Robust Denoising of Point-Sampled Surfaces

Jifang Li, Renfang Wang*
College of Computer Science and Information Technology
Zhejiang Wanli University
Ningbo 315100, China

Abstract: - Based on sampling likelihood and feature intensity, in this paper, a feature-preserving denoising algorithm for point-sampled surfaces is proposed. In terms of moving least squares surface, the sampling likelihood for each point on point-sampled surfaces is computed, which measures the probability that a 3D point is located on the sampled surface. Based on the normal tensor voting, the feature intensity of sample point is evaluated. By applying the modified bilateral filtering to each normal, and in combination with sampling likelihood and feature intensity, the filtered point-sampled surfaces are obtained. Experimental results demonstrate that the algorithm is robust, and can denoise the noise efficiently while preserving the surface features.

Key-Words: -Moving least squares surface; sampling likelihood; normal voting tensor; feature intensity; bilateral filtering; point-sampled surfaces denoising

1 Introduction

Point clouds have become increasingly popular in modelling and rendering applications[1-15] due to improved graphics hardware and technologies for the acquisition of point geometry. Point-sampled models without topological connectivity are normally generated by sampling the boundary surface of physical 3D objects with 3D-scanning devices. Despite the steady improvement in scanning accuracy, undesirable noise is inevitably introduced from various sources such as local measurements and algorithmic errors. Consequently, noisy models need to be denoised or smoothed before performing any subsequent geometry processing such as simplification, reconstruction and parameterization. It remains a challenging task to remove the inevitable noise while preserving the underlying surface features in computer graphics.

So far, various smoothing methods for mesh model have been proposed, please refer to [16-18] and the references therein. The most common techniques are based on Laplace smoothing. Taubin[19]introduced signal processing on surfaces that is based on the definition of the Laplacian operator on meshes and developed a fast and simple iterative Laplacian smoothing scheme. Desbrun et al.[20] extended this approach to irregular meshes using a geometric flow analogy. Ohtake et al.[21] extended the Laplace smoothing by combining geometry smoothing with parameterization regularization. Peng et al. [22] applied locally adaptive Wiener filtering to meshes. However, these techniques are all isotropic, and therefore indiscriminately smooth noise as well as salient features, leading to shrinkage or undesired distortion of the mesh shape. To compensate these drawbacks, Liu et al. [23] proposed a method that keeps the volume of each star of a vertex. Vollmer et al. [24] suggested a method that is based on the idea to push the vertices back to their previous positions. In order to reduces diffusion across edges, Hildebrandt et al.[17] proposed an anisotropic smoothing scheme. Inspired by anisotropic diffusion in image processing [25], feature-preserving mesh smoothing methods were introduced[26-28]. These methods modified the diffusion equation to make it nonlinear or anisotropic, thus could preserve sharp features. The work of [29-31] proposed diffusion-type smoothing on the normal field first, and then constructed the surface to match the new normals. Although these approaches are superior to those using isotropic techniques, they would cause significant computational times.

As a straightforward representation form for highly complex objects, the point-sampled model is obviously processed with less overhead in computation time and memory costs than mesh model. Thus, point-sampled model denoising has been an active research area. Like mesh denoising techniques, earlier methods such as Laplacian [8] for denoising point-sampled surfaces (PSS) are isotropic, which result commonly in point drifting and oversmoothing. So the anisotropic methods were also introduced. Clarenz et al. [32] presented a PDE-based surface fairing application within the framework of processing point-based surface via

PDEs. Lange and Polthier [33] proposed a new method for anisotropic fairing of a point sampled surface based on the concept of anisotropic geometric mean curvature flow. Based on dynamic balanced flow, Xiao et al. [34] presented a novel approach for fairing PSS. Other methods have also been proposed for denoising PSS. Algorithms that recently attracted the interest of many researchers are moving-least squares (MLS) approaches [15,35,36] to fit a point set with a local polynomial approximation; the point set surface can be smoothed by shifting point positions towards the corresponding MLS surface. The main problem of MLS-based methods is that prominent shape features are blurred while smoothing PSS.

Concerning the above problem of MLS approaches, this paper puts forward a robust denoising algorithm for PSS with feature preserving. Based on MLS surface, the sampling likelihood of sample point is first computed. In order to more efficiently preserve the surface features while denoising PSS, we adopt normal voting tensor to evaluate the feature intensity of sample point and apply the modified bilateral filtering to filter the normal of each sample point. The smoothed model is finally obtained via the combination of sampling likelihood and feature intensity.

The paper is organized as follows. Section 2 gives the method for computing the sampling likelihood and Section 3 describes measure of the feature intensity. Our denoising approach with feature preservation is described in Section 4. We compare our method with two denoising techniques in Section 5. Section 6 concludes the paper.

2 Computing the Sampling Likelihood Based on MLS

In this paper, we consider the probability that a 3D point is located on the sampled surface as the sampling likelihood. The sample point closer to the sampled surface should be characteristic of higher sampling likelihood than one being more distant. We approximately take the MLS surface approximating the k nearest neighbors $N_k(\boldsymbol{p}_i)$ of sample point $\boldsymbol{p}_i$ as the sampled surface. In the following we will briefly review the MLS surface and then describe how to compute the sampling likelihood.

2.1 Moving Least Squares Surface

Alexa et al.[1] proposed a representation of point-sampled model by fitting a local polynomial approximation to the point set using a MLS method. The result of the MLS-fitting is a smooth, 2-manifold surface for any point set. Given a point set $P=\{\boldsymbol{p}_i\}$, the continuous MLS surface S is defined implicitly as the stationary set of a projection operator $\psi(\boldsymbol{r})$ that projects a point onto the MLS surface. To evaluate ψ, a local reference plane $H=\{x\in\mathbb{R}^3|\boldsymbol{n}\cdot\boldsymbol{x}-D=0\}$ is first computed by minimizing the weighted sum of squared distances, i.e., $\arg\min_{\boldsymbol{n},\boldsymbol{q}}\sum_{p_i\in P}(\boldsymbol{n}\cdot\boldsymbol{p}_i-\boldsymbol{n}\cdot\boldsymbol{q})^2\theta(\|\boldsymbol{p}_i-\boldsymbol{q}\|)$, where $\boldsymbol{q}$ is the projection of $\boldsymbol{r}$ onto H and θ is the MLS kernel function $\theta(d)=\exp(-d^2/h^2)$, where h is a global scale factor. Accordingly, the local reference domain is given by an orthonormal coordinate system on H so that $\boldsymbol{q}$ is the origin of this system. Then a bivariate polynomial $g(u,v)$ is fitted to the points projected onto the reference plane H using a similar weighted least squares optimization. Here (u_i,v_i) is the representation of $\boldsymbol{q}_i$ in the local coordinate system in H, where $\boldsymbol{q}_i$ is the projection of $\boldsymbol{p}_i$ onto H. So, the projection of $\boldsymbol{r}$ onto S is given as $\psi(\boldsymbol{r})=\boldsymbol{q}+g(0,0)\cdot\boldsymbol{n}$. More details on the MLS method can be found in[37].

2.2 Definition of the Sampling Likelihood

Based on MLS, we first compute the distance of each point in $N_k(\boldsymbol{p}_i)$ to its projection onto the MLS surface and then define the sampling likelihood of $\boldsymbol{p}_i$. We take a third degree polynomial to approximate the cluster of $N_k(\boldsymbol{p}_i)$:

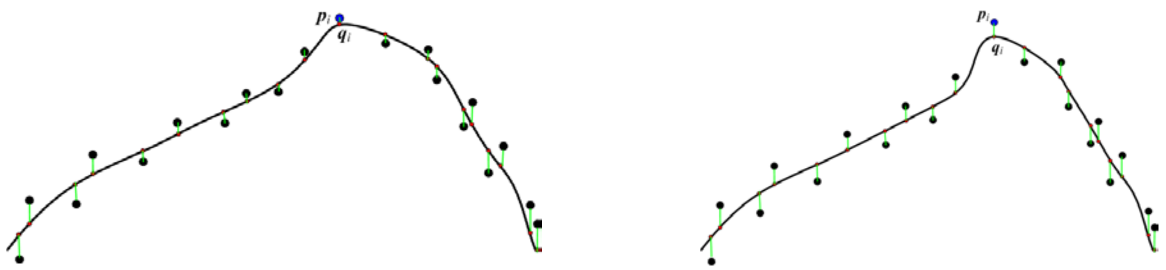
$$g(x,y)=a_9x^3+a_8y^3+a_7x^2y+a_6xy^2+a_5x^2+a_4y^2+a_3xy+a_2x+a_1y+a_0 \quad (1)$$

Let $\boldsymbol{q}_{ij}$ be the projection of $\boldsymbol{p}_{ij}\in N_k(\boldsymbol{p}_i)$ onto the above MLS surface $g(x,y)$ ($\boldsymbol{q}_{i0}$, i.e. $\boldsymbol{q}_i$, is the projection of $\boldsymbol{p}_i$) and define the sampling likelihood l_i of $\boldsymbol{p}_i$ as

$$\begin{cases} l_i=1\text{-}d_i/d_{\boldsymbol{max}} \\ d_i=\dfrac{\sum_{j=0}^{k-1}\theta(\|\boldsymbol{p}_{ij}-\boldsymbol{p}_i\|)\|\boldsymbol{p}_{ij}-\boldsymbol{q}_{ij}\|}{\sum_{j=0}^{k-1}\theta(\|\boldsymbol{p}_{ij}-\boldsymbol{p}_i\|)}, \\ d_{\boldsymbol{max}}=\max_{i\in N}\{d_i\} \end{cases} \quad (2)$$

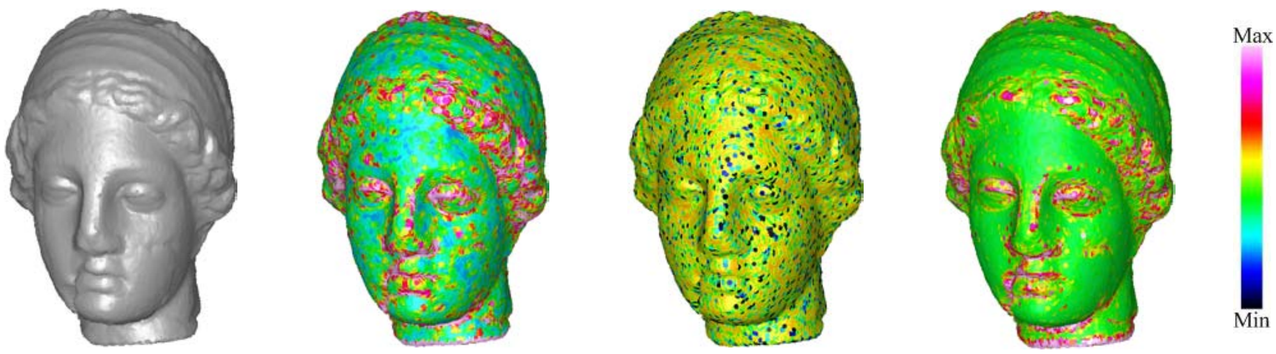
where N is the size of point-sampled model and d_i is the weighted-average distance of $\boldsymbol{p}_{ij}$ to its projection onto the MLS surface.

Obviously, the influence of $\boldsymbol{p}_{ij}$ on d_i decreases exponentially with Euclidean distance to $\boldsymbol{p}_i$. On the other hand, the smaller d_i is, the greater l_i. Consequently, the sampling likelihood l_i can effectively denote the value of probability that $\boldsymbol{p}_i$ is located on the sampled surface. Fig.1 illustrates the MLS surface approximating the k nearest neighbors $N_k(\boldsymbol{p}_i)$ of sample point $\boldsymbol{p}_i$. The MLS surface, approximating $N_k(\boldsymbol{p}_i)$ of sample point $\boldsymbol{p}_i$ with high sampling likelihood l_i, is shown in Fig.1a and the MLS surface, approximating $N_k(\boldsymbol{p}_i)$ of sample point $\boldsymbol{p}_i$ with low sampling likelihood l_i, is shown in Fig.1b. According to the size of Point-sampled model, we take $k \in [20, 35]$ to determine the sampling likelihood l_i. The related visualizations of the noisy Igea model, as shown in Fig.2a, are demonstrated in Fig.2. Fig.2b demonstrates mean curvature visualization of the noisy Igea model and the sampling likelihood visualization of noisy model is illustrated in Fig.2c. In this paper, all the point-sampled models are rendered by using a point-based rendering technique.



(a) (b)

Fig.1 MLS surface approximating point set $N_k(p_i)$. (a) The sample point p_i with high sampling likelihood l_i; (b) The sample point p_i with low sampling likelihood l_i.



(a) (b) (c) (d)

Fig.2 (a) Noisy Igea Model; (b) Mean curvature visualization of (a); (c) Sampling likelihood visualization of (a); (d) Feature intensity visualization of (a).

3 Measuring the Feature Intensity

Though achieving denosing of PSS by moving each sample point to its projection onto the corresponding MLS surface, surface features can not be efficiently preserved. So we first determine the feature intensity of sample point before denoising. In this paper, the feature intensity of sample point is measured by extending normal voting tensor applied to the extraction of sharp edge on 3D mesh [38] to point-sampled surfaces. A normal voting tensor $\boldsymbol{T}_i$ for a sample point $\boldsymbol{p}_i$ is defined by $\boldsymbol{T}_i = \sum_{ij\in N(i)} u_{ij}\, \boldsymbol{n}_{ij}^{'}\, \boldsymbol{n}_{ij}^{'T}$, where $N(i)$ is the index set of $\boldsymbol{p}_{ij}$ belonging to $N_k(\boldsymbol{p}_i)$, u_{ij} is a weight defined as $u_{ij} = \exp(-\| \boldsymbol{p}_{ij} - \boldsymbol{p}_i \| / \sigma_e)$. We take standard deviation σ_e as $\sigma_e = 2r/3$, r is the radius of the enclosing sphere of $N_k(\boldsymbol{p}_i)$. $\boldsymbol{n}_{ij}$ is $\boldsymbol{p}_{ij}$'s normal ($\|\boldsymbol{n}_{ij}\| = 1$) and $\boldsymbol{n}_{ij}^{'}$ is determined as $\boldsymbol{n}_{ij}^{'} = 2(\boldsymbol{n}_{ij}\cdot \boldsymbol{w}_{ij})\boldsymbol{w}_{ij} - \boldsymbol{n}_{ij}$, where $\boldsymbol{w}_{ij} = (\boldsymbol{p}_i - \boldsymbol{p}_{ij}) \times \boldsymbol{n}_{ij} \times (\boldsymbol{p}_i - \boldsymbol{p}_{ij})(\|\boldsymbol{w}_{ij}\|=1)$. From the definition, $\boldsymbol{T}_i$ is symmetric and positive semi-definite. Accordingly, its eigenvalues are real-valued and non-negative: $v_1 \geq v_2 \geq v_3 \geq 0$. Furthermore, the corresponding eigenvectors e_1, e_2 and e_3 form an orthonormals basis. So we define the feature intensity of sample point $\boldsymbol{p}_i$ as

$$s_i = \begin{cases} 1 & \| \bar{\boldsymbol{n}}_i \cdot \boldsymbol{e}_1 \| < \delta \\ 1 & v_3 > \alpha(v_1 - v_2) \wedge v_3 > \beta(v_2 - v_3), \\ (v_2 - v_3)/v_1 & \text{otherwise} \end{cases} \quad (3)$$

where $\bar{\boldsymbol{n}}_i = \sum_{ij\in N(i)} u_{ij}\, \boldsymbol{n}_{ij}^{'}$ is a weighted sum of voting normal $\boldsymbol{n}_{ij}^{'}$, and δ, α, β are positive real numbers. In this paper, we experimentally set δ, α, β to 0.2, 0.3 and 0.3, respectively.

The equation classifies each sample point into three types which correspond to the type of feature that the point belongs to, i.e. face, sharp edge, or corner. The feature intensity becomes about 1.0 if sample point lies on a sharp edge or a corner, or about 0.0 if it lies on a face. As a result, s_i indicates the geometric feature of surface at $\boldsymbol{p}_i$. The feature intensity visualization of noisy Igea model (Fig.2a) is illustrated in Fig.2d.

In order to measure the feature intensities for noisy model efficiently, we determine a neighbourhood of sample point through a growing process. Let $N_R(\boldsymbol{p}_i)=\{\boldsymbol{q}_{ij} \mid \boldsymbol{q}_{ij} \in P_n,\ \|\boldsymbol{q}_{ij} - \boldsymbol{p}_i\| \leq R = 10|E|\}$ be the set of the neighbors of $\boldsymbol{p}_i$ whose elements are within a fixed radius R bound centered at $\boldsymbol{p}_i$, where $|E| = \sum_{i=0}^{n-1} r_{i\,\min} \big/ n$ is the average edge length of the sample point set P_n, $n=|P_n|$ and $r_{i\,\min}$ is the distance between $\boldsymbol{p}_i$ and its nearest point. We initially compute a samller subset $N_k(\boldsymbol{p}_i)$ so that $N_R(\boldsymbol{p}_i)=\{\boldsymbol{p}_{ij} : \|\boldsymbol{p}_{ij} - \boldsymbol{p}_i\| \leq R \wedge \boldsymbol{p}_{ij} \in N_k(\boldsymbol{p}_i)\}$, where we typically take k=9. The set $N_R(\boldsymbol{p}_i)$ is augmented $N_R(\boldsymbol{p}_i) \leftarrow N_R(\boldsymbol{p}_i) \bigcup \tilde{N}_k(\boldsymbol{p}_{ij})$ for each element $\boldsymbol{p}_{ij} \in N_R(\boldsymbol{p}_i)$ not processed, where $\tilde{N}_k(\boldsymbol{p}_{ij})=\{\tilde{\boldsymbol{p}}_{ij} : \|\tilde{\boldsymbol{p}}_{ij} - \boldsymbol{p}_i\| \leq R \wedge \tilde{\boldsymbol{p}}_{ij} \in N_k(\boldsymbol{p}_{ij}) \wedge \tilde{\boldsymbol{p}}_{ij} \notin N_R(\boldsymbol{p}_i)\}$. This process is repeated for each new point in $N_R(\boldsymbol{p}_i)$ not processed until there are not more points to be added, then the final neighbourhood $N_R(\boldsymbol{p}_i)$ is determined. Observe that determining the neighbourhood in this way tends to eliminate points that are in different connected components or on different sides of a thin region, as shown in Fig.3.

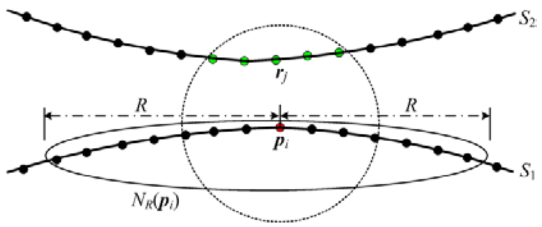


Fig.3 Determining the neighbourhood $N_R(\boldsymbol{p}_i)$ of sample point $\boldsymbol{p}_i$.

4 Denoising of PSS

The main idea of this paper's denoising algorithm is as follows: $\boldsymbol{p}_i$'s normal is first filtered by using the modified bilateral filtering. In combination with the sampling likelihood and feature intensity, the distance weight m_i of $\boldsymbol{p}_i$ is then determined when denoising PSS. $\boldsymbol{p}_i$ is moved in the filtered normal direction with an offset D_i so as to smooth PSS.

4.1 Normal Filtering

If $\boldsymbol{p}_i$ is moved in the vector $\boldsymbol{p}_i\boldsymbol{q}_i$ direction when smoothing PSS, the surface features should be blurred. In order to preserving those features more effectively, we first smooth $\boldsymbol{p}_i$'s normal according to the modified bilateral filtering.

Surface normals play an important role in surface denoising as surface features are best described with the first-order surface normals. It also is well-known that normal variations offer more intuitive geometric meaning than point position variations. A smooth surface can be described as one having smoothly varying normals whereas features such as sharp edges and corners appear as discontinuities in the normals. Thus unlike the bilateral filtering in [16], we design the following bilateral filtering with the normal-variation term to compute the filtered normal $\boldsymbol{n}_i^*$

$$\boldsymbol{n}_i^* = \frac{\sum_{ij\in N(i)} w_c(\|\boldsymbol{p}_{ij} - \boldsymbol{p}_i\|) w_s(\alpha_{ij}) \boldsymbol{n}_{ij}}{\sum_{ij\in N(i)} w_c(\|\boldsymbol{p}_{ij} - \boldsymbol{p}_i\|) w_s(\alpha_{ij})}, \quad (4)$$

Where $w(x)$ is a Gaussian kernel: $w_c(x)=\exp(-x^2/2\sigma_c^2)$ and $w_s(x)=\exp(-x^2/2\sigma_s^2)$. The normal variation α_{ij} is defined as $\alpha_{ij} = acos(\boldsymbol{n}_i \cdot \boldsymbol{n}_{ij})$ ($\|\boldsymbol{n}_i\|=|\boldsymbol{n}_{ij}\|=1$). Here, we take the parameter σ_c as $\sigma_c = r/2$ and σ_s as the standard deviation of the normal variation α_{ij}.

According to the equation, $\boldsymbol{n}_i^*$ is the weighted average of $\boldsymbol{p}_{ij}$'s normal where the weight of each normal is computed using a standard Gaussian function w_c in the spatial domain multiplied by an influence function w_s in the intensity domain that decreases the weight of normals with large normal variation. Therefore, $\boldsymbol{n}_i^*$ is influenced mainly by the sample points in $N_k(\boldsymbol{p}_i)$ that have a similar intensity.

4.2 Sample Point Filtering

Next, for each sample point $\boldsymbol{p}_i$ we find its smoothed position $\boldsymbol{p}_i^*$ by moving it along $\boldsymbol{n}_i^*$ with an offset D_i, i.e., $\boldsymbol{p}_i^* = \boldsymbol{p}_i + D_i \boldsymbol{n}_i^*$, where D_i is determined as $D_i = m_i \cdot \|\boldsymbol{q}_i - \boldsymbol{p}_i\|$. In terms of the sampling likelihood and feature intensity, define the distance weight m_i as $m_i = (-1)^\tau[\lambda e^{-l_i} + (1-\lambda)e^{-s_i}]$, where $\lambda (0 \le \lambda \le 1)$ is a user-adjustable parameter; τ is set to 0 when $\boldsymbol{n}_i^* \cdot \boldsymbol{p}_i \boldsymbol{q}_i > 0$, which indicates that $\boldsymbol{p}_i$ is moved along $\boldsymbol{n}_i^*$, otherwise τ is set to 1 and $\boldsymbol{p}_i$ is moved along $-\boldsymbol{n}_i^*$.

5 Experimental Results and Discussion

In our experiments, we use Microsoft Visual C++ programming language on a personal computer with a Pentium IV 2.8 GHz CPU and 1 GB main memory. We have implemented our denoising algorithm and another two denoising techniques: the Bilateral denoising (BIL) and the MLS-based denoising to compare their denoising results. We use three models in our comparison: a noisy Igea model with 134345 sample points (Fig.2a), a noisy Fandisk model with 97580 sample points (Fig.4a) and a noisy Dragon-head Model with 100056 sample points (Fig.6). For these models, table 1 presents the related statistics, where Iters. stands for the number of iterations, Max. Error is the maximum of distances between the original (noisy) points and their corresponding denoised points and Ave. Error is the average of distances.

We use two visualization schemes to compare the techniques with our method. The first scheme consists of coloring by the mean curvature. The second one measures the difference between the original and denoised point model, i.e., we visualize the differences in the positions of the corresponding sample points of the models $|\boldsymbol{p}_i^{\text{noisy}} - \boldsymbol{p}_i^{\text{denoised}}|$.

In Fig.5, we demonstrate a comparison of the denoised Fandisk models by MLS, BIL and our method. The denoised models are illustrated in the top row of Fig.5, and their corresponding mean curvature visualizations in the bottom row. As seen in Fig.5, our algorithm removes the high-frequency noise properly and achieves a more accurate result than MLS or BIL does. Fig.7 shows a comparison of MLS, BIL and our algorithm concerning feature preservation. Note that our alogrithm preserves sharp features more accurately than MLS or BIL does while producing a smooth result, as shown in the closer views of the upper jaw of the denoised model.

In Fig.8, we demonstrate a comparison of the denoised Igea models by MLS, BIL and our method. The denoised models are illustrated in the top row of Fig.8 and their corresponding mean curvature visualizations in the middle row. In Fig.8c, we show the denoising efficiency of our approach on the noisy Igea model (Fig.2a). It can be noticed that the high-frequency noise is properly removed, while fine details in hair, mouth and face regions are accurately preserved. At the same time, we demonstrate that our algorithm presents the best performance according to the entropy of the differences between the noisy and denoised models, as shown in the bottom row of Fig.8. From the Max. and Ave. errors in Table 1 we can also notice that our method outperforms its two rivals. As a result, our method produces the lowest oversmoothing when compared with the two other denoising techniques.

Since the MLS-based denoising shifts sample points to its projection onto the corresponding MLS surface, sharp features are significantly smoothed out. Although the bilateral denoising performs well in general, the sharp features are not able to be efficiently preserved as it actually uses a static window/kernel in the two domains. Due to take not only into account the sampling likelihoods of sample points but also the feature intensities while denoising PSS, our algorithm can deliver quality smoothing while preserving the surface features more efficiently than BIL or MLS. From the executing time listed in Table 1, we notice that our method is slower than MLS or BIL mainly because our method needs to compute the sampling likelihood and measure the feature intensity for each sample point.

Table 1 The related statistics

Fig.	Method	Iters.	Max. Error $(\cdot 10^{-4})$	Ave. Error $(\cdot 10^{-4})$	Exec time
5	BIL	2	64.7	14.2	6.38s
	MLS	1	57.2	12.1	70.26s
	Ours	2	49.5	11.3	103.51s
7	BIL	2	78.0	15.3	7.56s
	MLS	2	81.3	16.7	76.05s
	Ours	3	72.6	14.5	112.87s
8	BIL	1	50.1	5.8	8.84s
	MLS	1	43.8	6.7	140.33s
	Ours	1	35.3	4.3	184.62s

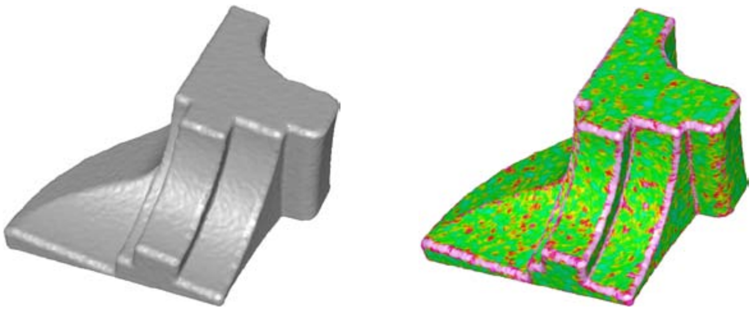
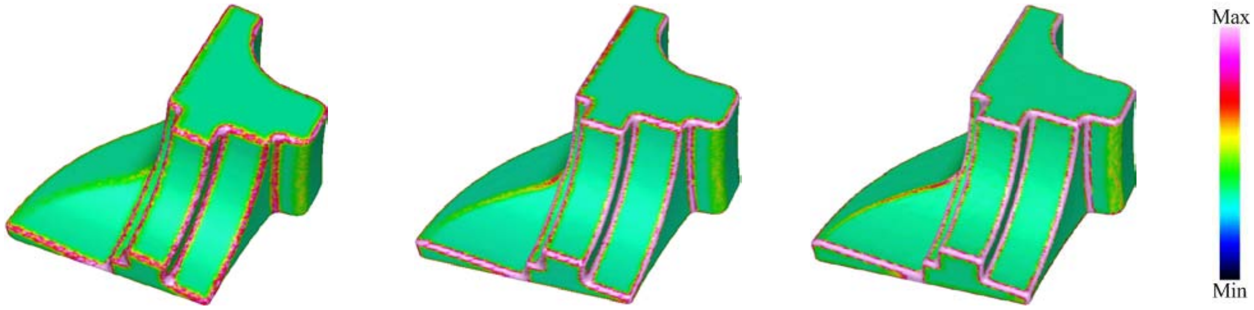


Fig. 4 Noisy Fandisk Model and mean curvature visualization



(b) MLS **(c) BIL** **(d) Ours**

Fig.5 Denoising the noisy Fandisk model. Top: the denoised models. Bottom: the corresponding denoised model colored by mean curvature.

6 Conclusion

In this paper, we presented a robust denoising algorithm for point-sampled surfaces. In terms of the MLS surfaces, the sampling likelihood of sample point is computed and the feature intensity of sample point is evaluated based on normal voting tensor. The point's normal is filtered by using the modified bilateral filtering. The point-sampled surfaces are smoothed by moving each sample point along its own filtered normal with an offset determined according to the combination of the sampling likelihood and feature intensity.

Our experimental results demonstrate that the proposed algorithm is robust, and can denoise the noise efficiently while preserving the surface features.

Acknowledgements

This project is supported by the Scientific Research Fund of Zhejiang Provincial Education Department of China under Grant Nos.20080511, 20080642.

References:

[1] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva T. Point set surfaces. In: Proceedings of IEEE Visualization, San Diego, California, 2001:21-28.

[2] Amenta N, Kil, Y. Defining Point set surfaces. ACM Trans. Graph, 2004, 23(3):264-270.

[3] Pfister H, Zwicker M, van Baar J, Gross M. Surfels: Surface elements as rendering primitives. In: Proceedings of ACM SIGGRAPH, 2000:335-342.

[4] Rusinkiewicz S, Levoy M. QSplat: A multiresolution point rendering system for large meshes. In: Proceedings of the SIGGRAPH, 2000:343-352.

[5] Botsch M, Kobbelt L. High-quality point-based rendering on modern GPUs. In: Proceedings of Pacific Graphics 2003:335-343.

[6] Pauly M, Keiser R, Kobblet L, Gross M. Shape modeling with point-sampled geometry. ACM Trans. Graph , 2003, 22(3):641-650.

[7] Pauly M, Keiser R, Gross M. Multi-scale feature extraction on poin-sampled surfaces. Computer Graphics Forum, 2003,22(3):281-289.

[8] Pauly M, Kobbelt LP, Gross M. Multiresolution Modeling of Point-Sampled Geometry. Technical Report, CS #379, ETH, Zurich 2002.

[9] Tobor I, Reuter P, Schlick C. Efficient Reconstruction of Large Scattered Geometric Datasets using the Partition of Unity and Radial Basis Functions. Journal of WSCG, 2004,12(1-3):467-474.

[10] Xiao C, Zheng W, Peng Q, Forrest, AR. Robust morphing of point-sampled geometry. Computer Animation and Virtual Worlds 2004,15:201-210.

[11] Miao Y Fen J, Xiao C, Li H, Peng Q. Detail-preserving local editing for point-sampled geometry. In:Proceeding of Computer Graphics International (CGI'06), 2006:673-681.

[12] Lin P, Lee T, Lin C. Free-form deformation for point-sampled surface. Journal of information science and engineering, 2007, 23(3):757-771.
[13] Pauly M, Kobbelt L, Gross M. Point-Based Multi-Scale Surface Representation. ACM Trans. on Graphics, 2006, 25(2):177-193.
[14] Chen Y, Lai S. A Partition-of-Unity Based Algorithm for Implicit Surface Reconstruction Using Belief Propagation. In:Proceedings of SMI '07, 2007,147-155.
[15] Daniels II J, Ha LK, Ochotta T, Silva CT. Robust Smooth Feature Extraction from Point Clouds. Proceedings of Shape Modeling International'07, 2007:123-136.
[16] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. In: Proc. of the Computer Graphics Annual Conf. Series, ACM SIGGRAPH, 2003:950-953.
[17] Hildebrandt K, Polthier K. Anisotropic filtering of non-linear surface features. Computer Graphics Forum, 2004, 23(3):391-400.
[18] Ji Z, Liu L, Wang G. Non-iterative global mesh smoothing with feature preservation International Journal of CAD/CAM, 2006, 6 (1):85-93.
[19] Taubin G. A signal processing approach to fair surface design. In: Proc. of the Computer Graphics. Annual Conf. Series, ACM SIGGRAPH,New York, NY, USA 1995:351-358.
[20] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proc. of the Computer Graphics. Annual Conf. Series, ACM SIGGRAPH,New York, NY, USA, 1999:317-324.
[21] Ohtake Y, Belyaev A, Bogaeski I. Polyhedral surface smoothing with simultaneous mesh regularization. In: Proceedings of Geometric Modeling and Processing, 2000:229-237.
[22] Peng J, Strela V, Zorin D. A simple algorithm for surface denoising. In: Proceedings of IEEE Visualization 2001.
[23] Liu X, Bao H, Shum H, Peng Q. A novel volume constrained smoothing method for meshes. Graphical Models, 2002, 64:169-182.
[24] Vollmer J, Mencl R, Müller H. Improved Laplacian smoothing of noisy surface meshes. In: Proceedings of Eurographics, 1999:131-138.
[25] Perona P, Malik J. Scale space and edge detection using anisotropic diffusion. IEEE Trans Pattern and Machine Intell, 1990,12 (7):629-639.
[26] Bajaj C L, Xu G. Anisotropic diffusion of surfaces and functions on surfaces. ACM Trans. on Graphics, 2003, 22(1):4-32.
[27] Clarenz U, Diewald U, Rumpf M. Anisotropic geometric diffusion in surface processing In: Proceedings of IEEE Visualization, 2000:397-405.
[28] Desbrun M, Meyer M, Schröder P, Barr A. Anisotropic feature-preserving denoising of height fields and bivariate data. In: Proceedings of Graphics Interface, 2000:145-152.
[29] Ohtake Y, Belyaev A, Seidel HP. Mesh smoothing by adaptive and anisotropic Gaussian filter. In: Proceedings of Vision, Modeling and Visualization, 2002:203-210.
[30] Taubin G. Linear anisotropic mesh filtering. IBM Research Technical Report, RC2213, 2001.
[31] Tasdizen T, Whitaker R, Burchard P, Osher S. Geometric surface smoothing via anisotropic diffusion of normals. In: Proceedings of IEEE Visualization, 2002:125-132.
[32] Clarenz U, Rumpf M, Telea A. Fairing of point based surfaces. In: Computer Graphics International(CGI'04), 2004:600-603.
[33] Lange C, Polthier K. Anisotropic smoothing of point sets. Comput.Aid Geomet.Des. 2005, 22 (7):680-692.
[34] Xiao C, Miao Y, liu S, Peng Q. A dynamic balanced flow for filtering point-sampled geometry. The Visual Computer, 2006, 22 (3):210-219.
[35] Mederos B, Velho L, de Figueiredo LH. Robust smoothing of noisy point clouds. In: Proc. of the SIAM Conf. on Geometric Design and Computing. Seattle: Nashboro Press 2003.
[36] Weyrich T, Pauly M, Keiser R, Heinzle S, Scandella S, Gross M. Post-processing of scanned 3D surface data. In: Proc. of Symp. on Point-Based Graphics 04, 2004: 85-94.
[37] Alexa M, Adamson A. On normals and projection operators for surfaces defined by point sets. In: Proc. of the Eurographics Symposium on Point-Based Graphics. Eurographics Association, 2004:149-155.
[38] Shimizu T, Date H, Kanai S, Kishinami T. A New Bilateral Mesh Smoothing Method by Recognizing Features In: Proc. of Ninth International Conference on CAD/CG'05, 2005:281-286.

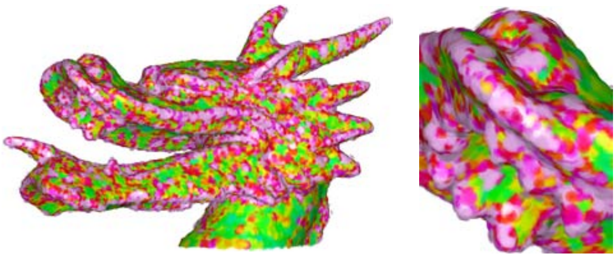
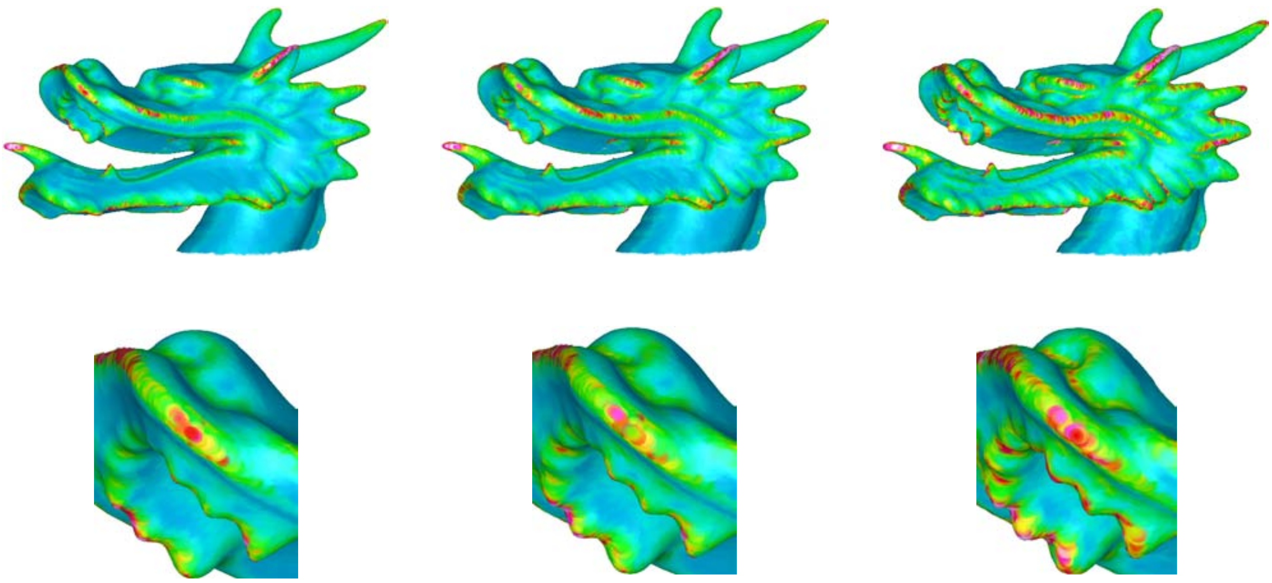
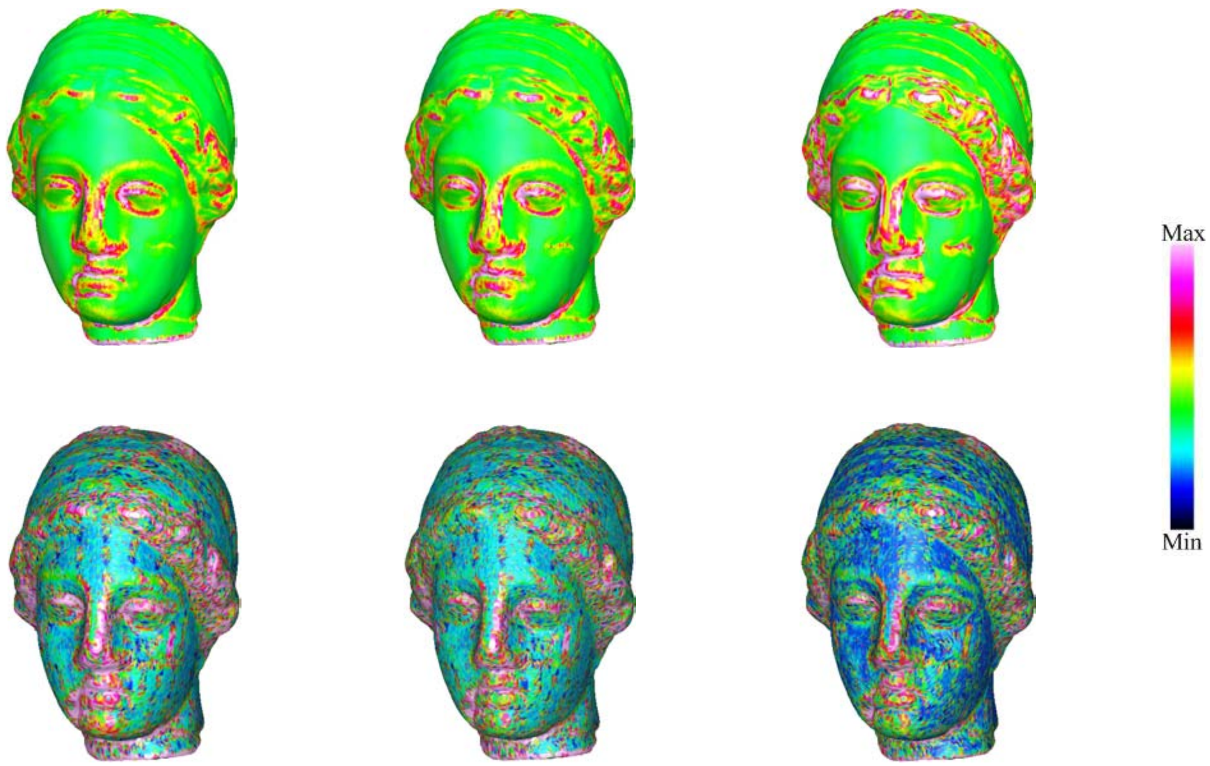


Fig.6 Noisy Dragon-head model is colored by mean curvature and the closer view of its upper jaw.



(b) MLS **(c) BIL** **(d) Ours**

Fig.7 Denoising noisy Drag-head model(Fig.3). Top: the denoised model colored by mean curvature. Bottom: the closer view of upper jaw of the corresponding denoised model.



(a) MLS **(b) BIL** **(c) Ours**

Fig.8 Denoising the noisy Igea model. Top: the denoised models. Middle: the corresponding denoised model colored by mean curvature. Bottom: the corresponding denoised model colored according to the entropy of the differences between the noisy and denoised models.