IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 8, 2022, accepted March 28, 2022, date of publication April 4, 2022, date of current version April 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164714

IMD-Net: A Deep Learning-Based Icosahedral Mesh Denoising Network

JAN BOTSCH, HARDIK JAIN, AND OLAF HELLWICH, (Senior Member, IEEE)
Computer Vision & Remote Sensing Group, Technische Universität Berlin, 10587 Berlin, Germany

Corresponding author: Hardik Jain (h.jain@campus.tu-berlin.de)

This work was supported by the German Research Foundation and the Open Access Publication Fund of Technische Universität Berlin.

ABSTRACT In this work, we propose a novel denoising technique, the icosahedral mesh denoising network (IMD-Net) for closed genus-0 meshes. IMD-Net is a deep neural network that produces a denoised mesh in a single end-to-end pass, preserving and emphasizing natural object features in the process. A preprocessing step, exploiting the homeomorphism between genus-0 mesh and sphere, remeshes an irregular mesh using the regular mesh structure of a frequency subdivided icosahedron. Enabled by gauge equivariant convolutional layers arranged in a residual U-net, IMD-Net denoises the remeshing invariant to global mesh transformations as well as local feature constellations and orientations, doing so with a computational complexity of traditional conv2D kernel. The network is equipped with carefully crafted loss function that leverages differences between positional, normal and curvature fields of target and noisy mesh in a numerically stable fashion. In a first, two large shape datasets commonly used in related fields, *ABC* and *ShapeNetCore*, are introduced to evaluate mesh denoising. IMD-Net's competitiveness with existing state-of-the-art techniques is established using both metric evaluations and visual inspection of denoised models. Our code is publicly available at https://github.com/jjabo/IMD-Net.

INDEX TERMS 3D surface mesh, deep learning, icosahedral CNN, mesh denoising network, noise filtering, spherical parametrization, U-net.

I. INTRODUCTION

The demand on fidelity and quality of 3D surface mesh models is steadily on the rise. Mesh models are becoming omnipresent in a variety of application domains, from archaeological preservation and reconstruction, over retail and reverse engineering, to various biomedical fields like neurology and orthodontics. 3D surface meshes can either be crafted by hand, which allows fine-grained control of their quality at the cost of significant time and resources, or they can be acquired using 3D scanning technologies, whose intrinsic physical imperfections inevitably introduce noise to the reconstruction process. The noise originates in the scanning process and indirectly affects the recovery of individual 3D points on the model surface which are then connected to generate a 3D mesh. The mesh denoising process is inherently constrained by the quality of the preceding mesh generation stage, but it can take advantage of geometric, topological and connectivity information of the mesh structure to produce high-fidelity mesh models.

The associate editor coordinating the review of this manuscript and approving it for publication was Mounim A. El Yacoubi.

A mesh vertex (or a face normal) can be denoised in a local fashion, using primarily information from a local neighborhood around a vertex (or face). With progressing research, it is found that increasing the *field of view* by including larger neighborhoods around a vertex or face of interest can have a beneficial impact on denoising results. However, there seems to be little work aiming at including information from beyond 2- or 3-ring neighborhoods in the denoising process. This is surprising, as the consideration of wider neighborhoods holds promise for denoising larger features more accurately. Also, an integration of increasingly global information about object shape, symmetries and surface structure in the denoising process could provide vital information to avoid introducing self-intersections and other anomalies that harm object fidelity and visual appeal.

Existing mesh denoising approaches generally have one thing in common, that they come with a set of model specific parameters, which require specific knowledge and careful tuning by a user [e.g., 1, 2, 3]. Often authors supply well working parameter defaults, but the fact remains that for each individual model user interaction might be required. This is a remnant from the recent past when denoising 3D mesh

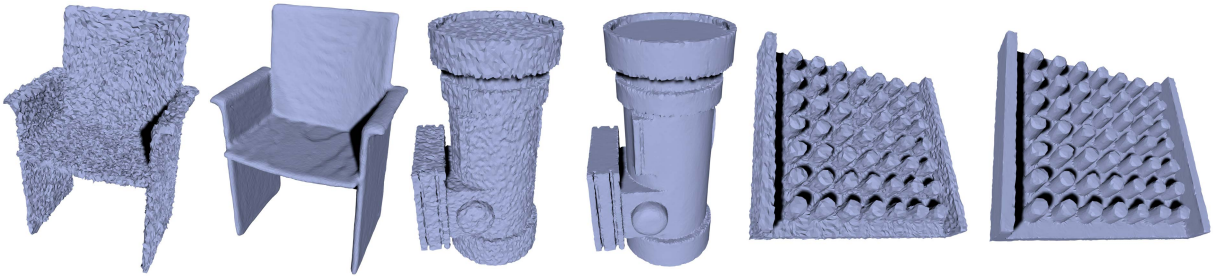

FIGURE 1. Three pairs of noisy meshes and their denoised version created by IMD-Net. IMD-Net is able to preserve fine details, avoid self-intersections in thin parts and accurately denoise smooth regions and sharp features.

models still was a rare task performed by specialists. But in today's digital landscape, where every mobile phone can be a 3D scanner and large datasets of 3D models need denoising, requiring user interaction is a deployment inhibitor and ought to be avoided.

Methods based on machine learning could provide parameter-free denoising at inference time. With an increasing number of noisy mesh models readily available, it also has become possible to incorporate information from numerous mesh models in the denoising process. Yet, the application of machine learning methods to mesh denoising has been lagging behind other fields such as image and natural language processing. This is primarily due to the irregular structure of mesh data. Approaches have been devised to overcome this irregular mesh structure, e.g., by voxelating meshes into 3D grids and applying 3D convolutions [4]. This, however, is undesirable: Surface meshes represent 2D-manifolds in 3D-space, hence an approach that operates in 3D-space and not exclusively on the manifold lacks efficiency.

Summarizing the above, a parameter-free, deep learning-based method that consumes a noisy input mesh and produces in one pass a denoised output mesh is sorely missing from the field. Ideally, such a method would integrate suitable amounts of local and global mesh information in the denoising process. Also, the method would optimally feature a computational complexity akin to neural networks used on other 2D-manifolds like images.

The contribution of this work includes (i) exploiting the homeomorphism between genus-0 meshes and spheres to regularize the surface on an icosahedral grid, (ii) the icosahedral mesh denoising network (IMD-Net), a novel denoising technique based on equivariant conv2D-layers allowing efficient and high-quality denoising of closed genus-0 meshes (iii) introducing two large CAD model datasets as benchmarks to mesh denoising, the *ABC* [5] and the *ShapeNetCore* [6] dataset, (iv) experiments that demonstrate the competitiveness of IMD-Net with state-of-the-art methods at various noise levels. Figure 1 shows denoising on three different meshes using the proposed IMD-Net, illustrating how well the our network preserves sharp features and fine details.

II. RELATED WORK

Mesh denoising has a history of more than three decades and numerous approaches have been put forward. Early attempts transfer ideas from related fields, most notably iterative Laplacian smoothing from mesh smoothing [10], [11] and various low-, band- and high-pass filters from signal processing [12]–[15]. These isotropic methods lack the means to preserve features and were quickly superseded by anisotropic techniques such as diffusion process-based methods [16]–[18] and the bilateral filter [19], [20]. Another successful line of anisotropic methods splits mesh denoising into *face normal filtering* and *vertex position updating* by numerically integrating the denoised face normal field.

A considerable work has gone into finding suitable filters, yielding Laplacian smoothing [21], mean, median and alpha-trimming filters [22]–[24] as well as fuzzy median [25], [26] and random walk filters [27]. These filters, however, do not consider the regularity of a mesh. Zheng *et al.* [1] devised a joint bilateral normal filter (BNF) which uses a Gaussian weighted average of distances and orientation differences to surrounding faces in order to denoise face normals. Zhang *et al.* [3] used patches around each face to compute a guidance normal and incorporate deviations from the guidance normal in the joint bilateral normal filter (GNF). This produces impressive results if the patch is chosen well, triggering suggestions to select patches that adapt to corners and edges [28] or minimize the angular difference within each patch [29]. Throughout normal filtering methods the patches tend to be small and strictly local. This prevents the denoising process from integrating potentially beneficial information from non-local features and global shape. Further, most methods require selecting parameters for individual models inhibiting their application to larger datasets.

Several methods reframe mesh denoising as *sparse optimization problems*: [30]–[33]. He and Schaefer [2] proposed a L_0-minimization combined with an edge-based feature-preserving shape operator to yield piece-wise flat surface regions that intersect at the preserved features. In [34], [35] the total variation of face normals is minimized to smooth the surface while minimizing distortion of volume and shape. Others devise two-stage algorithms that first compute a base mesh using a smoothing scheme [36] or a regularizer

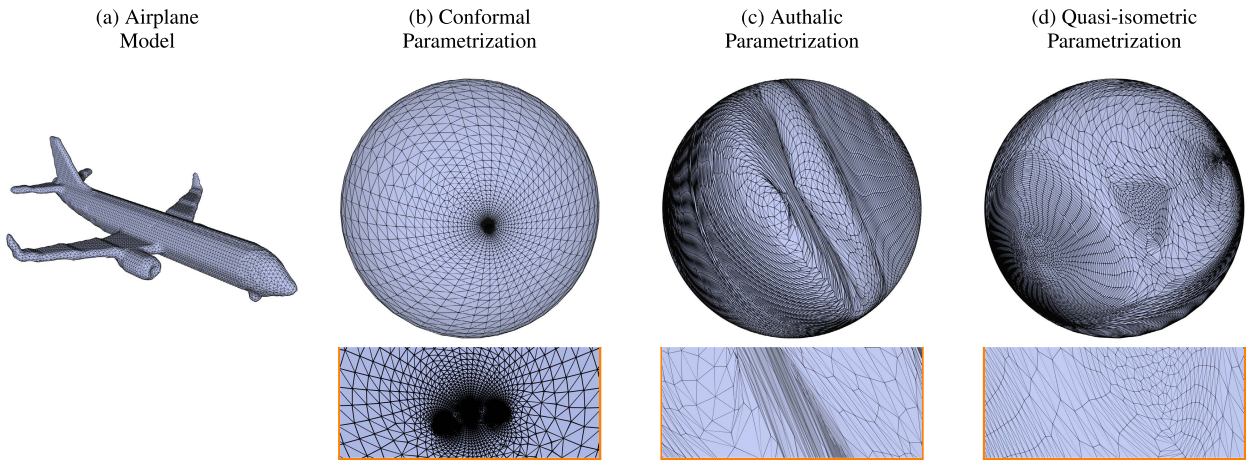

FIGURE 2. Spherical parametrizations of (a) airplane model. (b) A conformal parametrization produced with [7]. Extrusive features like the wings and the tail are all mapped to one dense cluster. (c) An authalic parametrization produced with [8]. Clusters are less dense, but thin, long-stretched triangles introduce large distortions. (d) A quasi-isometric parametrization produced with [9]. Few and sparse clusters with no thin, long-stretched triangles.

[37], [38] and then recover features from the residual between noisy and base mesh. The success of optimization-based methods often depends on prior assumptions on the noise distribution, and they require adjustment of (optimization) parameters to individual models, limiting their adaption to larger datasets.

As the focus on *feature preservation* increased, multiple schemes were proposed that classify vertices into features and non-features using tensor voting in combination with k-means clustering [39], eigenanalysis [40] or feature descriptors [41], [42] before applying a filtering technique. Arvanitis *et al.* [43] proposed a coarse-to-fine mesh denoising approach that uses graph spectral processing to preserve feature normals in the denoising process. Other techniques add feature detection to existing normal filtering methods: Yadav *et al.* [44] replaced the Gaussian similarity function of the bilateral normal filter [1] with Tukey's bi-weight function, which reduces the diffusion of sharp features; and Zhao *et al.* [45] deployed a graph-based feature detection to select optimal patches for computing guidance normals of Zhang *et al.* [3]; in [46] a base mesh and a feature-detecting saliency measure are employed to the same end. The results of feature-detection based methods are sensitive to their ability of correctly classifying features, which sometimes leads to misclassified features or introducing artificial ones.

A recent development in mesh denoising are methods that involve *machine learning*. Denoising autoencoders have been used in various applications of 1D noise filtering [47] as well as 2D image filtering. Wang *et al.* [48] transfer the bilateral normal filter from image filtering and apply it repeatedly to each face to compose geometric descriptors which are subsequently clustered. In a cascaded normal regression (CNR), a regression function is fitted to each cluster using a radial basis function network. In [49] this method is complemented by a reverse descriptor that aims to recover geometric features which were previously lost in the regression. Nousias *et al.* [50] pursued a geometric deep learning approach that employs a conditional variational autoencoder consisting of a Gaussian encoder and a Bernoulli decoder, followed by one step of bilateral filtering to remove small artifacts. In a work extending GNF, Zhao *et al.* [4] proposed NormalNet, a cascaded deep 3D-CNN that processes voxelated patches to estimate a guidance normal. The reported results look promising but are dimmed by the high computational complexity of a 3D-CNN. Using a graph convolutional network, [51] proposed an elegant and well performing two stage approach for mesh denoising. The deep normal filtering network (DNF-Net) [52] denoises a mesh split into patches by extracting local geometric features. It employs a multi-scale feature embedding unit that extracts features representing local geometric context and two residual learning units that aim to progressively attenuate noise. DNF-Net reports state-of-the-art denoising performance, but the patch creation and denoising are time-consuming which limits the number of patches (and meshes) that can reasonably be used for training, restricting the generalization potential of the network.

Method Components: The proposed IMD-Net is designed to efficiently and accurately denoise closed genus-0 meshes. It exploits the genus-0 property, which guarantees the existence of a bijective mapping between a mesh surface and the unit sphere. A spherical parametrization algorithm is employed to construct such a mapping. Parametrization algorithms can be distinguished by how well they preserve or minimize the distortion of intrinsic geometric metrics such as face angles: conformal mappings (Fig. 2b) [7], [53], face areas: authalic mappings (Fig. 2c) [8], [54] or both: isometric mappings. Unfortunately, isometric mappings generally do not exist between arbitrary genus-0 surfaces and spheres. In the context of this work, an algorithm is desired which is authalic but also does not create mapped triangles that are unnaturally stretched over large parts of the sphere

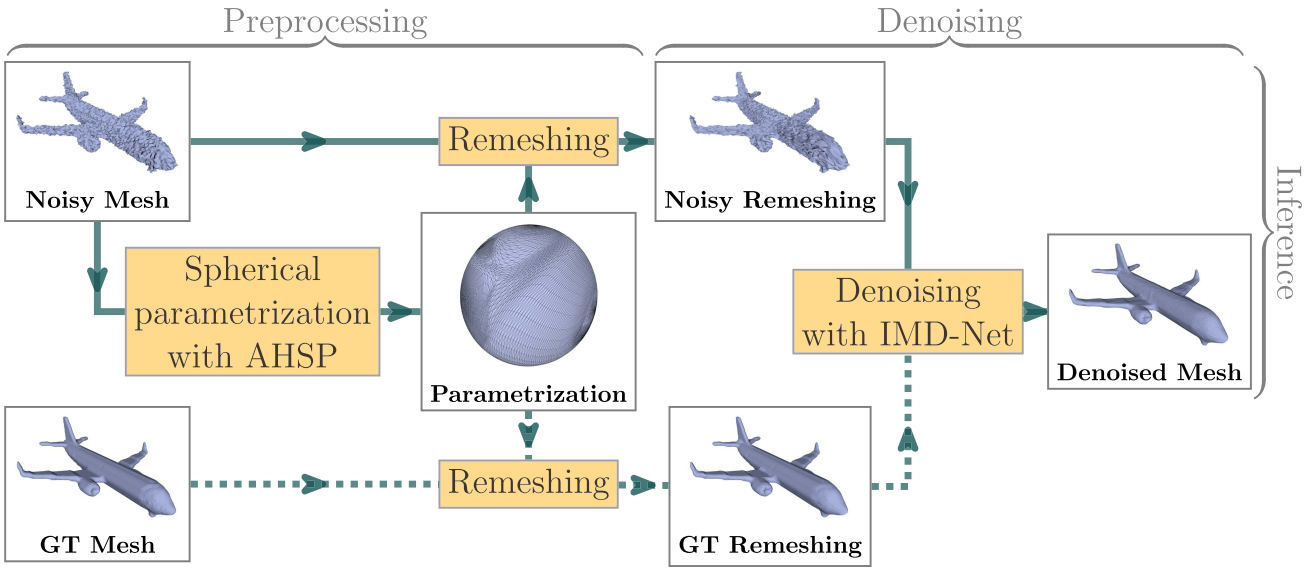

FIGURE 3. The high-level pipeline of the IMD-Net: A noisy genus-0 mesh is spherically parametrized using the AHSP [9]. Employing the parametrization, the noisy and ground truth (GT) meshes are remeshed as regular meshes. The noisy remeshing is denoised with IMD-Net. For network training, the ground truth remeshing is used in the loss function.

surface. Recently, Hu *et al.* [9] developed a quasi-isometric parametrization method based on progressive meshes, which exhibits such characteristics (Fig. 2d). Advanced hierarchical spherical parametrizations [9] has also been recently used as a remeshing tool in generative learning of icosahedral meshes [55]. The method has a low computational complexity and achieves state-of-the-art performance and is therefore our choice of parametrization.

IMD-Net denoises data mapped onto the unit sphere. In the context of mesh denoising, the most desirable properties of deep neural networks are translation, scale, and rotation equivariance. In [56] and [57] networks were suggested that are translation and scale equivariant on spherical data. Spherical CNNs [58] define a spherical cross-correlation that is rotation-equivariant. The correlation is designed to satisfy the generalized Fourier theorem and can be efficiently computed using a generalized Fast Fourier Transform (FFT) algorithm. Signal and filters, both defined on a spherical-polar grid, are Fourier transformed, cross-correlated, inherently rotated to achieve equivariance, and finally inverse transformed. The authors report improved results on spherical images and for 3D-object detection. However, the number and angle of filter rotations is coupled to the grid size, and the spherical-polar grid causes oversampling near the sphere poles. More importantly, a spherical CNN layer has minimal complexity of $\mathcal{O}(B^3)$ with B being the bandwidth of the grid, making the overall network computationally expensive. Cohen *et al.* [59] presented *IcosahedralCNN*, a network operating in the spherical domain by discretizing it as a subdivided icosahedron. The discretized surface is planarized into five padded, rectangular maps and arranged in an atlas. Hexagonal convolution filters are defined and kernel expansion in combination with weight sharing are applied to make the convolution gauge (and hence translation, scale and rotation) equivariant. Two layers are distinguished: A layer which takes non-gauge equivariant input features (singular) and outputs gauge-equivariant features (regular) is referred to as a *S2R-layer*; and a layer which has regular input and outputs features is referred to as a *R2R layer*. The low computational complexity as well as the guarantee of gauge equivariance convinced us to use the S2R and R2R building blocks from IcosahedralCNN to construct IMD-Net.

III. METHOD

The promise of this work is a novel deep learning-based denoising technique that efficiently processes and denoises closed genus-0 meshes. This task is made particularly challenging by the inherent irregularity of the mesh data structure, which is overcome in two stages, by spherically parametrizing meshes and remeshing them as regular, subdivided icosahedrons. The preprocessed meshes are fed to the denoising network. In the training phase, the network also receives preprocessed ground truth mesh which are used in the loss function. The high-level pipeline of the IMD-Net framework is presented in Fig. 3. Details of each aspect of the framework are explained in this section.

A. REMESHING

A closed genus-0 mesh $\mathcal{B}$ (e.g., Fig. 4a) possess the special property that its irregular mesh data can be mapped onto the regular domain of a unit sphere. This spherical parametrization $\mathcal{M}$ is used to create a remeshing of $\mathcal{B}$ as a subdivided icosahedron $\mathcal{I}_r = \{v, f\}$, where the number of vertices N_v depends on a selectable subdivision level r with

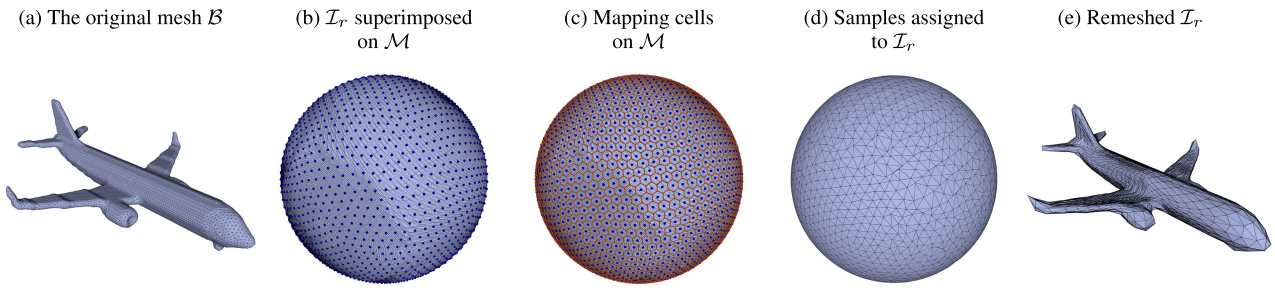

FIGURE 4. Illustration of remeshing of an airplane mesh at subdivision $r = 4$.

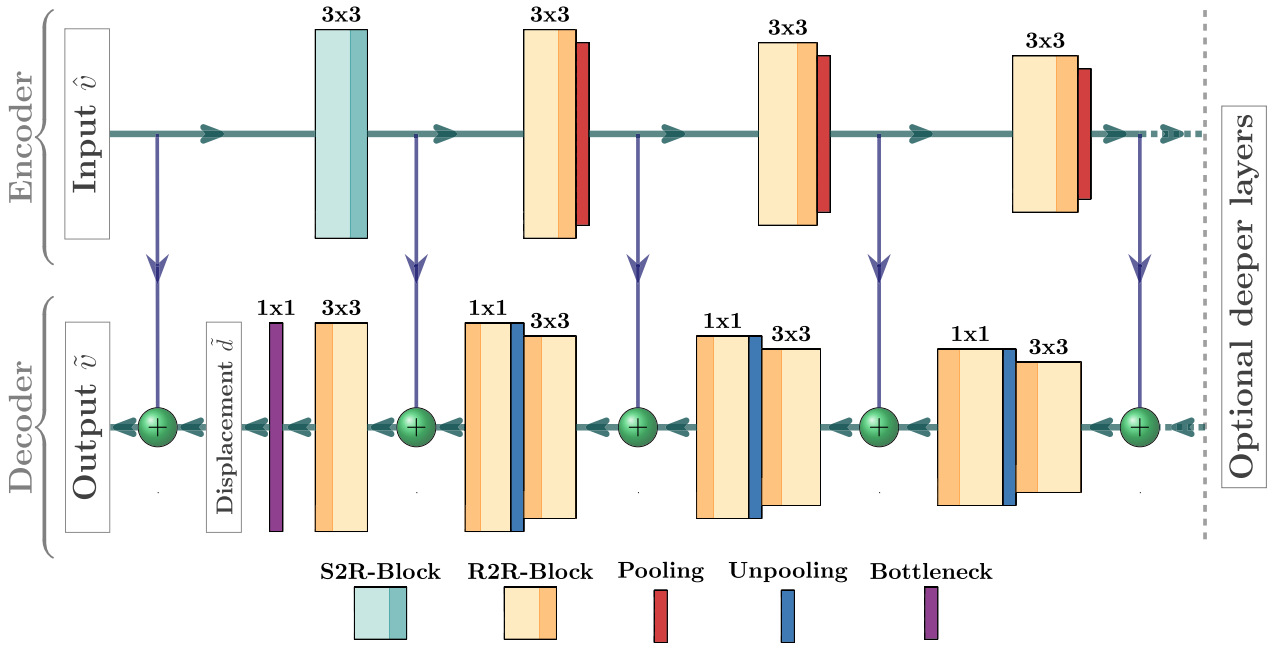

FIGURE 5. The compressed U-net architecture of the proposed IMD-Net.

$N_v = 5 \times 2^{2r+1} + 2$. To create a remeshing, first $\mathcal{I}_r$ is superimposed on $\mathcal{M}$ and the vertices of $\mathcal{I}_r$ are radially projected onto the sphere (Fig. 4b). $\mathcal{M}$'s surface is divided into non-overlapping *mapping cells* of near-equal size, one around each vertex of the superimposed icosahedron (Fig. 4c). The mapping cells are computed by using the faces of $\mathcal{I}_r$'s dual, i.e. the subdivided dodecahedron. Subsequently, the mapping cells are transferred from $\mathcal{M}$ to $\mathcal{B}$ and sorted into three mutually exclusive cell categories based on how many vertices of $\mathcal{B}$ are contained in a cell, namely into *one-vertex*, *zero-vertex* and *multi-vertex* cells. A position is sampled from the contained surface of each cell and assigned to the respective vertex of $\mathcal{I}_r$. One-vertex cells take the single vertex of $\mathcal{B}$ that is within the cell as sample. Zero-vertex cells contain no vertices, only faces and/or parts of faces, hence the centroid of the contained face(s) is used as sample. In multi-vertex cells the average of all contained vertices is taken as sample. In this case, relevant information might be lost, and therefore multi-vertex cells ought to be avoided, e.g., by using a sufficiently high subdivision. The output of the remeshing stage is the icosahedron $\mathcal{I}_r$ of which each vertex was repositioned to a sample location on $\mathcal{B}$ (Fig. 4e).

As shown in Fig. 3, a separate remeshing is created for the noisy $\hat{\mathcal{B}}$ and the ground truth $\mathcal{B}$ mesh, but both depend on the spherical parametrization of the noisy mesh, $\hat{\mathcal{M}}$. Using a shared parametrization ensures that the vertices and faces in both remeshings represent exactly corresponding surface positions and patches in the original meshes.

B. ICOSAHEDRAL MESH DENOISING NETWORK

IMD-Net can be assembled using the gauge equivariant S2R and R2R layers from IcosahedralCNN. The network consumes a remeshed noisy mesh, $\hat{\mathcal{I}}_r = \{\hat{v}, f\}$ and outputs a denoised (or estimated) mesh, $\tilde{\mathcal{I}}_r = \{\tilde{v}, f\}$. Specifically, the vertex positions of a subdivided icosahedron arranged as planarized atlases are the input ($\hat{v}$) and output ($\tilde{v}$) of the network. A modified U-net, referred to as *compressed U-net* and shown in Fig. 5, is chosen as architecture for IMD-Net. The architecture is based on the original U-net [60], but

the concatenation operations in the decoder, which concatenate feature maps coming from the upsampling and skip-connection paths, are removed. Instead, feature maps from the encoder are directly added to the respective decoder level via skip connections. The modification follows mathematically sound and empirically successful concept of residual neural networks [61]. In the context of mesh denoising, the number of learnable parameters are reduced by about 33% while maintaining the denoising performance.

The three-channel input is transformed to regular equivariant feature maps by a S2R block, composed of a 3×3 S2R-layer, a batch normalization layer [62] and ReLU. The batch normalization averages over groups of six feature maps to preserve equivariance [59]. The ReLU function operates pointwise and is therefore equivariant. In the encoder, the S2R-block is followed by multiple levels of R2R-blocks and pooling layers. In the decoder, the flow is reversed with corresponding R2R-Blocks and unpooling layers. A standard 1×1 conv-2D (bottleneck) layer produces the three channels of the displacement vectors which are added pointwise to the input vertices to yield denoised vertices.

C. LOSS FUNCTION

In the training phase, the network is provided a ground truth (or target) mesh $\mathcal{I}_r$ alongside the noisy mesh $\hat{\mathcal{I}}_r$. The faces f are identical and the vertices have the same ordering in a one-to-one correspondence. For any $i \in [0, N_v)$, $\hat{\mathbf{v}}_i$ is the noisy counterpart of the target vertex $\mathbf{v}_i$ and $\tilde{\mathbf{v}}_i$ is the network's estimate for the respective denoised vertex. As shown in Fig. 5, the network is trained to learn displacement vectors $\tilde{\mathbf{d}}_i$ and computes denoised vertices as $\tilde{\mathbf{v}}_i = \hat{\mathbf{v}}_i + \tilde{\mathbf{d}}_i$, which is faster than directly learning denoised vertices $\tilde{\mathbf{v}}_i$.

The loss function penalizes errors in the positions of denoised vertices (L_{pos}) as well as errors in first and second order properties of the surface (L_{cur}). L_{pos} guides the network to move vertices as close as possible to their target position (Eq. 1), denoising them in the process. Squaring the differences places greater significance on vertices that are far away from their target.

$$L_{\text{pos}}(\tilde{v}, v) = \frac{1}{3N_v} \sum_{i=0}^{N_v-1} \sum_{j:\, x,y,z} (\tilde{\mathbf{v}}_i^j - \mathbf{v}_i^j)^2 \tag{1}$$

L_{cur} is designed to minimize deviations of the surface's curvature (and the surface normals), thereby avoiding self-intersections. Normally, the second order discrete mean curvature is approximated at vertices using the Laplace-Beltrami cotangent operator. If, however, any triangle in the 1-ring around a vertex is close to degeneration and has corner angles approaching zero, the derivative of cotangent heads towards minus infinity, derailing the learning process. Fortunately, the mean curvature can be reformulated as an edge-based operator, outlined in Hildebrandt and Polthier [17]. The mean curvature computed for an edge e_i as $\mathbf{K}(e_i) = 2\,|e_i| \cos(\theta_{e_i}/2)\mathbf{n}_{e_i}$, where $|e_i|$ is the edge length, θ_{e_i} is the dihedral angle between the two faces k and l that form the edge and $\mathbf{n}_{e_i}$ is the edge normal computed as $\mathbf{n}_{e_i} = (\mathbf{n}_k + \mathbf{n}_l) / \|\mathbf{n}_k + \mathbf{n}_l\|$. Using the edge-based mean curvature (which also depends on face normals), a mean square error can be constructed by Eq. 2, where N_e is the number of edges and K^j the j-th component of the edge-based mean curvature vector. This loss guides the network to produce denoised surfaces that approximate second-order properties, namely edge-based mean curvatures, of the target surfaces.

$$L_{\text{cur}}(\tilde{e}, e) = \frac{1}{3N_e} \sum_{i=0}^{N_e-1} \sum_{j:\, x,y,z} (K^j(\tilde{e}_i) - K^j(e_i))^2 \tag{2}$$

L_{cur} accelerates the network's learning process and helps to create denoised surfaces that approximate well the local smoothness and global shape of the target surfaces. The combined loss function can then be formulated as $L_{\text{tot}}(\tilde{\mathcal{I}}_r, \mathcal{I}_r) = L_{\text{pos}}(\tilde{v}, v) + \alpha \cdot L_{\text{cur}}(\tilde{e}, e)$, where α allows adjusting the influence of the curvature loss.

IV. EXPERIMENTS AND DISCUSSIONS

A. DATASETS

Traditionally, a small selection of individual models has been used to evaluate and compare algorithms for mesh denoising [50]–[52]. To the best of the authors' knowledge there is no established dataset to evaluate mesh denoising algorithms. In this work, two independent benchmark datasets are selected for the experiments, the *ABC* [5] and the *ShapeNetCore* [6] dataset. While both datasets are composed of Computer-Aided-Design (CAD) models, they focus on vastly different classes and shapes. The ABC dataset is a collection of one million CAD models featuring mostly basic geometric shapes used in manufacturing, including screws, plates, rods and blocks. For this work, 10.5k models are selected at random, which contain on average 14.6k vertices and 29.2k faces with a standard deviation of 7.7k vertices and 15.3k faces. ShapeNetCore is a subset of the ShapeNet dataset and includes about 51.3k models in 55 common object categories such as airplanes, cars and tables. For the experiments, 25k models are selected with 9.7k vertices and 19.5k faces on average and a standard deviation of 1.4k vertices and 2.8 faces. When models are not genus-0, they are transformed into genus-0 meshes using the technique previously described in [63]. As is common practice, the model vertices are subjected to artificial Gaussian white noise along their normals $\mathbf{n}_{\mathbf{v_i}}$. The amplitude of the perturbation is chosen as a fraction of a model's mean edge length l_e yielding $\hat{\mathbf{v}}_i = \mathbf{v_i} + x \cdot \mathbf{n}_{\mathbf{v_i}}$, $x \sim \mathcal{N}(0, b \cdot l_e)$, where the factor b determines the level of noise and is used to generate low, medium and high noise by selecting b as 0.1, 0.2 and 0.5, respectively. Figure 6 shows the influence of different noise-levels on an example model.

After noise application, the noisy models and their respective ground truths are remeshed as subdivided icosahedrons with 6 subdivisions ($\mathcal{I}_6$) using the preprocessing scheme discussed in section III-A.

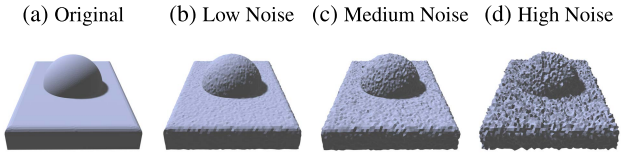

FIGURE 6. A model from *ABC* dataset in its original shape (a), subjected to low (b), medium (c) and high noise (d).

B. EVALUATION METRICS

The quality of denoising results is evaluated using two error metrics commonly deployed in mesh denoising. The mean angular difference, E_Θ (Eq. 3) measures the average difference between denoised and ground truth face normals. Where N_f denotes the number of faces in the mesh, $\tilde{\mathbf{n}}_i$ and $\mathbf{n}_i$ are the unit length normals of denoised and ground truth faces, respectively. A low error indicates a good recovery of the ground truth's shape and first-order surface properties.

$$E_\Theta = \frac{180^\circ}{N_f \cdot \pi} \sum_{i=0}^{N_f} \arccos\left(\tilde{\mathbf{n}}_i \cdot \mathbf{n}_i\right) \tag{3}$$

The mean distance error, E_D (Eq. 4) measures the average distance between two sets of vertices. Where N_v denotes the number of vertices in the mesh, $\tilde{\mathbf{v}}_i$ and $\mathbf{v}_i$ refer to the denoised and the ground truth vertices, respectively. A low E_D indicates volume and scale are well preserved in the denoising process.

$$E_D = \frac{1}{N_v} \sum_{i=0}^{N_v} \|\tilde{\mathbf{v}}_i - \mathbf{v}_i\|_2 \tag{4}$$

C. IMPLEMENTATION DETAILS

The icosahedral mesh denoising network, IMD-Net shown in Fig. 5 is implemented in python using PyTorch [64]. The implementation follows the outline of [59] to achieve convolution layers that produce equivariant feature maps on the surface of subdivided icosahedrons. The encoder path consists of one initial S2R-block producing 28 regular features, followed by three downsampling R2R-blocks outputting (44, 71, 114) regular features. The decoder block is arranged in reverse using three upsampling R2R-blocks outputting regular features and a final bottleneck layer that produces the output. IMD-Net is trained for 48 epochs with $\alpha = 10$ in L_{tot} at a learning rate of $4.5e^{-4}$ with Adam optimizer and a batch size of 8. The datasets are randomly split into training and test sets in an 80/20 ratio. The networks are trained for 48 epochs on the training set.

D. MESH DENOISING RESULTS

1) NON-LEARNING BASED METHODS

IMD-Net is compared against several state-of-the-art mesh denoising methods, namely with bilateral normal filtering (BNF) [1], guided normal filtering (GNF) [3], cascaded normal regression (CNR) [48] and non-local low-rank normal filtering (NLLR) [65]. The error metrics are computed for the test sets of *ABC* and *ShapeNetCore* datasets and are shown in Table 1. At first sight it becomes clear that IMD-Net outperforms its competitors in both metrics, independent of the noise level and dataset.

TABLE 1. Residual errors from different denoising methods for three noise levels on *ABC* and *ShapeNetCore* datasets.

Method ↓	ABC [5]					
Noise →	Low		Medium		High	
Error →	$E_\Theta[^\circ]$	$E_D[10^{-4}]$	$E_\Theta[^\circ]$	$E_D[10^{-4}]$	$E_\Theta[^\circ]$	$E_D[10^{-4}]$
BNF [1]	2.76°	2.80	4.24°	5.26	9.31°	11.70
GNF [3]	3.04°	3.26	4.28°	5.50	10.66°	15.87
CNR [48]	4.16°	3.34	6.89°	6.16	22.72°	21.79
NLLR [65]	3.89°	3.78	5.72°	6.33	11.22°	14.07
IMD-Net	2.14°	1.87	3.19°	3.34	5.79°	6.89
Method ↓	ShapeNetCore [6]					
Noise →	Low		Medium		High	
Error →	$E_\Theta[^\circ]$	$E_D[10^{-4}]$	$E_\Theta[^\circ]$	$E_D[10^{-4}]$	$E_\Theta[^\circ]$	$E_D[10^{-4}]$
BNF [1]	5.59°	8.68	9.23°	17.26	27.57°	38.81
GNF [3]	5.53°	8.54	8.73°	16.37	22.32°	38.43
CNR [48]	5.4°	8.61	8.5°	15.79	28.36°	37.66
NLLR [65]	5.42°	8.92	9.07°	17.40	26.91°	38.52
IMD-Net	3.97°	7.55	6.82°	14.26	11.91°	26.53

For low and medium noise levels, IMD-Net outperforms its closest competitor regarding E_Θ by 22.4-24.7% on *ABC* and by 19.7-26.5% on *ShapeNetCore*. This highlights IMD-Net's improved capability to recover the denoised normal field and its induced shape from noisy meshes. At high noise levels IMD-Net shows an even larger improvement of 37.8% and 46.6%, respectively. In addition, the network also achieves significantly lower values of E_D, e.g., a reduction of 41.2% and 29.5% for high noise levels. This implies that vertex positions denoised by IMD-Net are closer to the ground truth positions than with other algorithms and yield a more accurate volume recovery. The improved error metrics of IMD-Net indicates that the signal-to-noise ratio (SNR) at which it fails to distinguish true and noise-induced features is significantly higher than those of the competing algorithms. This can be credited to IMD-Net's design as a cU-net, which allows it to derive denoising filters combining various amounts of local and global information.

Most modern works, including the algorithms compared here ([1], [3], [48], [65]), first denoise the face normal field and then reposition the vertices to best fit the denoised normal field. When computed in sequence, a particular denoised normal field constrains the positions of the denoised vertices and defines a lower bound for E_D. This can contribute to an algorithm producing a lower E_Θ but a higher E_D than another one. IMD-Net, in contrast, is trained to denoise the vertex positions directly, taking into account derived quantities such as the face normal field and curvature in the loss function. This allows the network to optimize the output in regard to both error metrics, explaining the consistently low deviations from the ground truth.

With IMD-Net's quantitative progress and improved ability to generalize over large datasets evident, it is revealing to look at individual models in order to assess its impact on visual appearance, quality and fidelity of denoised models. Fig. 7 shows a qualitative comparison of denoising results at different noise levels. The magnified part of the object helps

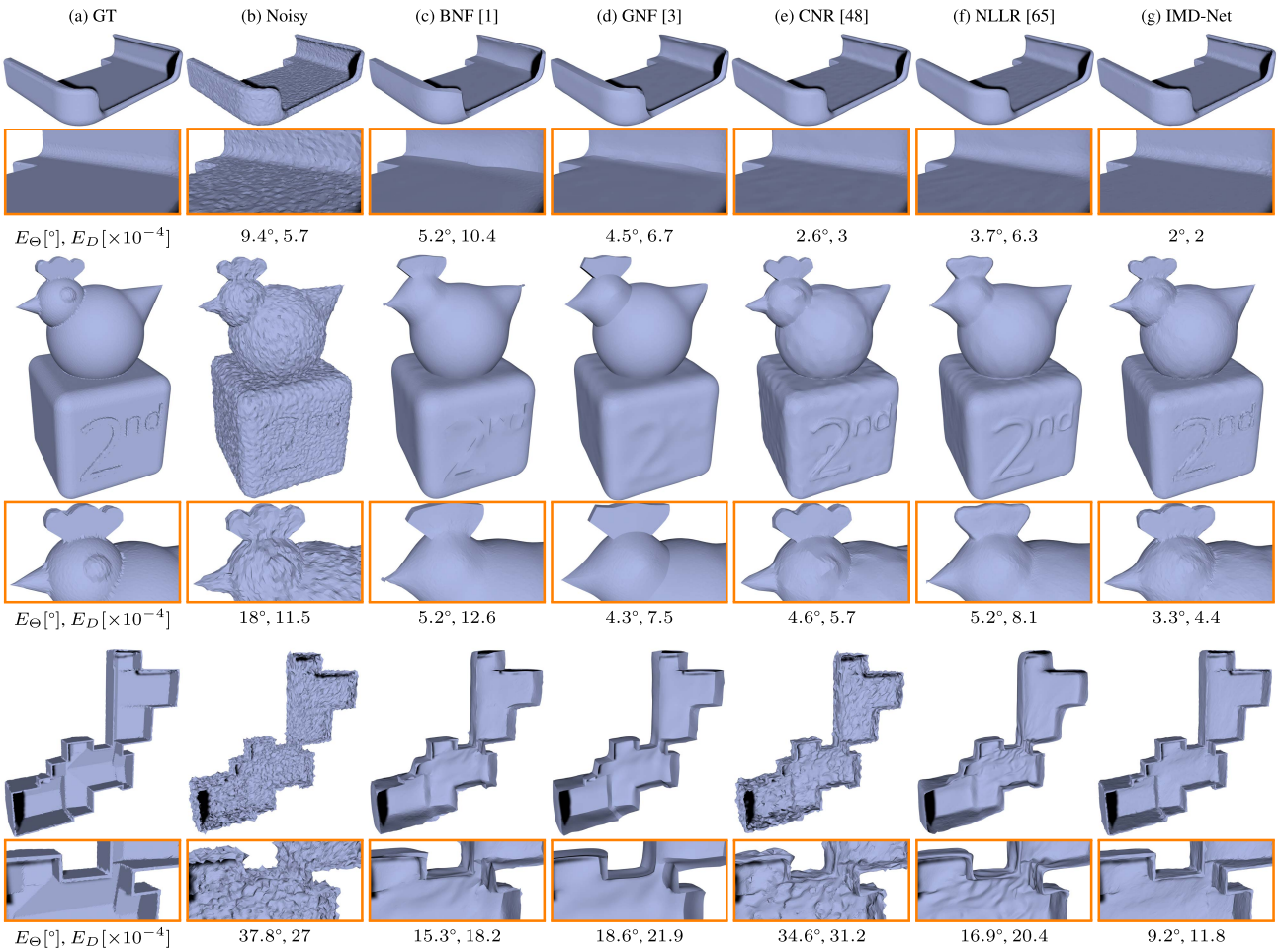

FIGURE 7. Illustration of denoising results on *ABC* dataset produced by competing denoising methods at different noise levels. Top to bottom: low, medium and high noise.

to understand how the algorithms under investigation handle flat areas, smooth transitions and sharp edges within a model. Most methods produce visually appealing results on parts of the surfaces. But some fail to accurately denoise smooth transitions and introduce artificial sharp edges of varying quality (Fig. 7c-d, top model). Others capture the transition well, but keep noisy oscillations in flat parts of the model (Fig. 7e-f, top model; Fig. 7c and f, bottom model). Again, others cause self-intersections in fine details of the mesh (Fig. 7c and f, middle model) or distort the volume noticeably (compare the size of the chicken's comb in Fig. 7, middle model). IMD-Net manages to recover flat surface, smooth transition as well as sharp edges with high fidelity. The proposed network conveniently avoids self-intersections, attributed to the use of second order properties L_{cur} in the loss function. Figure 8 shows some example models from *ShapeNetCore* [6] that were denoised by the competing denoising algorithms and IMD-Net at three different noise levels.

2) LEARNING BASED METHODS

Additionally, IMD-Net is compared to several learning-based methods, specifically to the data driven filtering using autoencoders (DFA) [50], denoising with facet graph convolutions (FGC) [51] and Deep Normal Filtering Network (DNF) [52]. FGC and DNF are pre-trained on 21 models from synthetic dataset of [48]. These methods require enormous computational resources and include time consuming pre- and post-processing stages, which restricts us from training them on large datasets (containing thousands of models) like the ones used in this work. Unlike, other learning based methods [50]–[52] which generate multiple patches from the shape and train the network using these patches, our proposed IMD-Net treats the complete mesh as an input. For all competing works the pre-trained networks published by the respective authors are used and for IMD-Net the network trained on the *ABC* dataset is used.

Fig. 10 presents denoising output of learning based methods on two shapes from *ABC* dataset and two shapes from test set of [48] synthetic dataset. DFA is pre-trained on eight CAD models and fails to accurately denoise the shapes from both the datasets. FGC and DNF perform quite well on test set of [48] (Fig. 10d-e, bottom two models) but fails to generalize on *ABC* dataset shapes (Fig. 10d-e, top two models) leaving traces of noisy oscillations in flat regions. IMD-Net, despite being trained on CAD models, performs well and rivals the performance of DNF and FGC on [48] test set

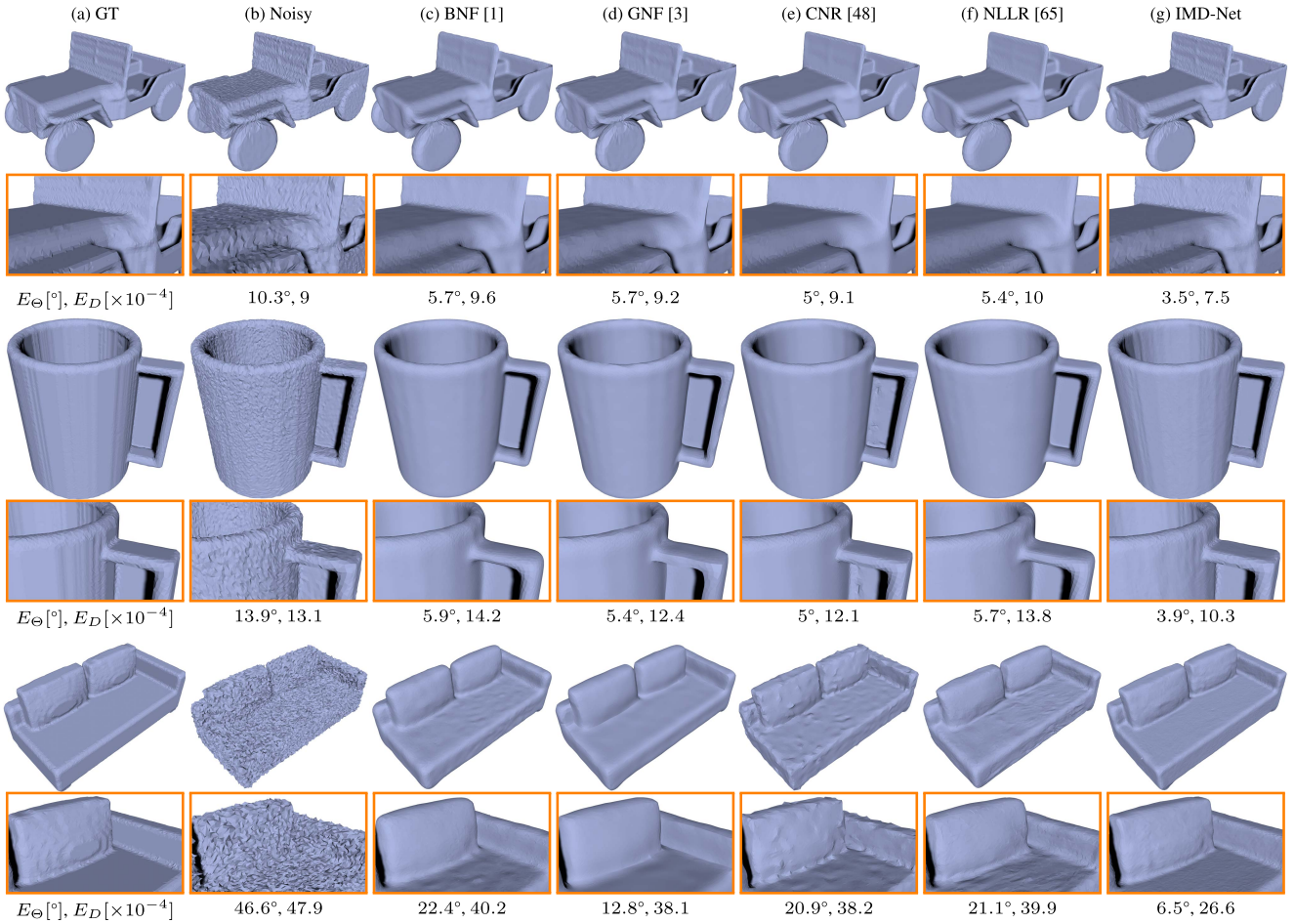

FIGURE 8. Denoised examples from *ShapeNetCore* produced by the competing denoising methods. From top to bottom: A car at low, a mug at medium and a sofa at high noise.

(Fig. 10f, bottom two models). We attribute this to the fact that IMD-Net is trained on large datasets of meshes (and not patches) hence generalizable to a wide variety of flat, curved or sharp features within a shape.

Also, IMD-Net produces result for a single model in few minutes (including pre-processing), whereas the other methods need between many minutes and hours. For e.g. the DNF required more than 100GB of intermediate storage, more than 80GB of RAM and a couple of hours for preprocessing, inference and postprocessing of the Armadillo model (Fig. 10 last row).

V. ABLATION STUDIES

The base configuration for IMD-Net, particularly the choices for the subdivison level, parametrization method, network architecture and the network depth, require both validation and justification. Therefore, three ablation studies are conducted, which explore the impact of these parameters on the performance.

A. SUBDIVISION LEVEL

To observe the influence of subdivision level r on the shape detail, models from *ABC* dataset [5] are subjected to medium noise and preprocessed using icosahedral meshes with subdivisions five (S5), six (S6) and seven (S7). The mean Hausdorff distance of remeshed *ABC* models at three subdivisions is shown in Fig. 9a accompanied by statistics on mesh sizes in Fig. 9b. In both figures a log scale is used and on this scale the curves are quasi-linear. This suggests that, in the given range of subdivisions, an increase of the subdivision level by one reduces the Hausdorff distance exponentially by a factor

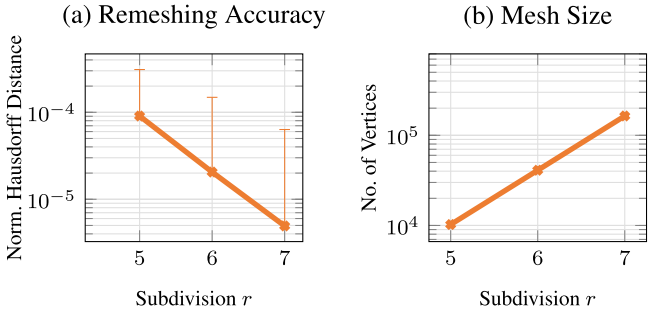

FIGURE 9. The normalized Hausdorff distance and mesh size of the remeshed *ABC* dataset at different subdivision level. $r = 6$ shows good agreement between Hausdorff distance and model size.

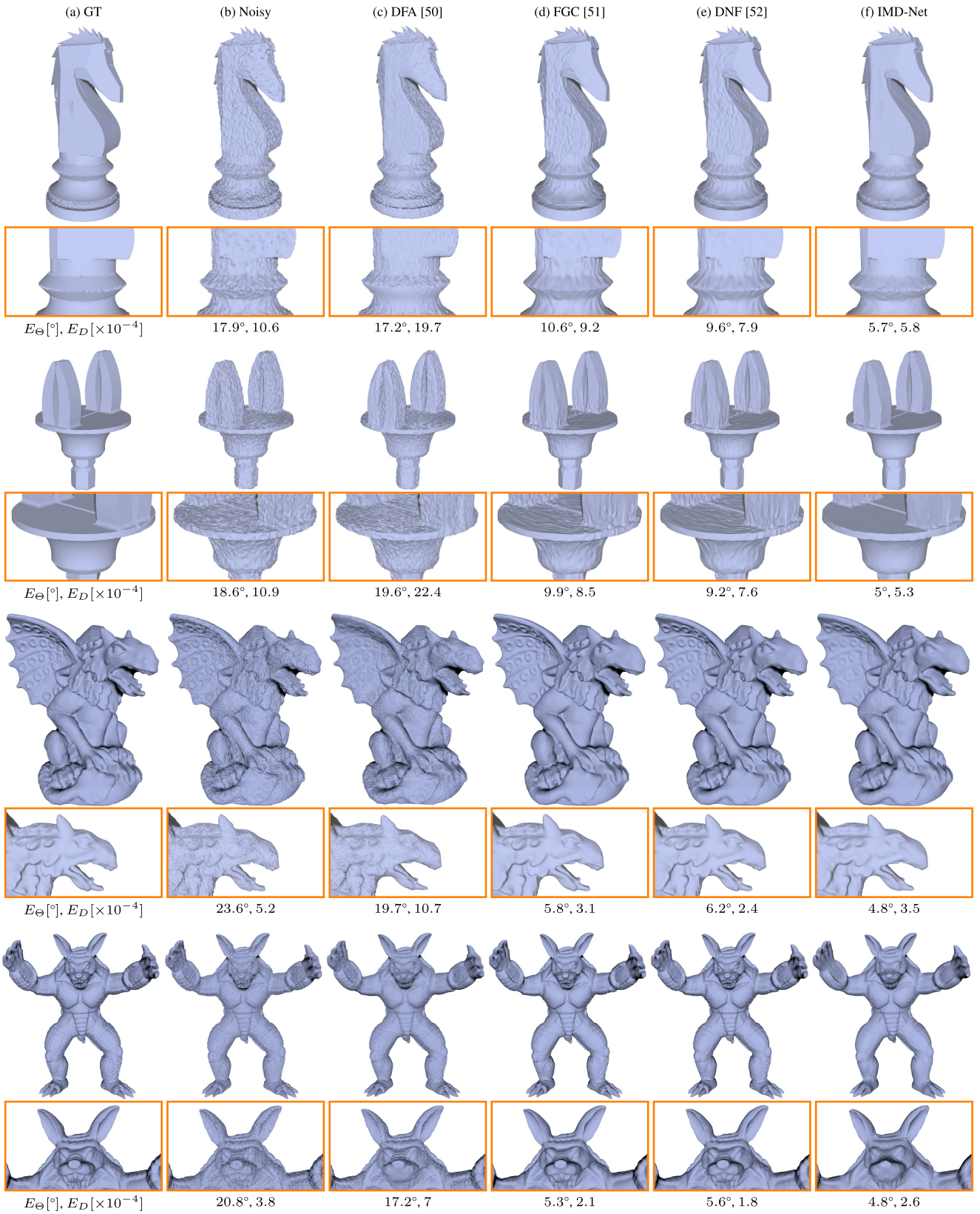

FIGURE 10. The *ABC* dataset (top two) and the synthetic dataset [48] (bottom two: gargoyle and armadillo) models subjected to medium noise and denoised by competing learning-based denoising methods.

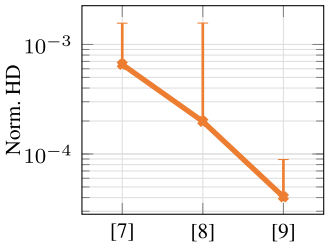

FIGURE 11. Mean and standard deviation of the normalized Hausdorff distance between the remeshed *ABC* dataset models at $r = 6$ and the ground truth for three different spherical parameterization methods: conformal [7], authalic [8] and quasi-isometric [9].

in the range of [0.2, 0.25]. It follows that the expected gain in accuracy shrinks exponentially with increasing subdivision. In absolute values, an increase from S5 to S6 reduces the Hausdorff distance by 7.01×10^{-5} and a further increase from S6 to S7 by 1.56×10^{-5}. This renders the increase from S5 to S6 about 4.5 times more efficient than the subsequent increase to S7.

At the same time, the vertex and face count in a remeshed model grow exponentially by a factor of about 4 with each increase of the subdivision level (Fig. 9b). This has severe speed and memory implications for any algorithm denoising the model as the size of the input data grows by the same factor. The implication for IMD-Net is that the required model size and memory resources grow exponentially with the subdivision. Together, the reciprocal nature of an exponentially slowing reduction in the Hausdorff distance and an exponential growing number of vertices and faces impose strict limits on feasible subdivisions. If the subdivision is too small, the accuracy of the remeshed model might be insufficient; if it is too big, the mesh size might make processing infeasible. In the given context, S6 is selected for denoising experiments, as it guarantees a good tradeoff between a small Hausdorff distance and a reasonable model size.

B. PARAMETRIZATION ALGORITHM

In computer graphics, different parametrization methods are compared with respect to the distortion the algorithms introduce in either the area or angles of faces. However, in this work, the parametrization is used as a preprocessing tool to remesh an input mesh onto a semi-regular grid. This approach, like any other remeshing technique, may result in the loss of shape details. Therefore, the different parametrization algorithms are compared with respect to the loss of 3D shape information caused by the parametrization and remeshing of an input mesh.

To observe the impact of different parametrization algorithms, models from the ABC dataset [5] were parametrized using three different parametrization schemes (conformal, authalic, quasi-isometric) and remeshed at subdivision level $r = 6$. The remeshed output shapes were then quantitatively compared to the input ground truth in terms of normalized Hausdorff distance, shown in Fig. 11.

The AHSP algorithm [9] applied in this work yields the smallest Hausdorff distance after remeshing. The other two methods, exemplifying conformal and authalic approaches, have approximately a 16.2 and 5.0 times higher Hausdorff distance when compared to the quasi-isometric AHSP approach. Further, a qualitative visualization of some models parametrized and remeshed using the three different methods is shown in Fig. 12. The clustering of extrusive shape features in conformal parametrizations results in incomplete shape preservation after remeshing. The authalic parametrization fares better, but it suffers from stretched triangles in the parametrization, and, as a consequence, the remeshed output misses shape details in regions of high Gaussian curvature. In contrast to both, AHSP consistently outputs detail and shape preserving remeshings and is therefore our choice of parametrization.

C. NETWORK ARCHITECTURE

In this ablation study, the cU-net is compared against the original U-net, and two other conceivable network architectures, derived from the base configuration. A ConvNet, which keeps the feature map size constant throughout the network (no pooling and unpooling) and drops the skip connections; and an autoencoder (AE), which shares the cU-net architecture in all but the missing skip connections. The four architectures are trained and tested in an identical setup using *ABC* dataset at medium noise level. Fig. 13 compares from left to right the denoising performance measured by E_Θ, the average GPU memory footprint and the model size in terms of trainable parameters.

The comparison of E_Θ in the leftmost chart reveals that the AE fails to learn to denoise meshes, reducing the error to only 18.36° where the other architectures achieve around 4°. This can be explained by the missing skip connections, which are essential for the given task. The other three architectures make local information directly available for vertex denoising, either through skip connections or by avoiding down- and up-sampling. These three architectures perform well, with cU-Net slightly outperforming the other two. However, it becomes obvious that Conv-Net is not a suitable choice when looking at the GPU memory usage, presented in Fig. 13b. With a batch size of only 4 it occupies a staggering 78.2% of the GPU memory, where cU-Net only requires about 50%. Since there is no pooling and unpooling in the network, the deeper blocks with many feature layers consume an enormous amount of memory. As it does not perform better than cU-Net or U-net, it can be argued that Conv-Net produces internally a lot of features that do not carry relevant information for denoising, rendering the other architectures more efficient.

Finally, the size of the different models, presented in Fig. 13c, explains why cU-Net is to be preferred as architecture over a vanilla U-net. By replacing the concatenation with an addition operation, the model size is reduced by about one third (927K vs 1.4M parameters) and performance is kept steady, clearly making cU-Net the architecture of choice.

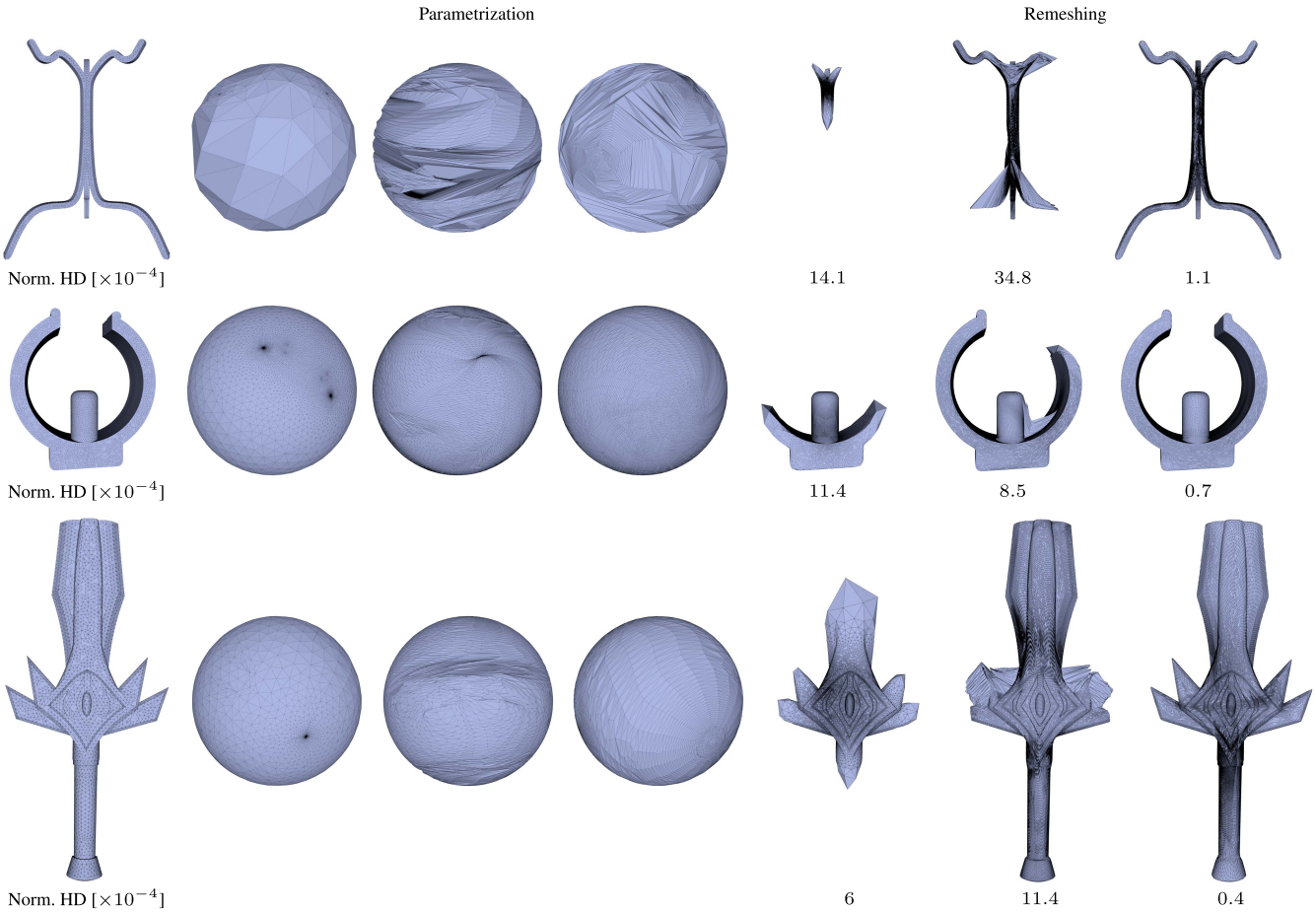

FIGURE 12. Models from the *ABC* dataset parametrized and remeshed at $r = 6$ using conformal [7], authalic [8] and quasi-isometric [9] respectively. Remeshings generated using AHSP best preserve details and shapes. The normalized Hausdorff distance (Norm. HD) is shown below the remeshed model for comparison.

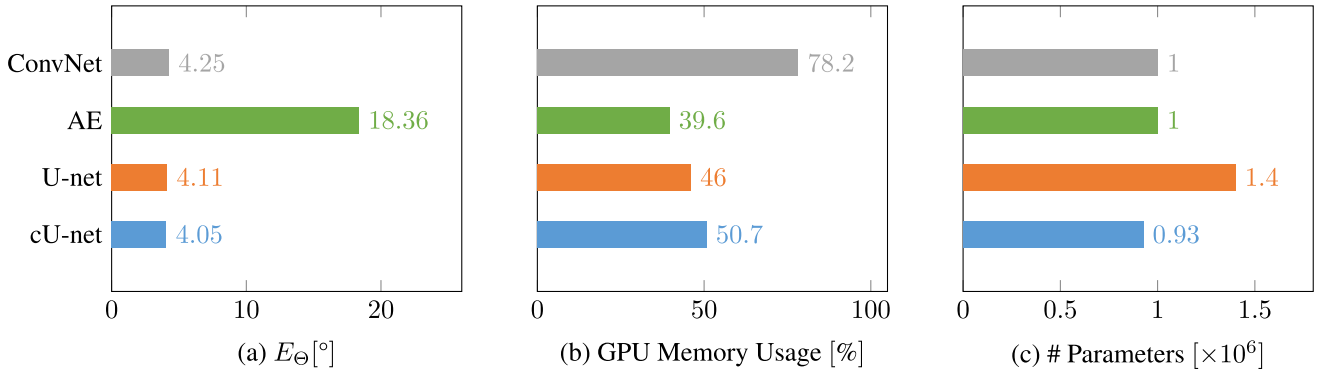

FIGURE 13. The network architectures ConvNet, AE, U-net and cU-net are compared in regard to E_Θ, GPU memory usage and model size. cU-net is the architecture of choice, as AE fails to reduce E_Θ, ConvNet consumes an infeasible amount of GPU memory and U-net has a larger model size at similar performance.

D. NETWORK DEPTH

Another ablation study is conducted to observe the influence of network depth on IMD-Net's denoising performance. The depth refers to the number of R2R-Blocks in encoder and decoder. The network depth is a crucial hyperparameter, determining the maximum receptive field of features and thereby influencing the amount of global information available for denoising. In order to allow a fair comparison, the four networks are normalized with respect to the model size, so that each model has about one million trainable parameters. The performance over 48 training epochs, measured by E_Θ and E_D, is plotted in Fig. 14. At the end of the training, E_Θ

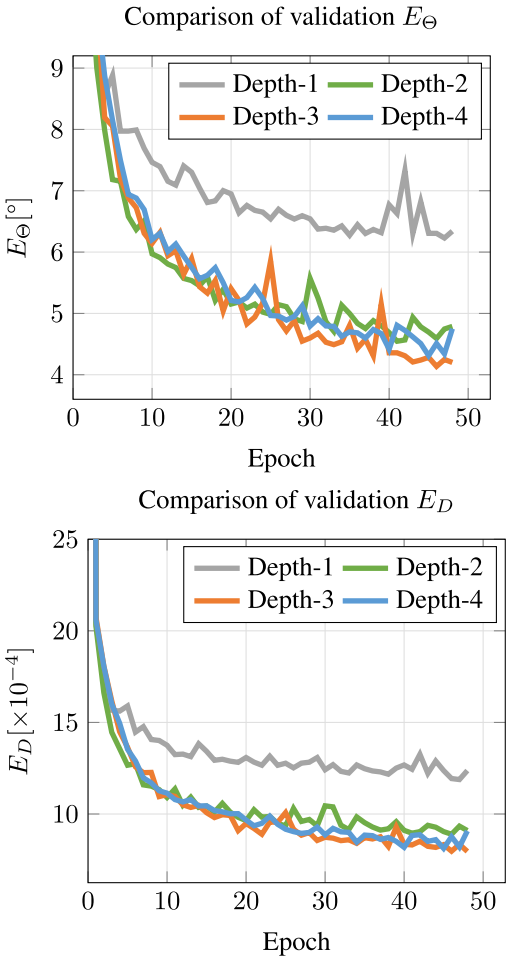

FIGURE 14. The validation error metrics of four cU-nets ranging from depth-1 to depth-4. Depth-3 performs consistently better than others on the two metrics.

is least on depth-3 network. With respect to the E_D, depth-3 and depth-4 networks have the lowest positional distance. Depth-4 network's slight decrease in performance indicates that additional blocks add model complexity but little useful information. This encourages using a depth-3 network, as it yields the best tradeoff between denoising performance and model complexity.

VI. CONCLUSION

This paper proposed a novel mesh denoising technique, the icosahedral mesh denoising network (IMD-Net), which is a deep neural network especially suited to denoise closed genus-0 meshes with high quality and fidelity in a single pass. IMD-Net consumes a noisy mesh, computes a remeshing and predicts denoised vertex positions while preserving and enhancing features of the original mesh. Beyond vertices and faces of a noisy mesh, no other explicit information about noise distribution or surface characteristics is needed. Training and inference of IMD-Net are exceptionally fast, as it is based on standard conv2D-layers. The nature of the proposed deep learning approach ensures that no parameters need tuning at inference. IMD-Net was trained and tested on two large-scale model datasets, *ABC* and *ShapeNetCore*, and compared to state-of-the-art learning and non-learning based algorithms. To the best of our knowledge, it is the first time mesh denoising algorithms are evaluated on such large datasets. The experiments showed that IMD-Net consistently outperforms other algorithms, both in terms of objective evaluation metrics and subjective visual inspections. Training on large dataset had the merit of being generalizable on shapes from different dataset, unlike other learning based methods. IMD-Net also illustrates that using the complete mesh as network input benefits the denoising process by utilizing global shape information. IMD-Net's performance advantage grows with increasing noise levels, standing proof of the positive impact that integrating local, non-local and global mesh information into the denoising process can have.

Future work includes exploring to mark a consistent seam on non genus-0 surfaces to parametrize them. This way, the remeshing approach proposed in this work can be utilized to remesh the surfaces and IMD-Net can be used to denoise meshes of higher genus.

REFERENCES

[1] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 10, pp. 1521–1530, Oct. 2013.
[2] L. He and S. Schaefer, "Mesh denoising via L_0 minimization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–8, Jul. 2013.
[3] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, "Guided mesh normal filtering," *Comput. Graph. Forum*, vol. 34, no. 7, pp. 23–34, 2015.
[4] W. Zhao, X. Liu, Y. Zhao, X. Fan, and D. Zhao, "NormalNet: Learning-based normal filtering for mesh denoising," 2019, *arXiv:1903.04015*.
[5] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "ABC: A big CAD model dataset for geometric deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9593–9603.
[6] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[7] P. T. Choi, K. C. Lam, and L. M. Lui, "Flash: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces," *SIAM J. Imag. Sci.*, vol. 8, no. 1, pp. 67–94, 2015.
[8] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science), vol. 9910. Amsterdam, The Netherlands: Springer, 2016, pp. 223–240.
[9] X. Hu, X.-M. Fu, and L. Liu, "Advanced hierarchical spherical parameterizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 6, pp. 1930–1941, Jun. 2018.
[10] D. A. Field, "Laplacian smoothing and Delaunay triangulations," *Commun. Appl. Numer. Methods*, vol. 4, no. 6, pp. 709–712, Nov. 1988.
[11] J. Vollmer, R. Mencl, and H. Müller, "Improved Laplacian smoothing of noisy surface meshes," *Comput. Graph. Forum*, vol. 18, no. 3, pp. 131–138, Sep. 1999.
[12] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn.*, New York, NY, USA, 1995, pp. 351–358.
[13] G. Taubin, T. Zhang, and G. Golub, "Optimal surface smoothing as filter design," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1064. Berlin, Germany: Springer, 1996, pp. 283–292.
[14] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*. New York, NY, USA, Jul. 1999, pp. 317–324.

[15] B. Kim and J. Rossignac, "GeoFilter: Geometric selection of mesh filter parameters," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 295–302, 2005.
[16] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," in *Proc. Vis. (VIS)*, Oct. 2000, pp. 397–405.
[17] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 391–400, 2004.
[18] M. Centin and A. Signoroni, "Mesh denoising with (Geo)Metric fidelity," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2380–2396, Aug. 2018.
[19] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 950–953, 2003.
[20] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 943–949, 2003.
[21] G. Taubin, "Linear anisotropic mesh filtering," IBM, Armonk, NY, USA, Res. Rep. RC2213, 2001, vol. 1, no. 4.
[22] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh smoothing via mean and median filtering applied to face normals," in *Proc. Geometric Modeling Process. Theory Appl. (GMP)*, Jul. 2002, pp. 124–131.
[23] H. Yagou, Y. Ohtake, and A. G. Belyaev, "Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding," in *Proc. Comput. Graph. Int.*, 2003, pp. 28–33.
[24] C.-Y. Chen and K.-Y. Cheng, "A sharpness dependent filter for mesh smoothing," *Comput. Aided Geom. Des.*, vol. 22, no. 5, pp. 376–391, Jul. 2005.
[25] Y. Shen and K. E. Barner, "Fuzzy vector median-based surface smoothing," *IEEE Trans. Vis. Comput. Graphics*, vol. 10, no. 3, pp. 252–265, May 2004.
[26] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 5, pp. 925–938, Sep./Oct. 2007.
[27] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Random walks for feature-preserving mesh denoising," *Comput. Aided Geometric Des.*, vol. 25, no. 7, pp. 437–456, Oct. 2008.
[28] T. Li, J. Wang, H. Liu, and L.-G. Liu, "Efficient mesh denoising via robust normal filtering and alternate vertex updating," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 11, pp. 1828–1842, Nov. 2017.
[29] J. Hurtado, M. Gattass, A. Raposo, and J. Coelho, "Adaptive patches for mesh denoising," in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2018, pp. 1–8.
[30] J. R. Diebel, S. Thrun, and M. Brünig, "A Bayesian method for probable surface reconstruction and decimation," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 39–59, Jan. 2006.
[31] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proc. 4th Int. Conf. Comput. Graph. Interact. Techn. Australasia Southeast Asia (GRAPHITE)*, 2006, pp. 381–389.
[32] L. Liu, C.-L. Tai, Z. Ji, and G. Wang, "Non-iterative approach for global mesh optimization," *Comput.-Aided Des.*, vol. 39, no. 9, pp. 772–782, Sep. 2007.
[33] K. Hildebrandt and K. Polthier, "Constraint-based fairing of surface meshes," in *Proc. Eurographics Symp. Geometry Process.*, 2007, pp. 1–10.
[34] H. Zhang, C. Wu, J. Zhang, and J. Deng, "Variational mesh denoising using total variation and piecewise constant function space," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 7, pp. 873–886, Jul. 2015.
[35] X. Wu, J. Zheng, Y. Cai, and C. W. Fu, "Mesh denoising using extended ROF model with L_1 fidelity," in *Computer Graphics Forum*, vol. 34. Hoboken, NJ, USA: Blackwell, Oct. 2015, pp. 35–45.
[36] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted ℓ_1-analysis compressed sensing," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 1–12, Mar. 2014.
[37] X. Lu, Z. Deng, and W. Chen, "A robust scheme for feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 3, pp. 1181–1194, Mar. 2016.
[38] X. Lu, W. Chen, and S. Schaefer, "Robust mesh denoising via vertex pre-filtering and L1-median normal filtering," *Comput. Aided Geometric Des.*, vol. 54, pp. 49–60, May 2017.
[39] M. Wei, L. Liang, W.-M. Pang, J. Wang, W. Li, and H. Wu, "Tensor voting guided mesh denoising," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 931–945, Apr. 2017.
[40] S. Yadav, U. Reitebuch, and K. Polthier, "Mesh denoising based on normal voting tensor and binary optimization," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2366–2379, Aug. 2018.
[41] M. Wei, H. Jin, X. Xie, L. Liu, and Q. Jing, "Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 10, pp. 2910–2926, Oct. 2019.
[42] X. Lu, S. Schaefer, J. Luo, L. Ma, and Y. He, "Low rank matrix approximation for 3D geometry filtering," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 4, pp. 1835–1847, Apr. 2022.
[43] G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Feature preserving mesh denoising based on graph spectral processing," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 3, pp. 1513–1527, Mar. 2019.
[44] S. Yadav, U. Reitebuch, and K. Polthier, "Robust and high fidelity mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 6, pp. 2304–2310, Jan. 2019.
[45] W. Zhao, X. Liu, S. Wang, X. Fan, and D. Zhao, "Graph-based feature-preserving mesh normal filtering," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 3, pp. 1937–1952, Mar. 2021.
[46] T. Li, W. Liu, H. Liu, J. Wang, and L. Liu, "Feature-convinced mesh denoising," *Graph. Models*, vol. 101, pp. 17–26, Jan. 2019.
[47] W.-H. Lee, M. Ozger, U. Challita, and K. W. Sung, "Noise learning-based denoising autoencoder," *IEEE Commun. Lett.*, vol. 25, no. 9, pp. 2983–2987, Sep. 2021.
[48] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.
[49] J. Wang, J. Huang, F. L. Wang, M. Wei, H. Xie, and J. Qin, "Data-driven geometry-recovering mesh denoising," *Comput.-Aided Des.*, vol. 114, pp. 133–142, Sep. 2019.
[50] S. Nousias, G. Arvanitis, A. S. Lalos, and K. Moustakas, "Fast mesh denoising with data driven normal filtering using deep autoencoders," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2019, pp. 260–263.
[51] M. Armando, J.-S. Franco, and E. Boyer, "Mesh denoising with facet graph convolutions," *IEEE Trans. Vis. Comput. Graphics*, early access, Dec. 17, 2020, doi: 10.1109/TVCG.2020.3045490.
[52] X. Li, R. Li, L. Zhu, C.-W. Fu, and P.-A. Heng, "DNF-net: A deep normal filtering network for mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 10, pp. 4060–4072, Oct. 2021.
[53] X. Gu and S.-T. Yau, "Global conformal surface parameterization," in *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Process.*, 2003, pp. 127–137.
[54] A. Dominitz and A. Tannenbaum, "Texture mapping via optimal mass transport," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 3, pp. 419–433, May 2010.
[55] H. Jain and O. Hellwich, "GenIcoNet: Generative icosahedral mesh convolutional network," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 64–73.
[56] W. Boomsma and J. Frellsen, "Spherical convolutions and their application in molecular modelling," in *Proc. NIPS*, vol. 2, 2017, p. 6.
[57] Y. C. Su and K. Grauman, "Learning spherical convolution for fast features from 360° imagery," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 530–540.
[58] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2018, pp. 1–15.
[59] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, "Gauge equivariant convolutional networks and the icosahedral CNN," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Feb. 2019, pp. 2357–2371.
[60] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351. Cham, Switzerland: Springer, May 2015, pp. 234–241.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2015, pp. 448–456.
[63] H. Jain, M. Wollhaf, and O. Hellwich, "Learning to reconstruct symmetric shapes using planar parameterization of 3D surface," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4133–4140.
[64] A. Paszke, S. Gross, F. Massa, and A. Lerer, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035.
[65] X. Li, L. Zhu, C. W. Fu, and P. A. Heng, "Non-local low-rank normal filtering for mesh denoising," *Comput. Graph. Forum*, vol. 37, no. 7, pp. 155–166, Oct. 2018.

JAN BOTSCH received the B.Sc. degree in engineering science from the Technical University of Munich, in 2015, and the M.Sc. degree in computer engineering from the Technical University of Berlin in 2021, both supported by a scholarship of the German Academic Scholarship Foundation. Since 2015, he has been working in the private and public sector on researching and applying image segmentation, 3D-reconstruction, texture mapping, and mesh processing techniques involving traditional algorithm engineering and machine learning.

HARDIK JAIN received the B.E. degree in electronics and instrumentation from the Shri Govindram Seksaria Institute of Technology and Science, Indore, India, in 2014, and the M.Tech. degree in instrumentation and signal processing from the Indian Institute of Technology (IIT) Roorkee, India, in 2016. He is currently pursuing the Ph.D. degree with the Computer Vision and Remote Sensing Group, Technische Universität Berlin, Germany. During his M.Tech. degree, he received DAAD IIT Master Sandwich Scholarship for completing dissertation with the Technische Universität Berlin. After his master's degree, he was a Junior Research Fellow with the Multimedia Analysis and Security Laboratory, Indian Institute of Technology Gandhinagar, India, from 2016 to 2017. He received DAAD Research Grants, in 2017. His research interests include 3D computer vision, 2D/3D machine learning, image analysis, and medical data registration. He was a recipient of the Best Student Award for Academic Excellence 2016 at IIT Roorkee.

OLAF HELLWICH (Senior Member, IEEE) received the M.Sc. degree in surveying engineering from the University of New Brunswick, Canada, in 1992, and the Ph.D. degree in linienextraktion aus SAR-daten mit einem Markoff-Zufallsfeld-Modell from the Technische Universität München, Germany, in 1997. He headed the Remote Sensing Group, Department of Photogrammetry and Remote Sensing, Technische Universität München. Since 2001, he has been a Professor with the Technische Universität Berlin, Germany, initially for photogrammetry and cartography, and since 2004, for computer vision and remote sensing. From 2007 to 2009, he was the Dean of the Faculty of Electrical Engineering and Computer Science, Technische Universität Berlin. He was a recipient of the Hansa Luftbild Prize of the German Society for Photogrammetry and Remote Sensing, in 2000. Since 2019, he has been a Principal Investigator in the Science of Intelligence Cluster under Germany's Excellence Strategy. His research interests include 3D object reconstruction, object recognition, synthetic aperture radar remote sensing, discovery and use of object shape priors in 3D reconstruction, and the use of vision in the modeling of intelligent systems.