



Revisiting Point Cloud Simplification: A Learnable Feature Preserving Approach

Rolandos Alexandros Potamias^(✉), Giorgos Bouritsas, and Stefanos Zafeiriou

Imperial College London, London, UK
{r.potamias,g.bouritsas,s.zafeiriou}@imperial.ac.uk

Abstract. The recent advances in 3D sensing technology have made possible the capture of point clouds in significantly high resolution. However, increased detail usually comes at the expense of high storage, as well as computational costs in terms of processing and visualization operations. Mesh and Point Cloud simplification methods aim to reduce the complexity of 3D models while retaining visual quality and relevant salient features. Traditional simplification techniques usually rely on solving a time-consuming optimization problem, hence they are impractical for large-scale datasets. In an attempt to alleviate this computational burden, we propose a fast point cloud simplification method by learning to sample salient points. The proposed method relies on a graph neural network architecture trained to select an arbitrary, user-defined, number of points according to their latent encodings and re-arrange their positions so as to minimize the visual perception error. The approach is extensively evaluated on various datasets using several perceptual metrics. Importantly, our method is able to generalize to out-of-distribution shapes, hence demonstrating zero-shot capabilities.

Keywords: Point clouds · Simplification · Sampling · GNN

1 Introduction

The progress in sensing technologies has significantly expedited the 3D data acquisition pipelines which in turn has increased the availability of large and diverse 3D datasets. With a single 3D sensing device [39], one can capture a target surface and represent it as a 3D object, with point clouds and meshes being the most popular representations. Several applications, ranging from virtual reality and 3D avatar generation [26, 46] to 3D printing and digitization of cultural heritage [43], depend from such representations. However, in general, a 3D capturing device generates thousands of points per second, making processing, visualization and storage of captured 3D objects a computationally daunting task. Often, raw point sets contain an enormous amount of redundant and putatively noisy points of points with low visual perceptual importance, which results

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20086-1_34.

into an unnecessary increase in the storage costs. Thus processing, rendering and editing applications require the development of efficient simplification methods that discard excessive details and reduce the size of the object, while preserving their significant visual characteristics. In particular, triangulation, registration and editing processes of real-world scans include an initial simplification step to achieve real-time meshing and rendering. In contrast to sampling methods that aim to preserve the overall point cloud structure, simplification methods attempt to solve the non-trivial task of preserving the semantics of the input objects [8]. Visual semantics refer to the salient features of the object that mostly correlate with human perception and determine its visual appearance in terms of curvature and roughness characteristics of the shape [27, 33]. As can be easily observed in an indicative case shown in Fig. 1, effortless sampling techniques, such as FPS or uniform sampling, can easily preserve the structure of a point cloud. However, the preservation of perceptually visual characteristics, especially when combined with structural preservation, remains a challenging task.

Traditional simplification methods manage to retain the structural and the salient characteristics of large point clouds [15, 42, 49] by constructing a point importance queue that sorts points according to their scores at every iteration. However, apart from being very time demanding, such optimizations are non-convex with increased computational requirements and can not be generalized to different topologies. On the contrary, an end-to-end differentiable neural-based simplification method could leverage the parallel processing of neural networks and simplify batches of point clouds in one pass [45].

In this study, we tackle the limitations of the literature on the task of point cloud simplification and propose the first, to the best of our knowledge, learnable point cloud simplification method. Motivated by the works that showcase the importance of perceptual saliency, we propose a method that preserves both the salient features as well as the overall structure of the input and can be used for real-time point cloud simplification without any prior surface reconstruction. Given that our method is fully differentiable, it can be directly integrated to any learnable pipeline without any modification. Additionally, we introduce a fast latent space clustering using FPS that could benefit several fields (such as graph partitioning, generative models, shape interpolation, etc.), serving as a fast (non-iterative) alternative to differentiable clustering, in a way that the cluster center selection is guided by the loss function. Finally, we highlight the limitations of popular distance metrics, such as Chamfer, to capture salient details of the simplified models, and we propose several evaluation criteria, well-suited for simplification tasks. The proposed method is extensively evaluated in a series of wide range experiments.

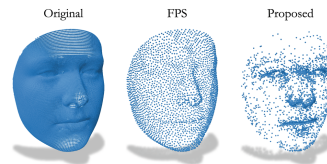


Fig. 1. Point cloud simplified using FPS (left) smooths out facial characteristics of the input whereas our method preserves salient features of the input (right).

2 Related Work

Mesh Simplification is a well studied field with long history of research. A mesh can be simplified progressively by two techniques, namely vertex decimation and edge collapse. Although the first approach is more interpretable, since it assigns importance scores to each vertex [36,51,52], it requires careful re-tessellation in order to fill the generated holes. Edge folding was first introduced in [18], where several transforms such as edge swap, edge split and edge collapse were introduced to minimize a simplification energy function. In the seminal works of [15,50], the authors associated each vertex with the set of planes in its 1-hop neighborhood and defined a fundamental *quadric* matrix quantifying the distance of a point from its respective set of planes. A Quadric Error Metric (QEM) was proposed to measure the error introduced when an edge collapses, ensuring that edges with the minim error are folded first. Several modification of the QEM cost function have been proposed to incorporate curvature features [22,23,38,62], mesh saliency [2,40,59] or to preserve boundary constrains [3] and spectral mesh properties [34,37]. Recently, [17] proposed a learnable edge collapse method that learns the importance of each edge, for task-driven mesh simplification. However, the edges are contracted in an inefficient iterative way and the resulting mesh can only be decimated approximately by two.

Point Cloud Simplification and Sampling: Similar to mesh simplification, iterative point selection and clustering techniques have also been proposed for point clouds [16,42,49,53,64]. In particular, *point cloud simplification* can be addressed either via mesh simplification where the points are fitted to a surface and then simplified using traditional mesh simplification objectives [1,14], or via direct optimization on the point cloud where the points are selected and decimated according to their estimated local properties [32,42,47,53,64]. However, similar to mesh simplification, computationally expensive iterative optimization is needed, making them inefficient for large scale point clouds. The point cloud simplification methodology presented in this paper attempts to address and overcome the inefficiencies of the aforementioned approaches using a learnable alternative that works with arbitrary, user-defined decimation factors.

In a different line of research, *sampling methods* rely on a point selection scheme that focuses on retaining either the overall structure of the object or specific components of the input. A huge difference between simplification and sampling is founded upon their point selection perspective. Sampling methods are usually utilized for hierarchical learning [48] in contrast to simplification methods that attempt to preserve as much of the visual appearance of the input even at very low resolutions. Farthest Point Sampling (FPS) [12], along with several modification of it, remains the most popular sampling choice and has been widely used as a building block in deep learning pipelines [47,48,61]. Nevertheless, as we experimentally showcase, FPS directly from the input xyz-space can not preserve sharp details of the input and thus is not suitable for simplification tasks. Recently, several methods [11,25] have been proposed as a learnable alternative for task-driven sampling, optimized for downstream tasks. However, they

require the input point clouds to have the same size which limits their usage to datasets with different topologies. In addition, they explicitly generate the sampled output using linear layers which is not scalable to large point clouds. Although the learnable sampling methods are closely related to ours, they only sample point clouds in a task-driven manner and as a result, the preservation of the high frequency details of the point cloud is not ensured.

Assessment of Perceptual Visual Quality: Processes such as simplification, lossy compression and watermarking inevitably introduce distortion to the 3D objects. Measuring the visual cost in rendered data is a long studied problem [9, 30]. Inspired by Image Quality Assessment measures, the objective of Perceptual Visual Quality (PVQ) assessment is to measure the distortion of an object in terms directly correlated with the Human Perceptual System (HPS). Several methods have been proposed, acting directly on 3D positions, to measure the PVQ using Laplacian distances [21], curvature statistics [24, 28, 54], dihedral angles [55] or per vertex roughness [9]. Several studies [10, 13, 29, 63] utilized crowdsourcing platforms and user subjective assessments to identify the most relevant geometric attributes that mostly correlate with human perception. The findings demonstrated that curvature related features along with dihedral angles and roughness indicate strong similarity with the HVS. In this work, motivated by the aforementioned studies we utilized curvature related losses and quality measures to train and assess the performance of the proposed model and we refer to them as perceptual measures.

3 Method

3.1 Preliminaries: Point Curvature Estimation

Calculating the local surface properties of an unstructured point cloud is a non-trivial problem. As demonstrated in [19, 42], covariance analysis can be an intuitive estimator of the surface normals and curvature. In particular, considering a neighborhood \mathcal{N}_i around the point $\mathbf{p}_i \in \mathcal{R}^3$ we can define the covariance matrix:

$$\mathbf{C} = \frac{1}{k-1} \sum_{i=1}^k (\mathbf{p}_{i_1} - \mathbf{p}_i)^T (\mathbf{p}_{i_1} - \mathbf{p}_i), \quad \mathbf{p}_{i_j} \in \mathcal{N}_i \quad (1)$$

Solving the eigendecomposition of the covariance matrix \mathbf{C} we can derive the eigenvectors corresponding to the principal eigenvalues that define an orthogonal frame at point \mathbf{p}_i . The eigenvalues λ_i measure the variation along the axis defined by their corresponding eigenvector. Intuitively, the eigenvectors that correspond to the largest eigenvalues span the tangent plane at point \mathbf{p}_i , whereas the eigenvector corresponding to the smallest eigenvalue can be used to approximate the surface normal \mathbf{n}_i . Thus, given that the smallest eigenvalue λ_0 measures the deviation of point \mathbf{p}_i from the surface, it can be used as an estimate of point curvature. As shown in [42], we may define:

$$\kappa(\mathbf{p}_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \quad \lambda_0 < \lambda_1 < \lambda_2 \quad (2)$$

as the local curvature estimate at point \mathbf{p}_i which is ideal for tasks such as point simplification. Using the previously estimated curvature at point \mathbf{p}_i we can estimate the mean curvature as the Gaussian weighted average of the curvatures around the neighborhood \mathcal{N}_i :

$$\bar{\kappa}(\mathbf{p}_i) = \frac{\sum_{j \in \mathcal{N}_i} \kappa(\mathbf{p}_j) \exp(-\|\mathbf{p}_j - \mathbf{p}_i\|^2/h)}{\sum_{j \in \mathcal{N}_i} \exp(-\|\mathbf{p}_j - \mathbf{p}_i\|^2/h)} \quad (3)$$

where h is a constant defining the neighborhood radius. Finally, we can define an estimation of the roughness as the difference between curvature and the mean curvature at point \mathbf{p}_i as: $\mathcal{R}(\mathbf{p}_i) = |\kappa(\mathbf{p}_i) - \bar{\kappa}(\mathbf{p}_i)|$

3.2 Model

The main building block of the proposed architecture is a graph neural network that receives at its input a point cloud (or a mesh) \mathcal{P}_1 with N points \mathbf{p}_i and outputs a simplified version \mathcal{P}_2 with M points, $M \ll N$. It is important to note that the simplified point cloud \mathcal{P}_2 does not need to be a subset of the original point set \mathcal{P}_1 . The proposed model is composed by three modules: the *Projection Network*, the *Point Selector* and the *Refinement Network*. Figure 2 illustrates the architecture of the proposed method.

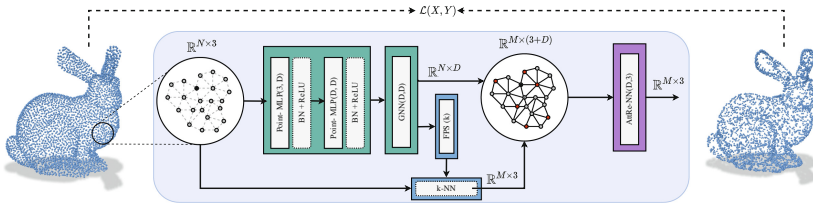


Fig. 2. Overview of the proposed method. Initially a point cloud (or a mesh) is passed through a projection network (green) and embedded to a higher dimensional latent space. FPS is used to select points from the set of latent representations (blue) that can be conceived as cluster centers of the input. Finally, a k-NN graph is constructed between the cluster centers and the input points that is used to modify their positions using the refinement layer (purple). (Color figure online)

Projection Network and Point Selector: In this study, we formulate sampling as a clustering problem. In particular, we aim to cluster points that share similar perceptual and structural features and express the simplified point cloud using the cluster centres. To do so, we designed a *Projector Network* that embeds (x, y, z) coordinates to a high dimensional space, where points with similar features are close in the latent space. In other words, instead of directly sampling from the Euclidean input space, we aim to sample points embedded to

a high dimensional latent space that captures the perceptual characteristics of the input. Clustering the latent space will create clusters with latent vectors of points that share similar perceptual characteristics.

Based on the observations that Farthest Point Sampling (FPS) provides a simple and intuitive technique to select points covering the point cloud structure [48], we built a sampling module on top of this sampling strategy, where points are sampled from a high dimensional space instead of the input xyz -space. Although any clustering algorithm could be adequate, we utilized FPS module since it covers sufficiently the input space without solving any optimization problem. Intuitively, using this formulation we are allowed to interfere the selection process and transform it to a learnable module that is trained to select point embeddings that cover the perceptual latent space, enabling the preservation of both structural and perceptual salient features of the input.

Projector Network comprises of a multi-layer perceptron (MLP) applied to each point independently, followed by a Graph Neural Network (GNN) that captures the local geometric properties around each point. The update rule of the GNN layer is the following:

$$\mathbf{f}'_i = \mathbf{W}_c \mathbf{f}_i + \frac{1}{\mathcal{N}_i} \sum_{j \in \mathcal{N}_i} \mathbf{W}_n \mathbf{f}_j \quad (4)$$

where \mathbf{f}_i denotes the output of the shared point-wise MLP for point \mathbf{p}_i and $\mathbf{W}_c, \mathbf{W}_n$ represent learnable projection matrices. The connectivity between points can be given either by the mesh triangulation or by a k-nn query in the input space (we used a small neighborhood of $k = 7$ as in [44]). Following the Projector Network, *Point Selector* utilizes FPS to select points, i.e. cluster centers, based on their latent representations, in order to cover the latent space. Given the cluster centers selected by FPS, a k-nn graph is constructed that connects the center points with their k-nearest neighbours, based on their 3D positions.

Attention-Based Refinement Layer: Cluster centers, their neighboring point positions along with their respective embeddings from the projection networks are passed to the attention-based refinement layer (AttRef) that modifies the positions of the cluster centers. This layer can be considered as a rectification step that given a neighborhood and its corresponding latent features, displaces the cluster center points in order to minimize the visual perceptual error. Given that the latent embeddings of each point can be considered as its local descriptor, the refinement layer generates the new positions based on the vertex displacements along with the neighborhood local descriptors. The final positions of the points as predicted by *AttRef* are defined as follows:

$$\mathbf{p}'_{c_i} = \mathbf{p}_{c_i} + \gamma \left(\frac{1}{\mathcal{N}_{c_i}} \sum_{j \in \mathcal{N}_{c_i}} \alpha_{ij} \phi([\mathbf{f}_j \| \mathbf{p}_j - \mathbf{p}_{c_i}]) \right) \quad (5)$$

where γ and ϕ are MLPs, \mathcal{N}_{c_i} the k-nearest neighbors of point \mathbf{p}_{c_i} (we used $k = 15$), \mathbf{f}_j the latent features of point \mathbf{p}_j and α_{ij} the attention coefficients

between center \mathbf{p}_{c_i} and point \mathbf{p}_j . The attention coefficients α_{ij} are computed using scaled dot-product [56], i.e. $\alpha_{ij} = \text{softmax}\left(\frac{\theta_q(\mathbf{f}_i)^T \theta_k(\mathbf{f}_j)}{\sqrt{d}}\right)$, where θ_q, θ_k are linear transformations mapping features \mathbf{f} to a d -dimensional space.

3.3 Loss Function

The selection of the loss function to be optimized is crucial for the task of simplification since we seek for a balance between the preservation of the object’s structure and its underlying salient features. A major barrier of most common distance metrics is the uniform weighting of points that can not reflect the perceptual differences between objects. As shown in many studies [20, 35, 58] the commonly used Chamfer distance (CD) between two point sets $\mathcal{P}_1, \mathcal{P}_2$ defined as:

$$d_{\mathcal{P}_1, \mathcal{P}_2} = \sum_{x \in \mathcal{P}_1} \min_{y \in \mathcal{P}_2} \|x - y\|^2 + \sum_{y \in \mathcal{P}_2} \min_{x \in \mathcal{P}_1} \|x - y\|^2 \quad (6)$$

can only describe the overall surface structure similarity between the two sets without taking into account the high frequency details of each point cloud. Figure 1 illustrates an example of such case. Similarly, the point to surface distance between points of a set \mathcal{P} and a surface \mathcal{M} as well as the Hausdorff distance can not preserve salient points of the object rather than its global appearance. To train our model task it is essential to devise a loss function that preserves both the salient features along with the structure of the point cloud.

Adaptive Chamfer Distance: As can be easily observed, the first term of Eq. (6) measures the preservation of the overall structure of \mathcal{P}_1 by \mathcal{P}_2 , in a uniform way. To break the uniformity of the first term of CD we introduce a weighting factor w_x in Eq. 7 that penalizes the distances between the two sets at the points with high salient features ensuring that they will be preserved at the simplified point cloud. We define the modified adaptive Chamfer distance as:

$$d_{\mathcal{P}_1, \mathcal{P}_2}^{Adapt} = \sum_{x \in \mathcal{P}_1} w_{\bar{\mathcal{K}}(x)} \min_{y \in \mathcal{P}_2} \|x - y\|^2 + \sum_{y \in \mathcal{P}_2} \min_{x \in \mathcal{P}_1} \|x - y\|^2 \quad (7)$$

where \mathcal{P}_1 denotes the initial point cloud, \mathcal{P}_2 the simplified one, and $w_{\bar{\mathcal{K}}(x)}$ a weighting factor proportional to the mean curvature $\bar{\mathcal{K}}$ at point x ¹. Since we only aim to retain salient points of \mathcal{P}_1 , we avoid applying a similar weighting factor to the second term of Eq. (6) to prevent the optimization process from getting trapped at local minima.

Curvature Preservation: Additional to the adaptive CD, we make use of a loss term to reinforce the selection of high curvature points of the input. To quantify the preservation of salient features of the input we introduce an error to measure the average point-wise curvature distance between the two point clouds:

¹ We define the weights w_x using the sigmoid of the normalized curvatures divided by a temperature scalar $\tau = 10$ to amplify high curvature values.

$$\mathcal{E}_c = \left(\frac{1}{|\mathcal{P}_1|} \sum_{x \in \mathcal{P}_1} \|\bar{\mathcal{K}}_1(x) - \bar{\mathcal{K}}_2(\text{NN}(x, \mathcal{P}_2))\|^2 \right)^{1/2} \quad (8)$$

where $\text{NN}(x, \mathcal{P}_2)$ denotes the nearest neighbour of x in set \mathcal{P}_2 , and $\bar{\mathcal{K}}(\cdot)$ the mean curvature. We refer to this error as Curvature Error (CE).

Overall Objective: We used a combination of the two aforementioned losses as the total objective to be minimized:

$$\mathcal{L}(\mathcal{P}_1, \mathcal{P}_2) = d_{\mathcal{P}_1, \mathcal{P}_2}^{\text{Adapt}} + \lambda \mathcal{E}_c, \quad (9)$$

where λ is used as a scaling factor set to 0.1. The first term ensures that the selected points cover the surface of the input, while the latter enforces the selection of high curvature points.

4 Evaluation Criteria

To assess the performance of the simplified models generated by our method in terms of visual perception we define several metrics that measure the similarity between the two point cloud models.

Roughness Preservation: Roughness describes the deviation of a point from the surface defined by its neighbours and has been identified as a salient feature in many visual perception studies [33, 57]. Similar to the curvature preservation loss, we calculate the roughness preservation error by substituting the curvature values with roughness in Eq. (8). We refer to this error as RE.

Point Cloud Structural Distortion Measure: Additionally to curvature and roughness preservation metrics, we also calculate the structural similarity score between the two point clouds that has been shown to highly correlate with human perception [31]. In particular, the point cloud Structural Distortion Measure (SDM) can be defined as:

$$D(\mathcal{P}_1, \mathcal{P}_2) = \frac{\alpha \mathcal{L}(p_i, \hat{p}_i) + \beta \mathcal{C}(p_i, \hat{p}_i) + \gamma \mathcal{S}(p_i, \hat{p}_i)}{\alpha + \beta + \gamma} \quad (10)$$

$$\mathcal{L}(p_i, \hat{p}_i) = \frac{\|\bar{\mathcal{K}}_1(p_i) - \bar{\mathcal{K}}_2(\hat{p}_i)\|}{\max(\bar{\mathcal{K}}_1(p_i), \bar{\mathcal{K}}_2(\hat{p}_i))}, \quad \mathcal{C}(p_i, \hat{p}_i) = \frac{\|\sigma_{\bar{\mathcal{K}}_1}(p_i) - \sigma_{\bar{\mathcal{K}}_2}(\hat{p}_i)\|}{\max(\bar{\mathcal{K}}_1(p_i), \bar{\mathcal{K}}_2(\hat{p}_i))} \quad (11)$$

$$\mathcal{S}(p_i, \hat{p}_i) = \frac{\|\sigma_{\bar{\mathcal{K}}_1}(p_i) \sigma_{\bar{\mathcal{K}}_2}(\hat{p}_i) - \sigma_{\bar{\mathcal{K}}_{12}}(p_i, \hat{p}_i)^2\|}{\sigma_{\bar{\mathcal{K}}_1}(p_i) \sigma_{\bar{\mathcal{K}}_2}(\hat{p}_i)} \quad (12)$$

where \mathcal{K}_1 , \mathcal{K}_2 , $\sigma_{\bar{\mathcal{K}}_1}$, $\sigma_{\bar{\mathcal{K}}_2}$, $\sigma_{\bar{\mathcal{K}}_{12}}(p_i, \hat{p}_i)$ are the mean, the gaussian-weighted standard deviation and the covariance of the curvatures for point p_i in \mathcal{P}_1 and its corresponding point \hat{p}_i in \mathcal{P}_2 , respectively. We establish the correspondence between the two point clouds using the 1-nearest neighbor for each point. The global similarity score is obtained using *Minkowski pooling* as suggested in [28].

Normals Consistency: Point normals are highly related to visual appearance as indicators of sharp and smooth areas. We measure the consistency of normals’ orientations between the two models using the cosine similarity loss:

$$\mathcal{E}_n = \frac{1}{|\mathcal{P}_1|} \sum_{\substack{x \in \mathcal{P}_1 \\ y \in NN(x, \mathcal{P}_2)}} 1 - \frac{\mathbf{n}_x \cdot \mathbf{n}_y}{\|\mathbf{n}_x\| \|\mathbf{n}_y\|} + \frac{1}{|\mathcal{P}_2|} \sum_{\substack{y \in \mathcal{P}_2 \\ x \in NN(y, \mathcal{P}_1)}} 1 - \frac{\mathbf{n}_x \cdot \mathbf{n}_y}{\|\mathbf{n}_x\| \|\mathbf{n}_y\|} \quad (13)$$

where \mathbf{n}_x denotes the normal at point x and $NN(x, \mathcal{P}_2)$ the nearest neighbour of x in set \mathcal{P}_2 , calculated as described in Sect. 3.1.

5 Experiments

In this section we extensively evaluate the proposed method with both quantitative and qualitative experiments.

Baselines. We compare our approach against several sampling and simplification methods including: uniform sampling (random), FPS, PointASNL adaptive sampling method [61], quadric error metric (QEM) simplification [15], spectral mesh simplification [37], feature preserving point cloud simplification [49] along with a top curvature points sampling (TCP) where the top-k curvature points are selected from the input point cloud. We failed to compare with recent simplification methods [34, 41] that rely on the eigendecomposition of the Laplacian matrix, since they entail an overwhelmingly large processing run-time and memory consumption (≈ 15 min for a mesh with ≈ 15 K points).

Datasets. We evaluated the proposed method using several publicly available 3D datasets, with different characteristics. The simplification benchmark TOSCA [6] dataset comprises 80 synthetic high-resolution meshes with 9 different deformable objects. It is an excellent candidate to assess feature-preserving simplification, since most of its meshes are non-smooth consisting of high curvature regions. Additionally, we used the popular ModelNet10 dataset [60] and the fixed topology high-resolution MeIn3D face dataset [5]. All datasets used were randomly split in 80%–20% train-test sets, taking care that none of the identities/shapes used for training are present in the respective test set.

Evaluation. We quantitatively evaluate the quality of the simplified point clouds in three folds. Primarily, we measure the low-level structural and perceptual measures, as described in Sect. 4, in Sect. 5.1. We additionally use an pre-trained objective classifier to measure the preservation of high-level semantics along with a user-study that aims to access the conceivable human perceptual similarity between the input and the simplified models. Furthermore, we access the importance of the major components of the proposed method using ablations studies, as well as the performance of the proposed method under noisy conditions and real-world scans. Due to space limitations several of the aforementioned experiments are reported on the supplementary material along with additional qualitative results.

5.1 Point Cloud Simplification

In this section, we showcase the simplification performance of the proposed method. For each dataset, we report both structural (i.e. CD, NC) as well as perceptual metrics (i.e. CE, RE, SDM) for the proposed and the baseline methods. Table 1 indicates the superiority of the proposed method to maintain the perceptual features of the input (i.e. low SDM) without sacrificing the overall structure (i.e. low CD) of the shape at three indicative simplification ratios. Results for larger simplification ratios can be found in the supplementary material. In contrast to TPC method where the selection of high curvature points leads to an increased Chamfer distance, the proposed method achieves a fair balance between structure (CD and NC) and saliency (SDM and RE). Additionally, the proposed method exhibits lower perceptual error (SDM) compared to FPS and PointASNL [61] methods that sample points directly from the xyz-space. This may be also observed in Fig. 3 where sampling from the perceptual latent space can effectively induce the preservation of the details of the input cloud.

Table 1. Simplification performance tested on TOSCA (top), ModelNet10 (middle) and MeIn3d (bottom) datasets. Best approaches are highlighted in **bold** and second best in **red**. We refer to the dataset used for training as “Proposed-Dataset”

Method	TOSCA											
	$N_s/N_{org} = 0.2$				$N_s/N_{org} = 0.1$				$N_s/N_{org} = 0.05$			
	CD	NC	RE($\times 10^{-4}$)	SDM($\times 10^{-4}$)	CD	NC	RE($\times 10^{-4}$)	SDM($\times 10^{-4}$)	CD	NC	RE($\times 10^{-4}$)	SDM($\times 10^{-4}$)
Random	1.63	0.312	4.45	6.07	3.35	0.342	4.91	10.7	6.68	0.369	5.71	19.2
TCP	51.3	0.625	4.99	9.52	129.4	0.732	6.42	17.8	172.5	0.793	6.20	32.4
FPS	0.81	0.307	4.71	5.13	1.93	0.341	4.82	9.64	3.94	0.321	5.56	18.3
QEM	1.35	0.291	4.01	5.36	2.64	0.310	4.79	10.4	4.77	0.338	5.53	18.4
Liu <i>et al.</i> [37]	2.17	0.358	4.39	5.39	3.12	0.331	4.96	10.4	5.62	0.441	5.96	18.5
Qi <i>et al.</i> [49]	2.49	0.303	4.45	7.37	3.46	0.353	4.51	13.18	6.15	0.372	5.34	23.18
Yan <i>et al.</i> [61]	1.17	0.301	4.27	5.41	2.54	0.321	4.48	9.51	5.14	0.357	5.27	18.1
Proposed-Meln3D	1.14	0.293	4.15	5.64	2.53	0.313	4.47	8.15	5.36	0.364	5.01	17.7
Proposed-ModelNet	1.15	0.310	4.01	5.53	2.51	0.312	4.81	9.72	5.19	0.341	4.99	17.4
Proposed-TOSCA	1.12	0.290	3.91	5.01	2.45	0.307	4.41	7.84	4.93	0.333	4.93	16.5
Method	ModelNet											
	$N_s/N_{org} = 0.2$				$N_s/N_{org} = 0.1$				$N_s/N_{org} = 0.05$			
	CD($\times 10^{-4}$)	NC	RE($\times 10^{-5}$)	SDM($\times 10^{-3}$)	CD($\times 10^{-5}$)	NC	RE($\times 10^{-5}$)	SDM($\times 10^{-3}$)	CD($\times 10^{-4}$)	NC	RE($\times 10^{-5}$)	SDM($\times 10^{-3}$)
Random	8.01	0.568	5.91	2.83	20.4	0.655	6.19	4.92	41.02	0.793	6.57	8.19
TCP	197.3	0.898	7.25	3.87	403.1	0.937	7.84	7.11	611.6	0.952	7.01	12.81
FPS	3.12	0.505	6.05	2.74	7.56	0.641	6.39	4.81	16.01	0.744	6.48	8.38
QEM	3.45	0.513	5.94	3.01	9.45	0.625	6.13	5.19	21.43	0.724	6.25	9.12
Liu <i>et al.</i> [37]	4.21	0.537	5.99	3.05	10.32	0.632	6.75	5.32	21.54	0.792	6.52	9.44
Qi <i>et al.</i> [49]	5.64	0.515	6.03	3.47	10.97	0.654	6.54	5.71	26.37	0.745	6.39	9.17
Yan <i>et al.</i> [61]	6.28	0.514	5.87	2.89	11.08	0.643	6.29	5.04	20.69	0.428	6.37	8.61
Proposed-Meln3D	4.02	0.531	5.93	2.86	29.31	0.610	6.08	4.76	45.12	0.701	6.33	8.02
Proposed-ModelNet	3.32	0.515	5.79	2.68	8.24	0.606	6.06	4.61	17.24	0.696	6.25	7.92
Proposed-TOSCA	4.35	0.523	5.77	2.72	9.42	0.603	5.91	4.64	22.18	0.688	6.04	7.96
Method	MeIn3D											
	$N_s/N_{org} = 0.2$				$N_s/N_{org} = 0.1$				$N_s/N_{org} = 0.05$			
	CD($\times 10^{-4}$)	NC	RE($\times 10^{-5}$)	SDM($\times 10^{-3}$)	CD($\times 10^{-5}$)	NC	RE($\times 10^{-5}$)	SDM($\times 10^{-3}$)	CD($\times 10^{-4}$)	NC	RE($\times 10^{-5}$)	SDM($\times 10^{-3}$)
Random	1.42	0.198	4.15	2.81	3.46	0.313	6.73	5.92	5.52	0.481	7.05	12.4
TCP	158.3	0.801	3.46	3.73	421.1	0.910	6.02	7.38	556.0	0.934	11.87	14.2
FPS	1.12	0.121	3.64	2.96	1.93	0.195	6.29	5.98	3.45	0.484	7.43	11.8
QEM	2.01	0.185	4.53	3.01	2.52	0.198	6.31	5.71	3.65	0.331	8.13	11.3
Liu <i>et al.</i> [37]	2.92	0.215	4.92	3.12	3.14	0.199	6.67	5.92	3.88	0.392	8.46	11.9
Qi <i>et al.</i> [49]	3.12	0.193	4.74	3.45	3.55	0.211	6.51	5.95	4.07	0.405	8.22	12.6
Yan <i>et al.</i> [61]	2.75	0.193	4.70	2.95	3.05	0.205	6.34	5.41	3.76	0.374	9.17	11.4
Proposed-Meln3D	1.24	0.128	3.15	2.30	2.01	0.192	5.69	4.91	3.25	0.305	6.47	10.6
Proposed-ModelNet	1.75	0.189	3.65	2.45	3.23	0.196	5.73	5.10	4.02	0.369	7.02	10.9
Proposed-TOSCA	1.54	0.168	3.29	2.41	2.32	0.194	5.98	5.06	3.82	0.342	6.49	10.8

Significantly, it can be observed in Table 1 and Fig. 4, the proposed method outperforms recent methods [37, 49, 61] under all metrics, and QEM in terms

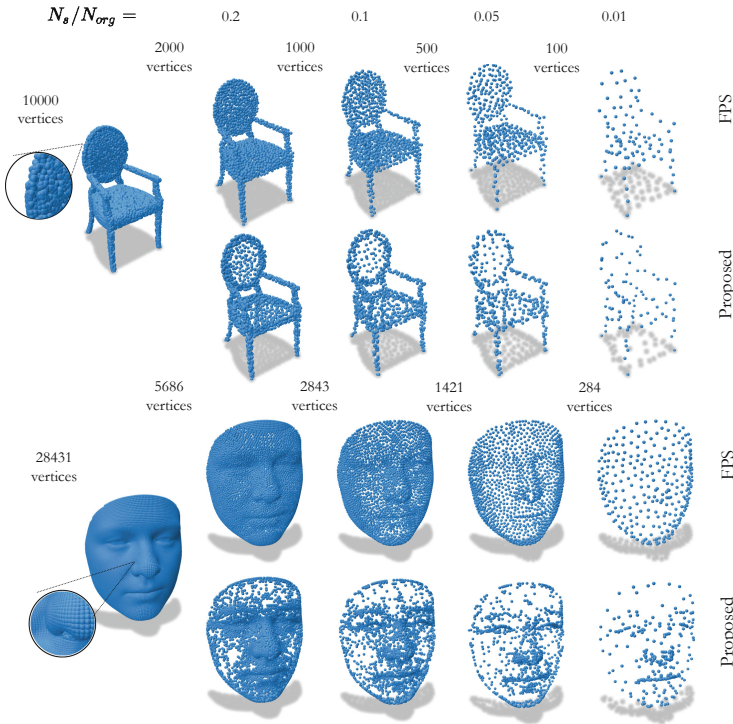


Fig. 3. Qualitative comparison between FPS (top row) and the proposed (bottom row) methods, at different simplification ratios. Differences between the two methods can be found at coarse and smooth areas, where the proposed model favours the preservation of high-frequency details of the input point cloud.

of perceptual error. Figure 5, demonstrates the superiority of our method to remarkably retain the salient points of the input using just 1% of the input is retained, in contrast to QEM that performs poorly at coarse areas of high curvature. The selection of salient points of the input is demonstrated in Fig. 3, where the proposed method favours point selection around the chair’s arm, the face’s eyes and nose in contrast to points at smooth areas, such as the forehead. Intuitively, smooth areas require only a few points to describe their associated planes compared to coarse areas that demand many points in order to preserve their curvature.

We also experimented with a cross-dataset generalization scenario where different datasets were used for training and testing the model (bottom rows in Table 1). Interestingly, it is observed that the proposed model can generalize well to out-of-distribution shapes and topologies indicating that it can be applied directly to any point cloud without fine-tuning, especially when trained with TOSCA or ModelNet dataset. We argue that this is due to the diversity of shapes and topologies as well as the presence of many rough regions at the training sets of these datasets that enforce the model to favour salient features.

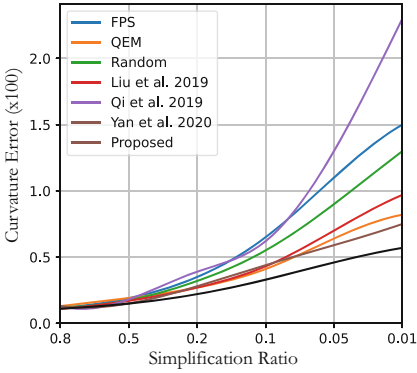


Fig. 4. Curvature preservation error comparison for the TOSCA dataset at different simplification ratios. Curvature error scales linearly for the proposed method.

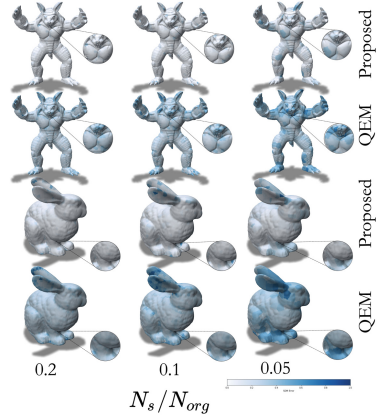


Fig. 5. Colored-coded curvature error comparison between QEM and the proposed method. Blue color corresponds to larger error. (Color figure online)

5.2 Computational Time

Inevitably, in addition to salient point preservation, a proper point cloud simplification method should confront real-time executions. Although time complexity is beyond the scope of this study, we assessed the time required for simplifying 80 high-resolution meshes from the TOSCA dataset. Since FPS, Uniform and TCP baseline methods do not require any significant computations, we compare the proposed method with the popular QEM approach using a highly optimized version from the MeshLab framework [7] and the official implementations of [49] and [37]. It is important to note that the code of the proposed method could be further optimized, using parallel programming. In particular, 80% of the computational time of the proposed method is acquired by k-NN search that could be further optimized whereas FPS takes 17 % and the rest 3% of the runtime is spent on the learnable modules. Figure 7 demonstrates that the required mean runtime of the proposed method decreases drastically across all experiments, as the desired simplification increases, requiring just a few seconds to simplify the input to 1% of its original size. In contrast, the methods of [49] and [37] require approximately a minute to simplify a single mesh which makes them impractical.

5.3 Mesh Simplification

As described in Sect. 2, mesh simplification is a long studied problem that has been tackled only by greedy algorithms. In this section, we will attempt to propose an alternative method that circumvents the greedy nature of simplification using the simplification technique proposed in Sect. 3. The proposed method,

without the need of any particular modification, can be easily extended for the task of mesh simplification when combined with a triangulation algorithm to transform the simplified vertices into a mesh structure. The process unfolds in two steps. Initially, mesh vertices are simplified by treating them as a point cloud but, instead of using a k-nn, the mesh adjacency matrix is utilized in order to determine point connectivity. In a second step, the remaining vertices are re-triangulated using an off-the-self triangulation algorithm such as Ball Pivoting [4]. Other triangulation algorithms, such as Delaunay, alpha shapes or Voronoi diagrams, could be also used but we observed that Ball Pivoting algorithm produces better results for small point clouds. Figure 6 shows visual results for the extension of the proposed simplification method to triangular meshes (for various simplification ratios).

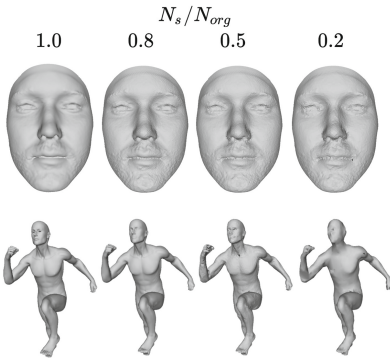


Fig. 6. Simplified meshes using the proposed method followed by Ball Pivoting Algorithm.

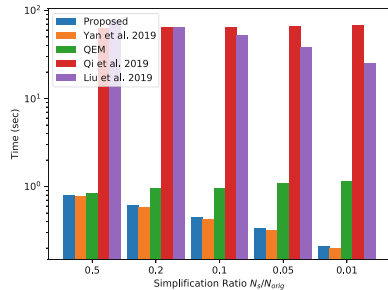


Fig. 7. Average time of simplification for the proposed and the baseline methods.

5.4 User-Study

To quantify the ability of the proposed method to select points that correlate with human perception we performed a user study, using the paired comparison protocol contrasting the proposed and one baseline method. In total, 50 participants were specifically asked to evaluate 18 point clouds and select one of the two simplified point clouds that mostly preserves the perceptual details of the reference one in terms of the overall shape and identity similarity. In average, as shown in Table 2, users selected 14 out of the 18 point clouds produced by the proposed method, as the ones preserving most of the visual features.

Table 2. User studies results of different methods. We average user preference scores (higher is better, results in %). Best results in bold.

Method	User choice
Ours vs QEM	73/27
Ours vs Liu <i>et al.</i> [37]	78/22
Our vs FPS	71/29
Average	74/26

5.5 Simplification Under Noise Conditions

To both quantitatively and qualitatively evaluate the performance of the proposed method in the presence of noise we fed the pretrained model with point clouds distorted with Gaussian noise ($\sigma = 1$). Quantitative experimental results are summarized in Table 3, with the proposed method to exhibit the best performance for almost all metrics (NC, RE, SDM) across all simplification ratios, and even the best second for structure preservation (CD). Such findings reveal the anti-noising capabilities of latent space sampling that, compared to xyz-space sampling, is less affected by the outlier noisy points. For qualitative comparisons along with results on LiDAR scans we refer the reader to the Supplementary material.

Table 3. Simplification performance tested on TOSCA dataset with addition of Gaussian noise. Best approaches are highlighted in **bold** and second best in **red**.

Method	$N_s/N_{org} = 0.2$				$N_s/N_{org} = 0.1$				$N_s/N_{org} = 0.05$			
	CD	NC	RE($\times 10^{-4}$)	SDM($\times 10^{-4}$)	CD	NC	RE($\times 10^{-4}$)	SDM($\times 10^{-4}$)	CD	NC	RE($\times 10^{-4}$)	SDM($\times 10^{-4}$)
Random	2.71	0.37	6.56	9.20	4.43	0.38	6.74	14.39	7.78	0.39	6.85	23.63
TCP	24.7	0.48	6.30	9.27	37.2	0.49	6.58	14.83	53.5	0.51	6.77	23.45
FPS	2.74	0.34	6.25	9.89	4.28	0.36	6.34	15.86	6.83	0.39	6.81	25.03
QEM	1.92	0.31	6.61	9.14	2.57	0.36	6.81	14.53	3.85	0.37	6.93	23.12
Liu <i>et al.</i> [37]	3.04	0.35	7.14	9.48	3.99	0.39	7.36	16.21	6.74	0.412	7.61	26.35
Qi <i>et al.</i> [49]	3.21	0.33	7.22	11.14	4.31	0.35	7.54	18.34	7.13	0.39	7.81	27.52
Yan <i>et al.</i> [61]	2.95	0.35	6.54	10.04	4.30	0.39	7.30	16.01	7.11	0.41	7.12	25.14
Proposed	2.50	0.33	6.13	8.81	3.96	0.35	6.24	14.42	6.46	0.37	6.34	22.31

6 Conclusion

Our work emphasises on the proposal of a learnable, neural-based simplification approach to overcome the inefficiencies of traditional greedy simplification methods. In this study we presented the first, to the best of our knowledge, learnable point cloud simplification method that aims at preserving salient features while at the same time retaining the global structural appearance of the input 3D object. Using three learnable modules we attain to simplify large-scale point clouds in real-time, addressing the literature limitations regarding computational complexity. As shown in an extensive series of both quantitative and qualitative experiments the proposed method not only outperforms its counterparts under most perceptual criteria but also exhibits zero-shot capabilities. Regarding future work, we plan to adapt the proposed method to mesh structures using a more sophisticated triangulation process. In particular, instead of using off-the-shelf triangulation algorithms on top of the point cloud simplification model, we aim to extend the proposed method to predict the triangulation of the simplified model utilizing the priors of the input mesh.

Acknowledgements.. Dr. Stefanos Zafeiriou acknowledges support from EPSRC fellowship Deform (EP/S010203/1).

References

1. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Point set surfaces. In: Proceedings Visualization, VIS 2001, pp. 21–29. IEEE (2001)
2. An, G., Watanabe, T., Kakimoto, M.: Mesh simplification using hybrid saliency. In: 2016 International Conference on Cyberworlds (CW), pp. 231–234. IEEE (2016)
3. Bahirat, K., Lai, C., McMahan, R.P., Prabhakaran, B.: Designing and evaluating a mesh simplification algorithm for virtual reality. *ACM Trans. Multimedia Comput. Commun. Appl.* **14**(3s) (2018)
4. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Visual Comput. Graphics* **5**(4), 349–359 (1999)
5. Booth, J., Roussos, A., Ponniah, A., Dunaway, D., Zafeiriou, S.: Large scale 3D morphable models. *Int. J. Comput. Vision* **126**(2), 233–254 (2018)
6. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: *Numerical Geometry of Non-rigid Shapes*. Springer, New York (2008). <https://doi.org/10.1007/978-0-387-73301-2>
7. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: Meshlab: an open-source mesh processing tool. In: Eurographics Italian Chapter Conference, Salerno, Italy, vol. 2008, pp. 129–136 (2008)
8. Cignoni, P., Montani, C., Scopigno, R.: A comparison of mesh simplification algorithms. *Comput. Graph.* **22**(1), 37–54 (1998)
9. Corsini, M., Larabi, M.C., Lavoué, G., Petřík, O., Váša, L., Wang, K.: Perceptual metrics for static and dynamic triangle meshes. In: Computer Graphics Forum, vol. 32, pp. 101–125. Wiley Online Library (2013)
10. Dong, L., Fang, Y., Lin, W., Seah, H.S.: Perceptual quality assessment for 3D triangle mesh based on curvature. *IEEE Trans. Multimedia* **17**(12), 2174–2184 (2015)
11. Dovrat, O., Lang, I., Avidan, S.: Learning to sample. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2760–2769 (2019)
12. Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.Y.: The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **6**(9), 1305–1315 (1997)
13. Feng, X., Wan, W., Da Xu, R.Y., Chen, H., Li, P., Sánchez, J.A.: A perceptual quality metric for 3D triangle meshes based on spatial pooling. *Front. Comp. Sci.* **12**(4), 798–812 (2018)
14. Galantucci, L.M., Percoco, G.: A multilevel approach to edge detection in tessellated point clouds. *CIRP Ann.* **54**(1), 127–130 (2005)
15. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 209–216 (1997)
16. Han, H., Han, X., Sun, F., Huang, C.: Point cloud simplification with preserved edge based on normal vector. *Optik Int. J. Light Electron Opt.* **126**(19), 2157–2162 (2015)
17. Hanocka, R., Hertz, A., Fish, N., Giryas, R., Fleishman, S., Cohen-Or, D.: Meshcnn: a network with an edge. *ACM Trans. Graph. (TOG)* **38**(4), 90:1–90:12 (2019)
18. Hoppe, H.: Progressive meshes. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 99–108 (1996)
19. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, pp. 71–78 (1992)

20. Jin, J., Patil, A.G., Xiong, Z., Zhang, H.: DR-KFS: a differentiable visual similarity metric for 3D shape reconstruction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 295–311. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_18
21. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 279–286 (2000)
22. Kim, S., Jeong, W., Kim, C.: Lod generation with discrete curvature error metric. In: Proceedings of Korea Israel Bi-National Conference, pp. 97–104. Citeseer (1999)
23. Kim, S.J., Kim, C.H., Levin, D.: Surface simplification using a discrete curvature norm. *Comput. Graph.* **26**(5), 657–663 (2002)
24. Kim, S.J., Kim, S.K., Kim, C.H.: Discrete differential error metric for surface simplification. In: Proceedings of 10th Pacific Conference on Computer Graphics and Applications, pp. 276–283. IEEE (2002)
25. Lang, I., Manor, A., Avidan, S.: Samplenet: differentiable point cloud sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7578–7588 (2020)
26. Lattas, A., et al.: Avatarme: realistically renderable 3D facial reconstruction “in-the-wild”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 760–769 (2020)
27. Lavoué, G.: A local roughness measure for 3D meshes and its application to visual masking. *ACM Trans. Appl. Percept.* (TAP) **5**(4), 1–23 (2009)
28. Lavoué, G.: A multiscale metric for 3D mesh visual quality assessment. In: *Computer Graphics Forum*, vol. 30, pp. 1427–1437. Wiley Online Library (2011)
29. Lavoué, G., Cheng, I., Basu, A.: Perceptual quality metrics for 3D meshes: towards an optimal multi-attribute computational model. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 3271–3276. IEEE (2013)
30. Lavoué, G., Corsini, M.: A comparison of perceptually-based metrics for objective evaluation of geometry processing. *IEEE Trans. Multimedia* **12**(7), 636–649 (2010)
31. Lavoué, G., Gelasca, E.D., Dupont, F., Baskurt, A., Ebrahimi, T.: Perceptually driven 3D distance metrics with application to watermarking. In: *Applications of Digital Image Processing XXIX*, vol. 6312, p. 63120L. International Society for Optics and Photonics (2006)
32. Leal, N., Leal, E., German, S.T.: A linear programming approach for 3D point cloud simplification. *IAENG Int. J. Comput. Sci.* **44**(1), 60–67 (2017)
33. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. In: *ACM SIGGRAPH 2005 Papers*, pp. 659–666 (2005)
34. Lescoat, T., Liu, H.T.D., Thiery, J.M., Jacobson, A., Boubekour, T., Ovsjanikov, M.: Spectral mesh simplification. *Comput. Graph. Forum* **39**(2), 315–324 (2020)
35. Li, C.L., Simon, T., Saragih, J., Póczos, B., Sheikh, Y.: LBS autoencoder: self-supervised fitting of articulated meshes to point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11967–11976 (2019)
36. Li, W., Chen, Y., Wang, Z., Zhao, W., Chen, L.: An improved decimation of triangle meshes based on curvature. In: Miao, D., Pedrycz, W., Ślęzak, D., Peters, G., Hu, Q., Wang, R. (eds.) *RSKT 2014*. LNCS (LNAI), vol. 8818, pp. 260–271. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11740-9_25
37. Liu, H.T.D., Jacobson, A., Ovsjanikov, M.: Spectral coarsening of geometric operators. *ACM Trans. Graph.* **38**(4) (2019)
38. Liu, X.L., Liu, Z.Y., Gao, P.D., Peng, X.: Edge collapse simplification based on sharp degree. *Ruan Jian Xue Bao(J. Softw.)* **16**(5), 669–675 (2005)

39. Lu, T., Chao, T.H.: A single-camera system captures high-resolution 3D images in one shot. *SPIE Newsroom* (2006)
40. Luan, W., Liu, C., Pang, H.: Skeleton-bridged mesh saliency and mesh simplification. In: 2020 4th International Conference on Computer Science and Artificial Intelligence, pp. 278–283 (2020)
41. Nasikun, A., Brandt, C., Hildebrandt, K.: Fast approximation of laplace-beltrami eigenproblems. In: *Computer Graphics Forum*, vol. 37, pp. 121–134. Wiley Online Library (2018)
42. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: *IEEE Visualization, VIS 2002*, pp. 163–170. IEEE (2002)
43. Pavlidis, G., Koutsoudis, A., Arnaoutoglou, F., Tsioukas, V., Chamzas, C.: Methods for 3D digitization of cultural heritage. *J. Cult. Herit.* **8**(1), 93–98 (2007)
44. Potamias, R.A., Neofytou, A., Bintsi, K.M., Zafeiriou, S.: Graphwalks: efficient shape agnostic geodesic shortest path estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2968–2977 (2022)
45. Potamias, R.A., Ploumpis, S., Zafeiriou, S.: Neural mesh simplification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18583–18592 (2022)
46. Potamias, R.A., Zheng, J., Ploumpis, S., Bouritsas, G., Ververas, E., Zafeiriou, S.: Learning to generate customized dynamic 3D facial expressions. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020. LNCS*, vol. 12374, pp. 278–294. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58526-6_17
47. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: deep learning on point sets for 3D classification and segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660 (2017)
48. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: Guyon, I., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc. (2017)
49. Qi, J., Hu, W., Guo, Z.: Feature preserving and uniformity-controllable point cloud simplification on graph. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 284–289. IEEE (2019)
50. Ronfard, R., Rossignac, J.: Full-range approximation of triangulated polyhedra. In: *Computer Graphics Forum*, vol. 15, pp. 67–76. Wiley Online Library (1996)
51. Rossignac, J., Borrel, P.: Multi-resolution 3D approximations for rendering complex scenes. In: Falcidieno, B., Kunii, T.L. (eds.) *Modeling in Computer Graphics*, pp. 455–465. Springer, Heidelberg (1993). https://doi.org/10.1007/978-3-642-78114-8_29
52. Schroeder, W.J., Zarge, J.A., Lorensen, W.E.: Decimation of triangle meshes. In: *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 65–70 (1992)
53. Shi, B.Q., Liang, J., Liu, Q.: Adaptive simplification of point cloud using k-means clustering. *Comput. Aided Des.* **43**(8), 910–922 (2011)
54. Torkhani, F., Wang, K., Chassery, J.M.: A curvature tensor distance for mesh visual quality assessment. In: *ICCV* (2012)
55. Váša, L., Rus, J.: Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes. In: *Computer Graphics Forum*, vol. 31, pp. 1715–1724. Wiley Online Library (2012)
56. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (2017)

57. Wang, Y., Zheng, J., Wang, H.: Fast mesh simplification method for three-dimensional geometric models with feature-preserving efficiency. *Sci. Program.* **2019** (2019)
58. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2mesh++: multi-view 3D mesh generation via deformation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1042–1051 (2019)
59. Wu, J., Shen, X., Zhu, W., Liu, L.: Mesh saliency with global rarity. *Graph. Models* **75**(5), 255–264 (2013)
60. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D shapenets: a deep representation for volumetric shapes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920 (2015)
61. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: robust point clouds processing using nonlocal neural networks with adaptive sampling. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5589–5598 (2020)
62. Yao, L., Huang, S., Xu, H., Li, P.: Quadratic error metric mesh simplification algorithm based on discrete curvature. *Math. Probl. Eng.* **2015** (2015)
63. Yildiz, Z.C., Oztireli, A.C., Capin, T.: A machine learning framework for full-reference 3D shape quality assessment. *Vis. Comput.* **36**(1), 127–139 (2020)
64. Zhang, K., Qiao, S., Wang, X., Yang, Y., Zhang, Y.: Feature-preserved point cloud simplification based on natural quadric shape models. *Appl. Sci.* **9**(10), 2130 (2019)