ASimp: Automatic High-Poly 3D Mesh Simplification for Preprocessing Based on QoE

Lehao Lin[1], Hong Kang[1], Yuqi Shi[1], Haihan Duan[2], Abdulmotaleb El Saddik[3], Wei Cai[4,*]
[1]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China
[2]Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen 518172, China
[3]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa K1N 6N5, Canada
[4]School of Engineering and Technology, University of Washington, Tacoma, WA 98402, United States
{lehaolin, hongkang1, yuqishi}@link.cuhk.edu.cn, duanhaihan@smbu.edu.cn, elsaddik@uottawa.ca, weicaics@uw.edu

2025 IEEE International Conference on Multimedia and Expo (ICME) | 979-8-3315-9495-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICME59968.2025.11209908

***Abstract*—Mesh simplification of 3D models can accelerate rendering, reduce storage space, and improve performance. However, for high-poly 3D models, there are ongoing concerns about potentially compromising the Quality of Experience (QoE), the need to set simplification ratios or parameters, and the time-consuming nature of the simplification process. To address these issues, we proposed a new mesh simplification for the preprocessing step. Based on the Quadratic Error Metric (QEM) simplification algorithm, we conducted human-centered 3D model comparison experiments to determine the optimal simplification ratio for high-poly 3D models in full body shots. From experimental data, we proposed and implemented ASimp, an automatic 3D mesh simplification scheme. In evaluation experiments, ASimp demonstrated rapid preprocessing speeds while ensuring QoE and the effectiveness of its simplification products. We hope that ASimp will contribute to the optimization of 3D models and find applications in fields such as cultural heritage, archaeology, visual effects, video games, medicine, metaverse, and beyond.**

***Index Terms*—Mesh Simplification, Human Perception, User Analysis, Quadric Error Metric, Quality of Experience**

I. Introduction

Advancements in computer hardware, software, and algorithms has led to significant demand for very high-poly and high-resolution 3D models across various fields [1] including the metaverse [2]. This is because such models can represent real-world objects with an extremely high level of detail (LOD) by utilizing a large number of polygonal faces and high-resolution textures. Using the high-poly 3D models will exert pressure on network bandwidth and user computer performance, including limited memory and restricted graphical rendering capabilities. Thus, they need to simplify the models, but excessive simplification can affect the Quality of Service (QoS) [3] and also lead to low Quality of Experience (QoE) [4]. So, an automatic preprocessing tool for 3D models is needed in these areas to balance the pressure and QoE.

Preprocessing is the most common approach when displaying or utilizing high-poly 3D models. It typically includes the following categories: Mesh Cleaning [5], Mesh Simplification, Mesh Smoothing [6], etc. This study will focus on the mesh simplification, aiming to fundamentally simplify the structure.

Currently, mesh simplification can be divided into two types: traditional mathematical simplification algorithms and deep learning-based simplification algorithms. We provide detailed descriptions of these algorithms in Section II. There is no optimal parameter guidance for different 3D model scenarios, and the simplification effects can only be perceived by the human eye. Although there are many evaluation metrics to quantitatively measure the simplification effects, such as chamfer distance (CD), curvature error, normal consistency, normal dissimilarity, F-score, etc. [7], these metrics are related to the degree and ratio of simplification. As the simplification ratio approaches 1, the metrics perform better. Moreover, numerical analysis cannot guarantee that the obtained simplified model still maintains a good QoE; for example, a better curvature error may lead to the loss of sharp corners that should be retained. Additionally, when the number of faces of a 3D model reaches millions, the processing time and memory consumption of simplification become significant challenges. Various learning-based simplification algorithms will also consume more resources.

Motivations. Based on the aforementioned challenges and issues, any solution should meet the following criteria: (1) Automatic: The process should not require manual parameter setting or adjustment. (2) QoE-friendly: The mesh simplification process and outcomes should be informed by user QoE knowledge and withstand user evaluation and assessment. (3) Efficiency: Given the high number of surfaces, the reduction process must be highly efficient.

Approach. Driven by the motivations above, we propose an automatic high-poly 3D mesh preprocessing simplification with human-centered comparative criteria, named ASimp, to balance the QoE and the model size. Firstly, we conducted interaction experiments comparing 3D models in the full body shot simplified at different ratios with the original model. Through approaching the QoE limit, we determined the optimal ratio for each 3D model. Secondly, we utilized the models and their simplification ratios with human feedback as training data to design and implement a neural network named ASimpNet. Thirdly, to address the high-poly problem, we integrated the QEM simplification algorithm [8] and optimized the simplification process to ensure that processing time does not unexpectedly increase due to the sample size or cause crashes due to memory limitations resulting. Finally, we evaluated the overall efficiency of the pipeline, and the performance of the

* Corresponding author: Wei Cai (weicaics@uw.edu).

final products. Through further human perception experiments, we validated that the simplified models maintain a QoE close to that of the original models.

The **contributions** of our work are as follows:

- We obtained data from human-computer interaction experiments to gauge users' perception and 3D quality assessment to explore the potential patterns between the face number and the ratio.
- To the best of our knowledge, we propose the first 3D mesh preprocessing simplification based on human-centered QoE by integrating QEM and Neural Network. We have implemented the entire simplification process, allowing users to achieve model simplification without the need to set any parameters. The relevant data and code are available at https://github.com/LehaoLin/ASimp.
- We conducted experiments to assess the efficiency, and the effectiveness of the proposed method. The results show that ASimp performs well in the preprocessing.

II. Related Work

As the mainstream of 3D mesh simplification algorithms, there are two directions, one is from the traditional mathematics, and the other is using deep learning.

Traditional mathematical mesh simplification methods can be classified to vertex clustering [9], the coplanar facets merging [10], and edge collapse [11]. In edge collapse, the Quadric Error Metric (QEM) algorithm [8] is popular and still one of the most common mesh simplification techniques [12], which can control the number of faces of the simplified product while keeping the general shape unchanged. But these algorithms require users to set the simplification ratio to adjust the final product, which causes trouble when facing a scenario where a large number of 3D models need to be processed. Another approach does not directly manipulate the original mesh but instead performs estimation-based reconstruction on a reduced sampled point cloud. However, the results are often of low accuracy. The work in [13] addresses the challenges associated with large-scale point cloud processing.

As for **the simplification by deep learning**, the core idea is to iteratively learn, extract and simplify the features of the model, such as Pitanuas et al. [14], PointTriNet [15], and PoNQ [16]. But there is a critical point in the practical application, which makes the ratio of 3D model tend to a fixed value. It leads to some high-complexity models unable to achieve a balance between the observable optimization degree and the calculation speed. Deep learning methods may encounter the challenge of excessively long processing time when dealing with high-poly models.

III. Methodology

A. Workflow

The overall workflow is illustrated in Fig. 1. Firstly, we conduct comparative experiments on 3D model experiences through preliminary research to determine the optimal simplification ratio based on user experience. Subsequently, we use the determined simplification ratio for each 3D model to train deep learning models, obtaining neural network models tailored to the optimal simplification ratio for each 3D model. Using this model, we implement the process of simplifying high-poly models. Finally, in the evaluation section, we conduct numerical and user studies to analyze the simplified models and the simplification process.

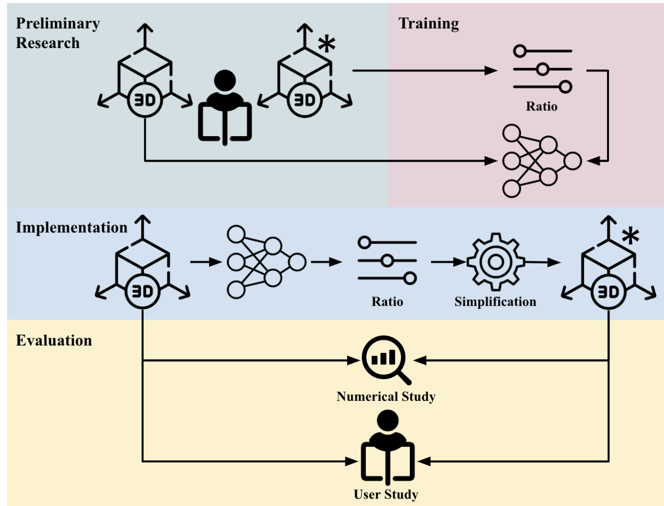


Fig. 1: The workflow overview.

B. Preliminary Research

We design an experiment to investigate the relationship between human perception and the quality of 3D models.

Presettings. Besides the number of faces, there are still other factors that influence human observation of 3D models. Therefore, we try to eliminate other factor impacts. Since the focus of this study is on the number of faces, we use the models without textures and set the color of the models to be uniformly white. Moreover, we set a black background to ensure that the brightness of the background does not affect participants' observations and enables them to accurately identify the model's edges. The observation equipment also has an impact on the observations. According to statistics on the most popular desktop displays currently[1], we use a 24-inch monitor with resolution 1920x1080, also known as 1080p.

Ethics. For this experiment, ethics approval was granted by the school's Institutional Review Board.

Dataset. Since the flourishing development of deep learning in recent years, many 3D model datasets [17] and platforms [18] have emerged. However, the majority of high-resolution 3D datasets have insufficient data volume and lack diversity in types. We selected 80 pieces of 3D models from the ThreeDScans Dataset[2] for experimentation, which meets the criteria of being open-source, sufficient number, and high-poly. Among these models, the minimum number of faces is 101,330, and the maximum is 3,997,268. This dataset utilizes 3D scans of artistic sculptures, featuring extremely high face counts and simplified materials, making it ideal for

[1]https://gs.statcounter.com/screen-resolution-stats/ accessed in Sept, 2024.
[2]https://threedscans.com/ accessed in Sept, 2024.

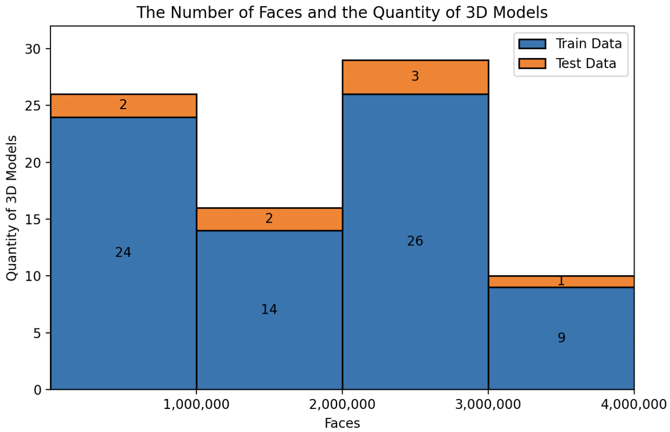


Fig. 2: The face distribution of the selected training set and test set. (Min: 101,330; Max: 3,997,268)

experiments involving face perception. Furthermore, it can be downloaded and utilized without any copyright restrictions.

Fig. 2 illustrates the distribution of the number of faces regarding the 3D models we selected. We divided the dataset into training and test sets with a ratio of 9:1. We selected 8 pieces of 3D models from the dataset as the test set, ensuring that the distribution of the number of faces in the test set reflects that of the overall dataset, and the remaining models constitute the training set. Subsequently, we reduce the number of faces for each model by intervals of 5%, generating 20 models with varying numbers of faces ranging from 5% to 100% of the original faces. These models were used for subsequent comparative experiments.

Experiment. To conduct comparative experiments on 3D models, we developed a web-based model comparison system. Users access the experimental interface by selecting the model name. The left side displays the model with 100% of its original number of faces, while the right side displays the model with varying numbers of faces. Participants can adjust the number of faces of the model on the right side using the slider above. We disabled the translation and scaling functions, allowing participants only to rotate and flip the models during observation. This is because allowing participants to infinitely zoom in on the models and observe details from very close distances would result in all models, regardless of their face count, appearing as distortion or basic triangles. Therefore, such a scenario is not within the scope of our experiment.

Participants observed the 3D model on the right, compared it with the one on the left, and adjusted the number of faces using the slider until they believed there was no significant difference compared to the model on the left. They should adjust the slider to achieve the minimum reduction percentage while ensuring that the model on the right does not exhibit significant jaggedness or blurring compared to the one on the left. Once the participant determined the minimum reduction percentage, they could click the submit button to submit their result. They could then choose another model that has not been experimented with for the next observation round.

Results. We collected experimental data from 24 participants, who are 10 undergraduate students and 14 postgraduate students. And each people spent about 2 hours to finish tasks. The relationship between the reduction ratio and the number of model faces obtained is shown in Fig. 3. Each data point represents the corresponding relationship between the number of faces and the reduction percentage of a particular 3D model.

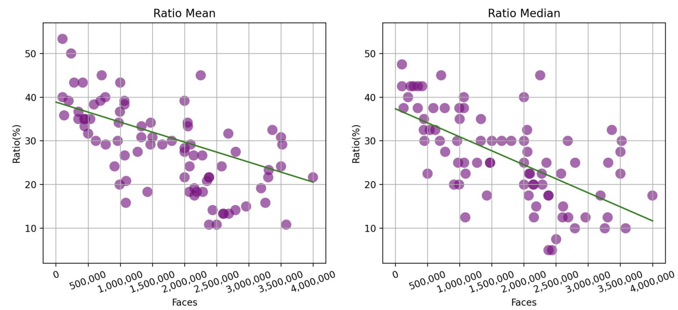


Fig. 3: The correspondence between the number of faces and the ratio: the left image shows the mean ratio for each model, while the right shows the median.

In the figure, we used a linear function $y = ax+b$ to perform regression fitting on the data points. Where x represents the total number of faces in the 3D model, y represents the obtained simplification percentage, and a and b are parameters of the linear function. To evaluate the fitting effects of both the mean and median, we introduced the Root Mean Square Error (RMSE) metric. The smaller the RMSE, the better the fitting effect of the data.

Fig. 4: The screenshot of the comparative experiment system. Both 3D models are in the full body shot. Left side is the original model, and right side is the model after simplification. The two 3D models on the left and right can rotate synchronously. When the participant rotates one of the models, the other would also rotate at the same angle.

From the fitting results, the linear relationship conforms to the following speculation: models with higher numbers of faces require smaller reduction percentages. However, since human perception of 3D models is not solely based on the number of faces but also on the shape of the mesh, the data points do not perfectly align with the trend of the linear plot. Therefore, in the following section, we utilize neural networks to extract shape features to introduce nonlinear characteristics to the regression for further exploration. Additionally, because the RMSE of the mean fitting data (= 7.447) is smaller

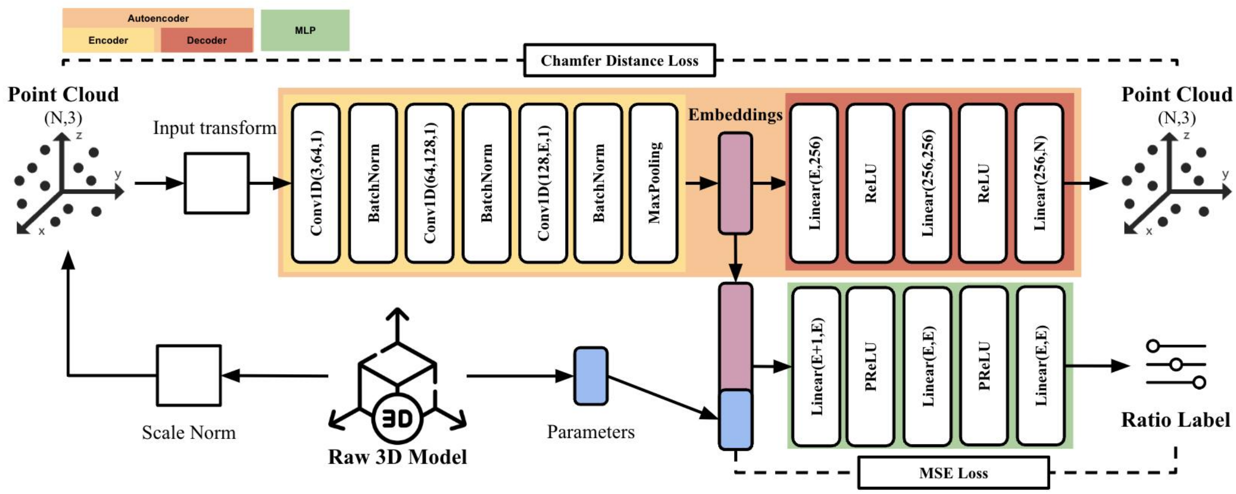


Fig. 5: The structure of ASimpNet.

than the median ($= 8.295$), indicating that the data is more concentrated around the fitted line, we use the mean ratio as the label for training the neural network dataset.

C. ASimpNet

To incorporate the geometric features of 3D models and obtain optimal and versatile simplification parameters, we propose a neural network architecture named ASimpNet. The structure of ASimpNet is illustrated in Fig. 5. It consists of two main components: an Autoencoder network and a multilayer perceptron (MLP) network. In this study, we only require the autoencoder structure for feature extraction of 3D models to identify differences in different model structures. PointNet [19] offers fast training speed and high performance, making it suitable for high-sampling point cloud data. Therefore, we solely utilize the PointNet structure as the autoencoder. PointNet is a universal continuous set function approximator:

$$f(x_1, x_2, ..., x_n) = \gamma(\max_{i=1,...,n}\{h(x_i)\}) \tag{1}$$

where $x_1, x_2, ...$ is a set of unordered point with $x_i \in \mathbb{R}^d$, γ and h are MLP networks. And the set function $f : \chi \rightarrow \mathbb{R}$ is invariant to input point permutations and can approximate any continuous set function [19]. We adopt Chamfer Distance as the criterion loss function to compare the decoded point cloud and the raw point cloud.

$$d(S_1, S_2) = \sum_{x\in S_1} \min_{y\in S_2} \left\|x - y\right\|_2^2 + \sum_{y\in S_2} \min_{x\in S_1} \left\|x - y\right\|_2^2 \tag{2}$$

where S_1 and S_2 represent the point set of two point clouds, x and y represent the single point. Based on the shape of the input for the MLP, we consider the size of input data, output types, ease of training, and training efficiency.

IV. Implementation

To ensure that the data fit within the memory constraints for training and considering the size of the dataset, we set the number of point cloud samples to 100,000. The model version with the lowest test loss and a relatively low train loss will be selected for the evaluation section. Regarding the original parameters of the 3D model, since the vertex number and edge number can be reflected in the face number, we choose the face number as the involved parameter of the following network.

A. Training Equipment

The training machine specifications for ASimp are as follows. Memory: 32GB, CPU: Intel i9-13900K, GPU: NVIDIA GeForce RTX 4080, Operating System: Ubuntu 22.04.

B. Data Augmentation

Due to the limited quantity of high-poly 3D model datasets, relying solely on the total face count parameter as input for the MLP may result in the network being tightly bound to the total face count, potentially leading to degraded performance. In our previous experiments, the label data obtained indicated that beyond a certain label ratio, users cannot distinguish between the simplified model and the original model. Therefore, we employ a resampling technique for data augmentation. We divide the length from “optimal” to “full face count” into $N = 5$ equal segments for resampling as additional inputs to the MLP. Thus, we update the respective simplification ratio to serve as the new label.

$$F_i = F_{raw} - \frac{i-1}{N}(F_{raw} - F_{opt});\ R_i = \frac{F_{opt}}{F_i} \tag{3}$$

where $i \in \{1, \ldots, N\}$, F_i and R_i denotes the new sample face number and the label ratio. F_{opt} and F_{raw} denotes the optimal face number and the raw face number.

V. Evaluation

We present the evaluation of our 3D mesh simplification scheme ASimp and the implemented ASimpNet model. We attempt to replicate the following works from their official implementation as baselines on our test dataset and machine: PoNQ [16], PointTriNet [15], since they all utilize neural networks, they fall under the category of learnable dimension reduction. Besides, we set the sample points 100,000 as ours.

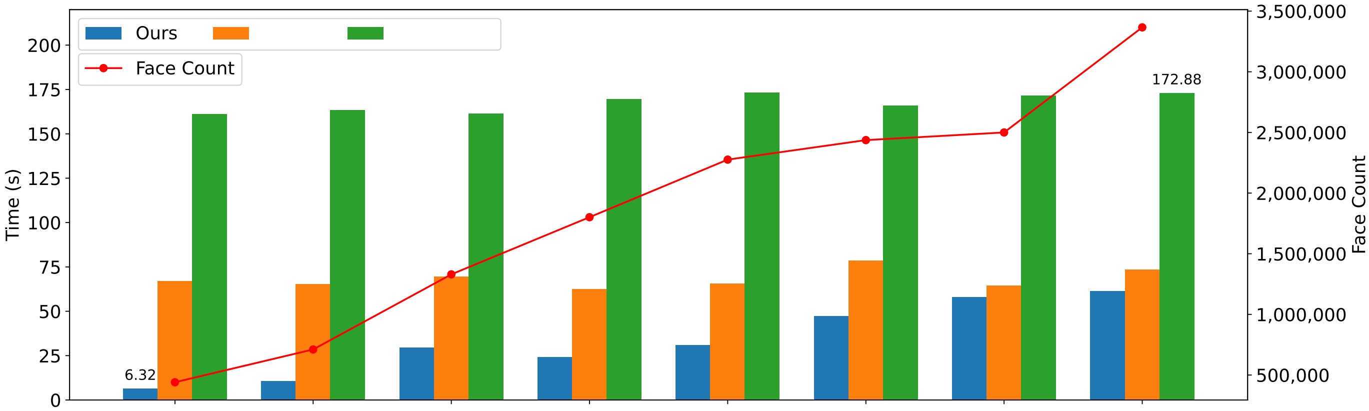


Fig. 6: It shows the comparison of time consumption during simplification with models from the test dataset. The names of 3D models are on the x-axis, and arranged from left to right in ascending order based on the number of surfaces. Each data in the bar chart represents the average of five trials.

Our evaluation aims at two research questions (**RQs**):
RQ1: How efficient of the entire ASimp preprocessing?
RQ2: How effective of the performance achieved?

A. *RQ1: Efficiency*

We compared baselines above with ASimp in time efficiency and face reduce proportion, where we processed with the CPU mode to simulate the common usage. The comparison result is illustrated in Fig. 6. Since the runtime and the number of output faces for the other two mesh simplification methods are related to the number of input sampling points, their runtime and output face count remain largely unchanged as the number of faces in the original 3D model increases. In contrast, our method shows a positive correlation between runtime and the number of faces in the original model, though both remain lower than those of the other methods. Additionally, the mesh simplification ratio in our approach is derived from values inferred by the neural network based on QoE data. Our outputs are not constrained by the settings of the sampling points, while the face number's upper limit of products from PoNQ and PointTriNet are fixed by their model structure design. The selected products from our work shows in Fig. 8.

B. *RQ2: Effectiveness*

Quantitative Assessment. To assess Asimp's simplification performance of simplified meshes, we sample 100K points from the surface and measure the chamfer distance (CD) and normal dissimilarity (NE) with the original mesh. What's more, we compute the MSE error between the first 200 eigenvectors of the laplacian (LE) between the original and the simplified product. To assess the triangulation performance, we use the percentage of non-watertight edges (WA) to ensure the reliability and usability of the meshes. And we compute the face ratio (FR) to compare the compression effect among the listed methods. The results are shown in Table I. We found that ASimp's products show the best performance in CD, NE, WA, LE, and processing time. But because of balancing the QoE and the face number, the FR is larger than others. Thus, we conduct the **ablation study** on ASimpNet. During the stage with the lowest test loss (indicating better generalizable optimal performance), we recorded their Chamfer distance (CD) train/test loss and the final ratio label's mean square error (MSE) train/test loss, shown in Table II.

TABLE I: Quantitative comparison of products from the baseline methods and ours. The data is mean values from the test dataset and after 5 trials. The lowest value of every column is highlighted in bold.

Method	CD	NE	WA	LE	FR	Time
PointTriNet	0.073	0.16	11.32	0.73	0.17	167.32
PoNQ	0.068	0.14	**0.00**	0.71	**0.16**	68.24
Ours	**0.005**	**0.04**	**0.00**	**0.25**	0.38	**33.47**

TABLE II: Results of the ablation study on ASimpNet.

	ASimpNet		**w/o scale norm**		**w/o batchnorm layers**	
	Train Loss	Test Loss	Train Loss	Test Loss	Train Loss	Test Loss
CD	0.0012	0.0090	2989.1	3591.3	0.0021	0.0093
MSE	161.94	287.49	149.61	496.82	194.16	290.84

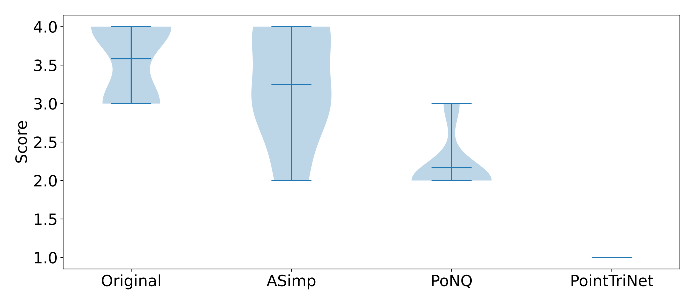
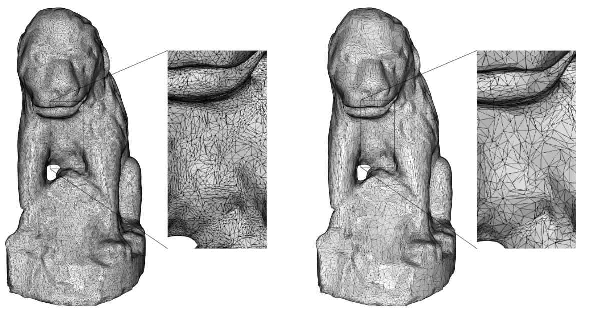


Fig. 7: The violin plot of QoE rating result from the user study.

Qualitative Assessment. We conducted a blind test by mixing simplified products from above experiments with the original 3D models. Similar to previous experiments, we disabled the zoom function and allowed participants to rotate 3D models during the observation. In each test round, participants were tasked with observing four 3D models: the original 3D model, ASimp product, PoNQ [16] product, and PointTriNet [15] product. Participants were then asked to score the 3D models from 1 to 4, where 4 represents the highest perceived quality, and 1 represents the lowest quality. Finally, we collected the data and grouped according to the method. The experimental results are shown in Fig. 7. This demonstrates that the QoE of ASimp's simplified product is very close to the original, indicating the effectiveness.



(a) Left: Original 101,330 faces; Right: Ours 36,368 faces.

(b) Left: Original 710,782 faces; Right: Ours 304,072 faces.

Fig. 8: The comparison showcases between original raw 3D models and ASimp-processed models. Black mesh wireframes are shown within them.

VI. Conclusion

In conclusion, our work introduces ASimp, an innovative, automatic, human-centered approach to 3D mesh simplification for preprocessing. Powered by the ASimpNet neural network, this method adeptly harmonizes the QoE with efficient simplification processes, ensuring that high-poly 3D models are rendered with optimal efficiency and keeping the visual fidelity. Through meticulous human-centered experiments in the model's full body shot, we have identified ideal simplification ratios that preserve the QoE, enabling the ASimpNet model to automate these adjustments seamlessly, leveraging both the ASimpNet model and QEM algorithms. Our findings demonstrate the ASimp's superior efficiency and effectiveness. This approach not only stands to improve the processing and use of 3D models in fields as diverse as cultural heritage, archaeology, and visual effects but also offers significant improvements in model storage, rendering speeds, and interactive user experiences.

Acknowledgment

This work is supported by the Open Topics of Key Laboratory of Blockchain Technology and Data Security, The Ministry of Industry and Information Technology of the People's Republic of China.

References

[1] F Rechichi and F Fiorillo, "Creating, visualizing and sharing high-poly and multi-textures models on the web," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 655–662, 2019.

[2] Haihan Duan, Jiaye Li, Sizheng Fan, Zhonghao Lin, Xiao Wu, and Wei Cai, "Metaverse for social good: A university campus prototype," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 153–161.

[3] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *Ieee Network*, vol. 24, no. 2, pp. 36–41, 2010.

[4] Ramesh Jain, "Quality of experience," *IEEE multimedia*, vol. 11, no. 1, pp. 96–95, 2004.

[5] Xiangrong Li, *Mesh modification procedures for general 3D non-manifold domains*, Ph.D. thesis, Citeseer, 2003.

[6] Antonio Bacciaglia, Alessandro Ceruti, and Alfredo Liverani, "Surface smoothing for topological optimized 3d models," *Structural and Multidisciplinary Optimization*, vol. 64, no. 6, pp. 3453–3472, 2021.

[7] Wenbin Zhao, Jiabao Lei, Yuxin Wen, Jianguo Zhang, and Kui Jia, "Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10256–10265.

[8] Michael Garland and Paul S Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 209–216.

[9] William J Schroeder, Jonathan A Zarge, and William E Lorensen, "Decimation of triangle meshes," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 1992, pp. 65–70.

[10] P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics*, vol. 22, no. 1, pp. 37–54, 1998.

[11] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, 1992, pp. 71–78.

[12] Michael Garland and Paul S Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," in *Proceedings Visualization'98 (Cat. No. 98CB36276)*. IEEE, 1998, pp. 263–269.

[13] Xianhe Jiao, Chenlei Lv, Junli Zhao, Ran Yi, Yu-Hui Wen, Zhenkuan Pan, Zhongke Wu, and Yong-jin Liu, "Weighted poisson-disk resampling on large-scale point clouds," *arXiv preprint arXiv:2412.09177*, 2024.

[14] Rolandos Alexandros Potamias, Stylianos Ploumpis, and Stefanos Zafeiriou, "Neural mesh simplification," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18562–18571.

[15] Nicholas Sharp and Maks Ovsjanikov, "Pointtrinet: Learned triangulation of 3d point sets," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 2020, pp. 762–778.

[16] Nissim Maruani, Maks Ovsjanikov, Pierre Alliez, and Mathieu Desbrun, "Ponq: a neural qem-based mesh representation," *arXiv preprint arXiv:2403.12870*, 2024.

[17] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris, "Deep learning advances in computer vision with 3d data: A survey," *ACM computing surveys (CSUR)*, vol. 50, no. 2, pp. 1–38, 2017.

[18] Lehao Lin, Haihan Duan, and Wei Cai, "Web3dp: A crowdsourcing platform for 3d models based on web3 infrastructure," in *Proceedings of the 14th conference on acm multimedia systems*, 2023, pp. 397–402.

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.