

Local Surface Descriptor for Geometry and Feature Preserved Mesh Denoising

Wenbo Zhao^{1,2}, Xianming Liu^{1,2*}, Junjun Jiang^{1,2}, Debin Zhao^{1,2}, Ge Li³, Xiangyang Ji⁴

¹Faculty of Computing, Harbin Institute of Technology, Harbin, China

²Peng Cheng Laboratory, Shenzhen, China

³School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China

⁴Department of Automation and BNRist, Tsinghua University, Beijing, China

Abstract

3D meshes are widely employed to represent geometry structure of 3D shapes. Due to limitation of scanning sensor precision and other issues, meshes are inevitably affected by noise, which hampers the subsequent applications. Convolutional neural networks (CNNs) achieve great success in image processing tasks, including 2D image denoising, and have been proven to own the capacity of modeling complex features at different scales, which is also particularly useful for mesh denoising. However, due to the nature of irregular structure, CNNs-based denoising strategies cannot be trivially applied for meshes. To circumvent this limitation, in the paper, we propose the local surface descriptor (LSD), which is able to transform the local deformable surface around a face into 2D grid representation and thus facilitates the deployment of CNNs to generate denoised face normals. To verify the superiority of LSD, we directly feed LSD into the classical Resnet without any complicated network design. The extensive experimental results show that, compared to the state-of-the-arts, our method achieves encouraging performance with respect to both objective and subjective evaluations.

Introduction

3D sensing and scanning techniques have been extensively employed to capture deformable surfaces of real physical objects, which are typically represented as meshes. These digital representations facilitate a variety of applications, including movie industry, digital games, virtual reality and so on. However, due to limitation of scanning sensor precision and computational issues, the derived 3D meshes inevitably contain noise, which hinders the subsequent applications. Mesh denoising is thus one of the most fundamental research topics in 3D geometry processing.

Mesh denoising is a challenging ill-posed problem. The desired denoising result owes to the ability of smoothing a noisy surface, as well as preserving the “original” object features, without introducing unnatural geometric distortions. Due to the brisk demand coming from different applications, there exist a rich literature on mesh denoising, ranging from filtering-based (Zheng et al. 2011; Zhang et al. 2015), optimization-based (He and Schaefer 2013; Wang et al.

2014) to the most recent data-driven methods (Wang, Liu, and Tong 2016; Li et al. 2020a). In particular, data-driven strategies for mesh denoising have gained great interest recently, boosted by the success of deep learning in 2D image denoising (Zhang et al. 2017).

Convolutional neural networks (CNNs) have massively impacted denoising tasks in 2D images, and are ubiquitous in many state-of-the-art approaches. CNNs have been proven to own the ability of modeling complex features at different scales, which is also very useful for mesh denoising since meshes contain multi-scale features, such as large-scale curvature changes, fillet radii, small scale details and corners. However, the application of CNNs-based denoising strategy to mesh denoising is not straightforward. CNNs require regular 2D grid as input, while meshes exhibit irregular 3D topology. Moreover, compared to point clouds, 3D meshes contain vertex connectivity in addition to vertex coordinates. These issues prevent the deployment of CNNs in mesh denoising. Accordingly, different from 2D image denoising in which CNNs-based strategy has become the basic methodology (Zhang et al. 2017), to the best of our knowledge, there are only a few deep learning based schemes for mesh denoising (Wang, Liu, and Tong 2016; Li et al. 2020a,b; Armando, Franco, and Boyer 2020; Zhao et al. 2021; Shen et al. 2021).

To denoise a mesh, a common practice among the most efficient methods is to first denoise the face normals and then update the vertex positions accordingly, so as to benefit from the scale invariance of the face normals (Fleishman, Drori, and Cohen-Or 2003; Sun et al. 2007; Armando, Franco, and Boyer 2020). Existing CNNs-based methods mostly adopt this strategy, which train networks to regress face normals. For instance, in (Li et al. 2020a), an end-to-end mesh normal denoising network called as DNF-Net is proposed, which takes patches of facet normals as inputs and directly outputs the corresponding denoised facet normals. However, in this method, the noisy normals are simply organized as matrices, which discard the surface geometry information, leading to lost in discriminative representation. To achieve satisfactory denoising performance, the proposed DNF-Net is carefully designed, which includes a tailored multi-scale feature embedding unit, a residual learning strategy to remove noise, and a deeply-supervised joint loss function.

Another feasible manner is to extend convolutional layers

*Corresponding author: csxm@hit.edu.cn

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

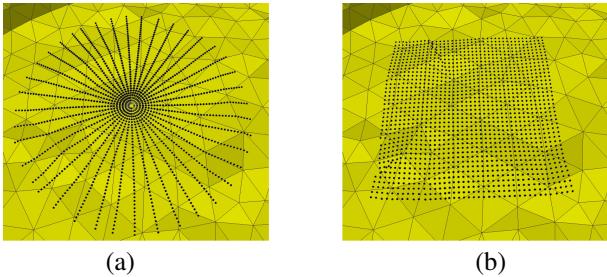


Figure 1: Sample 40×40 points with (a) uniformly distributed polar coordinates, in which the sampling density tends to be sparser with increased r . (b) with uniformly distributed virtual Cartesian coordinates, in which the sampling density remains unchanged with (vx, vy) .

of standard CNNs to graph, such that graph convolutional networks (GCNs) can be employed. For instance, in (Armando, Franco, and Boyer 2020), the noisy normals and the noisy positions are organized into a graph. A fully end-to-end learning strategy is then performed based on graph convolutions, which operates on a graph of facets, directly on the existing topology of the mesh, and follows a multi-scale design to extract geometric features at different resolution levels. The inputs of the network also discard the surface geometry information of faces. Moreover, this network operates on the whole mesh, and thus is limited in the ability of local feature preservation. Similarly, in (Shen et al. 2021), GCN-Denoiser is proposed very recently, which takes a triangular mesh as input and employs multiple GCNs to progressively regress the noise-free normals of the underlying surface patches. Although these methods achieve promising results, the performance of which largely relies on the power of GCNs. Unfortunately, GCNs itself is a tool under development, whose maturity is far from CNNs.

As reviewed above, it can be found that the state-of-the-art neural network based methods are relying on carefully-tailored CNN/GCN network. A natural question is raised: do we have to fall into the arms race of network architecture design like that are taking place in image denoising? In this paper, we revisit this issue and argue that the performance gain can also come from well-organized face normals. The main contributions of this work are highlighted as follows:

- We propose the local surface descriptor (LSD) to transform normals of local deformable surface into 2D grid representation. Compared to other deep learning based schemes, LSD contains both face normal and surface geometry information, and thus owns stronger discriminative representation ability.
- We propose the LSD-net, which straightforwardly leverages the classical Resnet for mesh denoising without the need of designing elaborated network structure, since LSD has regular 2D form.
- We provide extensive experimental comparison with the state-of-the-art methods on multiple datasets to show that our scheme achieves the best mesh denoising performance so far with respect to average objective metric and

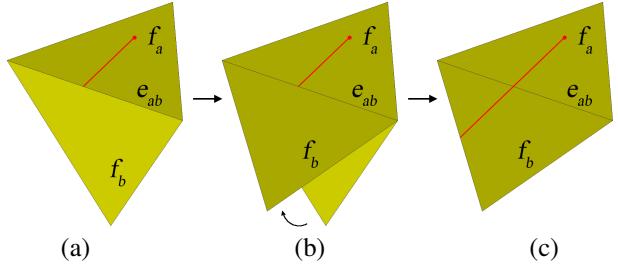


Figure 2: An example of geodesic propagation. (a) The geodesics starts from the centre of f_a and reaches e_{ab} . (b) f_b is rotated to the same plane as f_a around e_{ab} . (c) The geodesics is continued on f_b .

subjective metric, which demonstrates the power of LSD.

Local Surface Descriptor

In this section, we introduce in detail the generation process of local surface descriptor (LSD). The goal of LSD is to transform the local structure around a face from 3D to 2D while preserving the the geometric information, which is achieved by uniformly sampling a set of points on the 3D irregular surface and organizing the normals of sample points into a 2D matrix.

Notations. Given a face f in a noisy mesh with c as its center, which is regarded as the target face, we attempt to sample a set of points $\{p_{(i,j)}\}, i, j \in [-t_s, t_s]$ from the surface around c_i , where t_s is a parameter that controls the number of sampling points. The normals $\{\mathbf{n}_{(i,j)}\}$ of these points are then organized as the local surface descriptor \mathbf{L} of f , which further serves as the input to the denoising neural network.

Coordinate System Selection. There is a fundamental challenge for sampling points on mesh: how to build a coordinate system $\{\mathbf{s}\} = \{s_1, s_2, \dots, s_n\}$ to which a certain point on the mesh surface is projected to facilitate the sampling process? When the surface is planar, the sampling process can be straightforwardly done by building a 2D Cartesian coordinate system on the surface. However, this no longer works well for the case of noisy surface since it generally has non-trivial curvature. In (Kokkinos et al. 2012), an alternative approach was proposed to build a polar coordinate system $\{\mathbf{s}\} = \{r, \varphi\}$ on the mesh surface by shooting geodesic with the radial coordinate r and the angular coordinate φ , which has the ability of handling curved surface. The surface is then segmented into multiple bins, and the intrinsic equivalents in each bin are transformed as the shape context descriptor. However, there is a certain drawback associated with the use of polar coordinate. When sampling points are with uniformly distributed (r, φ) , the sampling density tends to be sparser with increased r , as shown in Fig. 1-(a), leading to the loss of geometry detail information. In contrast, as shown in Fig. 1-(b), in the Cartesian coordinate system $\{\mathbf{s}\} = \{x, y\}$, the sampling density remains unchanged with uniformly distributed (x, y) .

Local Surface Descriptor Generation. Considering the pros and cons of these two coordinate systems, we generate the local surface descriptor via a coordinate transforma-

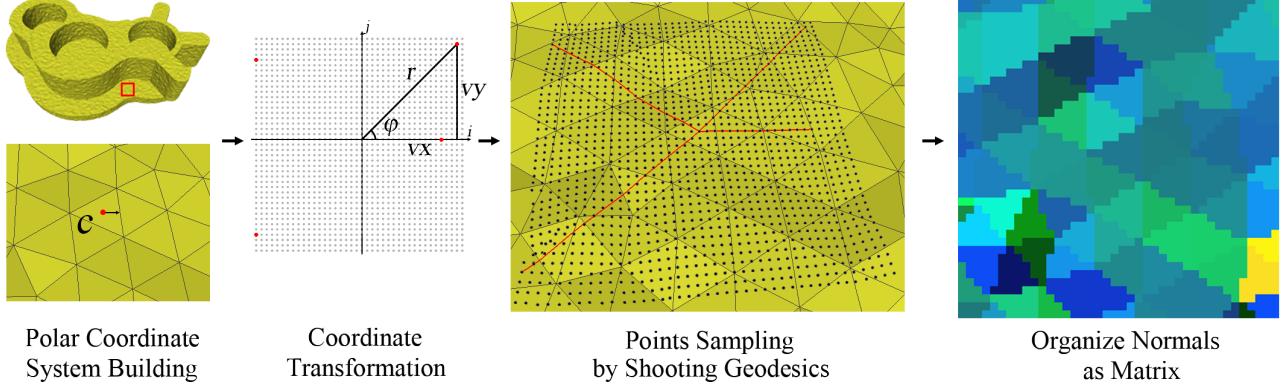


Figure 3: An illustration of LSD generation.

tion strategy, which integrates the respective superiority of the Cartesian and polar coordinate systems. As illustrated in Fig. 3, it includes the following main steps:

- **Step 1: Polar Coordinate System Building.** Firstly, a polar coordinate system $\{s\} = \{r, \varphi\}$ is built on f with c as the pole. Point sampling is done by shooting geodesic according to (r, φ) . The orientation of polar axis can be arbitrary. In this work, we set the ray from c to one of the midpoint of f 's edges as the polar axis.
- **Step 2: Coordinate Transformation.** Secondly, a polar coordinate set $\{(r, \varphi)\}$ of sampling points $\{p_{(i,j)}\}$ are generated via the coordinate transformation strategy. Specifically, a virtual Cartesian coordinate system $\{(vx, vy)\}$ is built firstly. Considering that the scales of meshes are various, the distance between sampling points should be adaptive to the scale, thus the corresponding virtual Cartesian coordinate of $p_{(i,j)}$ is set to:

$$(vx, vy) = \left(\frac{d_a}{p_s} i, \frac{d_a}{p_s} j \right) \quad (1)$$

where d_a is the average Euclidean distance of adjacent face centers in the noisy mesh, which can be regarded as an estimation of scale; p_s is the parameter that controls the precision of sampling. Then the polar coordinate (r, φ) of $p_{(i,j)}$ is computed as:

$$r = \sqrt{(vx)^2 + (vy)^2} \quad (2)$$

$$\varphi = \arctan \frac{vy}{vx}$$

The polar coordinate set $\{(r, \varphi)\}$ forms a square that can preserve uniform sampling density.

In this way, we achieve the consistency of sampling density regardless of the radial coordinate r , as illustrated in Fig. 1-(b).

- **Step 3: Points Sampling by Shooting Geodesics.** With the transformed polar coordinate set, we then perform point sampling on the mesh surface. It is worth noting that, for a point in the curved surface, even though we know its polar coordinate (r, φ) , it is not straightforward to be located. We thus leverage the following propagation

process for locating the points to sample: At the beginning, a geodesic is shot from c_i with angle φ . When the geodesic reaches an edge, it is further propagated to the adjacent face by the standard unfolding procedure (Bronstein, Bronstein, and Kimmel 2006; Kimmel and Sethian 1998), as shown in Fig. 2. Specifically, supposing that the geodesic is in f_a , the adjacent face is denoted as f_b and the edge between them is e_{ab} . f_b is then rotated to the same plane as f_a around e_{ab} , which enables the geodesic to be continued on the plane until it reaches another edge. This process is repeated until the total length of the geodesic is equal to r and the terminal point is regarded as the sampling point.

- **Step 4: Organize Normals as Matrix.** Finally, the normals $\{\mathbf{n}_{(i,j)}\}$ of all the sampling points $\{p_{(i,j)}\}$ are organized into a matrix according to their subscripts, which serves as the LSD \mathbf{L} . To enable the generated LSD have the property of rotation-invariance, we propose to apply a rotation matrix \mathcal{R} to normalize $\{\mathbf{n}_{(i,j)}\}$ in \mathbf{L} . Suppose \mathbf{n}_t is a fixed direction, we propose to rotate the normal \mathbf{n} of the target face f to \mathbf{n}_t , and the rotation angle is θ . Considering that the normal is corrupted with noise, we propose to use the average normal \mathbf{n}^* of the 2-ring faces around f as an estimate of \mathbf{n} . We then compute \mathcal{R} by the Rodrigues' rotation formula (Liang 2018):

$$\mathcal{R} = \mathbf{I} + (\sin\theta) \mathbf{N} + (1 - \cos\theta) \mathbf{N}^2 \quad (3)$$

where \mathbf{N} is the skew-symmetric cross-product matrix of $\mathbf{n}^* \times \mathbf{n}_t$. \mathcal{R} is further applied to $\{\mathbf{n}_{(i,j)}\}$ to obtain the normalized ones:

$$\tilde{\mathbf{n}}_{(i,j)} = \mathcal{R} \mathbf{n}_{(i,j)} \quad (4)$$

which are then organized as a matrix that serves as the final LSD $\tilde{\mathbf{L}}$.

An example of LSD generation is illustrated in Fig. 3. In this case, we set $t_s = 20$ and $p_s = 8$. To verify the effectiveness of shooting geodesics, four points are marked with red in Step 2, and the corresponding geodesics are drawn in Step 3. It can be observed that the geodesics can be propagated on the noisy surface, and the sampling points are regularly distributed. In Step 4, the LSD is visualized by normalized

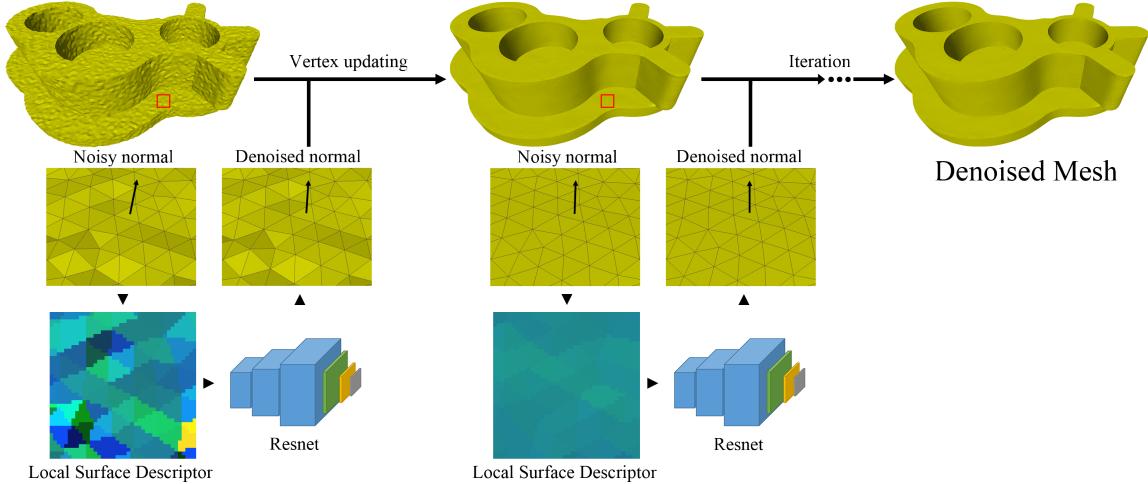


Figure 4: The framework of LSD-net. The noisy mesh will be iteratively processed. During each iteration, the LSD of each face is built firstly. Then it is fed to the Resnet to obtain denoised normal. Finally, the new coordinates of the vertices are updated according to the denoised normals.

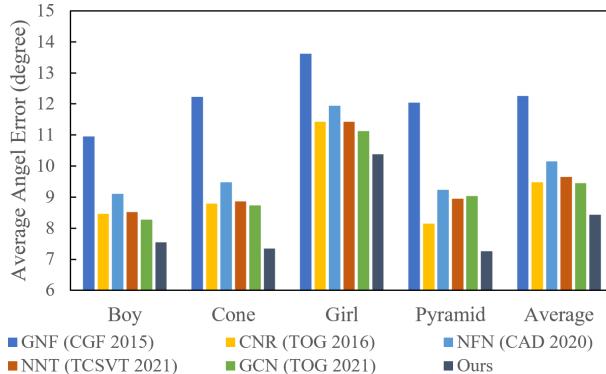


Figure 5: The objective comparison on the Kinect V1 dataset. The lower, the better.

normal values to RGB. As illustrated, the LSD is a piecewise smooth 2D grid representation, which preserves both the normal and geometry information.

Mesh Denoising Based on LSD

With the generated LSD, the irregular mesh structure is transformed to regular 2D grid. It facilitates the following normal denoising process, since we can directly leverage the popular CNNs-based network for our purpose. In this section, we apply the classical Resnet (He et al. 2016) on LSD for mesh denoising, which is referred to as LSD-net.

Overall Pipeline. LSD-net follows the popular iterative normal filtering framework, which is repeated for N_f times. Each iteration consists of the following three main steps: 1) The normalized LSD $\tilde{\mathbf{L}}$ of each face f is generated. 2) The denoised normal $\hat{\mathbf{n}}$ is obtained by employing a Resnet with $\tilde{\mathbf{L}}$ as the input:

$$\hat{\mathbf{n}} = \mathcal{R}^{-1} \text{RESNET}^{(k)} (\tilde{\mathbf{L}}) \quad (5)$$

where k represents the iteration number. Note that here the denoised normals should be restored by rotating with the inverse matrix of \mathcal{R} before vertex updating. 3) The position \mathbf{p}_v of vertex v is updated N_v times according to the restored denoised normals (Zhang et al. 2015):

$$\mathbf{p}'_v = \mathbf{p}_v + \frac{1}{|f_v|} \sum_{\mathbf{n} \in f_v} \hat{\mathbf{n}} (\hat{\mathbf{n}} \cdot (\mathbf{c} - \mathbf{p}_v)) \quad (6)$$

where $|f_v|$ is the set of faces that contain v as one of the vertexes, \mathbf{c} and \mathbf{n} are the centre and normal of a face in $|f_v|$.

Network Structure. We employ a Resnet with three residual blocks to process the input LSD $\tilde{\mathbf{L}}$. Each of the residual block contains seven 3×3 convolution layers. The first layer is with a stride of 2, which performs down-sampling. Three residual connections are introduced to connect the output of layer 1, 3, 5 to layer 3, 5, 7. The channel numbers of each block are 32, 48 and 64, respectively. Then, a global average-pooling layer and two fully connected (FC) layers are applied, the channel numbers of the two FC layers are 128 and 3 to directly generate the denoised normal. All the convolution and FC layers are equipped with ReLU and batch normalization, expecting the last FC layer which is equipped with Tanh to ensure that the output lies in [-1,1], aiming to output the denoised normal.

Network Training. We separately train three series of $\text{RESNET}^{(k)}$ corresponding to the training sets released by (Wang, Liu, and Tong 2016), including the Synthetic set (60 meshes), Kinect V1 set (72 meshes) and Kinect V2 set (72 meshes). For a certain series, the training input of $\text{RESNET}^{(1)}$ is obtained by randomly choosing 800,000 faces from the training set and generating their normalized LSDs, while their corresponding ground truth normals are employed as the targets. Notice that the ground truth normals should also be rotated with \mathcal{R} to ensure consistency. Then $\text{RESNET}^{(1)}$ is trained with the mean-squared-error (MSE) loss between the denoised normals and the target

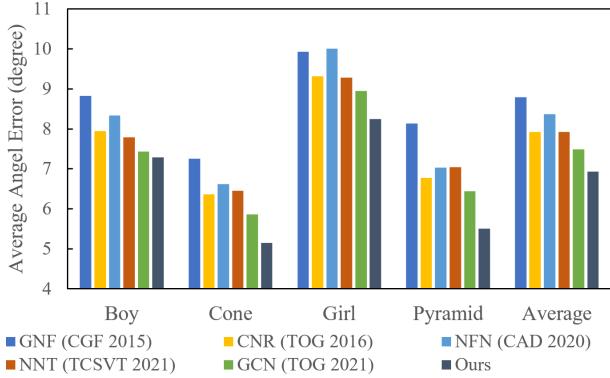


Figure 6: The objective comparison on the Kinect V2 dataset. The lower, the better.

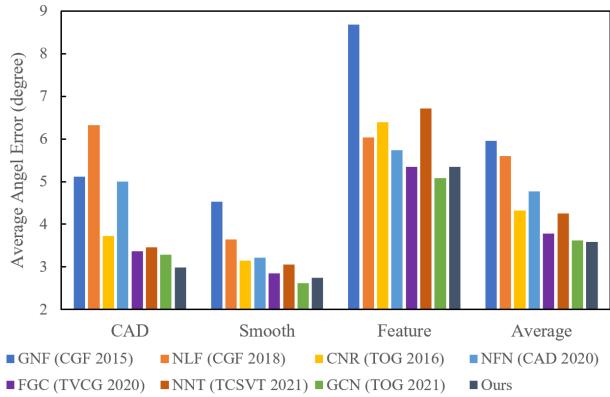


Figure 7: The objective comparison on the synthetic dataset. The lower, the better.

ones, using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate = 0.0001, batch size = 80). The following network $\text{RESNET}^{(k+1)}$ is trained with the same parameters on the new training set generated by applying $\text{RESNET}^{(k)}$ on the former training set.

Experimental Results

In this section, extensive experimental results are provided to demonstrate the superiority of our method compared with the state-of-the-arts. All the experiments are conducted on a server with two Tesla V100 GPUs.

Comparison Group. We compare our LSD-net with several state-of-the-art schemes in terms of both objective and subjective criterion:

- Learning-based schemes: 1) Cascaded normal regression (CNR) (Wang, Liu, and Tong 2016), 2) Facet graph convolutions (FGC) (Armando, Franco, and Boyer 2020), 3) Normalf-net (NFN) (Li et al. 2020b), 4) NormalNet (NNT) (Zhao et al. 2021), 5) GCN-Denoiser (GCN) (Shen et al. 2021).
- Traditional schemes: 6) Guided normal filtering (GNF) (Zhang et al. 2015), 7) Non-local low-rank normal filtering (NLF) (Li et al. 2018).

Dataset	Kinect V1	Kinect V2	Synthetic	Scanned
N_f	4	3	2	2
N_v	20	20	20	20

Table 1: The settings of N_f and N_v for different datasets.

The source codes and per-trained models of CNR, FGC, GNF and NLF are kindly released by their authors or implemented by a third party. We adopt their default parameters in experiments. The denoising results of NFN, NNT and GCN are provided by their authors.

Parameters Setting. For the parameters of generating LSD, we set $t_s = 40$ and $\alpha_c = 8$. The direction of \mathbf{n}_t can be arbitrarily chosen, so we set $\mathbf{n}_t = (1, 0, 0)$. Under these conditions, the proposed LSD contains a large region (8-9 ring around the central face) on the surface and provides enough information to infer the denoised normal.

We validate our scheme on four datasets: 1) Kinect V1 dataset; 2) Kinect V2 dataset; 3) Synthetic dataset; 4) Real-scanned dataset. The parameters of iteration are individually set for different datasets to achieve better results, which are shown in Table 1. In the following, we provide the detailed comparison results.

Results on Kinect Meshes

Dataset Details. We first provide the comparison on the Kinect V1/V2 datasets (Wang, Liu, and Tong 2016), which consist of 73 and 72 scanned meshes. These meshes are captured from 4 sculptures: *Boy* (24 meshes), *Girl* (25 Kinect V1 and 24 Kinect V2 meshes), *Cone* (12 meshes) and *Pyramid* (12 meshes). Since the authors of FGC only provide the pre-trained model of Synthetic dataset, and the source code of NLR can only process the watertight meshes, FGC and NLR are not included into this comparison.

Comparison Results. We first provide the objective comparisons of the Kinect V1 dataset in Fig. 5 and Kinect V2 dataset in Fig. 6 with respect to the average angle error. The meshes in the Kinect V1/V2 dataset have different noise patterns from the Gaussian noise. LSD is capable of indicating the pattern by preserving the geometry information in a large region. Therefore, LSD-net achieves the best performance on all the categories in the two datasets.

Then the subjective comparison of Kinect V1 dataset is shown in Fig. 8, including *Cone16* and *Pyramid02*. It can be observed that the noise has a step-like appearance, which will be treated as pseudo-feature by the traditional scheme GNF. On the other hand, most of the learning-based schemes can avoid this and recover the smooth part in the red box but failed on recovering the feature. Due to the preserving of geometry information, LSD-net can explore the hidden structure under noise, leading to the best feature recovery results. Similar denoising results can be observed in Fig. 9, including Kinect V2 meshes *Boy02* and *Girl24*, LSD-net consistently explores the feature under noise.

Results on Synthetic Meshes

Dataset Details. Secondly, we conduct the comparison on the synthetic dataset (Wang, Liu, and Tong 2016), which

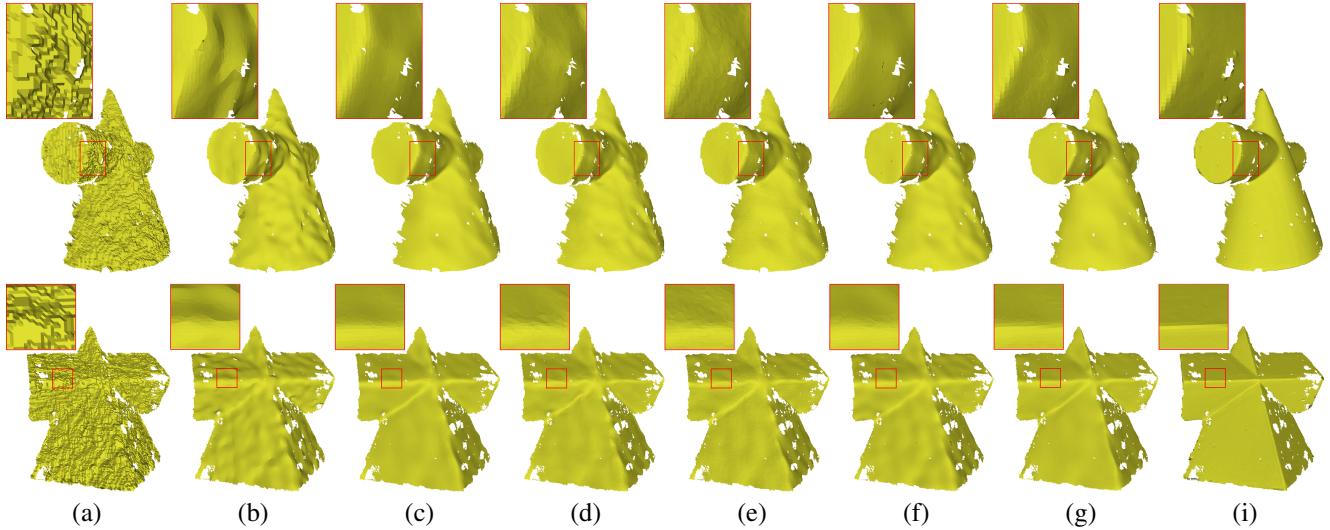


Figure 8: Denoising results of meshes *Cone16* and *Pyramid02* from Kinect V1 dataset.(a) noisy meshes; (b) to (g) denoising results of GNF, CNR, NFN, NNT, GCN and LSD-net; (i) ground truth. Please enlarge the PDF to see more details.

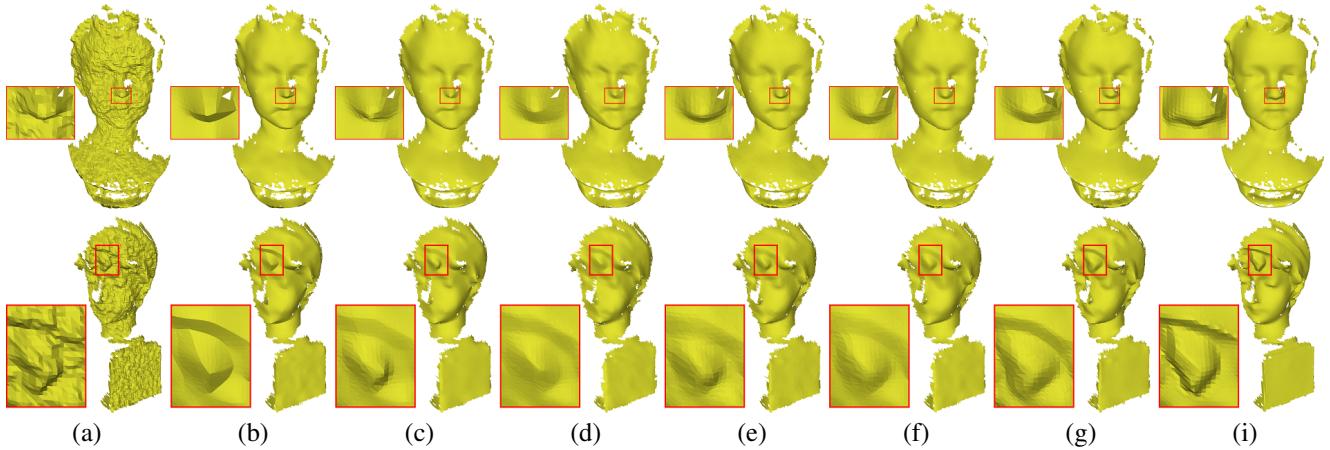


Figure 9: Denoising results of meshes *Boy02* and *Girl24* from Kinect V2 dataset. (a) noisy meshes; (b) to (g) denoising results of GNF, CNR, NFN, NNT, GCN and LSD-net; (i) ground truth. Please enlarge the PDF to see more details.

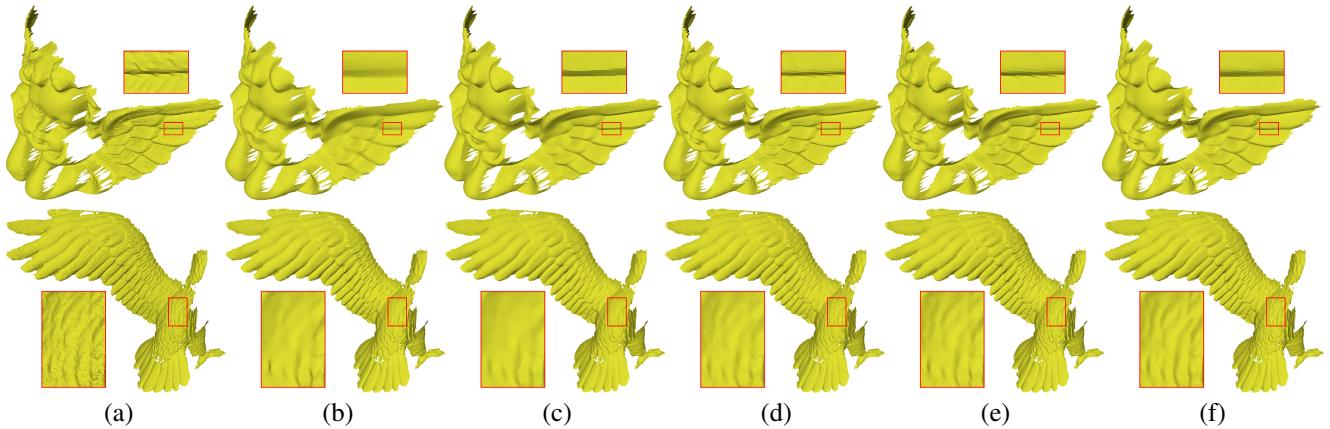


Figure 10: Denoising results of real-scanned meshes *Angel* and *Eagle*. (a) noisy meshes; (b) to (f) are the results of CNR, NNT, FGC, GCN and LSD-net. Please enlarge the PDF to see more details.

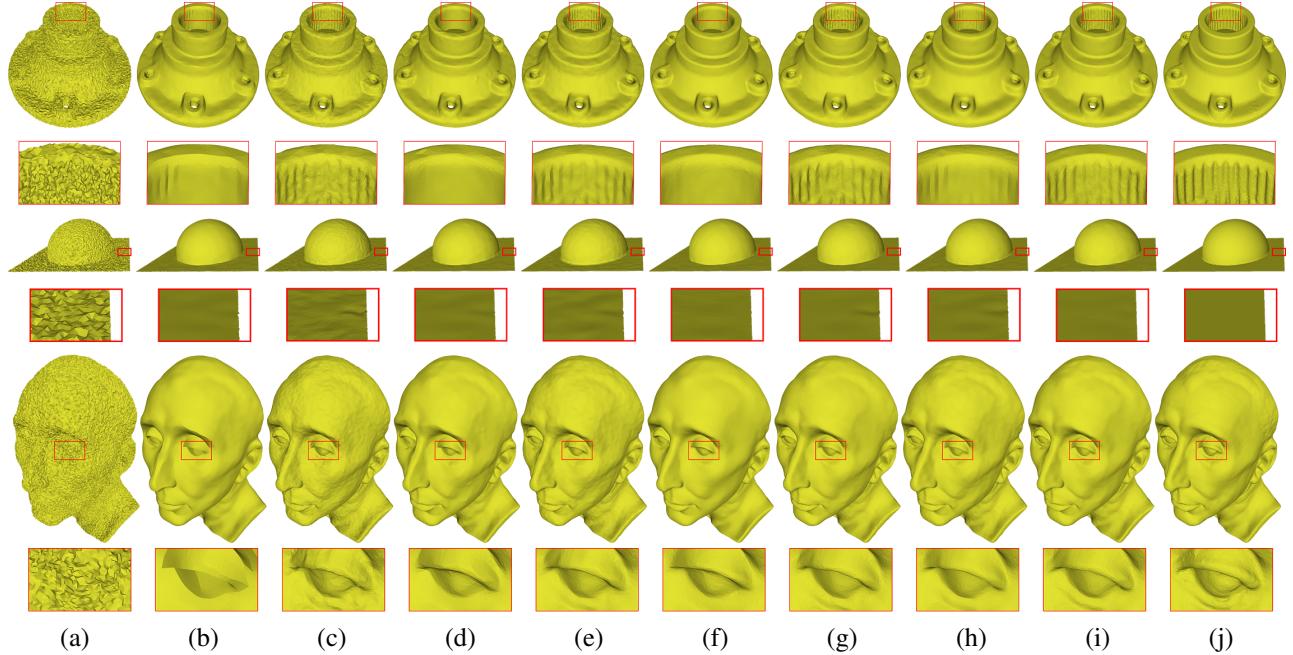


Figure 11: Denoising results of CAD mesh *Carter100K*, smooth mesh *Plane-sphere* and feature mesh *Nicolo* from synthetic dataset. (a) noisy meshes; (b) to (i) denoising results of GNF, NLF, CNR, NFN, NNT, FGC, GCN and ours; (j) ground truth.

consists of 29 noise-free meshes. These meshes are corrupted by Gaussian noise with three standard deviations: 0.1, 0.2 and 0.3 of the average edge length of each mesh. Therefore, there are 87 test meshes in total and they are divided into three categories: CAD (42 meshes with sharp edges), smooth (21 meshes with curved surface and plane), and feature (24 meshes with rich details). All the compared schemes are tested on this dataset.

Comparison Results. We first provide the objective comparison in Fig. 7. As illustrated, our scheme achieves the best average denoising performance compared with other state-of-the-art methods. It is worth noting that GCN introduces bilateral filtering-based normal refinement after obtaining the denoised normals to improve the overall performance. However, this filtering-based refinement cannot distinguish small-scale features from noise, which results in over-smoothing outputs as shown in Fig. 10 and 11.

Then we provide the subjective comparison in Fig. 11, including CAD mesh *Carter100K*, smooth mesh *Plane-sphere* and feature mesh *Nicolo*. For *Carter100K*, GCN over-smooths the internal gear in the red box which agrees with the findings above. NLF, NFN and FGC preserve the internal gear but fail on noise removal. For *Plane-sphere* and *Nicolo*, LSD-net shows the capability of recovering smooth region as well as detailed regions.

Results on Real-scanned Meshes

Dataset Details. Finally, we offer the comparison results on two real-scanned meshes, *Angel* and *Eagle*, which are provided by (Zhang et al. 2015) and (Yadav, Reitebuch, and Polthier 2018). There are no ground truth meshes provided,

and thus we cannot provide objective comparison results.

To evaluate the generalization capability, the compared methods are trained on the Synthetic dataset. Since the authors of NFN do not provide the denoising results of these meshes, it is not included into this comparison.

Comparison Results. As illustrated in Fig. 10, for *Angel*, CNR over-smooths the structure. FGC and GCN can preserve the structure but fail on removing the noise. LSD-net achieves a good balance between noise smoothing and feature preservation. For *Eagle*, LSD-net can preserve the small-scale feature well, while the other schemes over-smooth the region in the red box.

Conclusion

In this work, we presented a local surface descriptor, which is capable of transforming normals of local deformable surface into 2D grid representation. In this way, we can straightforwardly exploit the the classical Resnet for mesh denoising without the requirement of designing elaborated network structure. Extensive experimental results show that, compared to the state-of-the-art mesh denoising schemes, our method achieves the best average objective results and can effectively recover fine-scale features and avoid introducing pseudo-features in subjective comparisons.

Acknowledgements

This work was supported by National Key R&D Program of China under Grant 2019YFE0109600, and National Natural Science Foundation of China under Grants 61922027, 61932022, 62071155.

References

- Armando, M.; Franco, J.-S.; and Boyer, E. 2020. Mesh Denoising with Facet Graph Convolutions. *IEEE Transactions on Visualization and Computer Graphics*.
- Bronstein, A. M.; Bronstein, M. M.; and Kimmel, R. 2006. Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing*, 28(5): 1812–1836.
- Fleishman, S.; Drori, I.; and Cohen-Or, D. 2003. Bilateral mesh denoising. In *ACM Transactions on Graphics*, volume 22, 950–953.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.
- He, L.; and Schaefer, S. 2013. Mesh denoising via L0 minimization. *ACM Transactions on Graphics*, 32(4): 64.
- Kimmel, R.; and Sethian, J. A. 1998. Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences*, 95(15): 8431–8435.
- Kokkinos, I.; Bronstein, M. M.; Litman, R.; and Bronstein, A. M. 2012. Intrinsic shape context descriptors for deformable shapes. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 159–166.
- Li, X.; Li, R.; Zhu, L.; Fu, C.-W.; and Heng, P.-A. 2020a. DNF-Net: a Deep Normal Filtering Network for Mesh Denoising. *IEEE Transactions on Visualization and Computer Graphics*.
- Li, X.; Zhu, L.; Fu, C.-W.; and Heng, P.-A. 2018. Non-Local Low-Rank Normal Filtering for Mesh Denoising. In *Computer Graphics Forum*, volume 37, 155–166. Wiley Online Library.
- Li, Z.; Zhang, Y.; Feng, Y.; Xie, X.; Wang, Q.; Wei, M.; and Heng, P.-A. 2020b. NormalF-Net: Normal filtering neural network for feature-preserving mesh denoising. *Computer-Aided Design*, 102861.
- Liang, K. K. 2018. Efficient conversion from rotating matrix to rotation axis and angle by extending Rodrigues' formula. *arXiv preprint arXiv:1810.02999*.
- Shen, Y.; Fu, H.; Du, Z.; Chen, X.; Burnaev, E.; Zorin, D.; Zhou, K.; and Zheng, Y. 2021. GCN-Denoiser: Mesh Denoising with Graph Convolutional Networks. *ACM Transactions on Graphics*.
- Sun, X.; Rosin, P.; Martin, R.; and Langbein, F. 2007. Fast and effective feature-preserving mesh denoising. *IEEE Transactions on Visualization and Computer Graphics*, 13(5).
- Wang, P.-S.; Liu, Y.; and Tong, X. 2016. Mesh Denoising via Cascaded Normal Regression. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 35(6).
- Wang, R.; Yang, Z.; Liu, L.; Deng, J.; and Chen, F. 2014. Decoupling noise and features via weighted L1-analysis compressed sensing. *ACM Transactions on Graphics*, 33(2): 18.
- Yadav, S. K.; Reitebuch, U.; and Polthier, K. 2018. Mesh Denoising Based on Normal Voting Tensor and Binary Optimization. *IEEE Transactions on Visualization and Computer Graphics*, 24(8): 2366–2379.
- Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7): 3142–3155.
- Zhang, W.; Deng, B.; Zhang, J.; Bouaziz, S.; and Liu, L. 2015. Guided mesh normal filtering. In *Computer Graphics Forum*, volume 34, 23–34. Wiley Online Library.
- Zhao, W.; Liu, X.; Zhao, Y.; Fan, X.; and Zhao, D. 2021. NormalNet: Learning-based Mesh Normal Denoising via Local Partition Normalization. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.
- Zheng, Y.; Fu, H.; Au, O. K.-C.; and Tai, C.-L. 2011. Bilateral normal filtering for mesh denoising. *IEEE Transactions on Visualization and Computer Graphics*, 17(10): 1521–1530.