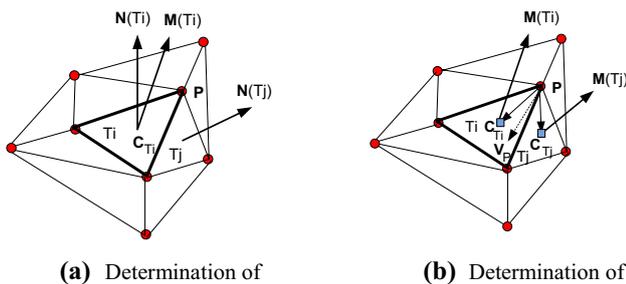


Fig. 11 Mean filter on a surface mesh

Gaussian filter is commonly used in image denoising through a convolution of an image with a Gaussian function :

$$G(x, y) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (5)$$

where x and y are the two-dimensional coordinates. In image processing, these two are the coordinates on an image plane. In the case of surface mesh, x and y two mean coordinates of a local coordinate system with x - y plane as the tangent plane at a particular surface point. σ is the standard deviation, which is a parameter in the surface smoothing process. If it is considered as a variable, then we have an adaptive Gaussian smoothing scheme. In such a case, σ can be a function of surface curvature. One way to implement a Gaussian filter on surface mesh is to modify the previous mean filter by adding a weighting factor into Eq. (3):

$$\mathbf{M}_{T_i} = \frac{1}{\sum_{j \in \text{Neighbor}(T_i)} D(T_j) A(T_j)} \sum_{j \in \text{Neighbor}(T_i)} D(T_j) A(T_j) \mathbf{N}(T_j), \quad (6)$$

$$D(T_j) = G(x_j, y_j) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{x_j^2 + y_j^2}{2\sigma^2}\right), \quad (7)$$

where $D(T_j)$ is the weighting factor for triangle T_j . x_j and y_j refer to the coordinates of the centroid of triangle T_j in a local coordinate system with its origin at the centroid of T_i . The remaining part of the Gaussian filter is similar to that of the previous mean filter.

The evaluation of different signal-processing filters can be conducted on the basis of an L^2 error metric of vertex position or surface normal between denoised mesh and the ground truth without noise. This means that synthetic noise is added to the mesh of ground truth for generating test cases. It is relatively difficult to carry out a quantitative comparison among different smoothing algorithms on surface meshes obtained from real-world sensors. In such cases, qualitative visual evaluation is still feasible.

Generally speaking, numerical experiments indicate that a mean or Gaussian filter performs better than Laplacian smoothing on continuous surfaces, while a median filter excels at the sharp edges or corners. There is no substantial difference between mean and Gaussian filters in the context of the previous implementations. Volume-preserving Laplacian smoothing is an exception and performs reasonably well. Different filters are normally implemented in an iterative way, in which there is no true convergence. In other words, users need to specify the number of iterations for a smoothing process.

4.2 Second-order geometric flow

Geometric flow is a gradient flow associated with a functional on a manifold. Mean curvature flow is a typical example of geometric flow on hypersurfaces in a Riemannian manifold (such as smooth surfaces in three-dimensional Euclidean space). The basic idea of curvature flow is that the evolution rate of a surface is controlled by the magnitude of curvatures at different surface points. Since it can be described by a parabolic partial differential equation, smoothing effect is expected. The surface curvature flow is expressed as

$$\frac{\partial \mathbf{x}_i}{\partial t} = -\kappa_i \mathbf{n}_i, \tag{8}$$

where κ_i and \mathbf{n}_i are the curvature and surface normal at a surface point, and \mathbf{x}_i . κ can be minimum principle curvature (κ_2), maximum principle curvature (κ_1), Gauss curvature ($\kappa_1 \kappa_2$), or mean curvature ($\frac{\kappa_1 + \kappa_2}{2}$).

In the cases of surface meshes, discrete curvatures and surface normal can be estimated by a least-squares fitting over a neighborhood of a surface point [41]. Alternatively, curvature normal may be effectively computed by an analytical formula based on angles and vertex coordinates in a one-ring neighborhood [42]. The curvature flow is closely related to a diffusion process represented by

$$\frac{\partial \mathbf{x}}{\partial t} = \Delta \mathbf{x}, \tag{9}$$

where Δ is a Laplace operator, and can be approximated by Eq. (1).

Existing methods of the second-order geometric flow include mean curvature flow [42–46], area decreasing [47], and Laplacian of mean curvature [48]. The mean curvature flow was often used on implicit surfaces [49–51]. In computer vision, curvature flows were used on curves [52–54]. In general, the mean curvature flow based on a one-ring neighborhood, although efficient, is less accurate, compared to the least-squares fitting approach. At sharp edges or corners, it needs a special treatment on the anisotropy (see Sect. 5.1).

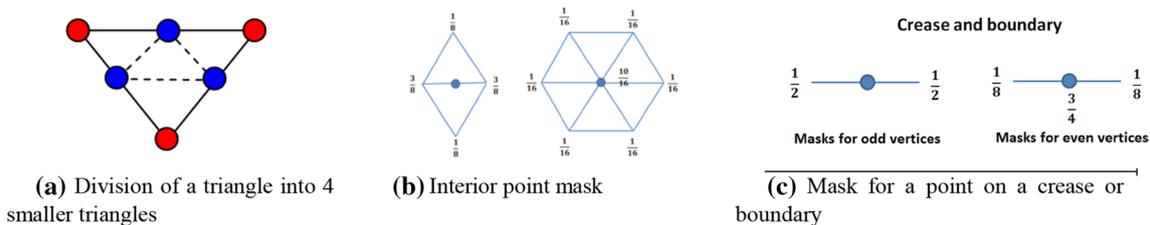


Fig. 12 Loop subdivision surface (Source: Doug James)

4.3 Multiresolution analysis

Multiresolution analysis was frequently used on triangle meshes [28, 38, 55–57]. The main purpose of this approach is to provide a multiresolution tool for editing surface meshes at different resolutions and surface smoothing can be a by-product of a subdivision process, which is a recursive procedure to replace each surface element (triangle or quadrilateral) with the same type of multiple smaller elements.

Loop subdivision surface [58] represents a smooth surface that is obtained from a recursive subdivision process to refine an irregular triangle mesh into a piecewise linear approximation of the underlying smooth surface. Note that approximation rather than interpolation is used to achieve local modification property without impacting the global shape. The underlying concept is use of spline refinement. At each level of subdivision, there are two main sub-tasks:

- division of each surface element into smaller ones (Fig. 12a);
- update on the coordinates of vertices via masks in either Figs. 12b or 12c.

Recursive execution of the above two steps leads to a convergence to the target smooth surface, as illustrated in Fig. 13. Note that extraordinary vertices (such as corner points) need a special treatment and the right mask in Fig. 12b can be extended to the cases, where there are n neighboring vertices.

Catmull–Clark subdivision surface is an earlier algorithm based on quadrilaterals [59]. Figure 14 illustrates the division of a quadrilateral into four smaller ones and masks for computing new coordinates of vertices at each level of subdivision.

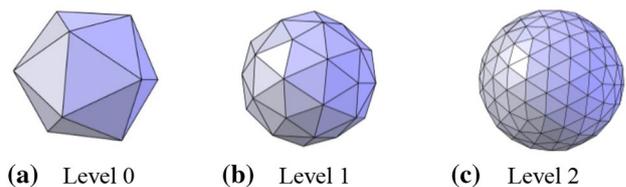


Fig. 13 Example of loop subdivision surface (Source: Wikipedia)

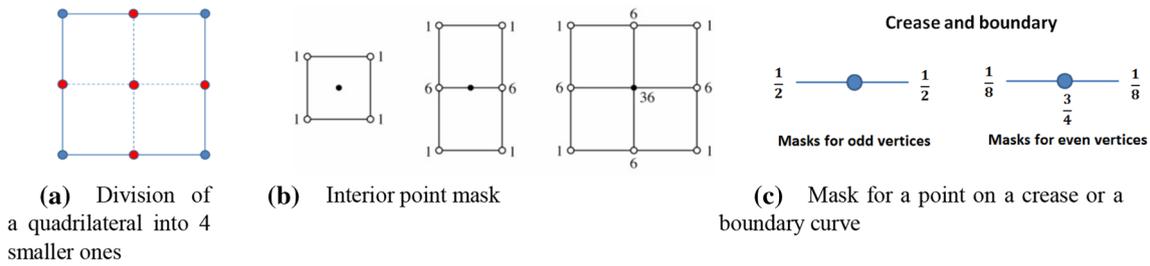


Fig. 14 Catmull–Clark subdivision surface. (Source: Doug James)

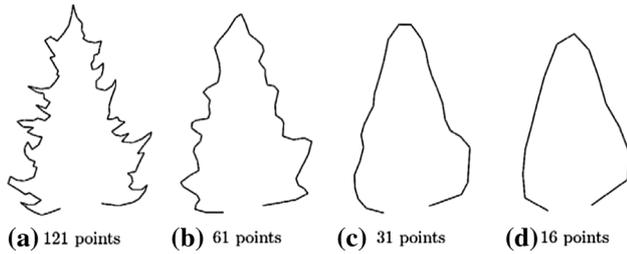


Fig. 15 Smooth reverse subdivision that starts from (a) and ends with (d) [62]

Subdivision normally starts from a coarse mesh and ends with a dense smooth mesh. If an input is a set of discrete data points, we need to convert the point cloud to a surface mesh first and then simplify the mesh to a coarse mesh [60, 61]. Alternatively, a reverse subdivision process can be used [62]. The reverse subdivision starts from a dense curve or surface and ends with a coarse curve or surface, as illustrated in Fig. 15.

In general, multiresolution analysis is more suited to surface editing or simplification rather than smoothing. To achieve smoothing, discrete data points need to be converted to surface mesh first and the quality of the mesh should be good. Some algorithms demand a regular surface mesh as a precondition, which can be considered as a limitation to this approach.

4.4 Least-squares fitting

Moving least-squares fitting [63] is a typical approach to remove surface noise, compared to a global least-squares fitting of the entire surface. As a local fitting, it is flexible to handle arbitrary shapes. The moving least-squares fitting within each local neighborhood consists of two main tasks:

- a. determination of a local coordinate system;
- b. update on the coordinates of all the discrete points within the neighborhood.

The neighborhood of a vertex \mathbf{p} can be constructed from its two-ring vertices of a surface mesh or K nearest neighboring points in a k - d tree. The k - d tree is a space-partitioning data structure for organizing discrete data points in 3D space [64] and particularly suited for non-uniformly distributed points, compared to an octree. Principal component analysis [65] can be applied for estimating the surface tangent plane at vertex \mathbf{p} through a covariance matrix over its neighboring discrete data points:

$$\mathbf{C}\mathbf{V} = \sum_{\mathbf{q} \in \text{Nbhd}(\mathbf{p})} (\mathbf{q} - \mathbf{p}) \otimes (\mathbf{q} - \mathbf{p}), \tag{10}$$

where $\text{Nbhd}(\mathbf{p})$ represents a set of neighboring vertices around vertex \mathbf{p} . \otimes is an outer product operator of two vectors. The eigenvectors $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and eigenvalues $(\lambda_1 \geq \lambda_2 \geq \lambda_3)$ of the matrix in Eq. (10) can be computed by Jacobi transformation [66]. The third eigenvector, \mathbf{v}_3 , represents a normal vector of the tangent plane, while \mathbf{v}_1 and \mathbf{v}_2 are a pair of base vectors for the tangent plane. A local coordinate system $(x-y-z)$ is then formed as a coordinate system $(\mathbf{v}_1 - \mathbf{v}_2 - \mathbf{v}_3)$.

With the determined local coordinate, a quadric patch may be used to perform a local least-squares fitting around vertex \mathbf{p} :

$$z = f(x, y) = a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6, \tag{11}$$

where coordinates (x, y, z) are measured in $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ directions, respectively. A higher order (greater than 2) surface patch is not necessary under the constraint of computational efficiency, while a linear patch is too simple to be accurate. The linear least-squares estimation [41] of six coefficients a_i for the quadric patch in Eq. (11) is expressed as

$$\mathbf{B}\mathbf{X} = \mathbf{Z}, \tag{12a}$$

$$\mathbf{B} = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1.0 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1.0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1.0 \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_6 \end{bmatrix}, \tag{12b}$$

in which n is the number of vertices, $\text{Neighbor}_{2\text{-ring}}(\mathbf{p})$. If the inverse of $\mathbf{B}^T\mathbf{B}$ exists, Eq. (12a) can be solved by Cholesky decomposition:

$$\mathbf{X} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{Z}, \tag{12c}$$

Otherwise, a singular value decomposition [66] can be used to solve Eq. (12a). Thus, the Cholesky decomposition is tried first and can be used as a conditional flag to determine which one to use. In all of our numerical experiments, there was no single case of failure for the Cholesky decomposition of $\mathbf{B}^T\mathbf{B}$.

After a local surface patch is computed, geometric correction of discrete data points within the neighborhood of vertex \mathbf{p} may be conducted in either surface normal direction or z coordinate direction, as illustrated in Fig. 16a. The correction is essentially to bring each data point back to the surface of the local least-squares fitted patch. The efficacy of the moving least-squares fitting can be seen in Fig. 16b, c, in which a noisy monkey saddle surface is effectively corrected a smooth surface.

Overall, the main strengths of the moving least-squares fitting include

- a. It is well suited to handling discrete noisy data points with arbitrary underlying smooth surface.
- b. It possesses a salient property of convergence with an iterative implementation, compared to other existing methods. Figure 17 demonstrates the convergence of the moving least-squares fitting with quadric local patches and monkey saddle case in Fig. 16b, in which MQ, MN, MD, GS, BL, and VL represent the moving least-squares fitting, mean filter, median filter, Gaussian filter, bilateral filter, and volume-preserving Laplacian, respectively.

The main weaknesses of the moving least-squares fitting are

- a. Its computational time is greater than most of existing methods by a factor of 2–10. Thus, it is suited only in the

cases, where the accuracy is crucial as in high-precision industrial inspection and manufacturing.

- b. It is not suited to geometric discontinuity (sharp edges and corners) without a special treatment. The special treatments are discussed in next section.

As a real-valued function, radial basis function (RBF) can be used in surface smoothing and reconstruction [67–73] via least-squares fitting. In essence, the surface of an object can be approximated by

$$s(\mathbf{x}) = \sum_{i=1}^N \omega_i \vartheta(\mathbf{x} - \mathbf{x}_i), \tag{13a}$$

where N is the number of radial basis functions $\vartheta(\mathbf{x} - \mathbf{x}_i)$ and \mathbf{x}_i represents each center for different RBFs. ω_i refers to a weight and the set of all weights can be estimated by linear least squares.

Carr [74] used the following equation for reconstructing 3D objects:

$$s(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=1}^N \omega_i \vartheta(\mathbf{x} - \mathbf{x}_i), \tag{13b}$$

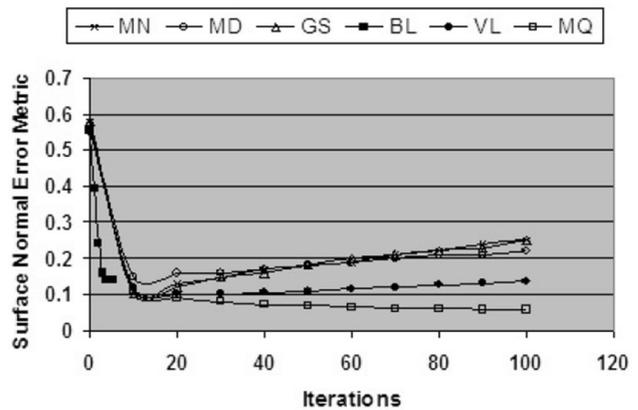


Fig. 17 Convergence of different denoising algorithms [41]

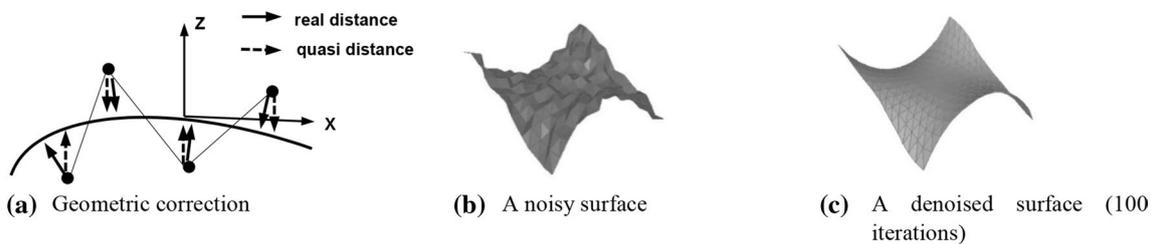


Fig. 16 Moving least-squares fitting of discrete data points [41]

where $p(\mathbf{x})$ is a low-degree polynomial and $\vartheta(\mathbf{x} - \mathbf{x}_i)$ takes a form of polyharmonic function. With noisy data, the following optimization problem is formulated:

$$\min_{s \in BL^{(2)}(\mathbb{R}^3)} \rho s^2 + \frac{1}{N} \sum_{i=1}^N (s(\mathbf{x}_i) - \mathbf{f}_i)^2, \tag{13c}$$

$$s^2 = \int_{\mathbb{R}^3} \left[\left(\frac{\partial^2 s(x)}{\partial x^2} \right)^2 + \left(\frac{\partial^2 s(x)}{\partial y^2} \right)^2 + \left(\frac{\partial^2 s(x)}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 s(x)}{\partial x \partial y} \right)^2 + 2 \left(\frac{\partial^2 s(x)}{\partial x \partial z} \right)^2 + 2 \left(\frac{\partial^2 s(x)}{\partial y \partial z} \right)^2 \right] dx, \tag{13d}$$

in which parameter ρ is positive for controlling a balance between smoothness of the first term and data fidelity of the second term. \mathbf{f}_i refers to the position of a noisy data point. Figure 19 shows the smoothing effect of ρ when it increases.

Ohtake proposed another variation of radial basis functions for approximating 3D scattered data [70]. An object surface is approximated by

$$s(\mathbf{x}) = \sum_{\mathbf{c}_i \in \mathbb{C}} [p_i(\mathbf{x}) + \omega_i] \vartheta_{\sigma_i}(\mathbf{x} - \mathbf{c}_i), \tag{14a}$$

where \mathbb{C} refers to a set of RBF centers, \mathbf{c}_i . $p_i(\mathbf{x})$ is a polynomial for local base approximation for center i , and ω_i is a weight for center i for describing local details. $\vartheta_{\sigma_i}(\mathbf{x} - \mathbf{c}_i)$ is a local radial basis function (Gaussian in particular) at center i with σ_i as a parameter of Gaussian kernel. To consider the partition of unity, $\vartheta_{\sigma_i}(\mathbf{x} - \mathbf{c}_i)$ can be replaced by

$$\Phi_{\sigma_i}(\mathbf{x} - \mathbf{c}_i) = \frac{\vartheta_{\sigma_i}(\mathbf{x} - \mathbf{c}_i)}{\sum_j \vartheta_{\sigma_j}(\mathbf{x} - \mathbf{c}_j)} \tag{14b}$$

A local RBF approximation is expressed as

$$\min \sum_j d_j \vartheta_{\sigma}(\mathbf{p}_j - \mathbf{c}_i) \mathbf{n}_j, \tag{14c}$$

$$d_j = \alpha_j \sum_{i=1}^K \mathbf{p}_j - \mathbf{p}_i^2, \tag{14d}$$

where \mathbf{n}_j is the unit normal of data point \mathbf{p}_j near the σ -neighborhood of center \mathbf{c}_i . d_j is a weight that is proportional to the sum of $\mathbf{p}_j - \mathbf{p}_i^2$ and \mathbf{p}_i refers to K nearest neighbors of \mathbf{p}_j . $\alpha_j \in [0, 1]$ is a confidence coefficient of \mathbf{p}_j . Equation (14c) means that the correction of data points is limited to the normal direction of each point.

Another global RBF approximation is

$$\min \frac{\sum_{i=1}^N d_i f(\mathbf{p}_i)^2}{L^2 \sum_{i=1}^N d_i} + \frac{T_{reg}}{M} \sum_{i=1}^M \left(\frac{\omega_i}{\sigma_i} \right)^2, \tag{14e}$$

in which N and M represent the number of data points and the number of RBF centers, respectively. $T_{reg} = 10^{-5}$. $f(\mathbf{x}) = 0$ refers to a zero-level set that approximates the data point set $\mathcal{P} = \{\mathbf{p}_i\}$.

5 Methodologies for reducing measurement error at geometric discontinuity

When laser scanning sensor, LIDAR, and digital ranger finder are used to measure arbitrarily-shaped objects, sharp edges or corners of these objects often cause severe measurement noise. Then, the shape evaluation of the objects is significantly influenced by the existence of these noisy points at geometric discontinuity. Figure 18 demonstrated two most prominent types of geometric discontinuities: C^0 and C^1 discontinuity. All the other higher order geometric discontinuities are less important in considering a curve as being approximately smooth. Classic optimization methods (such as gradient descent and quasi-newton methods) fail at geometric discontinuity, where no gradient information is available and the traditional piecewise least-squares fitting tends to generate a small magnitude gap (C^0 discontinuity).

5.1 Anisotropic diffusion

Many attempts have been tried to maintain sharp features in a smoothing process. Anisotropic diffusion is one typical approach that was first proposed by Perona and Malik [75] in image processing and later used in different applications: height fields [45], triangle meshes [43, 76–79], level-set surfaces [80], surface and function [81], surface reconstruction [18] via fitting [82–89] or fairing [90–93]. It essentially relies upon the partition of the problem domain into feature

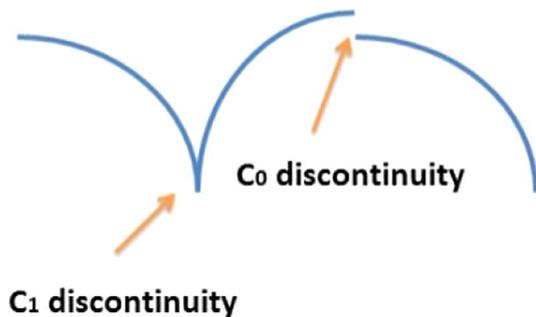


Fig. 18 Geometric discontinuity

Fig. 19 RBF approximation of noisy data points [74]

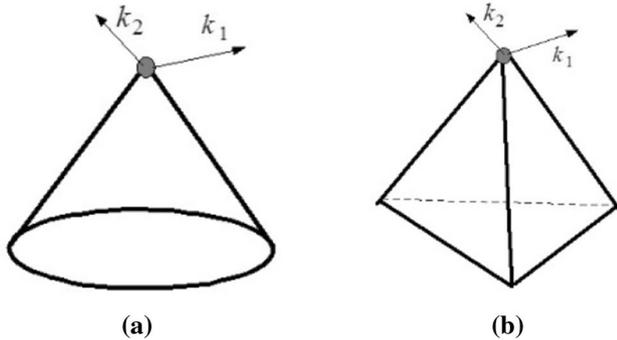
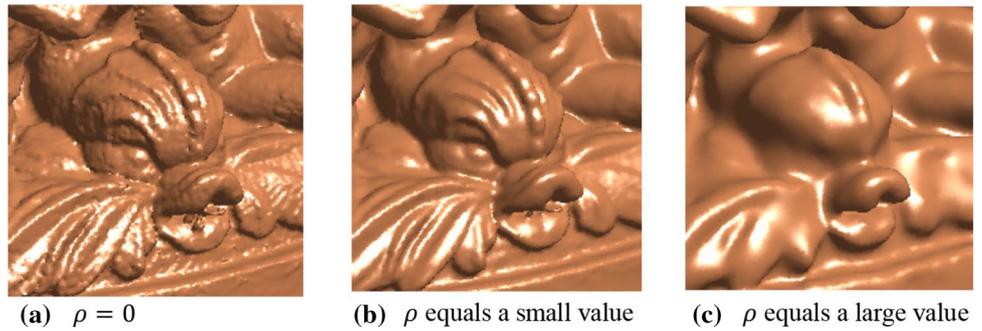


Fig. 20 Two cases of singular points, where anisotropic diffusion does not perform well [41]

and non-feature regions, and smoothing operation is applied only within the feature regions. A weighting function is normally used to avoid an explicit partition with a very small weighting factor for feature regions, where no operation is expected. This approach is not well suited to cases, where a large magnitude of noise exists in feature regions. In contrast to the attenuation, some researchers added an enhancement in feature regions [43, 44, 77]. This may potentially cause over-preserving features.

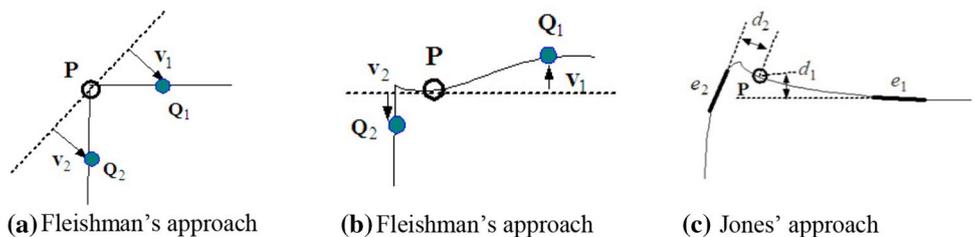
The feature regions can be identified using discrete curvatures. The curvature is normally small along the direction of edge and becomes large in the direction that is perpendicular to the edge. When the anisotropic diffusion is extended from 2D images to 3D surface meshes, one weakness of the approach is that it is suited to some singular points such as the tip of a cone, as shown in Fig. 20a, where both maximum and minimum principal curvatures are the same making the

directions of principal curvatures meaningless and misleading. As a result, no edge enhancement should be applied for, any edge sharpening treatment along the two directions (k_1, k_2). Another example is shown in Fig. 20b, where three edges meet at one point but are not orthogonal to each other. The anisotropic diffusion will not perform correctly at the tip point, where the directions of two principal curvatures do not match with the directions of at least two edges. One remedy is that all the edges are explicitly identified and individually processed by the concept of anisotropic diffusion.

5.2 Bilateral filter

Bilateral filter is another approach for maintaining sharp features in a smoothing process [94, 95]. It is basically a weighted Gaussian filter in which a feature-preserving weighting function is used. One variation of the weighting function is to penalize a large variation in the distance of neighboring vertices to the target tangent plane [95], and another variation is based on the distance between the center vertex and neighboring elements [94]. Both schemes work properly to a certain extent. However, there are some special cases, where the schemes do not perform well as expected. For instance, if the tangent plane in Fig. 21a is determined by the average of normal vectors of two neighboring planes, Fleishman’s approach tends to smooth out the feature. In another possible case, if the center vertex is at a short distance from the sharp feature, the scheme may or may not work depending upon the location of neighboring vertices. In Fig. 21b, vertex Q_2 contributes a vector v_2 in an incorrect way. On the other hand, the second variation (Jones’s scheme) works correctly if the center vertex is on a feature

Fig. 21 Three cases where bilateral filtering does not perform well [41]



edge. However, if the center vertex \mathbf{P} is at some distance from the feature edge, Jones' scheme does not work in an entirely correct way, because the distance d_2 may be even smaller than the distance d_1 in Fig. 21c, such that the center vertex \mathbf{P} makes a wrong contribution to the predictor.

Overall, bilateral filter or anisotropic diffusion is used for sharp edge enhancement [33, 43, 45, 75–77, 79–81, 94–97], and most of these approaches are limited in visual effect. Both approaches are not aimed at precise noise correction near sharp edges or corners in the fields of metrology and reverse engineering, where the denoising accuracy plays an important role for a successful mission.

5.3 Median filter

The conventional median filter is a non-linear filter that may avoid the above problems of anisotropic diffusion and bilateral filter at singular points. One way to implement it is a similar way as in the mean filter [30]. The only difference lies in Eq. (3), which is replaced with

$$\mathbf{M}_{T_i} = \mathbf{N}(T_j), \quad j \text{ is the median of a set } \{\mathbf{N}(T_j) \cdot \mathbf{N}(T_i) \mid j \in \text{Neighbor}(T_i)\}, \quad (15)$$

where $\mathbf{N}(T_j)$ has the same meaning as that in Eq. (3). A variant to this method is a weighted median filter, in which the determination of the median in Eq. (15) is conducted via assigning different weights (positive integers) to neighboring triangles based on either edge-connected neighbor or vertex-connected neighbor. The edge-connected neighbors should have a higher weight than the vertex-connected neighbors.

Since gradient information is not available at sharp features, discrete integrations or averaging operations are invalid at these locations. In the median filter, averaging operations are not necessary, and yet, it provides a reasonable quality of smoothing for high-curvature non-sharp feature regions. However, if the singular point itself contains a certain degree of noise, this filter does not perform well in such cases.

5.4 Piecewise least-squares fitting

Piecewise least-squares fitting means that the surface of an object is partitioned into several sub-regions in each of which no geometric discontinuity exists except its boundaries. Consequently, an independent fitting can be applied on each of these smooth regions. In [98], a moving least square fitting scheme was proposed for removing outliers and noise as a projection problem. This method performs well with any isolated outliers in the form of points and point clusters. Herein, “isolated” refers to the cases, where the outlier points or clusters are not attached to a main surface. A similar approach was reported in [18] with no sharp features

considered. Wang [99] used a mean shift clustering method to determine the best tangent plane in feature-preserving surface reconstruction. It should be more flexible than principal component analysis in analyzing a region around a corner point that is connected to three or more non-orthogonal faces.

One major problem with the piecewise regression is potential gaps at the location of discontinuity, as illustrated in Fig. 3a. Such a gap is extremely not desirable in the fields of metrology and reverse engineering, where precision has a high priority.

Reuter [100] developed an interactive approach that demanded users to specify sharp features manually as a precondition of the method. It becomes tedious for complex objects with many sharp edges or corners. A singularity indicator field (SIF) [101] was proposed to estimate the proximity of each point to discontinuity. It is essentially proportional to the distance between each data point and its projected surface point on the fitted smooth patch. A weighting factor is determined from the value of SIF and is used in a local fitting process. One potential problem arises in the cases, where a significant amount of outliers exists, such that it is difficult to distinguish between outliers and singular points at geometric discontinuity on the basis of SIF.

5.5 Constrained least-squares fitting

To overcome the gap problem of piecewise least-squares fitting, one solution is to use a constrained fitting at geometric discontinuity. However, since gradient information is not available at sharp features, tradition optimization methods such as gradient method become useless. Evolutionary computation is well suited in such cases where the approximate location of corner or edge points can be estimated by a high curvature [41] through principal component analysis. In

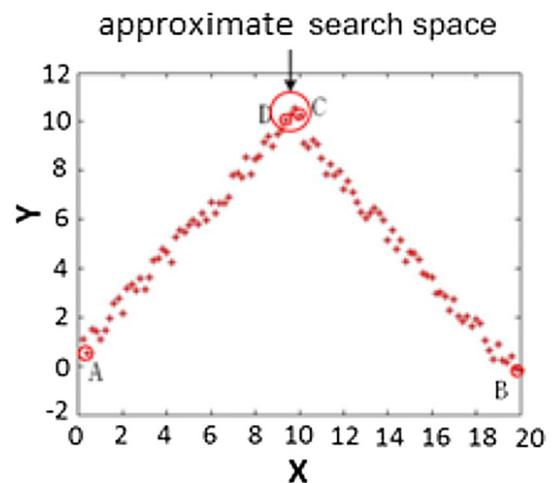


Fig. 22 Approximate search space of a corner point

the cases of two-dimensional and three-dimensional lines, a simple line propagation scheme was reported to find an approximate search space of corner points, as shown in Fig. 22 [102].

Within the approximate search space, an evolutionary computing method is well suited for the task of finding the optimal corner point to minimize the constrained least-squares fitting error. Evolutionary computation is a population-based optimization methodology that mimics the growth or development in a population [103]. In an iterative process, the optimization of the problem is achieved via an evolution of a working set of individuals (i.e., population) [104]. Because few or no assumptions are made on problems to be optimized, it is well suited for the irregular and noisy problems in which no derivative information is available at geometric discontinuity. Three typical approaches of evolutionary computing are (a) genetic algorithm (GA) [105], (b) evolution strategies [106, 107], and (c) evolutionary programming [108, 109]. Particle swarm is another variation of evolutionary computing [110].

Figure 23a demonstrates the transformation of the coordinate increments ($\Delta x, \Delta y, \Delta z$) of each 3D data point into an n -bit binary number (i.e., a chromosome). These binary bits can be generated using a random number generator. Figure 23b shows the result of genetic search, in which the corner point is the constraint for two coordinated least-squares

fitting processes with 10% reduction in fitting error, compared to the traditional fitting method.

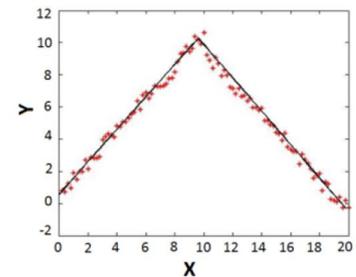
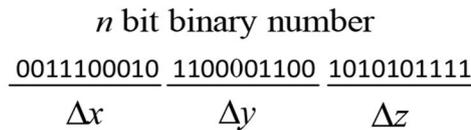
Figure 24a, b illustrates the result of particle swarm for two test cases. A comparison between particle swarm and genetic algorithm is given in Fig. 24c. In general, particle swarm provides a more stable solution, compared to genetic algorithm in the context of constrained least-squares fitting with discrete data points. However, it is difficult to generalize the constrained least-squares fitting for handling the intersection of arbitrary shapes. If feature lines or curves can be explicitly determined [111, 112], surface propagation then starts from the feature line or curve.

5.6 Hybrid smoothing

The basic idea of hybrid smoothing is to use two different algorithms, respectively, for feature and non-feature regions. Below is an example of implementing such an approach [41]:

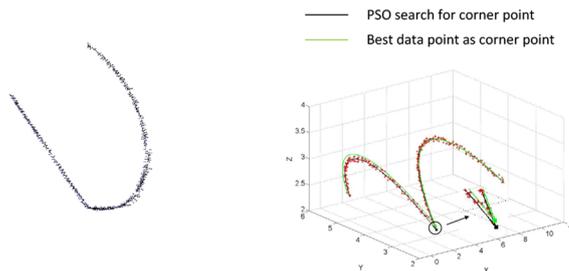
1. Use a feature-preserving pre-smoothing (median filter) that does not require any threshold and implicitly retains the sharp features. Adopt G^1 geometric discontinuity and curvature threshold as an indicator for partitioning feature and non-feature regions. Herein, feature regions refer to the areas in which either sharp edges or high

Fig. 23 Genetic search for the optimal corner point of constrained least-squares fitting [102]



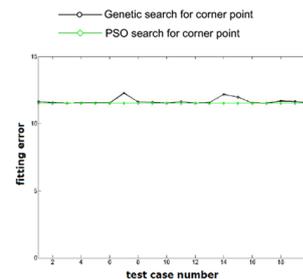
(a) A chromosome for a 3-D case.

(b) Genetic search result



(a) Circle-line case

(b) Circle-circle case



(c) Particle swarm versus genetic algorithm

Fig. 24 Constrained least-squares fitting of circle–line intersection via particle swarm

curvatures exist, while the remaining parts are called the non-feature regions.

2. Select a median filter for feature regions. In comparison with anisotropic diffusion algorithms, its main advantage is no need for the information on the directions of principal curvatures. Such information is unavailable at some singular points (e.g., apex of a cone). It also avoids some pitfalls of bilateral filters at sharp edges.
3. Design a second-order predictor as an accurate indicator for guiding a surface smoothing process in non-feature regions. Here, the predictor is essentially a local quadratic patch as in Eq. (11). To use this equation, a local principal component analysis is needed to determine the surface normal direction (i.e., the direction of z in a local coordinate system). The main benefit of the proposed second-order predictor is a better accuracy and convergence with curved surfaces than the first-order predictors, mean curvature flow, and Gaussian predictors.
4. Apply the second-order predictor in non-feature regions and the median filter in feature regions.

In an algorithmic format, the above hybrid approach is represented by the following procedure that calls two routines for median and second-order filters, respectively. The details of these two routines are given in [41].

ALGORITHM 1: Hybrid Denoising

- (a) perform feature-preserving pre-smoothing of a noisy input mesh
- (b) conduct partition of the resulting surface mesh obtained from step (a) and pass the partition information back to the input mesh
- (c) loop over all elements in the noisy input mesh
 - (c.1) if an element is in a feature region, execute *median_filter()* routine
 - (c.2) if the element is in a non-feature region, invoke *second_order_filter()* routine
 - (c.3) go back to the beginning of (c) and repeat in an iterative way

Figure 25 shows an example of the result of the hybrid smoothing. The efficacy of this method depends on the quality of the algorithms that is chosen for feature and non-feature regions. It also relies on the accuracy of identifying the feature regions.

5.7 Global optimization

Global optimization means that a unified formula of optimization is applied over the entire domain of discrete data points. A typical approach is l_1 -sparse reconstruction of sharp point set surface [113]. The optimization is divided into two parts. In part 1, the optimization of orientation is formulated as

$$N^{out} = \arg \min_N \sum_{(p_i-p_j) \in E} \omega_{ij} \|n_i - n_j\|_2, \text{ s.t. } \forall i \|n_i - n_i^{in}\|_2 \leq \gamma_n, \tag{16}$$

where N^{in} and N^{out} are two sets of surface normal for input and output, respectively. p_i and p_j represent two data points i and j , while n_i and n_j refer to the surface normal at these two points. E is an adjacency set each element of which is a pair (p_i, p_j) . γ_n is an angular threshold that is used to limit the change in the direction of normal at each surface point. ω_{ij} is a weight that has a smaller value at geometric discontinuity:

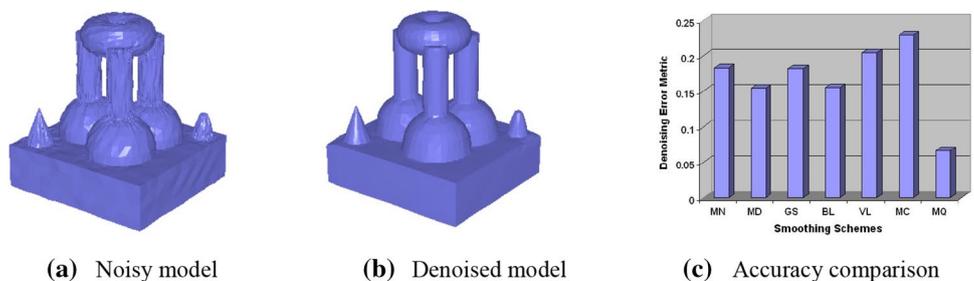
$$\omega_{ij} = e^{-\left(\frac{\theta_{ij}}{\sigma_\theta}\right)^4}, \tag{17}$$

in which σ_θ is a angular parameter (e.g., 10°) and θ_{ij} refers to an angle between initial normal of p_i and p_j .

In part 2, the optimization of vertex position is formulated in a similar way:

$$C^X(X, N^{out}, W, E) = \arg \min_X \sum_{(p_i-p_j) \in E} \omega_{ij} |n_{ij}^{out} \cdot (x_i - x_j)| \\ = \arg \min_t \|At + f\|_1, \text{ s.t. } \forall i \|t\|_2 \leq \gamma_x, \tag{18}$$

Fig. 25 Hybrid smoothing on a synthetic noisy model (MN, MD, GS, BL, VL, MC, and MQ represent mean filter, median filter, Gaussian filter, bilateral filter, volumetric Laplacian, mean curvature flow, and hybrid smoothing, respectively)



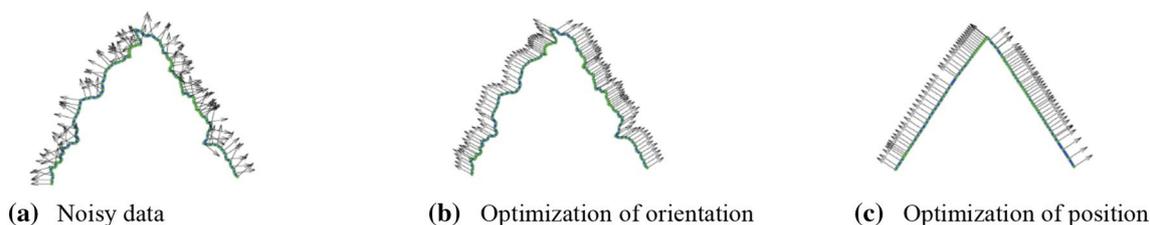


Fig. 26 l_1 -sparse reconstruction of sharp point set surface [113]

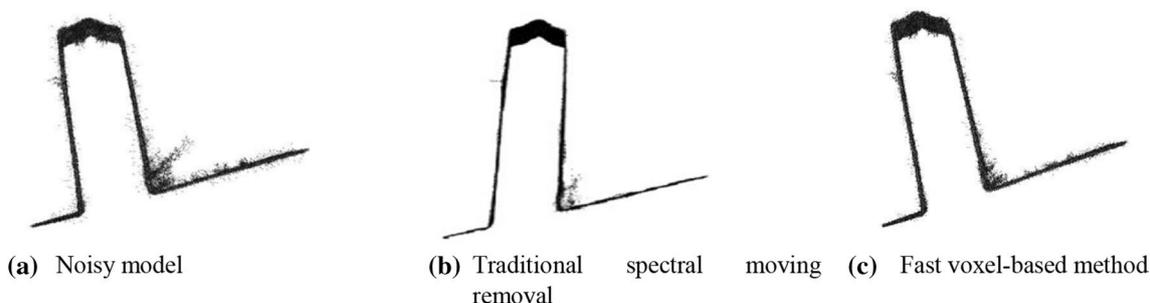


Fig. 27 Comparison between fast voxel-based method and traditional spectral moving removal

where t represents a set of parameters that are used to compute movement of each point p_i along its normal direction, n_i , during the optimization of position. γ_x is a threshold that is used to limit the position change at each surface point. $W = \{\omega_{ij}\}$ is a set of weights defined in Eq. (17) and n_{ij}^{out} represents the average of surface normal n_i^{out} and n_j^{out} . $C_{ij}^X(\dots)$ refers to a global penalty function and X is a set of position vectors for all the points.

An interior-point solver was used to solve the above convex optimization problems. Figure 26 illustrates the efficacy of this global optimization approach in the case of flat surfaces. However, l_1 minimization poses a limit on its capability in handling curved surfaces. In general, the denoised effect on curved surfaces with this method is not as good as the moving least-squares with local quadric patches.

6 Computational efficiency

Computing efficiency is an important issue in comparing different algorithms for correcting measurement outliers and noise. Several typing cases are discussed below.

6.1 Non-isolated outlier cluster

In [19], k-d tree was used as the main data structure for organizing discrete data points and guiding the surface propagation. Although the approach delivers excellent results of

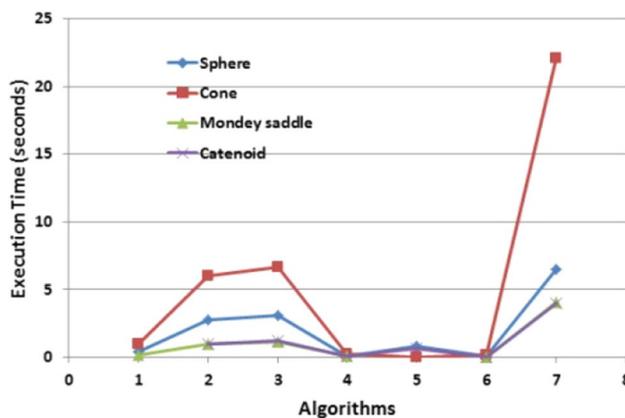


Fig. 28 Comparison of smoothing methods (Algorithm 1: mean filter; 2: median filter; 3: Gaussian filter; 4: bilateral filter; 5: mean curvature flow; 6: volume-preserving Laplacian; 7: least-squares fitting)

outlier removal, the computing cost is approximately ten times slower than the traditional methods.

To improve the computational efficiency, the domain of a point cloud model is subdivided into a limited number of voxels (volume elements) through a uniform partition in a 3D space [114]. A voxel is considered as the minimal unit of analysis and display. Specifically, all the numerical analyses are aimed at each single voxel and all the data points in a voxel are treated simultaneously as an outlier or true data point. This approach is proposed for three reasons:

Principle of locality: as for the tasks of surface outlier removal, data points in a small voxel are highly likely to have similar properties.

A reasonable partition of voxels provides us an effective way to analysis data clusters statistically and locally.

It is cheap to partition data clusters into voxels and work on them.

A surface propagation is also voxel-based, i.e., it traverses from one voxel to another. The propagation is based on two driving forces: distance and angle. Normally, the number of data points is not constant across different voxels and the maximum number of data points per voxel is controlled around 20.

Figure 27 indicates that the fast voxel-based method achieves a similar removal effect on the non-isolated outlier cluster, as compared to the traditional spectral moving removal. However, the format takes only 3.9 s, while the latter uses 34 s, leading to a ten times speed up for the fast voxel-based method.

6.2 Smoothing algorithm

Figure 28 illustrates a comparison of different smoothing methods in terms of execution time with four test models. Least-squares fitting is the most time-consuming method, while Gaussian filter ranks the second. Laplacian and mean filters are the least time-consuming methods and bilateral filter is also very fast, because only a few iterations are needed.

6.3 Genetic algorithm versus particle swarm

Figure 29 shows the comparison between genetic algorithm and swarm particle on constrained least-squares fitting of 3D laser scanning data. In general, both algorithms deliver similar accuracy in fitting error. The vertical axis in Fig. 29b refers to reduction in fitting errors of generic algorithm and swarm particle with respect to the traditional least-squares fitting. However, swarm particle is much faster than one implementation of genetic algorithm, as shown in Fig. 29a.

7 Concluding remarks and future directions

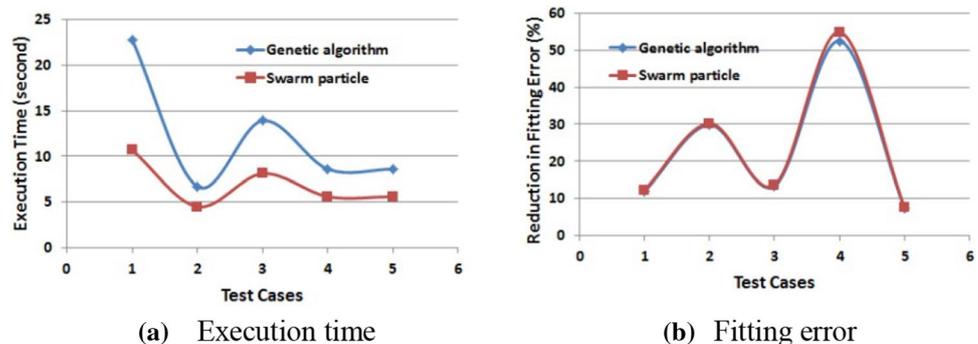
In this paper, we classify all the measurement errors into two categories: long-range outliers and short-range noise. An overview is given on existing methods for the geometric correction of discrete data points and/or the corresponding surface meshes. Since there are many excellent approaches in the past, it is impossible for the authors to present all of those methods. If one particular method is not cited, that does not mean the paper is unimportant.

For discrete long-range outliers, including isolated outlier clusters, existing methods are matured enough to handle those defects efficiently and cleanly. With respect to non-isolated outlier clusters, several methods can be used to remove most of outliers, but it is extremely difficult, if not impossible, to correct all the outliers. Normally, a second pass of smoothing algorithm is required for the complete correction of those non-isolated clusters.

As to short-range noise, signal-processing methods are easy to implement as an excellent tool for incremental smoothing. However, one problem is poor convergence, which may lead to oversmoothing. Anisotropic diffusion and bilateral filter can be executed with a small number of iterations and be used for feature enhancement. Their accuracy is, however, lower than that of piecewise least-squares fitting in dealing with sharp features. Constrained least-squares fitting can even further reduce the fitting error at geometric discontinuity, but is limited to some special geometric primitives such as line segments and circular arcs. Hybrid smoothing and global optimization are other two approaches in handling the smoothing of data points with sharp features. The accuracy of hybrid smoothing depends upon the identification of feature regions as well as which algorithm is selected for feature and non-feature regions. Some global optimization methods may be suited only to models with many planar surfaces with less desired effect on curved surfaces.

In terms of computational time, the data structure of a k-tree is less efficient than a set of pseudo voxels in removing non-isolated outlier clusters. Swarm particle is more

Fig. 29 Comparison between genetic algorithm and swarm particles



stable and faster than genetic algorithm in constrained least-squared fitting at geometric discontinuity.

With respect to future research, several possible directions are listed below:

Real-time design and modeling.

It is a desired feature to conduct real-time denoising during laser scanning on the fly. General-purpose graphics processing unit (GPGPU) and parallel computing are useful tools to achieve this goal.

Parallel smoothing and denoising.

Design of parallel denoising or smoothing algorithms is desired for big-data models.

Non-isolated outlier clusters at geometric discontinuity.

There is still a demand on good mechanisms or principles to handle the mingling between non-isolated outlier clusters and geometric discontinuity (sharp edges and corners).

Incomplete and dynamic point cloud.

Develop effective denoising algorithms for incomplete and dynamic point cloud data as in autonomous driving.

Acknowledgements This article was supported in part by U.S. National Science Foundation DMI-0514900, CMMI-0721625, ECCS-1039563, and IIP-1445355.

References

- Klinger P, Veit K, Hausler G, Karbacher S, Laboureaux X (2003) Optical 3D sensors for real applications—potentials and limits. Proceedings of the 8th International Rendez-vous for 3D Digitization and Modeling Professionals. April 23–24, Paris, France, 1–5
- Bernardini F, Bajaj CL, Chen J, Schikore DR (1999) Automatic reconstruction of 3D CAD models from digital scans. *Int J Comput Geometry Appl* 9(4 & 5):327–369
- Varady T, Martin RR, Cox J (1997) Reverse engineering of geometric models—an introduction. *Comput Aided Des* 29(4):255–269
- Dorsch RG, Hausler G, Herrmann JM (1994) Laser triangulation: fundamental uncertainty in distance measurement. *Appl Opt* 33(7):1306–1314
- Chan HL, Mitchell JP, Spence AD, Sklad MP, Capson DW (2005) Laser digitizer/stereo vision methods for simultaneous measurement/analysis of sheet metal forming strain/geometry. ASME 2005 Int Mech Eng Congress Exposition. 237–246
- Shen J, Yoon D, Liu Y, Orady E (2007) A planar measurement error synthesizer for touch probes and laser sensors. Technical Report 2007–1, University of Michigan–Dearborn
- Rioux M, Bechthold G, Taylor D, Duggan M (1987) Design of a large depth of view three-dimensional camera for robot vision. *Opt Eng* 26(12):1245–1250
- Baribeau R, Rioux M (1991) Influence of speckle on laser range finders. *Appl Opt* 30(20):2873–2878
- Tamura S, Kim EK, Close R, Sato Y (1994) Error correction in laser scanner three-dimensional measurement by two-axis model and coarse-fine parameter search. *Pattern Recogn* 27(3):331–338
- Smith KB, Zheng YF (1998) Accuracy analysis of point laser triangulation probes using simulation. *ASME J Manuf Sci Eng* 120(4):736–745
- Feng H, Liu Y, Xi F (2001) Analysis of digitizing errors of a laser scanning system. *Precision Eng* 25(3):185–191
- Curless B, Levoy M (1995) Better optical triangulation through spacetime analysis. Proceedings of IEEE International Conference on Computer Vision '95, 987–994
- Cederberg P, Olsson M, Bolmsjo G (2002) Virtual triangulation sensor development, behavior simulation and CAR integration applied to robotic arc-welding. *J Intell Rob Syst* 35(4):365–379
- Forest J, Salvi J, Cabruja E, Pous C (2004) Laser stripe peak detector for 3D scanners. A FIR filter approach. Proceedings of the 17th International Conference on Pattern Recognition. 3:646–649
- Xie H, McDonnell KT, Qin H (2004) Surface reconstruction of noisy and defective data sets. *IEEE Visualization* 259–266
- Kolluri R, Shewchuk JR, O'Brien JF (2004) Spectral surface reconstruction from noisy point clouds. Symposium on Geometry Processing. Nice, 11–21
- Weyrich T, Pauly M, Keiser R, Heinzle S, Scandella S, Gross M (2004) Post-processing of scanned 3D surface data. Proceedings of Eurographics Symposium on Point-Based Graphics. 85–94
- Schall O, Belyaev A, Seidel H (2005) Robust filtering of noisy scattered point data. Eurographics symposium on point-based graphics
- Shen J, Yoon D, Shehu D, Chang SY (2009) Spectral moving removal of non-isolated surface outlier clusters. *Comput Aided Des* 41(4):256–267
- Lloyd SP (1982) Least squares quantization in PCM. *IEEE Trans Inf Theory* 28(2):129–137
- Wang Y, Feng HY (2015) Outlier detection for scanned point cloud using majority voting. *Comput Aided Des* 62(C):31–43
- Fabio R (2003) From point cloud to surface: the modeling and visualization problem. *Int Arch Photogrammetry Rem Sens Spatial Inf Sci* 34(5):W10
- Ma J, Chen JS, Feng H, Wang L (2016) Automatic construction of watertight manifold triangle meshes from scanned point clouds using matched umbrella facets. Proceedings of 2016 CAD Conference and Exhibition. June 27–29, Vancouver, Canada, 1–6
- Digne J, Cohen-Steiner D, Alliez P, De Goes F, Desbrun M (2014) Feature-preserving surface reconstruction and simplification from defect-laden point sets. *J Math Imaging Vision* 48(2):369–382
- Taubin GA (1995) Signal Processing approach to fair surface design. Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. August 6–11, Los Angeles, California, USA, 351–358
- Taubin G (2001) Linear anisotropic mesh filtering. Technical Report RC22213. IBM Research Division
- Vollmer J, Mencl R, Muller H (1999) Improved Laplacian smoothing of noisy surface meshes. *Computer Graphics Forum* 18(3):131–138
- Kobbelt L, Campagna S, Vorsatz J, Seidel H (1998) Interactive Multi-resolution Modeling on Arbitrary Meshes. Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. July 19–24, Orlando, Florida, USA, 105–114
- Belyaev A, Ohtake Y (2003) A comparison of mesh smoothing methods. Israel-Korea Bi-National Conference on Geometric Modeling and Comput Graph. February 12–14, Tel-Aviv, Israel, 83–87
- Yagou H, Ohtake Y, Belyaev A (2002) Mesh smoothing via mean and median filtering applied to face normals. *Geometric Modeling Processing*. July 10–12, RIKEN, Saitama, Japan, 124–131
- Shen Y, Barner KE (2004) Fuzzy vector median-based surface smoothing. *IEEE Trans Visual Comput Gr* 10(3):266–277
- Yagou H, Ohtake Y, Belyaev A (2003) Mesh denoising via iterative alpha-trimming and non-linear diffusion of normals with

- automatic thresholding. Proceedings of Computer Graphics International 2003. July 9–11, Tokyo, Japan, 28–33
33. Ohtake Y, Belyaev A, Seidel H (2002) Mesh smoothing by adaptive and anisotropic Gaussian filter. Vision, modeling, and visualization 2002. November 20–22, Erlangen, 203–210
 34. Oliensis J (1993) Local reproducible smoothing without shrinkage. IEEE Trans Pattern Anal Mach Intell 15(3):307–312
 35. Peng J, Strela V, Zorin D (2001) A simple algorithm for surface denoising. IEEE Vis. October 21–26, San Diego, 107–112
 36. Alexa M (2002) Wiener filtering of meshes. Proceedings of Shape Modeling International. May 17–22, Banff, Alberta, Canada, 51–57
 37. Pauly M, Gross M (2001) Spectral processing of point-sampled geometry. Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. August 12–17, Los Angeles, California, USA, 379–386
 38. Guskov I, Sweldens W, Schroder P (1999) Multiresolution signal processing for meshes. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. August 8–13, Los Angeles, California, USA, 325–334
 39. Schall O, Belyaev A, Seidel H (2007) Error-guided adaptive Fourier-based surface reconstruction. Comput Aided Des 39:421–426
 40. Kazhdan MM (2005) Reconstruction of solid models from oriented point sets. Proceedings of Symposium on Geometry Processing. 73–82
 41. Shen J, Maxim B, Akingbehin K (2005) Accurate correction of surface noises of polygonal meshes. Int J Numer Methods Eng 64(12):1678–1698
 42. Desbrun M, Meyer M, Schroder P, Barr AH (1999) Implicit fairing of irregular meshes using diffusion and curvature flow. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. August 8–13, Los Angeles, USA, 317–324
 43. Clarenz U, Diewald U, Rumpf M (2000) Anisotropic geometric diffusion in surface processing. Proceedings of IEEE Visualization. October 8–13, Salt Lake City, USA, 397–405
 44. Clarenz U, Rumpf M, Telea A (2004) Robust feature detection and local classification for surfaces based on moment analysis. IEEE Trans Visual Comput Gr 10(5):516–524
 45. Desbrun M, Meyer M, Schroder P, Barr AH (2000) Anisotropic feature-preserving denoising of height fields and bivariate data. Gr Interface. Montreal, 145–152
 46. Clarenz U, Dziuk G, Rumpf M (2003) On generalized mean curvature flow in surface processing. In: Karcher H, Hildebrandt S (eds) Springer, pp 217–248
 47. Sun Y, Page DL, Paik JK, Koschan A, Abidi MA (2002) Triangle mesh-based surface modeling using adaptive smoothing and implicit texture integration. Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission. June 19–21, Padova, Italy, 588–597
 48. Schneider R, Kobbelt L (2000) Generating fair meshes with gl boundary conditions. Proceedings of Geometric Modeling and Processing. April 10–12, Hong Kong, China, 251–261
 49. Whittaker RT (1998) A level-set approach to 3D reconstruction from range data. Int J Comput Vision 29(3):203–231
 50. Zhao HK, Osher S, Merriman B, Kang M (2000) Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. Comput Vis Image Underst 80(3):295–314
 51. Gomes J, Faugeras OD (2000) Level sets and distance functions. Proceedings of Sixth European Conference on Computer Vision. June 26–July 1, Trinity College Dublin, Ireland, 588–602
 52. Cao F. Geometric curve evolution and image processing. Springer:2003
 53. Sapiro G, Tannenbaum A (1993) Affine invariant scale-space. Int J Comput Vision 11(1):25–44
 54. Moisan L (1998) Affine plane curve evolution: a fully consistent scheme. IEEE Trans Image Process 7(3):411–420
 55. Lounsbery M, DeRose T, Warren J (1997) Multiresolution analysis for surfaces of arbitrary topological type. ACM Trans Gr 16(1):34–73
 56. Lee A, Moreton H, Hoppe H (2000) Displaced subdivision surfaces. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. July, New Orleans, Louisiana, USA, 85–94
 57. Razdan A, Mongkolnam P, Farin G (2003) Reverse engineering using a subdivision surface scheme. Proceedings of 8th 3D Modeling Symposium. April 23–24, Paris, France, 10
 58. Loop C (1987) Smooth subdivision surfaces based on triangles. M. Sc. The University of Utah
 59. Catmull E, Clark J (1978) Recursively generated B-spline surfaces on arbitrary topological meshes. Comput Aided Des 10(6):350–355
 60. Garland M, Heckbert PS (1997) Surface simplification using quadric error metrics. Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. August 3–8, Los Angeles, California, USA, 209–216
 61. Kanai T (2001) MeshToSS: converting subdivision surfaces from dense meshes. Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV-01). November 21–23, Stuttgart, Germany, 325–332
 62. Sadeghi J, Samavati FF (2011) Smooth reverse loop and Catmull-Clark subdivision. Graph Models 73(5):202–217
 63. Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. Math Comput 37(155):141–158
 64. Bentley JL (1990) K-d trees for semidynamic point sets. Proceedings of 6th Annual Symposium on Computational Geometry, 187–197
 65. Hoppe H, De Rose T, Duchamp T, MacDonald J, Stuetzle W (1993) Mesh optimization. ACM SIGGRAPH Comput Gr Proc 27:19–26
 66. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C: the art of scientific computing. Cambridge University Press, Cambridge
 67. Beatson RK, Bui HQ (2003) Mollification formulas and implicit smoothing. Research Report UCDMS 2003/19. Dept. of Mathematics and Statistics, Univ. of Canterbury
 68. Carr JC, Beatson RK, McCallum BC, Fright WR, McLennan TJ, Mitchell TJ (2003) Smooth surface reconstruction from noisy range data. Proceedings of Graphite 2003. February 11–14, Melbourne, Australia, 119–126
 69. Ohtake Y, Belyaev A, Seidel H Multi-level partition of unity implicits. July 27–31, San Diego, 2003; 27–31
 70. Ohtake Y, Belyaev A, Seidel H (2004) 3D scattered data approximation with adaptive compactly supported radial basis functions. Proceedings of Shape Modeling International 2004. June 7–9, Genova, Italy, 31–39
 71. Savchenko VV, Pasko AA, Okunev OG, Kunii TL (1995) Function representation of solids reconstructed from scattered surface points and contours. Comput Gr Forum 14(4):181–188
 72. Turk G, O'Brien JF (1999) Variational implicit surfaces. Technical Report GIT-GVU-99-15. Georgia Institute of Technology
 73. Morse BS, Yoo TS, Rheingans P, Chen DT, Subramanian KR (2001) Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. Proceedings of International Conference on Shape Modeling and Applications '01. May 7–11, Genova, Italy, IEEE Computer Society Press, 89–98
 74. Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, Evans TR (2001) Reconstruction and representation of

- 3D objects with radial basis functions. Proceedings of ACM SIG-GRAPH. 67–76
75. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
 76. Meyer M, Desbrun M, Schroder P, Barr AH (2003) Discrete differential-geometry operators for triangulated 2-manifolds. In: *visualization and mathematics III*. Springer, Heidelberg, pp 35–57
 77. Zhang H, Fiume EL (2002) Mesh smoothing with shape or feature preservation. In: Vince J, Earnshaw R (eds) *Advanced in modeling, animation, and rendering*. Springer, pp 167–182
 78. Ohtake Y, Belyaev A, Bogaevski IA (2001) Mesh regularization and adaptive smoothing. *Comput Aided Des* 33(11):789–800
 79. Hildebrandt K, Polthier K (2004) Anisotropic filtering of non-linear surface features. *Comput Gr Forum* 23(3):391–400
 80. Tasdizen T, Whitaker RT, Burchard P, Osher S (2002) Anisotropic geometric diffusion in surface processing. *IEEE Visualization*. October 27–November 01, Boston, 2002; 125–132
 81. Bajaj C, Xu G (2003) Anisotropic diffusion of subdivision surfaces and functions on surfaces. *ACM Trans Gr* 22(1):4–32
 82. Loop C, DeRose T (1990) Generalized B-spline surfaces of arbitrary topology. Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques. August, Dallas, Texas, USA, 24(4):347–356
 83. Ma W, Kruth JP (1995) Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Comput Aided Des* 27(9):663–675
 84. Eck M, Hoppe H (1996) Automatic reconstruction of B-spline surfaces of arbitrary topological type. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. August 4–9 ECK95, New Orleans, Louisiana, USA, 325–334
 85. Forsey DR, Bartels RH (1988) Hierarchical B-spline refinement. Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques. August 1–5, Atlanta, Georgia, USA, 22(4):205–212
 86. Forsey DR, Bartels RH (1995) Surface fitting with hierarchical splines. *ACM Trans Gr* 14(2):134–161
 87. Krishnamurthy V, Levoy M (1996) Fitting smooth surfaces to dense polygon meshes. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. August 4–9, New Orleans, Louisiana, USA, 313–324
 88. Pfeifle R, Seidel H (1996) Fitting triangular B-splines to functional scattered data. Proceedings of Graphics Interface '95. May 17–19, Quebec, Quebec, Canada, 26–33
 89. Sarkar B, Menq CH (1991) Smooth-surface approximation and reverse engineering. *Comput Aided Des* 23(9):623–628
 90. Zhang C, Zhang P, Cheng F (2001) Fairing spline curves and surfaces by minimizing energy. *Comput Aided Des* 33(13):913–923
 91. Sapidis N, Farin G (1990) Automatic fairing algorithm for B-spline curves. *Comput Aided Des* 22(2):121–129
 92. Kjellander JAP (1983) Smoothing of bicubic parametric splines. *Comput Aided Des* 15(5):288–293
 93. Poliakov JF, Wong YK, Thomas PD (1999) An automatic curve fairing algorithm for cubic B-spline curves. *J Comput Appl Math* 102(1):73–85
 94. Jones TR, Durant F, Desbrun M (2003) Non-iterative, feature-preserving mesh smoothing. Proceedings of the 30th Annual Conference on Computer Graphics and Interactive Techniques. July 27–31, San Diego, California, USA, 943–949
 95. Fleishman S, Drori I, Cohen-Or D (2003) Bilateral mesh denoising. Proceedings of the 30th Annual Conference on Computer Graphics and Interactive Techniques. July 27–31, San Diego, California, USA, 950–953
 96. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. Proceedings of IEEE ICCV. January 4–7, Bombay, India, 836–846
 97. Choudhury P, Tumblin J (2003) The trilateral filter for high contrast images and meshes. Proceedings of the Eurographics Symposium on Rendering. Leuven, Belgium, 186–196
 98. Fleishman S, Cohen-Or D, Silva C (2005) Robust moving least-squares fitting with sharp features. *ACM Trans Gr* 24(3):544–552
 99. Wang J, Yu Z, Zhu W, Cao J (2013) Feature-preserving surface reconstruction from unoriented, noisy point data. *Comput Gr Forum* 32(1):164–176
 100. Reuter P, Joyot P, Trunzler J, Boubekur T, Schlick C (2005) Surface reconstruction with enriched reproducing kernel particle approximation. Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics, 79–87
 101. Lipman Y, Cohen OR, Levin D (2007) Data-dependent MLS for faithful surface approximation. Proceedings of the Fifty Eurographics Symposium on Geometry Processing. 59–67
 102. Shi X, Shen J (2016) Genetic search for optimally-constrained multiple-line fitting of discrete data points. *Appl Soft Comput* 40(2):236–251
 103. Beyer HG (2001) *The theory of evolutionary strategies*. Heidelberg
 104. Goldberg DE, Korb B, Deb K (1989) Messy genetic algorithms: motivation, analysis, and first results. *Complex Syst* 3(5):493–530
 105. Holland JH (1975) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U. Michigan Press, Ann Arbor
 106. Beyer HG, Schwefel HP (2002) Evolution strategies—a comprehensive introduction. *Nat Comput* 1(1):3–52
 107. Fogel DB (1994) An introduction to simulated evolutionary optimization. *IEEE Trans Neural Networks* 5(1):3–14
 108. Chellapilla K (1998) Combining mutation operators in evolutionary programming. *IEEE Trans Evol Comput* 2(3):91–96
 109. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102
 110. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks. Perth, WA, 4:1942–1948
 111. Daniels JI, Ha LK, Ochotta T, Silva CT (2007) Robust smooth feature extraction from point clouds. *IEEE International Conference on Shape Modeling and Applications*. 123–136
 112. Jenke P, Kruckeberg B, Straber W (2008) Surface reconstruction from fitted shape primitives. Proceedings of Vision Modeling and Visualization. 31–40
 113. Avron H, Sharf A, Greif C, Cohen-Or D (2010) L1-sparse reconstruction of sharp point set surfaces. *ACM Trans Graphics* 29(5):135-1–135-20
 114. Chen Y, Chen H, Shen J (2016) Fast voxel-based surface propagation method for outlier removal. 2016 CAD Conference and Exhibition. Vancouver, Canada, 1–5