Visual Comput (2006) 22: 147–157
DOI 10.1007/s00371-006-0372-0

ORIGINAL ARTICLE

Guofei Hu
Qunsheng Peng
A. Robin Forrest

Mean shift denoising of point-sampled surfaces*

Published online: 31 January 2006

G. Hu[1] (✉) · Q. Peng
State Key Lab. of CAD & CG,
Zhejiang University, Hangzhou 310027,
P.R. China
guofei.hu@ge.com

A.R. Forrest
School of Computing Sciences, University
of East Anglia, Norwich, NR4 7TJ, UK

Abstract This paper presents an anisotropic denoising/smoothing algorithm for point-sampled surfaces. Motivated by the impressive results of mean shift filtering on image denoising, we extend the concept to 3D surface smoothing by taking the vertex normal and the curvature as the range component and the vertex position as the spatial component. Then the local mode of each vertex on point-based surfaces is computed by a 3D mean shift procedure dependent on local neighborhoods that are adaptively obtained by a kdtree data structure. Clustering pieces of point-based surfaces of similar local mode provides a meaningful model segmentation. Based on the adaptively clustered neighbors, we finally apply a trilateral point filtering scheme that adjusts the position of sample points along their normal directions to successfully reduce noise from point-sampled surfaces while preserving geometric features.

Keywords Point-sampled surface · Denoising · Mean shift filter · PCA · Surface variation

1 Introduction

Point-sampled models are normally generated by sampling the boundary surface of physical 3D objects with range sensing devices. While adhering to this kind of model, the objects are represented simply by the geometry of the sample points without additional topological connectivity such as vertex valence or adjacency. Due to a variety of physical factors and limitations of the acquisition procedure, these raw point models obtained are prone to various kinds of undesirable noise and distortions. It is then necessary to denoise the data to improve the model's quality before carrying on further digital geometry processing, such as reconstruction, parameterization, morphing, etc. As a straightforward representation form for highly complex sculptured objects, the point-sampled geometry is obviously processed with less overhead in computation time and memory costs than triangular meshes. The disadvantage of denoising a mesh is that the connectivity information that implicitly defines the surface topology needs to be maintained all the time: once a vertex of the mesh is removed, its linked edges and faces must be removed immediately. Thus, denoising point-based models is preferred to denoising a mesh. The purpose of 3D denoising is to remove the effects caused by isolated noises from the derived surface whilst preserving the appearance of geometrically sharp features and minimizing distortion locally or globally. This still remains a challenging problem in computer graphics.

In recent years, a variety of point-based denoising and smoothing techniques have been presented, and they can be classified into the following three groups:

- Generalizing techniques of FFT and spectral filters in image setting to point-based surfaces via local sur-

[1] *Present address*: GE Global Research, 1800 CaiLun Rd, Zhangjiang Hi-Tech Park, Shanghai 201203, China

* This work is supported by the National Natural Science Foundation of China (grant No.60103017) and the National Key Basic Research and Development Program (grant No.2002CB312101). A.R. Forrest is funded by The Royal Society-NNSFC China-UK Project 13230/Q807.

face parameterization and regular resampling techniques [22].
- Interpolating or approximating raw point clouds with smooth surfaces such as NURBS patches, optimized meshes, bivariate polynomial surfaces [2], or other higher order surfaces [30], local parametric surfaces [24], global implicit surfaces [6], extremal surfaces [3], etc.
- Directly extending 2D filters into 3D settings by applying local position estimating techniques iteratively or non-iteratively, isotropically or anisotropically, based on statistics, differential geometry theory, and approximation theory, and so on.

The third category is more attractive because it is simple and straightforward. Techniques for image denoising such as Laplacian, nonlinear diffusion, Wiener filtering and bilateral filtering commonly serve as foundations for 3D surface denoising and smoothing algorithms.

The mean shift filter, introduced first by Comaniciu and Meer [11], is a dynamic nonlinear filter linked to the bilateral filter, and achieves a high quality discontinuity-preserving filtering by identifying local modes in the joint spatial-range domain. It is a general nonparametric technique for the analysis of a complex multimodal feature space and the delineation of arbitrarily shaped clusters [12], and it has a wide variety of applications for some computer vision tasks such as image filtering [5, 11], image segmentation [8], non-rigid object tracking [13], and 3D video segmentation [29]. Inspired by the mean shift filtering of images, we here formulate a 3D mean shift filter to denoise point-sampled geometry with geometric feature-preservation.

2 Related work

In the context of mesh denoising/smoothing/fairing, many techniques adopt the concept of constrained energy minimization using either the domain of direct neighbors or the global mesh, such as the membrane energy and thin-plate energy of a parametric surface. Apart from these surface variational methods, researchers have also developed various efficient methods to remove high frequency information from meshes taking into consideration the problems of volume-shrinkage, vertex drifting and feature-preservation. These methods include $\lambda|\mu$ filters [27], the HC algorithm [28], mean curvature flow [14], the anisotropic geometric diffusion [9], PDE-based diffusion [4], Wiener filtering [25], bilateral mesh filtering [15, 19, 20], and so on. Most of them could be directly transferred to point-based surfaces. For pure point-based geometry, many techniques have been proposed recently. To restore the object surface in the presence of blur and noise, Pauly et al. [22] applied a least-squares optimal filter (Wiener filter) for point-sampled geometry after local parameterization and regular resampling. A real 3D Wiener filter was introduced by Alexa [1] via an adaption of Wiener interpolation for mesh geometry, and could easily be formulated for point-based geometry. The approach proves to be a powerful way of general filtering for manifold geometry. Lange and Polthier [21] present a new method for anisotropic fairing of a point sampled surface based on the concept of anisotropic geometric mean curvature flow. Clarenz et al [10] present a PDE-based surface fairing application within the framework of processing point-based surfaces via PDEs. Hu and Peng [17] propose an anisotropic denoising algorithm for point-sampled models taking into account the local spatial geometry as well as normal and curvature information. All of the above methods are somewhat feature-preserving, however, they all need to set a user-defined parameter (the k nearest neighbors) for processing, and the experimental value of k must be tailored, for instance between 6 and 20, to ensure stable computations and to avoid excessive smoothing of the density estimation.

2.1 Contributions

Our main contribution in this paper is to introduce a mean shift-based anisotropic denoising algorithm for point-sampled surfaces. We extend mean shift filtering in the image setting to 3D surface smoothing by taking into account the vertex normal and curvature as the range component, while taking position as the spatial component. We find the local mode of each sample point on point-based surfaces via a 3D mean shift procedure. By clustering adjacent sample points of similar local modes, we provide a meaningful segmentation of the point model. The neighbors of each sample point are collected under spatial and range constraints. Finally, we introduce a trilateral point filtering algorithm, which successfully performs noise removal whilst preserving geometric features, such as salient ridge and ravine structures, by adjusting the position of each sample point along its normal direction.

3 Mean shift denoising

3.1 Mean shift procedure

In the setting of 3D geometry, the spatial and range domains are slightly different from those of images. As shown in Fig. 1, for instance, the 3D surface is parameterized onto an 2D plane domain (Fig. 1a), and we usually regard the 3D position of a vertex as spatial information, while we regard the normal and curvature of the local surface as range information or feature space information. Regardless of parameterization, we here extend the mean shift technique directly to 3D filtering of point-based geometry in the 3D domain.

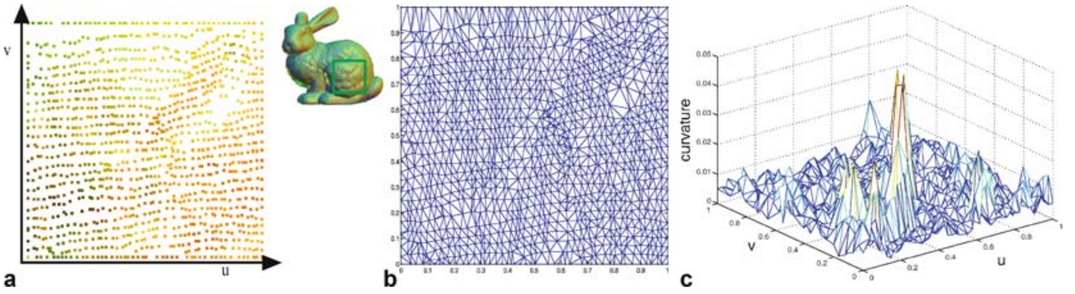

Fig. 1a–c. Feature space of 3D geometry:. **a** planar parameterization of the selected region; **b** Delaunay triangulation of the point cloud (**a**); **c** feature space (distribution of curvature information) of the selected region

Assume now that the data points $\boldsymbol{p}_i$ are the *generalized points* of the input raw point model $P \subset R^3$, and the vector components of $\boldsymbol{p}_i$ contain both the spatial position information $\boldsymbol{v}_i = (x_i, y_i, z_i)$ and range information including the normal vector $\boldsymbol{n}_i$ and the mean curvature H_i of vertices, written as

$$\boldsymbol{p}_i = (\boldsymbol{v}_i, \boldsymbol{n}_i, H_i), \tag{1}$$

with $i = 1, 2, \dots, n$, and n being the number of points in P. Here the dimension of vector $\boldsymbol{p}_i$ is 7. The k nearest neighboring points of generalized points $\boldsymbol{p}_i$ are denoted by $N(\boldsymbol{p}_i) = \{\boldsymbol{q}_{i1}, \boldsymbol{q}_{i2}, \dots, \boldsymbol{q}_{ik}\}$.

Thus, *the mean shift vector* of $\boldsymbol{p}_i$ can be expressed as

$$M_v(\boldsymbol{p}_i) = \frac{\sum_{j=1}^{k} g\left(\left\|\boldsymbol{p}_i^r - \boldsymbol{q}_{ij}^r\right\|\right)(\boldsymbol{q}_{ij} - M(\boldsymbol{p}_i))}{\sum_{j=1}^{k} g\left(\left\|\boldsymbol{p}_i^r - \boldsymbol{q}_{ij}^r\right\|\right)}, \tag{2}$$

where $g(\cdot)$ could be either a Gaussian kernel or an Epanechnikov kernel; $\boldsymbol{p}_i^r = (\boldsymbol{n}_i, H_i)$ is the range part of $\boldsymbol{p}_i$; and $M(\boldsymbol{p}_i)$ is called *the mean shift point* associated with $\boldsymbol{p}_i$, and $M(\boldsymbol{p}_i)$ could be initialized to coincide with $\boldsymbol{p}_i$; $M_v(\boldsymbol{p}_i)$ is the mean shift vector associated with $M(\boldsymbol{p}_i)$. Then we define the *mean shift procedure* as the repeated movement of data points to the sample means, written as

$$M(\boldsymbol{p}_i) := M(\boldsymbol{p}_i) + M_v(\boldsymbol{p}_i). \tag{3}$$

From Eq. 3, it is well-known that the mean shift procedure is an iterative algorithm that actually moves a sample point along the direction of the maximum increase in the density gradient with the simultaneous processing of both the spatial and the range domains. The convergence of lattices in an image setting has been proven in [12] and has been extended to arbitrary dimensions of the data source, consequently it is also convergent when the spatial dimension is 3. By running the procedure for all $i = 1, 2, \dots, \infty$, each data point iterates to a local mode in the joint spatial range domain, and the mean shift procedure has provided a stable local mode detection for the point-sampled model (see Fig. 2b).

3.2 Range component estimation

Before introducing the algorithm, we describe how to initialize the range component of generalized points. There exist various methods to estimate normals and curvatures of the point-sampled geometry [16, 18]. An obvious idea for estimating the normal at $\boldsymbol{v}_i$ is to find a plane approximating the unorganized spatial neighboring points $N(\boldsymbol{v}_i) = \{\boldsymbol{q}_{ij}^s\}$ by least-square fitting. Assume that the plane is $D : \boldsymbol{r}_i \cdot \boldsymbol{n}_i - d = 0$. $\boldsymbol{n}_i$ is estimated by solving the following least-square optimiza-

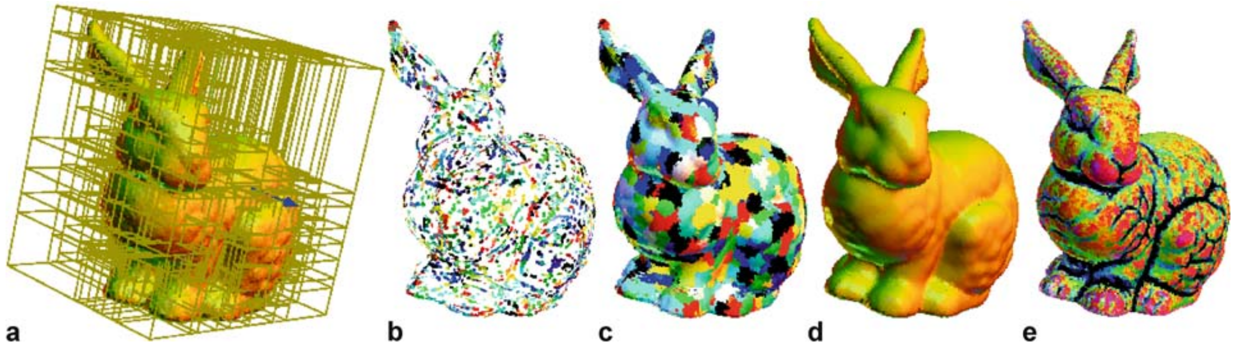

Fig. 2a–e. Processing pipeline: **a** noisy bunny with kdtree partitioning; **b** point set of local modes; **c** segmentation with consistent octree clustering; **d** denoising result of our algorithm; **e** mean curvature evaluation of the smoothed bunny

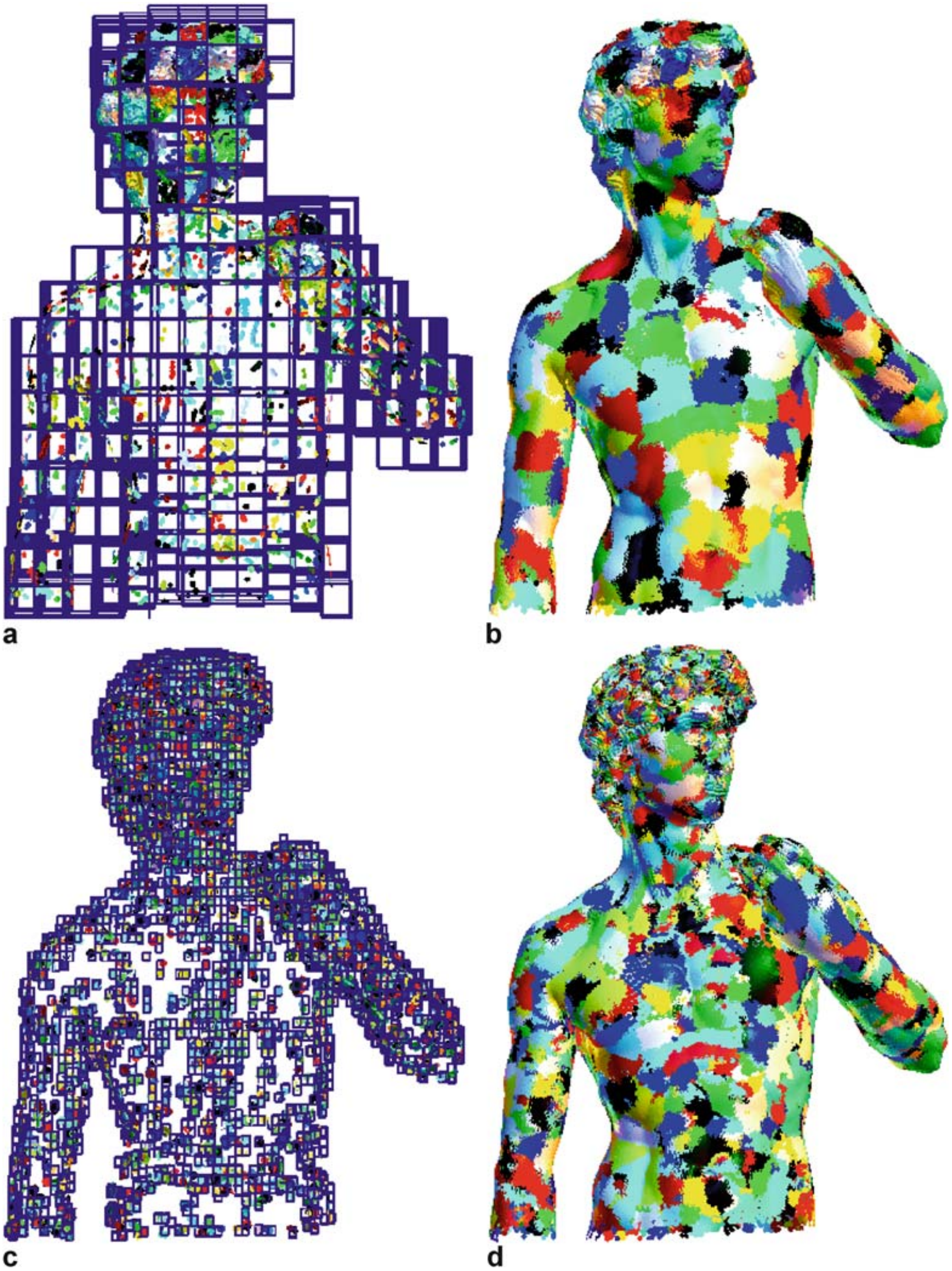

Fig. 3. Octree-based clustering with different bounding cell size r for the same set of local modes; the models are rendered with a point splatting renderer

tion:

$$\min\Big(\sum_{j=0}^{k}\big(\boldsymbol{q}^{s}_{i,j}\cdot\boldsymbol{n}_i-d\big)^2\Big), \tag{4}$$

where k is the number of neighboring points around $\boldsymbol{v}_i$. A 3×3 covariance matrix could be defined based on the theory of principal component analysis (PCA):

$$C_i=\begin{bmatrix}\boldsymbol{q}^{s}_{i,1}-\overline{\boldsymbol{v}}_i\\ \boldsymbol{q}^{s}_{i,2}-\overline{\boldsymbol{v}}_i\\ \cdots\\ \boldsymbol{q}^{s}_{i,k}-\overline{\boldsymbol{v}}_i\end{bmatrix}^{T}\begin{bmatrix}\boldsymbol{q}^{s}_{i,1}-\overline{\boldsymbol{v}}_i\\ \boldsymbol{q}^{s}_{i,2}-\overline{\boldsymbol{v}}_i\\ \cdots\\ \boldsymbol{q}^{s}_{i,k}-\overline{\boldsymbol{v}}_i\end{bmatrix}, \tag{5}$$

where C_i is a symmetric positive semi-definite matrix, and $\overline{\boldsymbol{v}}_i$ is the centroid of $N(\boldsymbol{v}_i)$. The normal of $\boldsymbol{v}_i$ is chosen to be the unit vector $\boldsymbol{e}_{i,1}$, which corresponds to the minimal eigenvalue of C_i.

$$\boldsymbol{n}_i = \boldsymbol{e}_{i,1}. \tag{6}$$

In [23] Pauly et al. showed that surface variation is closely related to mean curvature, and here the curvature on $\boldsymbol{v}_i$ is taken as the surface variation,

$$H_i = \lambda_{i,1}/(\lambda_{i,1}+\lambda_{i,2}+\lambda_{i,3}), \tag{7}$$

where $\lambda_{i,j}$, $j = 1, 2, 3$ are eigenvalues of C_i and satisfy $\lambda_{i,1} \leq \lambda_{i,2} \leq \lambda_{i,3}$.

3.3 Algorithm

Features and noise are both local geometric details in point-based geometry. It is not trivial to distinguish between them. In surface theory, it is well-known that letting the surface flow in the normal direction with the speed taken as the mean curvature of the surface makes the surface smoother and smoother, finally approximating to a minimal surface [26]. Desbrun et al. [14] formulated discrete mean curvature flow of meshes based on level set theory. For point-based representation, the mean curvature flow definitely prevents vertex-drift in point models, but unfortunately, it is not good at geometric feature-preservation, and even leads to oversmoothing. The bilateral mesh filter presented by Fleishman et al. [15] performs well to some extent in discontinuity-preservation. However it actually uses a static window/kernel in the two domains, and its parameters for spatial Gaussian, range Gaussian, kernel size and number of iterations are often user-defined. In order to obtain a dynamic window, we generalize the 3D mean shift filter, which has a more powerful adaptation to the local geometry of the point model.

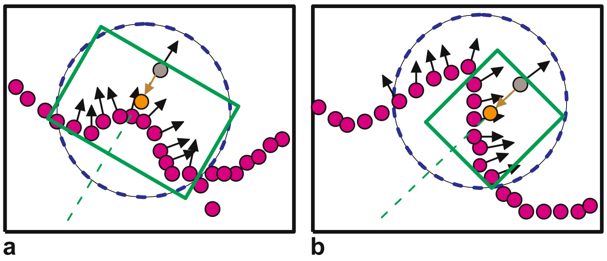

Fig. 4. Vertex position estimation according to its influence region. The black arrows, blue dotted circles and green boxes, respectively, indicate vertex normals, spatial k-nearest neighbors and the adaptive neighbors called influence region. Grey points adhered with noise are moved to new positions of orange points where the noise is diffused

The detailed mean shift denoising algorithm consists of four processing stages.

Step 1. Initialization. Construct a kdtree structure for the point model and search neighbors $\{\boldsymbol{q}^s_{ij}\}$ for each $\boldsymbol{v}_i$ in the spatial domain, then initialize range component $\boldsymbol{p}^r_i$ by principal component analysis of the spatial neighbors $\{\boldsymbol{q}^s_{ij}\}$. The range bandwidths $h^r = (h^r_1, h^r_2)$ are usually defined as positive values related to normal and curvature, respectively; for formulation convenience, we write h^r directly. Let L^i_∞ be the infinity norm of the difference between range vectors in the spatial neighborhood of $\boldsymbol{p}_i$. We take h^r as the mean of L^i_∞ for all points. The range neighborhood $\{\boldsymbol{q}^r_{ij}\}$ of $\boldsymbol{p}_i$ is also a neighboring point set that satisfies the following constraint:

$$\frac{\left\|\boldsymbol{p}^r_i - \boldsymbol{q}^r_{ij}\right\|}{h^r} < 1, \tag{8}$$

where $\|\cdot\|$ is the Euclidean norm. Finally, update the neighbors of $\boldsymbol{p}_i$ as the intersection of point sets of the spatial neighbors and the range neighbors

$$\{\boldsymbol{q}_{ij}\} = \left\{\boldsymbol{q}^s_{ij}\right\} \cap \left\{\boldsymbol{q}^r_{ij}\right\}. \tag{9}$$

Step 2. Mean shift procedure. Repeat the mean shift procedure discussed in Sect. 3.1 until convergence. In our implementation, the convergence condition is that $M_v(\boldsymbol{p}_i)$ is less than a specified value ε. The convergence point is called the local mode, denoted by

$$M_{\text{conv}}(\boldsymbol{p}_i) = (\boldsymbol{v}_{i,\text{conv}}, \boldsymbol{n}_{i,\text{conv}}, H_{i,\text{conv}}) \tag{10}$$

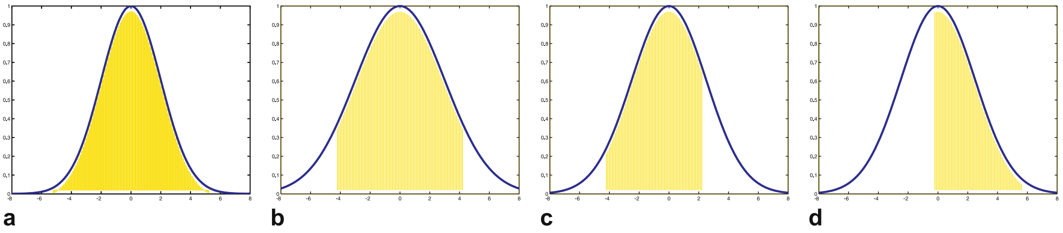

Fig. 5a–d. Uniform neighbors vs. adaptive neighbors: the blue curve is the curve of the Gaussian kernel function and yellow shadows indicate the influence regions. **a** and **b** are both uniform depending on isotropically selected neighbors, and σ is 2.0 and 3.0, respectively; the neighbors in **c** and **d** are anisotropic with $\sigma = 2.5$

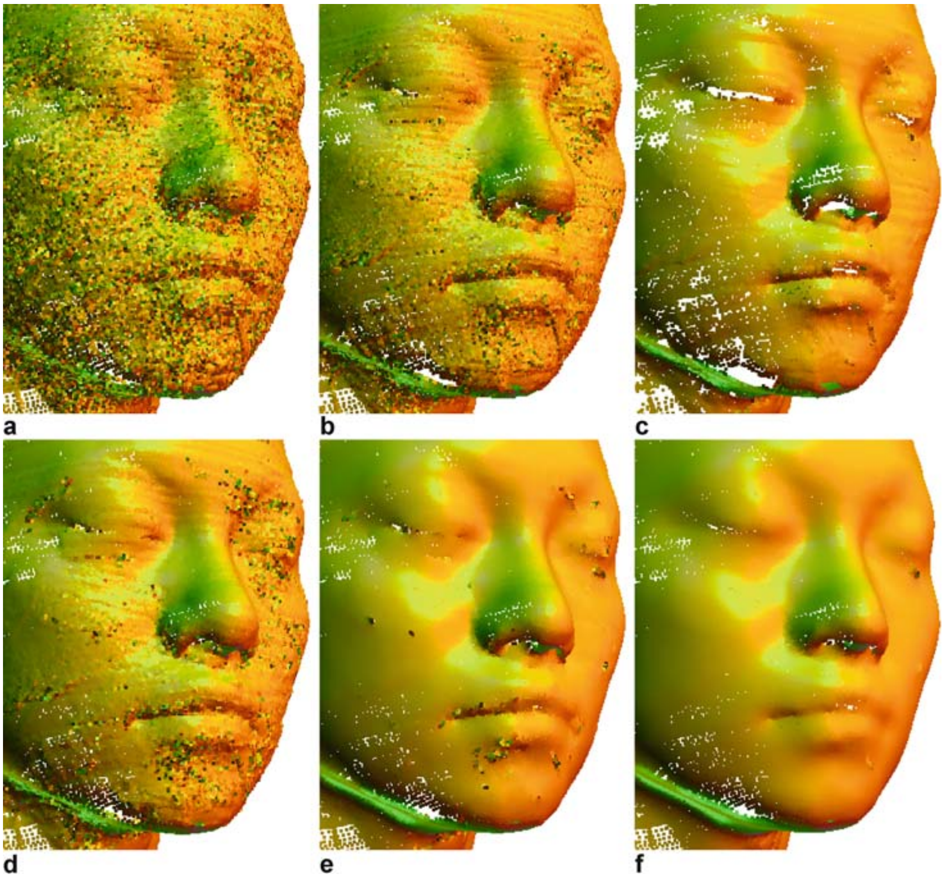

Fig. 6a–f. Comparisons with Laplacian denoising: **a** noisy point model of holehole; **b** noise-removal after 10 iterations of Laplacian smoothing; **c** vertex-drifting and oversmoothing heavily after 100 iterations of Laplacian smoothing; **d** result from one iteration of the mean shift filter; **e** some geometric features caused by multiple scans in the same areas of the face are still preserved after 10 iterations of mean shift denoising with $k = 10$; **f** the undesired geometric details on the face are cleared with k increasing to 20. Here the parameter k means the spatial k-nearest neighbors instead of the influence region in which adaptive neighbors are not known before the process of clustering

The mean shift procedure accounts simultaneously for both the generalized points with components of geometric information and feature attributes to determine the neighborhood of each sample point. Obviously, the normals of the point-sampled surface are filtered when we update the point $\boldsymbol{p}_i$ according to the range information carried by the local mode

$$\boldsymbol{p}_i := (\boldsymbol{v}_i, \boldsymbol{n}_{i,\text{conv}}, H_i), i = 1, 2, \dots, n. \tag{11}$$

The resulting modes are adherents of the local geometry.

Step 3. Clustering. Build clusters of points whose modes are similar. Generally, the neighborhood size k is an essential parameter for good shape smoothing results. In [24], Pauly et al. suggest we select a $k \in [6, 20]$ in the spatial domain. Actually point distribution of models is always irregular and seems anisotropic, thus it is necessary for k to take a non-uniform value for each point on the same model. Here we merge points to compose the range neighbors of $\boldsymbol{p}_i$ whose mean shift points are approximately the same. Figure 3 illustrates clustering with different bounding cell size r. Then for each $\boldsymbol{p}_i$, we take the points that are the intersection of the range neighbors $\{\boldsymbol{q}^r_{ij}\}$ and the spatial neighbors $\{\boldsymbol{q}^s_{ij}\}$ as new neighbors, denoted by $\{\boldsymbol{q}^*_{ij}\}$ and these new points act as the influence region of $\boldsymbol{p}_i$ in *Step 4*.

Step 4. Vertex estimation. After updating the range component of $\boldsymbol{p}_i$ and its neighbors, we apply trilateral filtering in the influence region associated with a fixed local mode. Compared with bilateral filtering, we not only separate spatial and range signals to determine the local area with geometric coherence, but also introduce a curvature-related kernel to smooth high gradient regions efficiently. Our approach is slightly different from the trilateral normal filtering proposed by Choudhury and Tumblin [7].

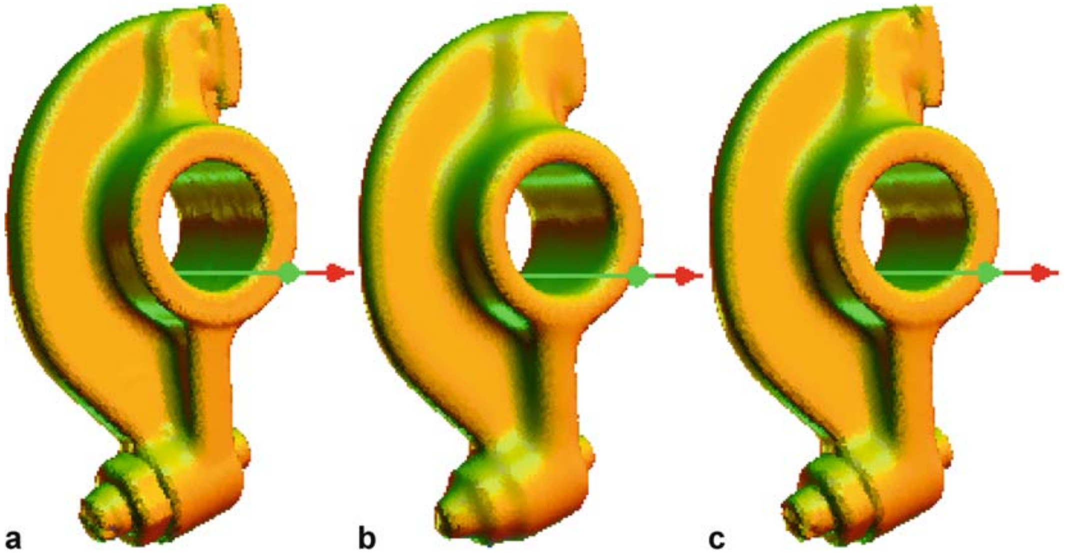

Fig. 7. a Original rockArm model; **b** smoothing result without normal clustering; **c** smoothing result with normal clustering

Table 1. Basic information of models and list of the computational time for different models and different iterations. Times do not include the time to load models and to estimate normals and curvature. Time1 indicates the time of finding the local mode in the mean shift procedure, and Time2 is the vertex estimating time. L_2 and L_∞ are, respectively, the L_2-norm and the L_∞-norm of vertex drifting values in the tangential direction. All input data are scaled into a unit bounding box in advance

Model	Figure	Verts.	Inital k	Time1	r	Iters.	Time2	L_2	L_∞
Bunny	Fig. 2	35k	12	19s	0.05	1	1s	0.0092	0.0278
David	Fig. 3	121k	15	108s	–	–	–	–	–
Holehole	Fig. 6e	101k	10	80s	–	10	11s	0.0277	0.0657
Holehole	Fig. 6f	101k	20	85s	–	10	16s	0.0348	0.0748
Fandisk	Fig. 8d	174k	15	–	–	5	13s	0.0313	0.0725
Dog	Fig. 9b	195k	12	130s	0.03	1	5s	0.0243	0.0457
Dog	Fig. 9c	195k	30	139s	0.03	1	10s	0.0243	0.0549

The curvatures are considered as second-order properties of the 3D geometry, and the performance of the curvature-related kernel for the regions near the salient ridge and ravine structures is satisfactory.

$$\boldsymbol{v}_i := \boldsymbol{v}_i + w_i \boldsymbol{n}_i \tag{12}$$

satisfy

$$w_i = \frac{\sum_{\boldsymbol{q}^*_{ij}} \langle \boldsymbol{n}_i, \boldsymbol{v}_i - \boldsymbol{q}^*_{ij} \rangle g_i}{\sum_{\boldsymbol{q}^*_{ij}} g_i}, \tag{13}$$

and

$$g_i = g_{\sigma_1}(d_i) g_{\sigma_2}(h_i) g_{\sigma_3}(e_i), \tag{14}$$

where $g_\sigma(\cdot)$ is a Gaussian kernel, d_i is the distance of $\|\boldsymbol{v}_i - \boldsymbol{q}^*_{ij}\|$, h_i is the projection of the vector $(\boldsymbol{v}_i - \boldsymbol{q}^*_{ij})$ onto the normal $\boldsymbol{n}_i$, and e_i is the inverse of the curvature difference between $\boldsymbol{v}_i$ and $\boldsymbol{q}^*_{ij}$.

Adaptive neighbors. For large point data sets, the selection of neighbors is a trade-off between computational time and smoothing quality. When choosing a small and uniform spatial size of a neighborhood, it costs less in computational time, but smooths the model poorly; on the other hand, if we choose a large and uniformly spaced neighborhood, the model is oversmoothed. We use the kdtree to search for the k nearest neighbors, for instance $k = 12$, instead of a neighboring spatial ball, which is unfortunately dependent on the sampling density of the point model. Furthermore, our clustering scheme provides an adaptive neighbor searching method, as shown in Figs. 4 and 5, where different influence regions are used adaptively to remove noise from vertices. Applying trilateral filtering to the adaptive neighborhoods greatly improves the smoothing capabilities of the mean shift filter in high gradient regions. Although the algorithm we present is ordinarily non-iterative whilst denoising, we can also use the resulting adaptive neighbors as inputs to estimate range information, then iteratively perform

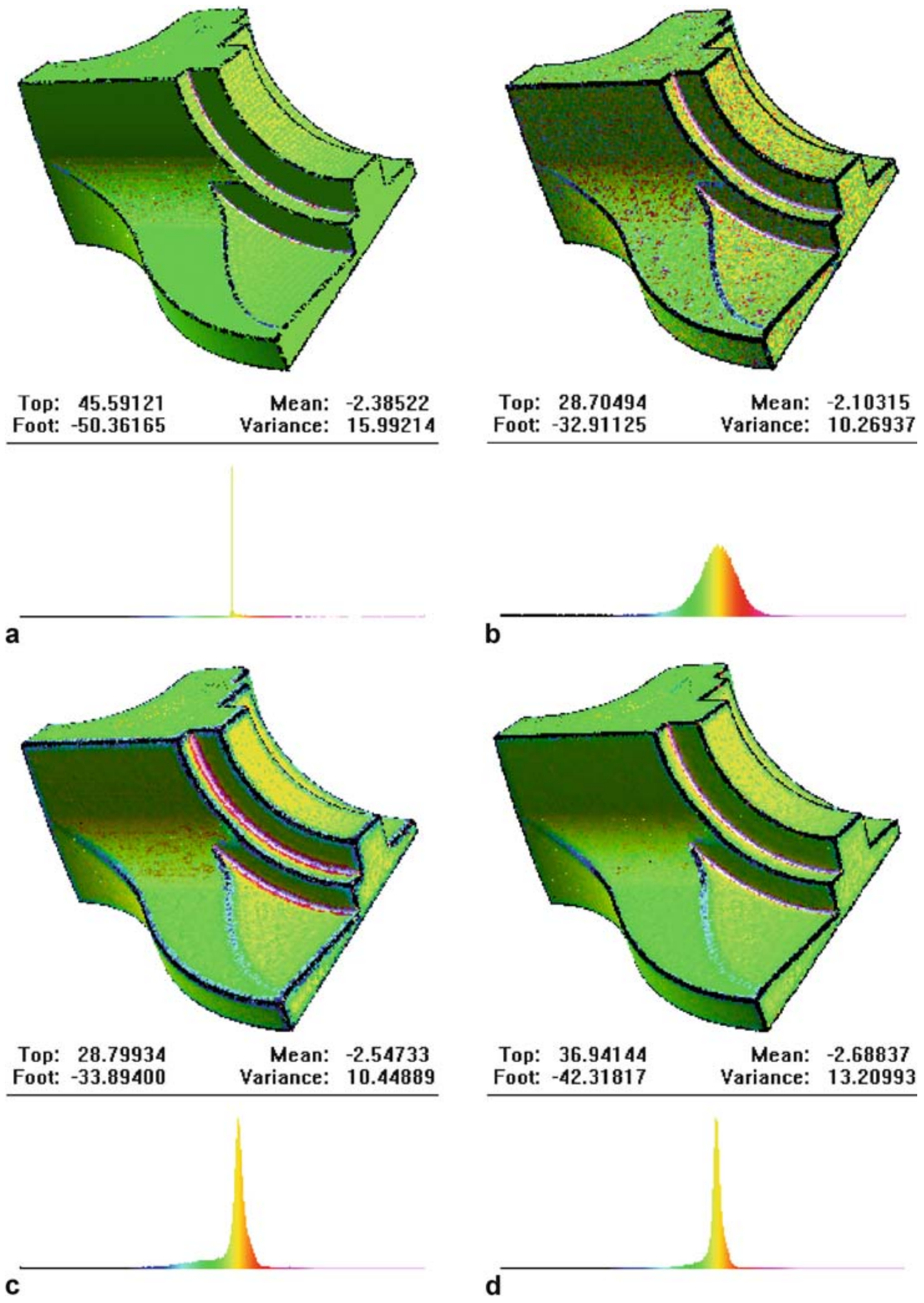

Fig. 8. Comparisons with bilateral filtering

the four steps of the algorithm to make the point model smoother.

Experimental results. We have implemented the mean shift denoising described, Laplacian smoothing, bilateral point denoising and trilateral denoising in Microsoft Windows Pro XP using the C++ language. Our implementations run on a 1.6G PIV with 512M of memory. A variety of parameters are used in our algorithms to give good results. For comparison purposes, we initially select $k = 10$ for both the Laplacian smoothing operator and the mean shift filter in Fig. 6b–e, and $k = 20$ in Fig. 6f. Multi-scale denoising results with different neighbor sizes are given in Fig. 9, where the point models are normalized to unit cube. The spatial neighbors are obtained by setting $k = 12$ in the second column $k = 30$

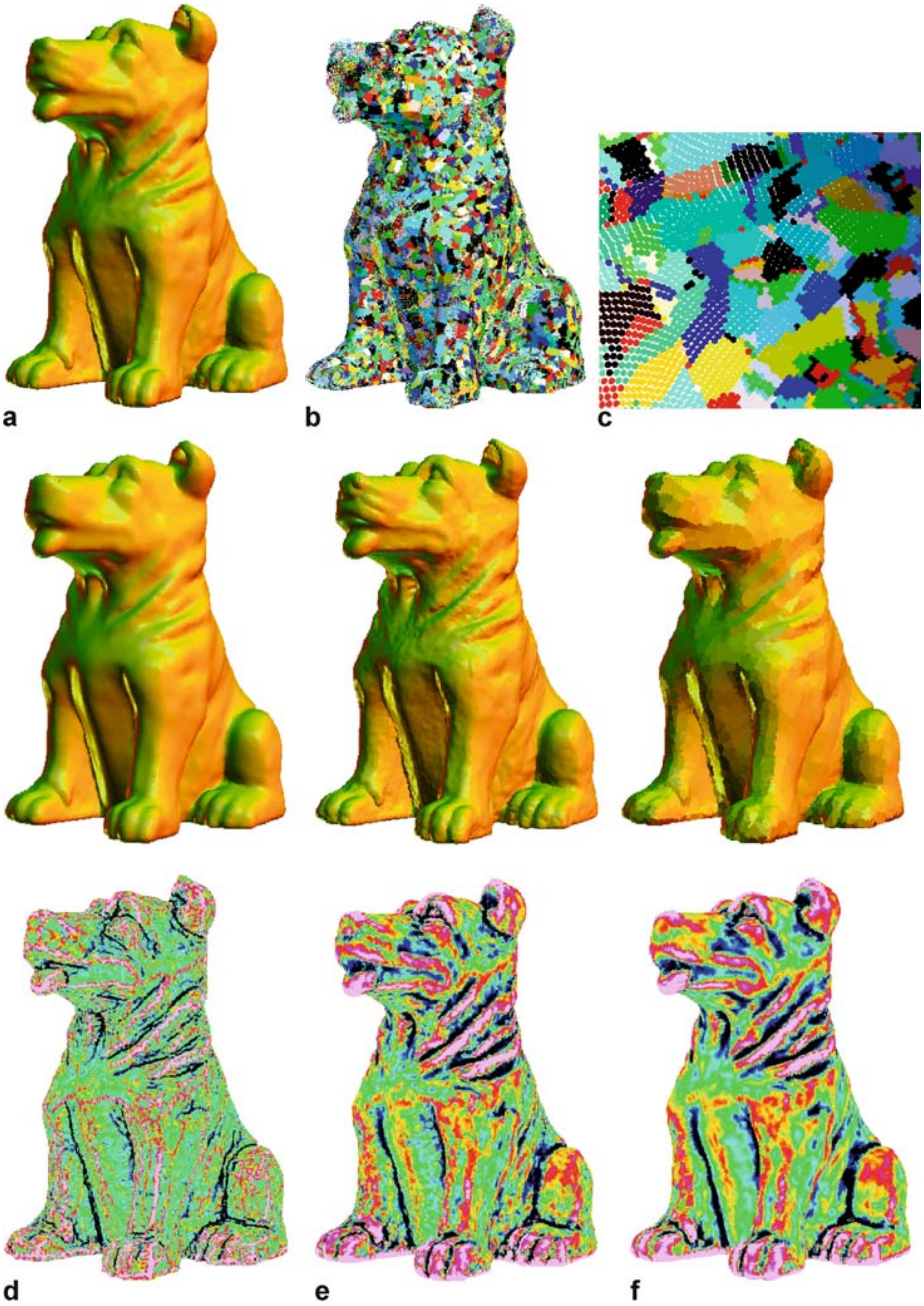

Fig. 9. Multi-scale denoising with different initial neighbor size (all with one iteration)

in the third column. To search the local mode, we set $h_1^r = 0.02$ related to the bandwidth of curvatures, $h_2^r = 0.9$ for normal range bandwidth and the convergence threshold $\varepsilon = 0.001$, which is proportional to the average radius of the bounding sphere of $N(\boldsymbol{p}_i)$. The octree cell size is set to $r = 0.01$ to get the clustering results (see Fig. 9b and c). In the vertex estimation phase, these parameters should be tailored to fit different models. The second row of Fig. 9 shows the results of mean shift denoising, which is feature-preserving, while the third row shows the corresponding visualization results for the mean curvature. Unlike surface variation, we here first reconstruct the local geometry of k neighboring points into O-strip polygonal meshes, then adopt the more exact curvature tensor estimating scheme presented in [27], instead of PCA analysis, and then calculate mean curvature by averaging principal curvatures. Figure 8 shows the denoising results by our trilateral vertex estimation and bilateral filtering [15]. For the sake of comparison, we set the same value for common parameters such as $k = 15$ and five iterations for each. It is clearly demonstrated from the histograms for the normalized mean curvature of original (a), noisy (b), after bilateral filtering (c), without considering the curvature-related term $g_{\sigma_3}(e_i)$ and after trilateral filtering (d) that the trilateral filter adopted in the processing of mean shift filtering offers better performance for appearance-preserving problems when denoising point models. Figure 7 shows our vertex estimation results with and without normal clustering, for the sake of comparison, all parameters like the initial k value, σ and iterations are identical. It is obvious that features such as edges and corners of the rockArm model are preserved with normal clustering.

4 Conclusions

This paper extends mean shift filtering in the image setting to 3D surface smoothing by taking into account the vertex normal and curvature as range component, as well as position as the spatial component. A new approach to mean-shift-based anisotropic denoising of point-sampled surfaces has been presented, which can readily be formulated for mesh-based geometry and even for general 3D geometry. By conducting meaningful clustering according to the local mode of point-based surfaces and the trilateral vertex estimation, our new approach performs stably. While undesirable noise in the point data set is successfully removed, the local shape features are preserved. Vertex shifting along the normals avoids local shape corruption. Note that when taking $\sigma_3 = \infty$ and fixing the spatial neighbors without considering range neighbors, mean shift filtering is reduced to bilateral filtering. An interesting direction for future research is to apply the mean-shift procedure for 3D geometry to segmentation, simplification and feature detection of point-sampled surfaces.

Acknowledgement We are grateful to Rui Wang for his valuable and constructive comments. Also special thanks are due to Yutaka Ohtake for the color-coding scheme, to Xiaoyi Zhong for the hole-hole model, to the Stanford 3D Scanning Repository for their data and the anonymous reviewers for their valuable comments.

References

1. Alexa, M.: Wiener filtering of meshes. In: Wyvill, G. (ed) Proceedings of the Shape Modeling International 2002, pp. 51–57, IEEE Computer Society, Banff, Canada (2002)
2. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S.: Point set surfaces. Proceedings of the Conference on Visualization '01, pp. 21–28, IEEE Computer Society, San Diego, CA (2001)
3. Amenta, N., Kil, Y.: Defining Point set surfaces. ACM Trans. Graph.. **23**(3), 264–270 (2004)
4. Bajaj, C., Xu, G.: Anisotropic diffusion of surfaces and functions on surfaces. ACM Trans. Graph.. **22**(1), 4–32 (2003)
5. Barash, D., Comaniciu, D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. Image Video Computing **22**(1), 73–81 (2004)
6. Carr, J., Beatson, R., Cherrie, J., Mitchell, T.: Reconstruction and representation of 3D objects with radial basis functions. Proceedings of ACM SIGGRAPH 2001, pp. 67–76 (2001)
7. Choudhury, P., Tumblin, J.: The trilateral filter for high contrast images and meshes. Proceedings of Eurographics 2003, pp. 186–196, Aire-la-Ville, Switzerland (2003)
8. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low-level vision. Proceedings of 16th International Conference on Pattern Recognition, pp. 150–155 (2002)
9. Clarenz, U., Diewald, U., Rumpf, M.: Anisotropic geometric diffusion in surface processing. In: Ertl, T., Hamann, B., Varshney, A. (eds) Proceedings of IEEE Visualization 2000, pp. 397–405, Salt Lake City, UT (2000)
10. Clarenz, U., Rumpf, M., Telea, A: Fairing of point based surfaces. Proceedings of Computer Graphics International, June 16–19, 2004 Crete, Greece, pp. 600–603 (2004)
11. Comaniciu, D., Meer, P.: Mean shift analysis and applications. IEEE International Conference on Computer Vision, September 20–27, 1999, Kerkyra, Greece, pp. 1197–1203 (1999)
12. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Patt. Anal. Mach. Intell. **24**(5), 603–619 (2002)
13. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. IEEE Conference on Computer Vision and Pattern Recognition, pp. 142–151, Hilton Head Island, SC, (2000)
14. Desbrun, M., Meyer, M., Schroder, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. Proceedings of SIGGRAPH99, pp. 317–324, Los Angeles, CA (1999)
15. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. Proceedings of SIGGRAPH03, pp. 950–953, San Diego, CA (2003)
16. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. Proceedings of ACM SIGGRAPH 92, pp.71–78, Chicago (1992)
17. Hu, G., Peng, Q.: Anisotropic denoising of point-sampled models (in Chinese). J. Software **15**, 215–221 (2004)
18. Hu, G., Peng, Q.: Bilateral estimation of normal on point-sampled models. International Conference on Computational Science and Its Applications, May 9–12, Singapore, pp. 758–768 (2005)
19. Hu, G., Peng, Q., Forrest, A.: Robust mesh smoothing. J Comput. Sci. Technol. **19**(4), 521–528 (2004)
20. Jones, T., Durand, F., Desbrun, M. Non-iterative, feature preserving mesh smoothing. Proceedings of SIGGRAPH03, pp. 943–949, San Diego, CA (2003)
21. Lange, C., Polthier, K. Anisotropic fairing of point sets. Comput. Aid Geomet. Des. **22**(7), 680–692 (2005)
22. Pauly, M., Gross, M.: Spectral processing of point-sampled geometry. Proceedings of ACM SIGGRAPH 2001, pp. 379–386 (2001)
23. Pauly, M., Gross, M., Kobblet, L.: Efficient Simplification of Point-Sampled Geometry. IEEE Visualization 02, pp. 163–170, IEEE Computer Society (2002)
24. Pauly, M., Keiser, R., Kobblet, L., Gross, M.: Shape modeling with point-sampled geometry. ACM Trans. Graph. **22**(3), 641–650 (2003)

25. Peng, J., Strela, V., Zorin, D.: A simple algorithm for surface denoising. IEEE Visualization 2001, pp. 107–112, San Diego, CA (2001)
26. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Exper. Math. **2**(1), 15–36 (1993)
27. Taubin, G.: A signal processing approach to fair surface design. Proceedings of ACM SIGGRAPH 1995, pp.351–358 (1995)
28. Vollmer, J., Mencl, R., Müller, H.: Improved laplacian smoothing of noisy surface meshes. Comput. Graph. Forum **18**(3), 131–138 (1999)
29. Wang, A., Xu, Y., Shum, H., Cohen, M.: Video tooning. ACM Trans. Graph. **23**(3), 574–583 (2004)
30. Welch, W., Witkin, A.: Free-form shape design using triangulated surfaces. Comput. Graph. **28**, 247–256 (1994)

GUOFEI HU was born in 1977. He is currently a lead researcher at the GE Global Research Center, Shanghai, China. He received his B.Sc. in Applied Mathematics from Zhejiang University in 1999, and his Ph.D. in Computer Graphics from Zhejiang University in 2005. His research interests include discrete differential geometry and digital geometry processing, especially filter design on point-based geometry.

QUNSHENG PENG is Professor of Computer Graphics at Zhejiang University. His research interests include realistic image synthesis, computer animation, scientific data visualization, virtual reality, feature modeling, and parametric design. Prof. Peng graduated from Beijing Mechanical College in 1970 and received his Ph.D from the Department of Computing Studies, University of East Anglia, in 1983. He is currently the Director of the State Key Lab of CAD& CG at Zhejiang University and is serving as a member of the editorial boards of several Chinese journals.

A. ROBIN FORREST is a Professorial Fellow in the School of Computing Sciences, University of East Anglia, U.K. He graduated in Mechanical Engineering from the University of Edinburgh in 1965 and obtained his Ph.D. at the University of Cambridge in 1968. His current research interests include point-based geometric modeling, information visualization, and computational geometry.