

IMAGE-BASED 3D MESH DENOISING THROUGH A BLOCK MATCHING 3D CONVOLUTIONAL NEURAL NETWORK FILTERING APPROACH

Gerasimos Arvanitis¹, Aris S. Lalos² and Konstantinos Moustakas¹

¹Electrical and Computer Engineering Department, University of Patras, Greece

²Industrial Systems Institute, “ATHENA” Research Center

Emails: {arvanitis, moustakas}@ece.upatras.gr, lalos@isi.gr

ABSTRACT

Throughout the years, several works have been proposed for 3D mesh denoising. Nevertheless, despite their reconstruction quality, there are still challenges related to the preservation of the fine surface features. Motivated by the impressive results of image denoising by 3D transform-domain collaborative filtering (CF), we extend it to 3D mesh denoising. CF is also capable of revealing the finest details shared by grouped blocks while preserving at the same time the unique features of each block. A new promising approach suggests unrolling the computational pipeline of CF into a convolutional neural network (CNN) structure increasing significantly the efficiency of this solution. In this paper, we successfully extend and apply this method to 3D meshes making a transition from face normals to pixels. Extensive evaluation studies carried out using a variety of 3D meshes verify that the proposed approach achieves plausible reconstruction outputs and provides very promising results.

Index Terms— 3D mesh denoising, CNNs of BM3D filtering, Image-based denoising of 3D normals.

1. INTRODUCTION

Despite the impressive results of recent 3D mesh denoising approaches [1, 2, 3], algorithms that are used in the area of image processing continue to inspire the area of 3D mesh processing. Motivated by this observation, we focus on converting the problem of mesh to image denoising using very robust approaches that have been used and successfully tested in this area. 3D transform-domain CF (e.g., 3D block matching - BM3D [4]) is an approach that achieves state-of-the-art image denoising performance by providing a 3D estimation that consists of jointly filtered grouped image blocks. CF is a special procedure that deals with 3D groups formed by similar 2D fragments of the image, including the following steps: 3D transformation of a group, shrinkage of the transform spectrum, and inverse 3D transformation. The CF of the grouped blocks, reveals the details which are common between blocks, since for each pixel, we obtain several different estimates that need to be combined. This method has been successfully applied in a large number of applications

including image/video denoising [5, 6] deblurring, superresolution, and compression. CF approaches process the noisy images by successively extracting reference blocks and by matching blocks that are similar to the reference one, forming a 3D group matrix. After performing a 3D transform to the group formed by the overlapping blocks and attenuating the noise by hard thresholding, an aggregation step is taking place in order to form the estimate of the whole image. Recent approaches [7] suggest substituting the 3D transform-domain processing with traditional CNNs, outperforming the original BM3D approaches in terms of computational efficiency. The BM3D filtering of images and its adoption to meshes have an intuitive formulation, which leads to a simple data-driven method that addresses issues that stem from the two dimensions to manifolds in three dimensions. The data-driven methods have some significant benefits as compared to the traditional mesh denoising methods [8, 9, 10], since they do not require the searching of ideal values per parameter for each model. However, their limitations mainly rely on the large dataset, required for the training process, and the corresponding training complexity. Additionally, in real case scenarios, the testing data have generally a different form from the training data (due to different light conditions, device characteristics, etc.) making the implementation of the CNN process in real applications, to be problematic. A possible solution to this problem is to remove any objective characteristic from the data (i.e., training and testing), in order to be independent of geometry constraints (density, connectivity, etc.), different quality of the scanner devices, or other external conditions that could affect the captured 3D model.

In this paper, we take into account all the aforementioned drawbacks, introducing a novel pipeline for 3D mesh denoising that efficiently exploits the benefits of the 3D transform-domain CF, used for image denoising. Our main contributions can be summarized as follows:

- The proposed data-driven method gives the flexibility for fully automatic runtime execution, without the need for exhaustive searches of ideal parameters per model.
- Our method has no special requirements, related to the form of the 3D mesh, since it operates with equal-sized images uniformly, encoding the useful geometrical in-

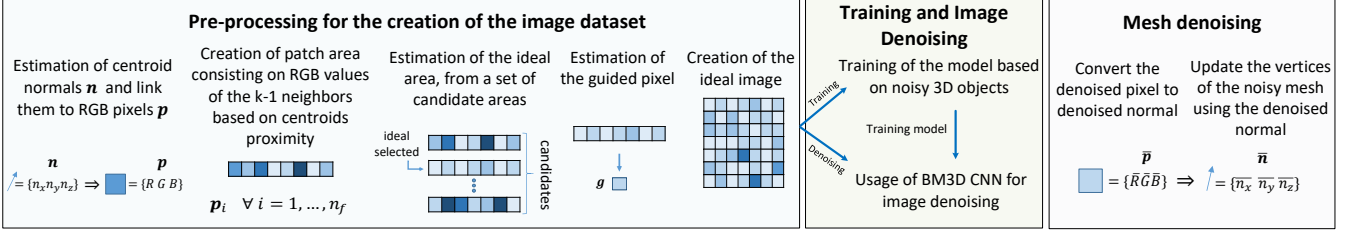


Fig. 1. Pipeline of the proposed method for image-based 3D mesh denoising using BM3D CNN filtering.

formation of any mesh.

- The proposed image-based approach results in a more efficient training process, avoiding the need for large datasets, exploiting efficiently geometrical coherence.
- We show how a method, inspired by robust and well-known approaches used in the area of the image processing, can be also efficiently used in the area of 3D mesh processing.

The rest of this paper is organized as follows: In Section 2, we discuss in detail each step of the proposed method. Section 3 presents the experimental results in comparison with other methods and in Section 4 we draw the conclusions.

2. IMAGE-BASED 3D MESH DENOISING

The initial step of the proposed pipeline is dedicated to the training of the CNNs. After that, the trained model is used to denoise any new noisy 3D object. The most significant process is a pre-processing step that generates the appropriate and uniformly-used form of the equal-sized images which efficiently encode the geometrical information of any noisy mesh. Fig. 1 briefly illustrates the pipeline of the proposed schema. We work with triangle meshes \mathbf{M} consisting of n vertices \mathbf{v} and n_f faces f . A noisy mesh $\hat{\mathbf{M}}$ can be written as: $\hat{\mathbf{v}} = \mathbf{v} + \hat{\mathbf{z}}$ where $\hat{\mathbf{z}}_i \in \mathbb{R}^{3 \times 1} \forall i = 1, \dots, n_f$ represents the noise vector. Each f_j face is represented by its corresponding normal \mathbf{n}_i and centroid point \mathbf{c}_i . Each unit centroid normal $\mathbf{n}_i = \{n_x, n_y, n_z\}, \forall i = 1, \dots, n_f$, where $|n_x|, |n_y|, |n_z| \in [-1, 1]$ and $\sqrt{(n_x + n_y + n_z)^2} = 1$, corresponds to a RGB pixel \mathbf{p}_i . However, a pixel of an image can not take negative values, for this reason we normalize the values of the normals in order to lie in a range of $[0, 1]$. Then, we link the $\{RGB\}$ components of each pixel to the values $\{n_x, n_y, n_z\}$, respectively. In this way, we can represent normals \mathbf{n}_i through pixels \mathbf{p}_i intensities, which is a fundamental concept in our approach.

2.1. Creation of Images Based on Normals

We start assuming that for each centroid \mathbf{c}_i there is a patch (enclosed in yellow border lines as shown in Fig. 2) which is created based on its proximity with the $k-1$ geometrical nearest neighboring centroids. Then, we estimate the normals of the corresponding faces and we formulate an image with size $k \times k$, where k is an odd number $\{k = 2q + 1 : q \in \mathbb{Z}\}$. In the central cell $(k+1)/2$ of the first row, we place the “normal-of-interest” (in red color). The normal \mathbf{n}_{12} of the closest centroid

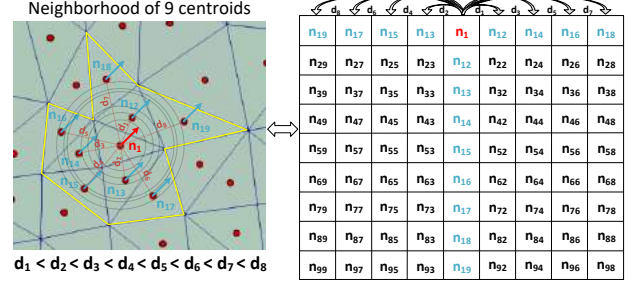


Fig. 2. Example of how an image 9×9 is created, using the normals of a neighborhood of 9 centroid points.

(i.e., d_1) is placed in the first right cell, then the normal \mathbf{n}_{13} of the next closest centroid (i.e., d_2) is placed in the first left cell, and the completion of the row continues, in turn, using the normals of all centroids, where $d_1 < d_2 < \dots < d_{k-1}$. The next i rows are completed in the same manner assuming, however, that the central cell represents the normal of the i closest centroid of \mathbf{n}_1 , increasingly for each row.

2.2. Creation of the Dataset Consisting of Ideal Images

Image-guided filtering has been presented in many papers providing superior results [11, 12, 13]. Additionally, in the area of 3D mesh denoising, guided normals have demonstrated also a remarkable denoising performance as it is also validated in [14, 2, 15]. In this work, we follow the same line of thought but using guided pixels and ideal-selected images, for the filtering step of the mesh denoising.

2.2.1. Estimation of the Ideal-selected Row and Guided Pixel

As we discussed earlier, for each centroid normal \mathbf{n}_i of the mesh, we estimate an image \mathcal{I}_i , the first row \mathcal{I}_{1i} of which represents the nearest neighborhood area called “patch”. Nevertheless, the same pixel \mathbf{p}_i may be presented in more than one row/patches since each neighborhood area is overlapping the neighboring areas of other normals. In fact, some of these rows may better represent the “color” of a pixel. For this reason, we collect a set $\mathcal{S}_i = \{\mathcal{I}_{1i_1}, \mathcal{I}_{1i_2}, \dots, \mathcal{I}_{1i_{n_p}}\}$ of n_p candidate rows and our main objective is to find which one of the $\mathcal{I}_{1i_j} \forall j = 1, \dots, n_p$, is the ideal representative of the pixel \mathbf{p}_i [14], in terms of the similarity of the color. For each one of these candidate rows \mathcal{I}_{1i_j} , we estimate the corresponding covariance matrix $\mathbf{C}_{ij} = \mathcal{I}_{1i_j}^T \mathcal{I}_{1i_j} \in$

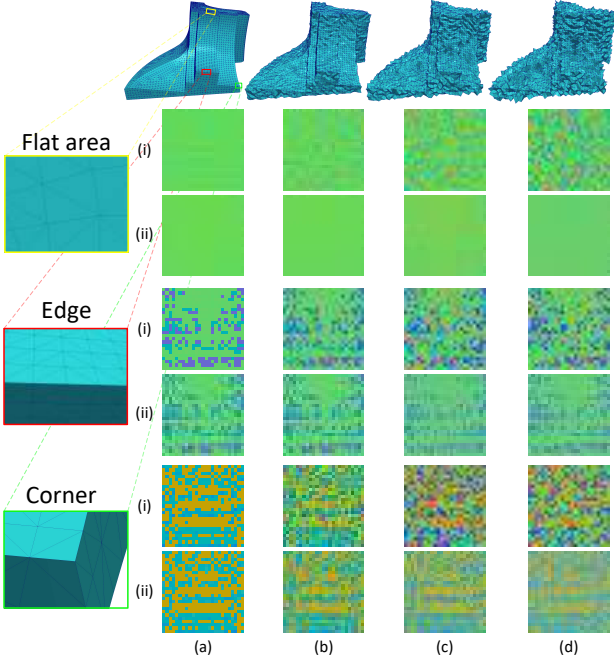


Fig. 3. Images of 25×25 pixels, representing patches of different features, affected by different level of noise: (a) original model, (b) $\sigma_E = 0.2$, (c) $\sigma_E = 0.4$, (d) $\sigma_E = 0.6$. The images are created: (i) using normals, as presented in Fig. 2, (ii) using the guided pixels, as presented in subsection 2.2.1.

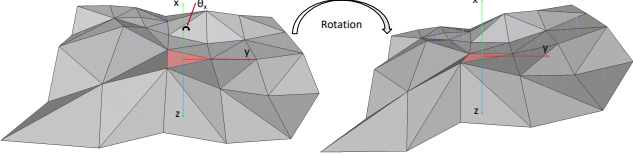


Fig. 4. Example of patch rotation.

$\mathbb{R}^{3 \times 3} \forall i = 1, \dots, n_f, \forall j = 1, \dots, n_p$ and we decompose it $\mathbf{C}_{ij} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}$ to its eigenvectors \mathbf{U} and eigenvalues $\mathbf{\Lambda} = \text{diag}(\lambda_{1ij}, \lambda_{2ij}, \lambda_{3ij})$. Then, for the estimation of the ideal-selected row \mathcal{I}_{1i}^* , two parameters are investigated: (a) the norm $s_{ij} = \|\lambda_{1ij} - \lambda_{2ij}\|_2$ of the first 2 eigenvalues and (b) the maximum color difference $h_{ij} = \max(\|\mathbf{p}_i - \mathbf{p}_l\|_2) \forall \mathbf{p}_l \in \mathcal{I}_{1ij}$ between the i pixel and the other pixels of the same row. Among all candidates rows, we pick this one with the smallest value of Eq. (1):

$$\mathcal{I}_{1i}^* = (\mathcal{I}_{1ij} \mid \min(s_{ij}h_{ij})) \quad (1)$$

$\forall i = 1, \dots, n_f, \forall j = 1, \dots, n_p$. Finally, the guided \mathbf{g}_i pixel is estimated as the average pixel of this ideal row:

$$\mathbf{g}_i = \frac{\sum_{\mathbf{p}_j \in \mathcal{I}_{1i}^*} \mathbf{p}_j}{\left\| \sum_{\mathbf{p}_j \in \mathcal{I}_{1i}^*} \mathbf{p}_j \right\|_2} \quad \forall i = 1, \dots, n_f \quad (2)$$

2.2.2. Estimation of Ideal Images Based on Guided Pixels

For the creation of the ideal images, we follow the procedure presented in the subsection 2.1, but using the guided pixels \mathbf{g}_i instead of the normals \mathbf{n}_i . In Fig. 3, we present examples of images created by the RGB representation of: (i) the normals, and (ii) the guided pixels, for different features (i.e., flat area, edge and corner) of Fandisk model affected by different level of Gaussian noise with intensity $\sigma_E = \frac{\sigma}{E_{\text{mean}}}$, where σ is the variance of the Gaussian function, and E_{mean} is the average edge length [2]. As we can observe, the images, created by the guided pixels (Fig. 3-(ii) (b)-(d)), have a similar presentation with the original, despite the different levels of noise. On the other hand, images that have been created using the centroid normals are very noisy and differ from the original, making difficult to be denoised, even in cases of flat areas. This observation is very important since it shows that it is not necessary to train different models for different noise patterns, as other data-driven methods do. Instead, a generic model would be also sufficient. This is especially useful in cases where the level or type of noise is not known a-priori.

2.3. BM3D CNN for Ideal Images Denoising

2.3.1. Basic Characteristics of BM3D

BM3D uses three basic steps: (1) “Block matching” which tries to find groups of similar patches, (2) “3D wavelet shrinkage” which denoises each one of these groups in the 3D wavelet transform domain and (3) “Patch group aggregation” is an averaging procedure in which all the estimated patches return to their original positions reweighed each pixel appearing in many instances since images are created by overlapping patches [4]. For a more efficient block matching step, we use a feature classification, similar to [14]. More specifically, we classify each i centroid into three different categories of features (i.e., flat area (F), edge (E) and corner (C)), using a κ -means ($\kappa = 3$) clustering, applied to the vector $\lambda_i = [\lambda_{1i1} \lambda_{2i1} \lambda_{3i1}]$ of its eigenvalues, as described in subsection 2.2.1. To create a more relevant dataset, we rotate all normals of each i patch about an angle θ_{x_i} , as shown in Fig. 4, so that the i “normal-of-interest” (i.e., lied in the red face), to be equal to the constant vector $[1 \ 0 \ 0]$. In this way, a more coherent dataset is created, accelerating the learning process. Otherwise, patches with every possible normal direction would need to be provided, resulting in the creation of very large training dataset, increasing also the execution time of the learning step.

2.3.2. Characteristics of the used BM3D CNN

The presented CNN (Fig. 5), similar to [7], represents the second step of the BM3D (i.e., 3D wavelet shrinkage) and consists of two convolution layers and a nonlinear transform layer. More specifically, the first convolution layer is used as the wavelet transform step and the second as the inverse

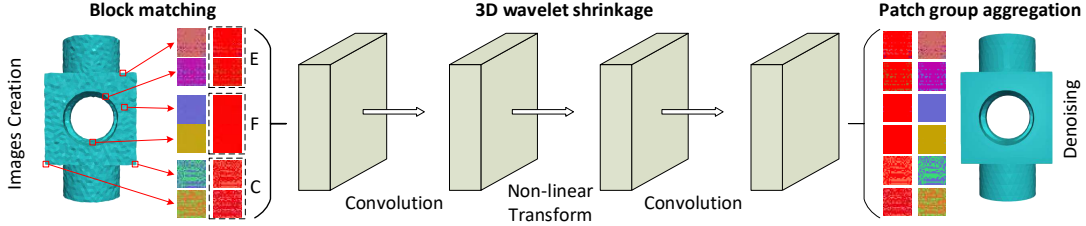


Fig. 5. Patches of the mesh are converted to images and then they are fed into the CNN for denoising.

wavelet transform step. They are generally presented as:

$$\mathbf{I}_{l+1} = W_l * \mathbf{I}_l \quad (3)$$

where $*$ indicates multidimensional convolution, \mathbf{I}_l denotes the group of images for the $l \in \{1, 3\}$ layer and W_l is the filter with the weights of the l layer after the concatenate process. The nonlinear transform layer is used as the wavelet shrinkage step. Similar to [7], the radial basis functions (RBFs) is used instead of regular hard thresholding or Rectified Linear Unit (ReLUs), according to:

$$\mathbf{I}_3 = W_2 e^{\frac{|\mathbf{I}_2 - \mu|^2}{\sigma^2}} \quad (4)$$

2.4. 3D Mesh Reconstruction by Updating Vertices

After the estimation of the denoised pixels $\bar{\mathbf{p}}$, which is the central pixel of the first row of the denoised images, we convert the RGB values to the (n_x, n_y, n_z) values of the corresponding denoised centroid normals $\bar{\mathbf{n}}$. Then, we reconstruct the denoised mesh updating its vertices, using the following robust and well-known equation [3]:

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} + \frac{\sum_{j \in \Psi_i} \bar{\mathbf{n}}_j (\langle \bar{\mathbf{n}}_j, (\mathbf{c}_j^{(t)} - \mathbf{v}_i^{(t)}) \rangle)}{|\Psi_i|} \quad (5)$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ represents the inner product of \mathbf{a} and \mathbf{b} , (t) represents the t -th iteration and the matrix Ψ_i represents the first-ring area of the vertex \mathbf{v}_i . Algorithm 1 summarizes the basic steps of the proposed process.

3. EXPERIMENTAL ANALYSIS AND RESULTS

3.1. Experimental Setup, Datasets and Metrics

The experiments were carried out using an Intel Core i7-4710 CPU @ 3.60GHz PC with 16 GB of RAM. For the experiments, a variety of 3D meshes were used [16], affected by different levels of noise. For the training process, we use only 5 different models, consisting of ~ 1.2 million (M) centroid points in total. This means that we have 1.2 M very relevant instances for the training which seem very effective, according to the experimental results. For the evaluation of the reconstructed results, we use the following metrics: (i) the average one-sided Hausdorff distance HD from the denoised mesh to the known ground-truth mesh, (ii) the metric θ representing the average angle difference between the normals of

Algorithm 1: BM3D CNN for 3D Mesh Denoising

Input : Noisy 3D model $\hat{\mathbf{M}} \in \mathbb{R}^{n \times 3}$;

Output: Denoised 3D model $\bar{\mathbf{M}} \in \mathbb{R}^{n \times 3}$;

- 1 Estimate the n_f centroid normals \mathbf{n} and convert them to corresponding pixels \mathbf{p} ;
- 2 **for** $i = 1, \dots, n_f$ **do**
- 3 Estimate the ideal-selected row \mathcal{I}_{1i}^* via Eq. (1) and the guided pixel \mathbf{g}_i according to Eq. (2);
- 4 Estimate the filtered image \mathcal{I}_i^* based on guided pixels;
- 5 **end**
- 6 Create 3 different types of images, based on the features classification, for a more efficient block matching;
- 7 Image denoising based on BM3D CNN via Eqs. (3)-(4);
- 8 Convert denoised pixels $\bar{\mathbf{p}}$ to denoised centroid normals $\bar{\mathbf{n}}$;
- 9 Reconstruct the denoised 3D mesh $\bar{\mathbf{M}}$ according to Eq. (5);

the ground truth and the reconstructed model [14] and (iii) heatmap visualization highlighting, in different colors, the difference $|\mathbf{M} - \bar{\mathbf{M}}|$ between original \mathbf{M} and reconstructed $\bar{\mathbf{M}}$ mesh.

3.2. Experimental Results

The quality performance of the proposed technique is compared with other well-known and robust techniques of the literature, namely (a) non-iterative smoothing (NIS) [17], (b) fast and effective (FAE) [3] mesh denoising, (c) bilateral normal filtering (BNF) [18], (d) guided normal filtering (GNF) [2], (e) two-stage graph spectral processing (TSGSP) [14]. We also compare the proposed approach to data-driven methods such as (f) deep autoencoders denoising (DAD) [19], and cascade normal regression [16] in two different approaches: (g) by using only a small dataset of models (CNR) (the exact same dataset used) and (h) (CNR+) by using a larger dataset. Our method is denoted as (OUR) and (GEN OUR) represents our approach but using a general training model, having been trained by different levels of noise at the same time. In Fig. 8, we present the original and three corresponding noisy 3D models which have been affected by three different level of noise (i.e., $\sigma_E = 0.3, 0.4, 0.5$). We provide enlarged details of each reconstructed model, for easier comparison among the methods, and we also provide the metric θ . The benefits of our approach are more apparent when the level of noise is $\sigma_E > 0.3$. For a lower level of noise, the results are comparable with those of the other methods. We can make similar

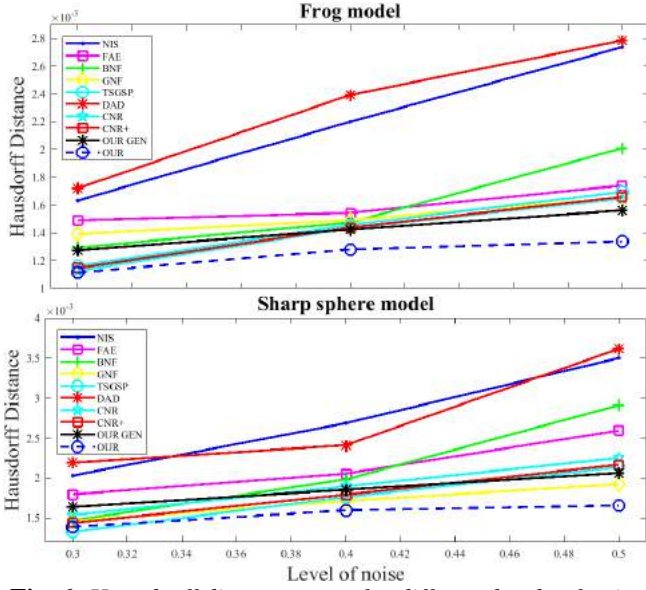


Fig. 6. Hausdorff distance error for different levels of noise.

conclusions observing the heatmaps of Fig. 7, which visualize in different colors the differences between the reconstructed and original mesh, in three different levels of noise. Big differences are represented by red color while small differences are represented with blue. Finally, in Fig. 6, we present plots of Hausdorff distance (HD) error per different level of noise, for the reconstructed models of different approaches. For a fairer comparison between data-driven and other conventional methods, we set fixed values at the parameters, which provide good results in all levels noise, and we do not search for the ideal values per each parameter and model. Our method avoids the over-smoothing of the models and at the same time, preserves the special characteristics of the features.

4. CONCLUSIONS

In this work, we presented an image-based 3D mesh denoising approach using BM3D CNN filtering applied in color images. We take advantage of the stable and robust behavior of BM3D, which has exhaustively tested for image denoising, in combination with the fast and effective behavior of CNNs. To achieve this, we create an appropriate form of images, representing patches of neighboring normals. Experimental analysis verified the correctness of our assumption while comparison with other traditional state-of-the-art methods demonstrates the potential of our approach. The proposed method has many advantages, such as (i) only fixed values of parameters are used, (ii) it requires a relatively small dataset for the training process, (iii) a general model, trained by different levels of noise, can be used for denoising, providing acceptable results.

Acknowledgements

Research supported by the European Union research and innovation program

Horizon 2020 Marie Skłodowska WARMEST :IoW Altitude Remote sensing for the Monitoring of the state of cultural hERitage Sites: building an inTe-grated model for maintenance, under Grant Agreement No. 777981. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826299.

5. REFERENCES

- [1] S. K. Yadav, U. Reitebuch, and K. Polthier, "Mesh denoising based on normal voting tensor and binary optimization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 8, pp. 2366–2379, Aug 2018.
- [2] Wangyu Zhang, Bailin Deng, Juyong Zhang, Sofien Bouaziz, and Ligang Liu, "Guided mesh normal filtering," in *Computer Graphics Forum*. Wiley Online Library, 2015, vol. 34, pp. 23–34.
- [3] X. Sun, P. Rosin, R. Martin, and F. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 925–938, Sept 2007.
- [4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.
- [5] F. Xue, J. Li, and T. Blu, "An iterative sure-let deconvolution algorithm based on bm3d denoiser," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 1795–1799.
- [6] B. Shi, Q. Lian, S. Chen, and X. Fan, "Sbm3d: Sparse regularization model induced by bm3d for weighted diffraction imaging," *IEEE Access*, vol. 6, pp. 46266–46280, 2018.
- [7] D. Yang and J. Sun, "Bm3d-net: A convolutional neural network for transform-domain collaborative filtering," *IEEE Signal Processing Letters*, vol. 25, no. 1, pp. 55–59, Jan 2018.
- [8] Lei He and Scott Schaefer, "Mesh denoising via l0 minimization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, July 2013.
- [9] S. K. Yadav, U. Reitebuch, and K. Polthier, "Robust and high fidelity mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.
- [10] M. Wei, J. Huang, X. Xie, L. Liu, J. Wang, and J. Qin, "Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.
- [11] L. Li, X. Guo, W. Feng, and J. Zhang, "Soft clustering guided image smoothing," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, July 2018, pp. 1–6.
- [12] X. Guo, Y. Li, J. Ma, and H. Ling, "Mutually guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.
- [13] Z. Lu, B. Long, K. Li, and F. Lu, "Effective guided image filtering for contrast enhancement," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1585–1589, Oct 2018.
- [14] G. Arvanitis, A. Lalos, K. Moustakas, and N. Fakotakis, "Feature preserving mesh denoising based on graph spectral processing," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.
- [15] A. S. Lalos, E. Vlachos, G. Arvanitis, K. Moustakas, and K. Berberidis, "Signal processing on static and dynamic 3d meshes: Sparse representations and applications," *IEEE Access*, vol. 7, pp. 15779–15803, 2019.
- [16] Peng-Shuai Wang, Yang Liu, and Xin Tong, "Mesh denoising via cascaded normal regression," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232:1–232:12, Nov. 2016.
- [17] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 943–949, July 2003.
- [18] Y. Zheng, H. Fu, O. K. Au, and C. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1521–1530, Oct 2011.
- [19] G. Arvanitis, A. Lalos, and K. Moustakas, "Feature-aware and content-wise denoising of 3d static and dynamic meshes using deep autoencoders," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, July 2019, pp. 97–102.

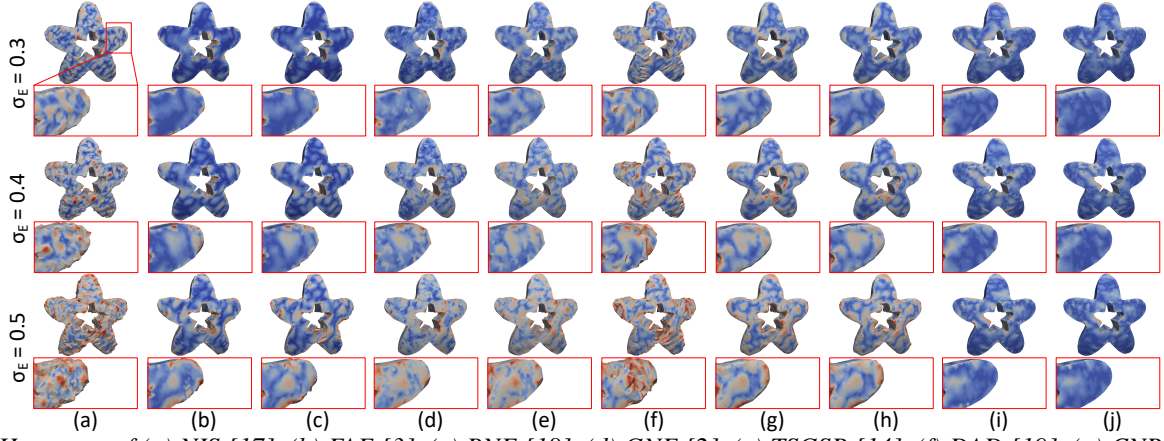


Fig. 7. Heatmaps of (a) NIS [17], (b) FAE [3], (c) BNF [18], (d) GNF [2], (e) TSGSP [14], (f) DAD [19], (g) CNR [16], (h) CNR+, (i) GEN OUR, (j) OUR.

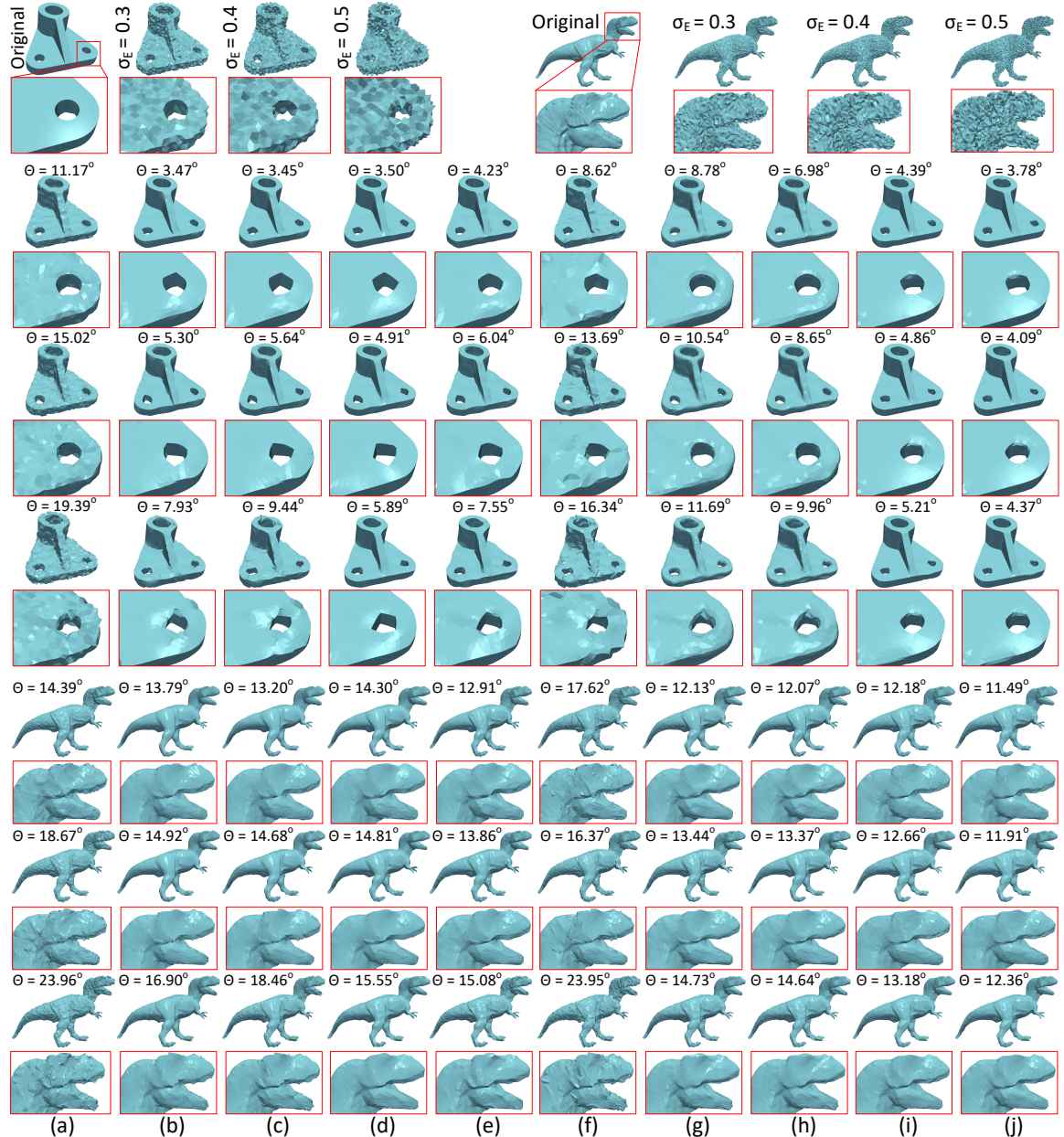


Fig. 8. Denoising results of: (a) NIS [17], (b) FAE [3], (c) BNF [18], (d) GNF [2], (e) TSGSP [14], (f) DAD [19], (g) CNR [16], (h) CNR+, (i) GEN OUR, (j) OUR.