Deep learning-based determination of optimal triangles number of graphic object's polygonal model

Oleksandr Romanyuk[1,†] and Yevhen Zavalniuk[1,*,†]

[1] *Vinnytsia National Technical University, Khmelnytske shose 95, 21021, Vinnytsia, Ukraine*

Abstract

In the article, the neural network-based method of predicting the optimal triangles count of scene object's polygonal model is proposed. The necessity of polygonal models simplification for providing the highly productive visualization of three-dimensional scenes is analyzed. The main approaches to polygonal models simplification are discussed. The main geometrical-spatial factors that determine the optimal polygons number of object's surface are indicated. The existing methods of the direct simplified model generation, iterative model generation of optimal complexity, model generation relative to the parameters of temporal rendering equation, prediction of the optimal local density of a particular polygon are described. The advantages and disadvantages of described methods are given. The need in complementing the methods of polygonal model simplification with the prediction of model's optimal polygons number is justified. The proposed method of optimal polygons number prediction that lies in two-branch neural processing of object's vector and volume data is described. The development of dataset of 24000 samples that is based on ShapeNet dataset for neural network training is described. The process of optimization of the proposed neural network architecture's parameters with the usage of the optimization library is characterized. The developed method of optimal triangles number prediction is verified through calculating the accuracy metrics of test dataset approximation. It is shown that the proposed method is more accurate than guessing the triangles number and using feedforward neural networks. The illustrative examples of optimal triangles number prediction for the cars and planes models are provided. In the result, the developed neural network provides faster and more effective polygonal model simplification at the stage of geometrical transformations of graphics pipeline.

Keywords

Rendering, polygonal model, model simplification, neural prediction, convolutional neural network, multilayer perceptron.

1. Introduction

The most important requirements of modern graphics systems are high realism and high productivity of scene visualization. The productivity of three-dimensional image creation depends on the implementation details of the main graphics pipeline stages, which include geometrical transformations and rendering.

The increase in rendering stage productivity is possible by means of developing effective shading methods [1, 2, 3], using simple bidirectional reflectance distribution functions [4, 5], simplifying vector normalization procedures [6, 7], and rendering parallelization.

The productivity of image creation is also determined by the number of polygons, into which the object surfaces are divided. Due to the limited computational capabilities of visualization systems, polygonal mesh simplification methods are used. The two main polygonal mesh simplification methods [8] are removing part of the vertices and merging the edge vertices into new vertices (edge collapse). The disadvantage of the first method is the need for repeated tessellation after vertices removal. In the second method, the base of the polygonal model is conserved.

It is reasonable to simplify the polygonal mesh to the lowest number of polygons at which the losses in visualization quality are imperceptible. The optimal polygons number of an object's surface depends on its geometrical complexity and relative position to the camera. The objects with high surface curvature are modeled in more detail, the distant objects from the camera are represented by a small number of polygons.

The automatic determination of reasonable objects' polygons numbers allows to increase the productivity of three-dimensional image creation and the effectivity of the scene's geometrical characteristics optimization.

Therefore, the development of methods for the three-dimensional figures' optimal polygons numbers prediction is needed.

2. Related Works

The most researched direction of polygonal models simplification is the direct generation of their simpler versions. In the polygonal model generation method, proposed by R. A. Potamias et al. [8], the sampling of vertex points, the prediction of polygons' edges, and the classification of polygons' faces are combined. The basis of the method lies in applying graph neural networks (GNN). In the first step, GNN is used for the selection of points, which approximate the object's surface the most accurately. In the second step, the probability of edge formation between two vertices is predicted. In the third step, the probability that the proposed face belongs to a simplified model is predicted. The method is universal, provides the single-pass polygonal model generation, for the first time the learning of point set and polygonal model generation is provided. However, the user needs to choose the needed polygonal model simplification level by himself.

The method, proposed by N. K. Jayakodi et al. [9], provides the images-based polygonal model generation at mobile devices, taking into account the optimal polygon number ("polycount") with available energetic resources. Firstly, using a convolutional neural network (CNN), the feature vector is extracted from the image. Next, for the model generation, the graph convolutional network (GCN) [10] is used. The multilevel system of polygonal mesh generation is applied. At each level, the binary mask of the projected image, which is based on the predicted model, and the binary mask of the original image are compared. If binary masks are not similar enough, the transition to the higher level of generation, which corresponds to a bigger number of object's surface polygons, is carried out. The threshold values for binary mask comparison are determined at each generation level, taking into account the energy costs. As a result, the generation of simple objects requires less computational resources. The approach lacks the

direct prediction of polygons number, it leads to iterative generation of the surface's polygonal mesh.

The method, developed by N. Tack et al. [11], also provides a highly productive generation of polygonal models for mobile devices. The equation for evaluating an approximate time of object rendering is proposed, it depends on the time of code initialization, projective transformations, vertex shading, triangle, row and pixel rasterization, number of vertices and triangles, probabilities of vertices shading and clipping. If the allowed time of object rendering is set, the optimal number of triangles is determined from the equation. After this, from the special plot the necessary level of detail and threshold angle between the polygon's vertices normals (determines the need in the subdivision of the polygon) are determined. The implementation of the level of detail increase is based on uniform or non-uniform (takes into account the curvature or silhouettes) triangles subdivision. The disadvantage of the approach is the dependence of distinct parameters' contributions to the complete time of visualization on the chosen renderer and the evolution of rendering algorithms.

R. Kanapady et al. [12] introduced a method to assess the optimal density of specific polygon within an object's polygonal model. Triangle density in the study is defined as the radius of the circle, drawn around the triangle. The optimal radius is the maximum radius, in the direction of which the approximation of the given surface characteristic is accurate enough. The prediction of triangles densities is carried out using the feedforward neural network with one hidden layer.

The method, proposed by Z. Zhang et al. [13], lies in predicting the maximum threshold area of a particular polygon relative to a given low-polygonal surface model. For the area prediction, the usage of three neural networks is possible: fully connected, ResNet1, and ResNet2. In order to provide the desired number of polygons the K coefficient is used. The equal distribution of surface approximation error is provided – the bigger polygons are characterized by lower error density.

The disadvantage of the last two methods is that only the local density of the polygonal model is predicted. Data about optimal local density provide the generation of the simplest possible polygonal mesh with acceptable accuracy of original surface reconstruction. However, during the simplification of the polygonal model, the initial solution usually is already characterized by the density distribution of particular polygons. Then, with the condition of the main surface's features conservation by the simplification algorithm, it is only enough to determine the optimal number of triangles. In addition, these methods are aimed at solving the material sciences tasks, that lie outside the computer graphics field.

In the method, proposed by A.-M. Cretu et al. [14], the generation of simplified polygonal model with the preservation of object's interest regions is investigated. For the object's polygonal model simplification the gas neural networks are used. The novelty of method is the prediction of the model's optimal faces number for different levels of detail. The prediction of faces number is made using two-layer feedforward neural network, the hidden layer of which includes 40 neurons. The input of neural network takes the polygonal model, the output data are predicted faces numbers. During the training dataset creation, for every resolution interval the number of faces that provided the most effective original model approximation was chosen. One of the method's disadvantages is that the usage of simple feedforward networks doesn't provide the comprehensive learning of objects' local features [15]. It is more appropriate to use more complex neural networks types, in particular, convolutional networks that allow to learn

the hierarchical spatial features [15] of three-dimensional figure. The approach can be improved through determination of the connection between object's polygons number and human perception of final image quality, and by considering the given coordinates of object and camera.

Thus, the task of predicting the reasonable surface's polygons number is insufficiently researched. Only in two of the discussed methods the direct determination of the surface model's optimal triangles number is present. However, the first of the specified methods is characterized by a large number of parameters for determination and is more connected with the features of the graphics pipeline, than with the objects' locations and surface features. The second of the specified methods is not effective enough for learning the local and spatial features of three-dimensional graphic figures. In addition, in most of the reviewed methods neural network-based processing of object surfaces' features is applied. The advantage of neural network usage for polygonal model simplification is the possibility of complex patterns, that are connected with the figure surface's geometry, learning.

Therefore, there is a need for the development of a neural method that determines the optimal number of triangles for an object's surface, considering both the object's location and complexity. The purpose of the article is the neural network development for predicting the optimal polygons number of graphic scene's object that will provide more efficient polygonal model simplification.

3. Proposed Methodology for Predicting the Optimal Triangles Number of Graphic Object

In the article, the relation between the geometrical complexity of the object, its location in the scene, and the optimal polygons number of the object's surface is discussed. As polygons, the triangles are considered, taking into account their simplicity and most frequent usage for surface tessellation.

3.1. The General Structure of Neural Network for Predicting the Optimal Polygons Number

The geometrical features of an object's surface are represented as voxelized tensor of dimensions $48 \times 48 \times 48$. The advantages of this approach are the structuredness of data and the effective representation of structural-spatial relations between surface points.

The features of the object's location in the scene are represented by a vector of dimension 5×1 that includes the data about the object's and camera's coordinates in the Blender Cartesian system. The vector also includes the number of triangles of the original polygonal model. In this study, the constant z-coordinates of the camera and objects are used for simplification.

The optimal number of polygons is defined as the minimal number of triangles, at which the NMSE error between the original-model-based image and the simplified-model-based image doesn't exceed 0.001. NMSE (Normalized Mean Square Error) is calculated using the formula [16]:

$$\frac{\sum_i (R_1(i) - R_2(i))^2 + (G_1(i) - G_2(i))^2 + (B_1(i) - B_2(i))^2}{\sum_i R_1(i)^2 + G_1(i)^2 + B_1(i)^2}, \tag{1}$$

where R, G, B – the values of RGB model channels, i – number of pixel, 1 and 2 – the indices of original and new images respectively.

Threshold NMSE value 0.001 means that images differ significantly for an observer [16].

The proposed method (Fig. 1) for determining the optimal number of triangles lies in the usage of two neural network branches. The first branch («first nn-branch») is designed for the transformations of the object's surface voxel tensor. The second branch ("second nn-branch") is designed for the transformations of the object's features vector. Both branches are merged into the resulting neural network branch ("unified nn-branch") that provides the optimal number of triangles prediction. The general structure of the proposed neural network is shown in Fig. 1.
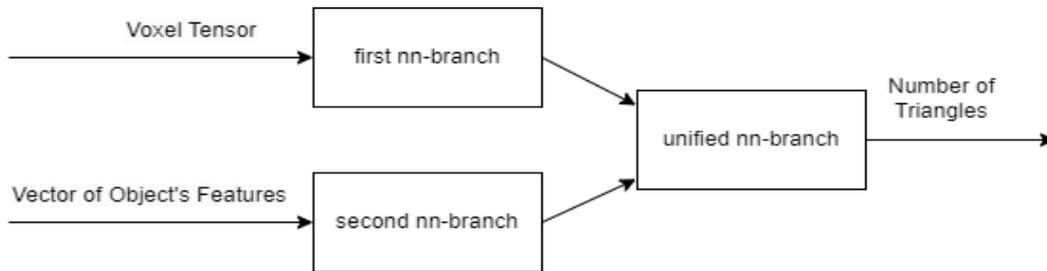
Figure 1: The structure of neural network for predicting the polygonal model's optimal number of triangles.

3.2. The Development of Dataset for Neural Network Training

In the study, a sample of 2000 converted to .stl format figures from the ShapeNet dataset [17] is used. The 400 objects from five file size intervals ($[0, 2), [2, 4), [4, 6), [6, 8), [8, 10]$ Mb) were selected in order to ensure that the dataset is balanced in terms of figure complexity levels. The base of the sample includes the figures of transport objects (Fig. 2).

Figure 2: Examples of used figures from ShapeNet dataset.

For each of the figures, 12 cases of its mutual location with the camera were considered. The camera was located at 4 directions relative to the object along the x, y axes of the coordinate system. The object was located at three discrete remoteness levels relative to the camera. Overall, there are 24000 combinations of objects and their locations relative to the camera.

For each of the combinations, 10 levels of triangles number reduction ($\{0.001, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 1.0\}$) were considered. The Blender modifier Decimate was applied in the mode of edge collapse. Fig. 3 shows the example of the figure «Plane» for different numbers of triangles.

Figure 3: Visualized figure «Plane» for different percentages of polygonal model simplification.

For each of the 24000 specified combinations, the 10 images, which correspond to different polygonal model simplification percentages, were formed. Using the NMSE metric, it was determined, which image is the last (in the direction of simplifying) that deviates from the original by less than 0.001. Accordingly, the triangles number of the model, at the basis of which the chosen image was formed, was determined. The found number of triangles is considered to be optimal.

As a result, the dataset of 24000 objects' feature vectors (see «The General Structure of Neural Network for Predicting the Optimal Polygons Number») and target optimal triangles numbers was created.

Web tool Drububu.com was used for the formation of 2000 voxelized figure tensors. Fig. 4 shows examples of voxelized figures from the ShapeNet dataset.
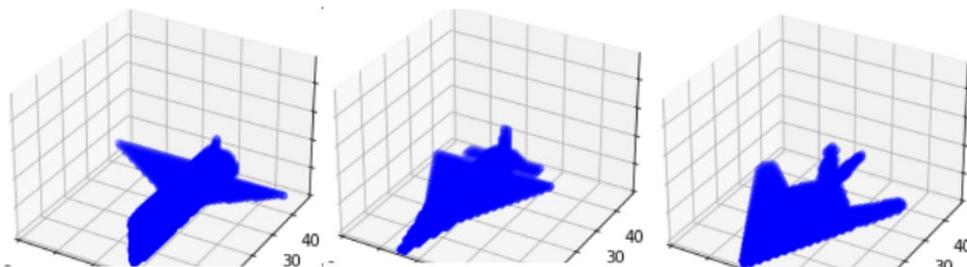
Figure 4: Examples of voxelised figures from ShapeNet dataset

75% of dataset are used for training, 15% are used for validation, 10% are used for testing.

3.3. The Development of Neural Network Architecture

The neural network branch for the object's feature vector processing is implemented using multi-layer perceptron (MLP) [18], branch for the object surface's voxel tensor processing is implemented using convolutional neural network (CNN) [19, 20]. These neural network types

are the standard solutions [15] for vector and volumetric data processing respectively. The output values of both branches serve as input values for MLP that merges them.

For the determination of optimal architectures for the specified neural network branches the optimization Python library Optuna was used.

For the optimization of first-branch MLP, the parameters were the number of neuron layers (interval of values $[1,10]$) and the number of neurons in layer ($[4,128]$). The 100 optimization experiments were conducted, validation MSE (mean squared error) was set as the error metric for minimization, validation MAE (mean absolute error) was set as error change tracking metric, the recommended by documentation sampler TPESmapler and pruner HyperbandPruner were used.

The resulting MLP includes 6 layers with 127, 16, 57, 60, 18, 35 neurons. MSE, MAE relative to the target triangles numbers were 30597088 and 3409 respectively.

For the CNN optimization, the parameters were the numbers of filters (categorical values $\{16,32,64,128\}$) for four convolutional layers. The size of the convolution kernel was set as $3\times3\times3$, and the size of pooling windows was set as $2\times2\times2$. The 50 experiments were conducted, validation MSE and MAE were set as metrics of error minimization and error change tracking. The same sampler and pruner were used.

The resulting numbers of filters were 32, 128, 128, 128. MSE, MAE were 16505085 and 2280 respectively.

For the optimization of MLP that merges neural network branches the parameters were the number of neurons in the layer, which receives the information from MLP ($\{1,4,16\}$), number of neurons in the layer, which receives the information from CNN ($\{1,4,16\}$, this layer is located after the layer with 512 neurons), number of the following layers ($[1,4]$) and number of neurons inside them $[4,128]$.

The resulting architecture of merging MLP includes 16-neuron layers for receiving information from both branches, 1 layer with 76 neurons. The final node of architecture is the neuron for result calculation. After 20 experiments MSE, MAE were 10997652 and 2009 respectively.

Fig. 5 shows the optimized architecture of the proposed neural network.

The last layer of neural network is linear. After convolutional and fully connected layers ReLU activation function is used.

3.4. Neural Network Training

For neural network training, the following parameters were set: number of iterations – 100, the criterion of early stoppage – validation error is not improving for 10 iterations, batch size – 32. MSE was selected as the error metric for minimization, MAE was selected as the metric for the error tracking. Adam optimizer was used.

The training was performed using Kaggle TPU VM v3-8.

Validation error MSE stopped improving after 71 training epochs. Fig. 6 shows the plot of training and validation MAE changes during the neural network training.
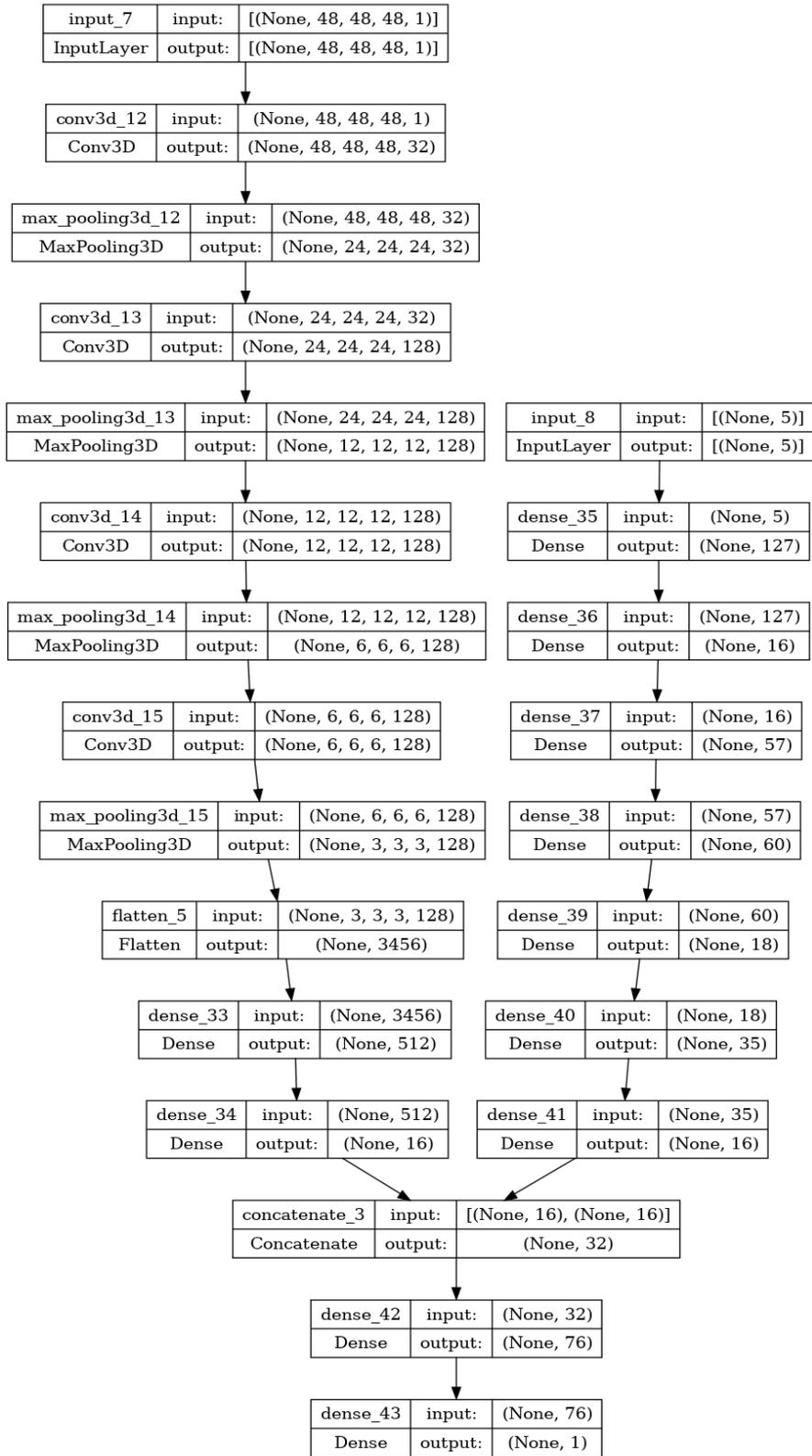
Figure 5: Optimized structure of neural network.
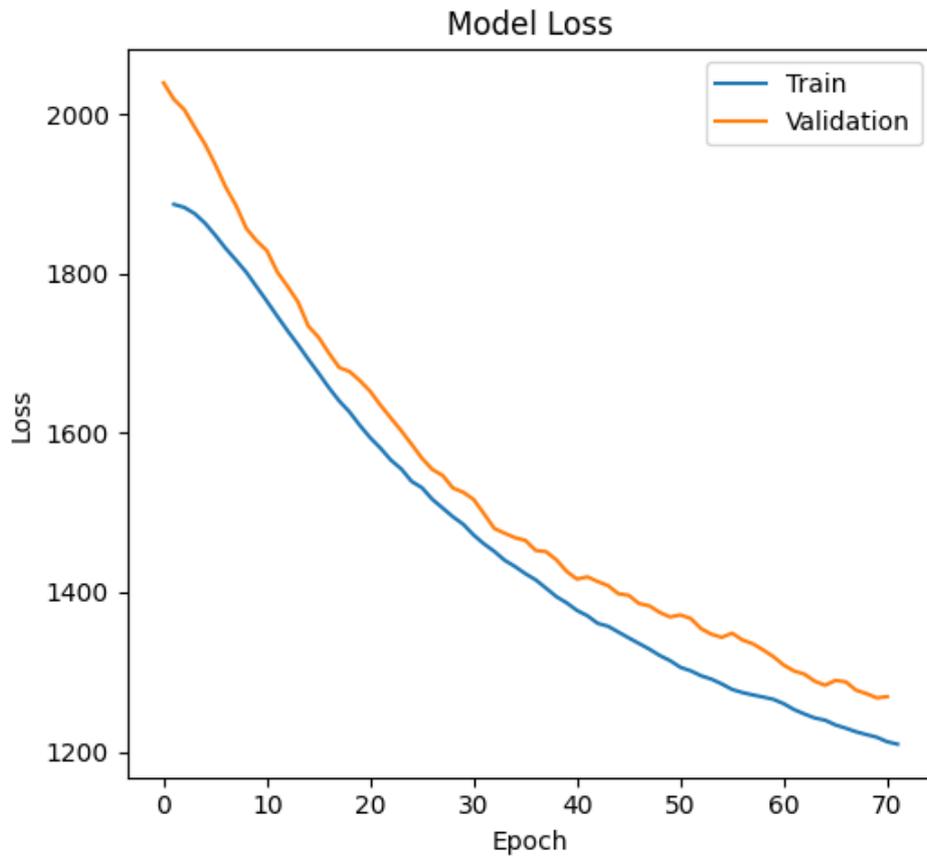
Figure 6: The plot of training and validation MAE error changes during neural network training.

In the Table 1, the final error metrics values after neural network training are listed.

Table 1

Final values of error metrics after training

Metric	Type	Value
MSE	Training	6159425
MSE	Validation	6597670
MAE	Training	1186
MAE	Validation	1212

3.5. Neural Network Testing on the Test Dataset

Test dataset includes 10% of full dataset samples. Fig. 7 shows the plot with the predicted and target values of optimal triangles numbers of polygonal models. Metric MAE relative to the test dataset equals 1322.

Table 2 shows the MAE values by intervals of target triangles numbers of objects.

Table 2

MAE metric values by intervals of target triangles numbers

Interval of target triangles numbers	MAE
10 − 10 000	880
10 000-20 000	2521
20 000-30 000	5191
30 000+	6563

NRMSE (Normalized Root Mean Square Error) [21] of approximating the target values set equals

$$NRMSE = \frac{100 \cdot \sqrt{MSE}}{y_{real\_max} - y_{real\_min}} = \frac{100 \cdot \sqrt{7884324}}{87857 - 10} = 3.2\%.$$
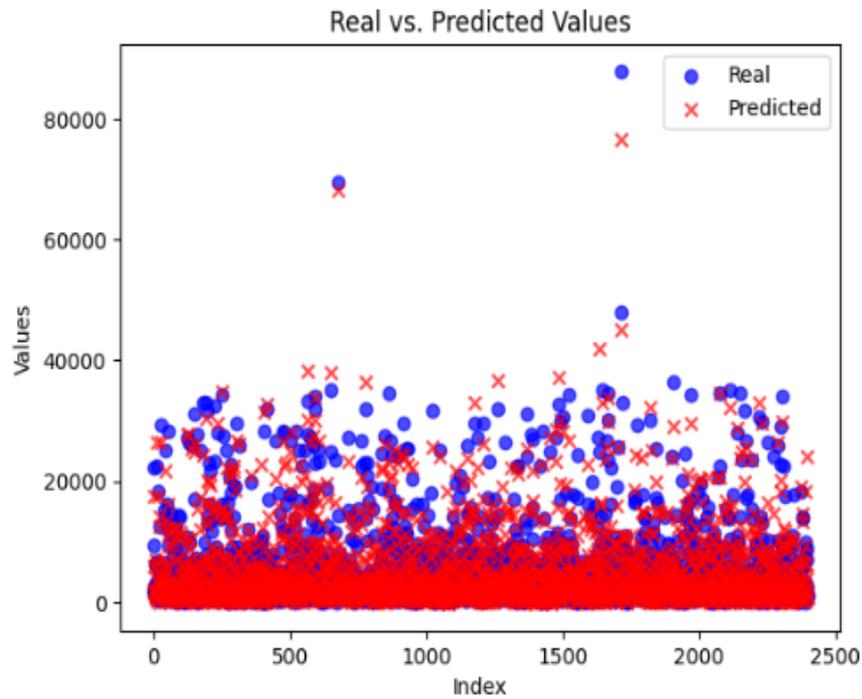
Figure 7: The plot with the predicted and target values of optimal triangles numbers.

The average speed of triangles number prediction for one figure was 0.09 s when using Kaggle CPU, 0.008 s when using Kaggle TPU VM v3-8.

Fig. 8 shows the examples of predicted optimal triangles numbers for the surfaces of three-dimensional cars and planes models.

The predicted optimal values are circled with a red line. The target triangles numbers are located below the images in the middle.

4. The Analysis of Results

RMSE value of the test dataset approximation is 3.2% of the dataset's target values range. Hence, the highly accurate prediction of the optimal object's triangles number is provided.

Let's compare the results of the model's predictions with guessing (guessing the constant value for each case). It was determined that the most accurate approximation of test dataset values is achieved with the number 3000. Then, MAE equals 4018. Therefore, the proposed model provides four times more accurate predictions of polygonal models' triangles numbers than simple guessing. In addition, the comparison of the proposed neural network architecture usage and the usage of the architecture, used in [14], for predicting the optimal polygons number was conducted. Two-layer feedforward neural network was used for predicting the optimal triangles numbers of objects from the created dataset at the three distance levels from the camera. In a result, MAE was 3767. The proposed model was also compared with more complex neural network ResNet1, used in [13], that includes fully connected layers of size 32, 32, 64, 128, 64, 32, 1, and residual connection. As the ResNet1 input the data vector and voxel tensor were used. Then, the number of both network parameters is approximately the same. MAE value was 1623.

In comparison with guessing, two-layer feedforward neural network, ResNet1 the decrease in approximating error of test dataset by 67%, 65% and 19% is provided respectively. The increase in the accuracy of triangles number prediction is explained by the usage of CNN for figures' local features learning and MLP for the corrective consideration of objects' spatial parameters. Therefore, the combined CNN and MLP usage provides more accurate learning of three-dimensional objects' features and predicting for them optimal triangles numbers.
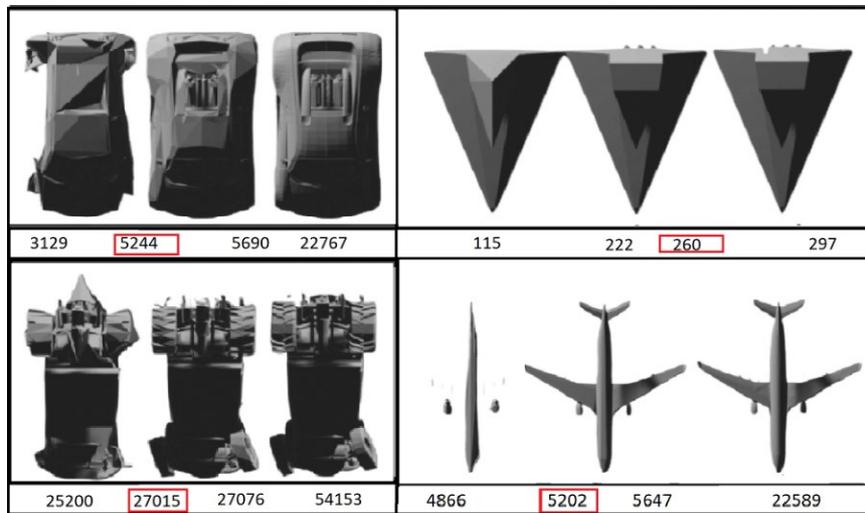
Figure 8: Examples of predicting the optimal triangles numbers for cars and planes models.

5. Discussion

Within the work, the precision of determining the optimal triangles number is improved by combining the optimized architectures of MLP, CNN and using NMSE metrics. In further research, the proposed approach can be improved.

The vector of the object's features includes simple data about the figure (initial number of polygons, coordinates of object and camera). The usage of more complex features (number of points of interest, mean points curvature) will raise the accuracy of dataset approximation. The research on encoding vector data into efficient representation is promising. For the representation of three-dimensional surfaces, it is planned to compare the usage of voxel tensors, polygon vertices arrays, and point clouds. For more precise triangles number prediction the enlargement of the dataset, which includes mainly the transport figures, is desirable. The diversification of mutual camera and object locations is needed. The weights learned on the transport figures are advisable for transfer learning usage. The neural comparison of image with the reference image is more flexible and promising than the usage of similarity metrics. Another possible direction of future research is the parallelization of triangles number prediction.

6. Conclusions

In the article, the neural network-based approach to determining the optimal triangles number of the object's polygonal model is proposed. The approach is based on two-branch processing of volumetric and vector object's features. The direct triangles number prediction is provided based on the learning of local surface and hierarchical spatial patterns of the object, considering the peculiarities of human perception, which is absent in the analyzed methods of polygonal model simplification.

The most promising directions of future research are comparing different methods of object representation, applying the developed neural network in transfer learning for a wider range of objects.

References

[1] O. Romanyuk, O. Romanyuk, R. Chekhmestruk, P. Mykhaylov, V. Maidaniuk, T. Korobeinikova, The Concept and Means of Adaptive Shading, in: 2022 12th International Conference on Advanced Computer Information Technologies. Conference Proceedings, ACIT'2022, IEEE, Ruzomberok, Slovakia, 2022, pp. 33–38. doi: 10.1109/ACIT54803.2022.9913105.

[2] Y. Hu, Y. Yuan, R. Wang, Z. Yang, H. Bao, Multirate Shading with Piecewise Interpolatory Approximation, Computer Graphics Forum 41(7) (2022) 255–266. doi: 10.1111/cgf.14674.

[3] B. Mohanto, A. Islam, E. Gobbetti, O. Staadt, An integrative view of foveated rendering, Computers & Graphics 102 (2022) 474–501. doi: 10.1016/j.cag.2021.10.010.

[4] O. Romanyuk, Y. Zavalniuk, S. Pavlov, R. Chekhmestruk, Z. Bondarenko, T. Koval, A. Kalizhanova, A. Iskakova, New Surface Reflectance Model with the Combination of two Cubic Functions Usage, Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 13(3) (2023) 101–106. doi: 10.35784/iapgos.5327.

[5] L. Bayrak, Investigation of Object Lighting Techniques by Using Two Directional Reflection Distribution Functions, Master's thesis, Firat University, Elazığ, Turkey, 2018 (in Turkish).

[6] D. Um, Solid Modeling and Applications. Rapid Prototyping, CAD and CAE Theory, 2nd. ed., Springer Cham, Cham, Switzerland, 2018.

[7] T. Akenine-Möller, E. Haines, N. Hoffman, A. Pesce, M. Iwanicki, S. Hilaire, Real-Time Rendering, 4th. ed., CRC Press, Boca Raton, FL, 2018.

[8] R. A. Potamias, S. Ploumpis, S. Zafeiriou, Neural Mesh Simplification, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, IEEE, New Orleans, LA, 2022, pp. 18562–18571. doi: 10.1109/CVPR52688.2022.01803.

[9] N. K. Jayakodi, J. Rao Doppa, P. P. Pande, PETNet: Polycount and Energy Trade-off Deep Networks for Producing 3D Objects from Images, in: 2020 57th ACM/IEEE Design Automation Conference, DAC 2020, IEEE, San Francisco, CA, 2020, pp. 1–6. doi: 10.1109/DAC18072.2020.9218525.

[10] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: 5th International Conference on Learning Representations, ICLR 2017, ICLR, Toulon, France, 2017, pp. 2713–2727. doi: 10.48550/arXiv.1609.02907.

[11] N. Tack, F. Morán, G. Lafruit, R. Lauwereins, 3D graphics rendering time modeling and control for mobile terminals, in: Proceedings of the ninth international conference on 3D Web technology, Web3D '04, ACM, Monterey, CA, 2004, pp. 109–117. doi: 10.1145/985040.985056.

[12] R. Kanapady, S. K. Bathina, K. K. Tamma, C. Kamath, V. Kumar, Determination of an initial mesh density for finite element computations via data mining, Report UCRL-JC-144765, Lawrence Livermore National Lab., Livermore, CA, 2001.

[13] Z. Zhang, Y. Wang, P. K. Jimack, H. Wang, MeshingNet: A New Mesh Generation Method Based on Deep Learning, in: Computational Science – ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part III, ICCS 2020, Springer Nature Switzerland, Cham, Switzerland, 2020, pp. 186–198. doi: 10.1007/978-3-030-50420-5_14.

[14] A.-M. Cretu, M. Chagnon-Forget, P. Payeur, Selectively densified 3D object modeling based on regions of interest detection using neural gas networks, Soft Computing 21 (2017) 5443–5455. doi: 10.1007/s00500-016-2132-z.

[15] F. Chollet, Deep Learning with Python, 1st. ed., Manning Publications Co., Shelter Island, NY, 2018.

[16] S. Pavlov, S. Romaniuk, N. Titova, W. Wójcik, O. Romanyuk, L. Koval, The System for Formation of Human Face Images for Operations Planning, in: Proceedings of the II International Scientific and Technical Conference "MODERN TECHNOLOGIES OF BIOMEDICAL ENGINEERING", MTBE 2023, Odesa Polytechnic National University, Odesa, Ukraine, 2023, pp. 145–148 (in Ukrainian).

[17] 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, F. Yu, ShapeNet: An Information-Rich 3D Model Repository, 2015. URL: https://arxiv.org/abs/1512.03012.

[18] A. Subasi, Practical Machine Learning for Data Analysis Using Python, 1st. ed., Academic Press, 2020.

[19] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, 1st. ed., The MIT Press, Cambridge, MA, 2016.

[20] P. Radiuk, An Approach to Accelerate the Training of Convolutional Neural Networks by Tuning the Hyperparameters Of Learning, Computer systems and information technologies 2 (2020) 32–37. doi: 10.31891/CSIT-2020-2-5.

[21] Jedox.com, Error Metrics: How to Evaluate Your Forecasts, 2021. URL: https://www.jedox.com/en/blog/error-metrics-how-to-evaluate-forecasts/#nrmse.