No Reference 3D mesh quality assessment using deep convolutional features

Zaineb Ibork, Anass Nouri, Olivier Lézoray, Christophe Charrier, Raja Touahni

No Reference 3D mesh quality assessment using deep convolutional features

Zaineb Ibork[1,2], Anass Nouri[1], Olivier Lézoray[2], Christophe Charrier[2], Raja Touahni[1]

[1]*SETIME Laboratory, Information Processing and A.I Team, Faculty of Sciences, Ibn Tofail University, Kénitra, Morocco*
[2]*Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France*
{zaineb.ibork,anass.nouri, touahni.raja}@uit.ac.ma {olivier.lezoray, christophe.charrier}@unicaen.fr

***Abstract*—3D meshes have gained significant interest in computer vision community due to their use in several applications such as virtual reality, gaming, heritage preservation, etc. However these 3D contents might be altered in the pre-processing steps like acquisition, compression or denoising. In this context, visual quality assessment algorithms can be used to quantify the amount of distorsions that affect a 3D mesh and hence degrade its visual rendering. We introduce a no-reference mesh quality assessment index based on deep convolutional features named DCFQI (Deep Convolutional Features Quality Index). Leveraging the power of deep learning, particularly transfer learning, allows the proposed approach to score visual quality without the need of reference content, hence emulating the human vision. By rendering a 3D mesh into 2D views and patches, a pre-trained convolutional neural network is used to automatically extract deep features from the latters. The obtained features are used in a Multi Layer Perceptron (MLP) to predict the objective quality score. Two learning strategies are presented and compared for blind quality estimation. Obtained results in terms of correlation with subjective human scores of quality demonstrate the superiority of the proposed index over existing methods.**

***Index Terms*—3D mesh, Visual Quality Assessment, Convolutional Neural Network, Deep learning, Transfert Learning.**

I. Introduction

Perceptual visual quality assessment of 3D meshes, known as 3D Mesh Visual Quality Assessment (MVQA), has gained significant interest in recent years due to the widespread use of 3D models in various applications, ranging from computer graphics to virtual reality, augmented reality, 3D printing, industrial design, engineering, and cultural heritage preservation. As 3D meshes usually undergo different lossy geometry processing operations, distortions can occur, impacting their visual quality and possibly the performance of computer vision applications. While subjective visual quality assessment by human observers is a reliable method, it is expensive, laborious, and time-consuming [1] Objective visual quality assessment methods offer a viable solution to these challenges. They can be categorized based on the availability of a reference object. Full-Reference (FR) methods require a complete reference, No-Reference (NR) or Blind methods do not have any reference information, and Reduced-Reference (RR) methods have partial reference information, such as extracted features. Existing perceptually driven methods predominantly focus on FR [2]–[5] and RR methods [6], [7] to evaluate perceived quality. However, in practical scenarios, a reference is not always available, necessitating the development of no-reference methods.

Recently, CNNs (Convolutional Neural Networks) have been widely adopted for NR MVQA [8]–[10]. However, processing meshes directly with CNNs is challenging due to their complex polygon-based structure and the reliance of CNNs on Euclidean discrete convolutions. In order to address this limitation and to mimic the subjective evaluation performed by the human visual system, we propose the following solution: we generate 2D projections of each mesh from multiple viewpoints and divide these projected images into overlapping patches to preserve edge information.

Next, while many state-of-the-art approaches are patch-based, we aim in this study to introduce a protocol that incorporates both views and patches. We independently fed the extracted patches and rendered views into a pre-trained CNN for feature extraction. To predict the visual quality score, we employ an additional MLP (Multi-Layer Perceptron). Finally, we compute the quality score of a mesh by aggregating the scores obtained from the images at the patch or view levels.

The structure of this paper is outlined as follows. Section II provides a description of the data preparation and the proposed method. Section III describes the experimental setup, including details about the database used, the validation protocol and a comparative discussion of the results. Finally, we conclude with remarks and perspectives on the topic.

II. Deep convolutional features Quality Index

A. Flowchart

Given a 3D mesh whose visual quality has to be evaluated, the proposed approach renders several 2D projection views by varying the point of view around the mesh. The obtained 2D views are normalized and cropped in order to minimize the amount of white background in the image. The obtained views can be subsequently divided into four overlapping patches. Each 2D view (or each patch extracted from the 2D views) is fed to a pre-trained convolutional network (VGG 16) [11] in order to obtain a feature vector as the default densely connected classifier is removed from the neural network. Consequently, this feature vector is used to estimate the

This work received funding from PHC TOUBKAL TBK/22/142-CAMPUS N°47259YH.

view or patch quality. As each 3D mesh M_i is described as a set of rendered 2D views or patches, we average the obtained objective quality scores to quantify the final 3D mesh visual quality. In the sequel, we will describe the process of preparing our two databases (view and patch). We will begin by rendering the 3D meshes, followed by the patching of the corresponding 2D views.

B. 3D mesh rendering

As alluded above, our approach considers 2D rendered views and 2D patches of a 3D mesh in order to assess the percieved quality of a 3D mesh. Given a database of N 3D meshes, the goal is to render each 3D mesh $M_{i,i\in[0,N[}$ in order to obtain 2D views/patches at different viewing angles. To ensure that each mesh is positioned in a similar way in the rendered views, its centroid is placed at the origin of the coordinate system. This enables to have comparable renderings for all the meshes, which is mandatory to predict reliable quality scores from the extracted features representing the views/patches.

a) 2D views: A 3D mesh M_i is rendered from 11 viewpoints by systematically changing the azimuth (θ_a) and elevation (θ_e) angles by $\frac{\pi}{3}$ (60 degrees) for each viewpoint. Figure 1 illustrates this process.

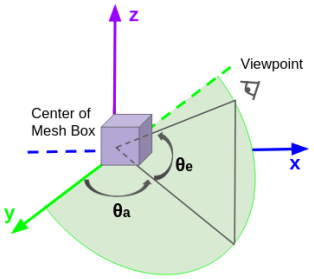

Fig. 1. Illustrating camera position in the rendering process: Azimuth angle (θ_a) in the horizontal plane with $z = 0$ and Elevation angle (θ_e) from the xy plane with $y = 0$.

The Elevation angle is fixed to 0 degree while varying the Azimuth (and vice versa) for capturing the views, ensuring smooth transitions. The camera position and the distance to object are manually fixed to ensure that the object appears close to the camera, hence maximizing finest details and visibility. This process allows us to create a comprehensive dataset of 2D views, showcasing various perspectives and important object details. An example of the resulting rendered 2D views of the Armadillo 3D mesh from the Liris/Epfl General Purpose datase [1] can be seen in Figure 2.

b) 2D views optimization and decomposition: The resulting rendered views have a large resolution of 1024×1024. This size was fixed to capture discriminating details of prime importance for visual quality assessment. However, these images also contain a significant amount of white background. To minimize the impact of the white background, useless for quality assessment, we crop and resize the images to include only the mesh bounding box, effectively removing most of the surrounding white background. As this cropping operation depends on the mesh bounding box size, the resulting images' sizes can be different. To avoid this, we resize all the images to 512×512.

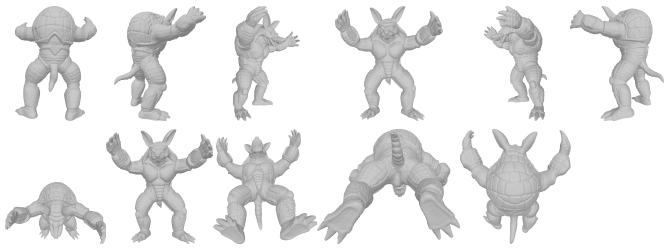

Fig. 2. Armadillo's 11 rendered views: the views in the first row are obtained by fixing $\theta_e = 0$ and graduating θ_a by 60 degrees at a time. We switch this process in the second row.

If the 2D views are interesting to capture the details from 3D meshes, only $11 \times N$ views are obtained. Such number of views might be not sufficient to leverage the power of deep learning architectures. To cope with this, four overlapped patches are extracted from each 2D views. In contrast to the approach of [10] that extract very small patches of 32×32, we consider larger patches of size 288×288. Indeed, having patches of very small sizes has many drawbacks. First, small patches do not always contain sufficient information for quality assessment. Second, these patches can be only made of background and specific strategies are needed to eliminate them [12]. Third, this does not ensure that the extracted number of patches per view is always the same, which creates an unbalanced dataset. To ensure a better coverage of all the information, especially at the connection between adjacent patches, patches are extracted with an overlap of 20%. Figure 3 illustrates the decomposition of a 2D view into four overlapped patches.

To sum up, given a database of N meshes, we construct two databases B_k with $k \in \{\text{view}, \text{patch}\}$. B_{view} contains $N \times 11$ images, whereas B_{patch} contains $N \times 11 \times 4$ images. Whatever the database, each constituting image is normalized between 0 and 1. In [12], the authors performed a local contrast normalization, while in the proposed approach, we have preferred a global normalization performed on the L^* lightness channel in the CIELAB color space. Indeed, the perceived lightness (L^*) is nonlinear as the human perception and its normalization may optimize the quality assessment process.

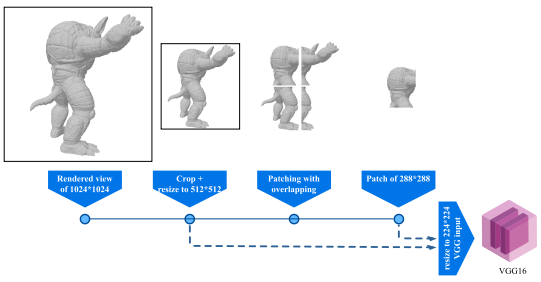

Fig. 3. Transformation Process: From View to Patches.

C. Learning and regression

Once the database B_k of rendered images is constructed, our goal is to score the quality of each mesh M_i from its rendered images I_j^i. To do so, features are extracted from the rendered images using a pre-trained VGG16 convolutional network (after resizing the images to 224×224). This network was considered as being the most efficient feature extractor with respect to other models such as AlexNet and ResNet [12]. The flowchart of our proposed approach is presented in Figure 4. Each image is fed into the pre-trained VGG16 model that acts as a feature extractor ϕ by saving its output before its dense layers. Consequently, a feature vector of size $7 \times 7 \times 512$ is obtained, then flattened in order to obtain a vector of size 25088. This feature vector $\phi(I_j^i)$ is used as an input to a shallow MLP to perform the regression task of quality assessment. The goal is to score the quality $\text{PMOS}_k(I_j^i)$ of an image I_j^i from a database B_k based on its corresponding feature vector $\phi(I_j^i)$. During training, the reference MOS for each image I_j^i corresponds to the 3D mesh M_i. Obviously, $\text{PMOS}_k(I_j^i)$ should be close to $\text{MOS}(M_i)$. The shallow MLP contains a single hidden layer of 512 neurons with a ReLu activation function, and a single neuron output layer. After the dense layer, a dropout layer with a rate of 0.5 is added to prevent over-fitting. In order to initialize the weights of the dense layer, we employed the Glorot uniform initialization method [13]. This method initializes the weights from a uniform distribution within a specific range, which is determined by the number of input and output neurons of the layer.

To predict the quality score of an input image, the MLP is trained on either B_{view} or B_{patch} (next section will present the training/evaluation protocol). Once the training has been performed, the quality estimation of a mesh M_i is computed as:

$$\text{PMOS}(M_i) = \frac{1}{n_k} \sum_{j=1}^{n_k} \text{PMOS}_k(I_j^i) \tag{1}$$

where $k \in \{\text{view}, \text{patch}\}$, and I_j^i is the j^{th} patch or view among the n_k images of the mesh M_i ($n_{\text{view}} = 11$ and $n_{\text{patch}} = 44$). The quality of a mesh is therefore the average of the predictions $\text{PMOS}_k(I_j^i)$ obtained for all its rendered images.

III. Experiments

A. Database

In this work, we have considered the LIRIS/EPFL General-Purpose database [1], which is a widely used dataset in the field of MVQA. This database contains 88 mesh models divided into 4 reference meshes (named Dinosaur, Armadillo, RockerArm and Venus) and 84 distorted versions. With this database we therefore have $N = 88$ meshes. There are 21 distortions for each reference mesh model. These distortions are generated through the application of smoothing and noise addition techniques on either smooth areas, rough areas or intermediate areas (between rough and smooth regions). A MOS (Mean Opinion Score) has been obtained for each model by averaging the subjective scores obtained from 12 human observers. The score range is between 0 (for good quality) and 10 (for bad quality). Figure 5 illustrates the reference Dinosaur mesh on the left and its corresponding distorted mesh with noise added on rough and intermediate areas.

B. Evaluation protocol

As presented in the previous Section, we can learn to predict the quality of a mesh M_i from its rendered images I_j^i that come either from B_{view} or B_{patch}. This means that we can predict the quality of a mesh either from 11 2D views or 44 2D views' patches. We will compare these two methodologies as well as with the state of the art. The following subsections describe the evaluation protocol we used.

To evaluate the quality assessment performance, we consider two standard measures: the Spearman Rank-Order cOrrelation Coefficient $SROOC$ and the Pearson Linear Correlation Coefficient $PLCC$. These metrics are commonly used in the field of visual quality assessment to measure the agreement and similarity between predicted scores and ground truth values. They will serve as a basis for comparing the performance of our proposed method with existing no-reference mesh quality assessment metrics. By quantifying the relationship between the predicted and actual scores or rankings, these coefficients provide valuable insights on the performance of prediction models.

The $SROOC$ coefficient (r_s) is a statistical measure used to assess the strength and direction of the monotonic relationship between the predicted quality scores $\text{PMOS}(M_i)$ and the reference mean opinion scores $\text{MOS}(M_i)$.

On the other hand, the $PLCC$ metric r_p assesses the linear relationship or correlation between the predicted scores and the ground truth values. Both metrics range from -1 to 1. A value of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. By using these metrics, we will be able to compare the performance of the patch-based and view-based methodologies, as well as benchmark them against state-of-the-art approaches.

C. Base Model

We initiated our experiments with the MLP Regression (MLPR) model presented in the previous section and depicted in Figure 4. The model was trained for a fixed number of 20 epochs using the RMSprop optimizer with a learning rate fixed to $1E{-}3$. Based on extensive tests, we have found that the best correlation scores are obtained with a batch size of one-third of the training set size. This parameter will be fixed this way for all the trials. In order to evaluation the model accuracy, we perform the Leave-One-Mesh-Out Cross-Validation procedure (LOMO-CV). At training, all the meshes are considered except one mesh and its distorted version.

Figure 6 presents the training and testing process for each trained MLPR using LOMO-CV. In this approach, the images associated to a specific mesh are all excluded (*i.e.*, 22 meshes) from the training process. The trained neural network is then

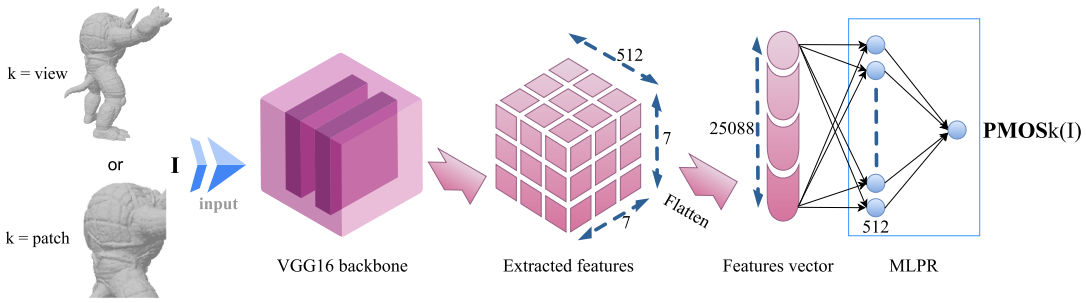

Fig. 4. The pipeline of the proposed quality assessment index that estimate the quality of a rendered image (2D view or 2D view patch).

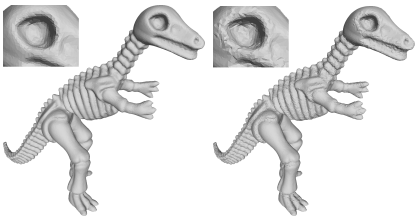

Fig. 5. Dinosaur reference 3D mesh (left) and its distorted version (right) with noise addition on rough and intermediate areas. The top-left subfigure presents a cropped and zoomed area.

tested on these excluded images to evaluate its performance as they represent unseen data. This LOMO-CV process is repeated for each 22 meshes (a reference mesh and its distorted versions) in the dataset. By employing cross-validation, we ensure an objective assessment of the MLPR model as it is evaluated on strictly independent data that they have not been trained on. For one fold, we obtain quality predictions $\text{PMOS}_k(I_j^i)$ for all the images of this fold. Therefore, we can evaluate the results directly at the image (2D view or 2D view patch) level or at the mesh level. We will differentiate these two different evaluation ways by the names **concat** and **average**. Indeed, as shown in Equation 1, to obtain the quality of a given mesh M_i we have to average all the predictions from its rendered views or patches. To achieve this, we group the prediction scores by mesh and compute the average score for each. This aggregation process allows us to consolidate the information from multiple predictions into a single score that reflects the overall quality of the mesh. By transitioning from the concatenation scores to the average scores by mesh, we obtained a more focused and interpretable evaluation of the quality of each individual mesh in our dataset. This approach provided a more fine-grained understanding of the performance of our model on a per-mesh basis.

Table I presents the results of the base model (with 20 epochs) in term of SROOC values. We can remark that the Spearman correlation values for ($\mathbf{r_s}$ concat) are relatively low on both databases $\mathbf{B}_{\text{view}}$ and $\mathbf{B}_{\text{patch}}$, especially for the Dinosaur mesh and its distorted versions, where the prediction can be much enhanced by averaging the prediction of all the patches. However, when we aggregate the predictions on an average basis per mesh, it provides better interpretability and aligns with our ultimate goal of evaluating mesh quality. It is logical that the patch protocol yields better results since the dataset is larger . The only exception is the Venus model, which shows a slightly better correlation with the view-based protocol.

	On $\mathbf{B}_{\text{view}}$		On $\mathbf{B}_{\text{patch}}$	
Configuration	$\mathbf{r_s}$ **concat**	$\mathbf{r_s}$ **average**	$\mathbf{r_s}$ **concat**	$\mathbf{r_s}$ **average**
Armadillo out	**0.704**	0.832	0.657	**0.949**
Dinosaur out	0.01	0.292	**0.253**	**0.779**
Venus out	0.895	**0.953**	**0.896**	0.942
RockerArm out	0.583	0.929	**0.727**	**0.949**
Average	0.548	0.751	**0.633**	**0.904**

TABLE I
SROOC VALUES FOR THE BASE MODEL TRAINED FOR 20 EPOCHS ON B_{VIEW} OR B_{PATCH} DATABASES.

In order to enhance the results, we employ in our implementation the Early Stopping technique which is a form of regularization that helps prevent overfitting and improves generalization of the trained model. This also allows us to identify the optimal number of epochs that may lead to better results. For each methodology (view-based or patch-based), the optimal number of epochs is fixed after a patience delay that determines the number of epochs to wait before stopping the training process if the monitored metric does not improve. Results are presented in Table II. This enables to drastically boost the results. If we compare our results with [12] that also considers VGG16 for feature extraction but with patches of very small size of 32×32 and a fixed number of epochs of 40, they obtained an average $\mathbf{r_s}$ of 0.925 while our approach achieves 0.945. This shows that the minimization process can be prone to local minima and early stopping combined to a patience delay can help to cope with. Larger patches also help to enhance the results as they capture more details.

	On $\mathbf{B}_{\text{view}}$			On $\mathbf{B}_{\text{patch}}$		
	Early stopping (patience=30)			Early stopping (patience=60)		
Configuration	**epochs**	$\mathbf{r_s}$ **concat**	$\mathbf{r_s}$ **average**	**epochs**	$\mathbf{r_s}$ **concat**	$\mathbf{r_s}$ **average**
Armadillo out	77	0.815	0.963	307	**0.939**	**0.989**
Dyno out	211	0.142	0.789	680	**0.189**	**0.814**
Venus out	71	0.933	0.984	311	**0.971**	**0.995**
RockerArm out	110	**0.957**	0.962	400	0.836	**0.981**
Average	–	0.712	0.924	–	**0.734**	**0.945**

TABLE II
SROOC VALUES FOR THE BASE MODEL TRAINED WITH AN EARLY STOPPING ON B_{VIEW} AND B_{PATCH} DATABASES.

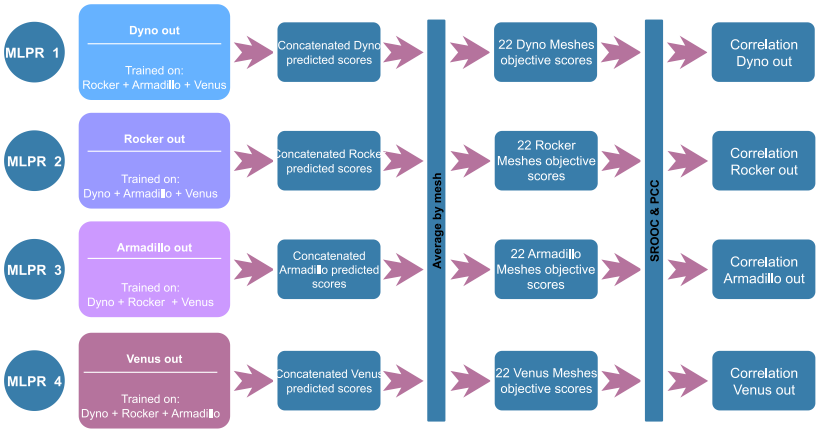

Fig. 6. Resulting neural networks issued from the leave-one-mesh-out cross-validation (LOMO-CV).

D. Cumulative Model

So far, we have shown in the previous section that our approach is competitive with similar approaches of the state-of-the art (in particular with [12] but more comparisons will be shown in the next section). As we have used a LOMO-CV training, we are able to measure how good our proposed approach is. However, we are not able to use a resulting network to evaluate the quality of new unseen 3D meshes, since four MLPRs were trained to predict the objective quality scores for each fold. In this section, we propose a novel learning strategy that allows to obtain a single neural network that can be used to assess visual quality of 3D meshes that don't belong to the LIRIS/EPFL General-Purpose database and that can go beyond the performances of the models obtained by LOMO-CV. To do so, we consider a cumulative training of which the principle is depicted in Figure 7. This learning strategy begin by training and testing the MLPR model with Glorot initializat on the first fold for 1000 epochs. Then, the same MLPR is used for training on the next fold and so on. As a consequence, the final cumulative MLPR can be used to perform future predictions on unseen data and this also helps in improving its accuracy. However if we measure the performances of this final model, this obviously leads to overestimated results as this cumulative training was gradually trained over the entire dataset. To mitigate this effect and better evaluate the performances of the Cumulative Model (CM), we re-train it using LOMO-CV in order to obtain a final Retrained Cumulative Model (RCM). To do so, on each fold, a new MLPR is initialized with the weights of the CM and is trained for a fixed number of epochs. The latter is determined by finding the global minimum of the average of the losses of all folds during the cumulative training. Table III presents the results of RCM in term of Spearman correlation. As expected, the CM model performs better than the Base model we have developed in the previous section. Refitting it enables to have a better evaluation of its generalization abilities.

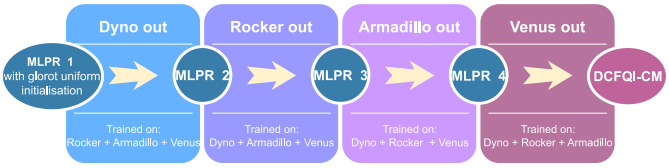

Fig. 7. Cumulative Model Training.

	On $\mathbf{B}_{\text{view}}$				On $\mathbf{B}_{\text{patch}}$			
	CM		RCM		CM		RCM	
Configuration	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$
Armadillo out	0.993	0.999	0.992	0.998	0.998	0.999	**0.997**	**0.999**
Dyno out	0.999	0.998	0.995	**0.998**	0.998	0.999	**0.998**	**0.998**
Venus out	0.999	0.999	0.997	0.998	1	1	**0.999**	**0.999**
RockerArm out	0.986	0.995	0.986	0.992	0.994	0.997	**0.991**	**0.996**
Average	0.994	0.997	0.992	0.996	0.997	0.998	**0.996**	**0.998**

TABLE III
SROOC AND PCC CUMULATIVE MODEL (CM) VS RE-TRAINED CUMULATIVE MODEL (RCM) RESULTS TRAINED ON DATABASES B_{VIEW} OR B_{PATCH}.

E. Comparison with the state-of-the-art

There has been recently several proposed metrics for no-reference quality assessment of 3D meshes. In this Section we compared our approach with several existing state-of-the-art no-reference methods [8]–[10], [12], [14]–[18]. The correlation values $\mathbf{r_s}$ and $\mathbf{r_p}$ of our methods (base or cumulative models), as well as those of the existing methods, are presented in Table IV. We mention that the correlations in the "All Meshes" columns were calculated between the objective scores of all objects and their corresponding MOS. Our proposed methods demonstrate high correlation scores. Both our patch-based (DCFQI-PBM) and view-based (DCFQI-VBM) Base Model methods outperform CNNs-CMP [12] for the Armadillo, Venus, and RockerArm models. Moreover, they exhibit competitive correlations on the entire database, with correlation values of $rs = 92.4\%$ and $rp = 92.1\%$. CNNs-CMP [12] uses a complex approach which relies the combination of features from three pre-trained models (VGG/AlexNet/ResNet) combined with saliency patch-based selection. Our approach shows that using larger patches (or

		Armadillo		Dyno		Venus		RockerArm		All Meshes	
Type of method	**Metrics**	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$	$\mathbf{r_s}$	$\mathbf{r_p}$
Features based	BMQI [14]	20.1	-	83.5	-	88.9	-	92.7	-	78.1	-
	BMQA-GSES [15]	98.7	80	99.2	80.4	98.8	80.1	**99.5**	**99.9**	90.5	87.9
	NR-GRNN [16]	87.1	97.3	91.2	94.1	86.3	85	78.6	74.8	86.2	88.7
	MVQ-GCN [17]	91.8	92.5	87.7	84.5	93.7	91.9	89.6	88.4	89.3	88.6
	NR-CNN 1 [8]	87.2	84.3	86.4	86.2	92.2	85.6	91.3	85.2	83.6	82.7
	NR-SVR [18]	76.8	91.5	78.6	84.1	85.7	88.6	86.2	86.6	81.5 8	7.8
View based	DCFQI-VBM	96.3	98.2	78.9	89	98.4	99.5	96.2	95.7	90.4	89.9
	DCFQI-VCM	99.2	99.8	99.5	**99,8**	99.7	99.8	98.6	99.2	96.5	96.6
Patch based	CNN-BMQA [9]	89.8	91.4	91.6	92.2	94.6	93.8	91.9	93.9	90	92
	NR-CNN 2 [10]	93.4	95.6	86.2	84.3	94.1	90.3	80.4	82.2	81.7	82.5
	CNNs-CMP [12]	95.8	95.6	93.6	92.9	93.4	91.3	94.5	95.2	92.6	91.3
	DCFQI-PBM	98.9	98	81.4	98.1	99.5	99.7	98.1	97	92.4	92.1
	DCFQI-PCM	**99.7**	**99.9**	**99.8**	**99.8**	**99.9**	**99.9**	99.1	99.6	**99.1**	**99.4**

TABLE IV
COMPARISON OF OUR PROPOSED DCFQI VIEW/PATCH BASED BASE AND CUMULATIVE MODELS (DCFQI-VBM & DCFQI-VCM / DCFQI-PBM & DCFQI-PCM RESPECTIVELY) WITH THE STATE OF THE ART NO-REFERENCE METRICS.

even views) with a careful optimization can be as effective. Finally, both our proposed patch-based (DCFQI-PCM) and view-based (DCFQI-VCM) Cumulative Models surpass the state-of-the-art techniques for all meshes, except for the RockerArm model, where we slightly fall short behind BMQA-GSES [16] method. Nevertheless, the overall performance of our methods is superior to the latter.

IV. CONCLUSION

In this paper, we presented a no-reference mesh quality assessment approach. It renders the mesh in 2D views that can be subsequently divided in patches. From these images, deep features are extracted by the pre-trained VGG16 CNN and fed into a MLP that performs quality prediction. This base model is competitive with the state-of-the-art, even at the view level. Finally a cumulative training has been proposed to obtain a single final model for prediction that goes beyond the state-of-the-art. Future works will consider combining both view and patch predictions.

REFERENCES

[1] G. Lavoué, E. D. Gelasca, F. Dupont, A. Baskurt, and T. Ebrahimi, "Perceptually driven 3d distance metrics with application to watermarking," in *Applications of Digital Image Processing XXIX*, vol. 6312. SPIE, 2006, p. 63120L.

[2] N. Aspert, D. S. Cruz, and T. Ebrahimi, "MESH: measuring errors between surfaces using the hausdorff distance," in *ICME*, 2002, pp. 705–708. [Online]. Available: https://doi.org/10.1109/ICME.2002.1035879

[3] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Comput. Graph. Forum*, vol. 17, no. 2, pp. 167–174, 1998. [Online]. Available: https://doi.org/10.1111/1467-8659.00236

[4] G. Lavoué, "A multiscale metric for 3d mesh visual quality assessment," *Comput. Graph. Forum*, vol. 30, no. 5, pp. 1427–1437, 2011. [Online]. Available: https://doi.org/10.1111/j.1467-8659.2011.02017.x

[5] L. Vása and J. Rus, "Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes," *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1715–1724, 2012. [Online]. Available: https://doi.org/10.1111/j.1467-8659.2012.03176.x

[6] M. Corsini, E. D. Gelasca, T. Ebrahimi, and M. Barni, "Watermarked 3-d mesh quality assessment," *IEEE Trans. Multim.*, vol. 9, no. 2, pp. 247–256, 2007. [Online]. Available: https://doi.org/10.1109/TMM.2006.886261

[7] K. Wang, F. Torkhani, and A. Montanvert, "A fast roughness-based approach to the assessment of 3d mesh visual quality," *Comput. Graph.*, vol. 36, no. 7, pp. 808–818, 2012. [Online]. Available: https://doi.org/10.1016/j.cag.2012.06.004

[8] I. Abouelaziz, M. E. Hassouni, and H. Cherifi, "A convolutional neural network framework for blind mesh visual quality assessment," in *ICIP*, 2017, pp. 755–759. [Online]. Available: https://doi.org/10.1109/ICIP.2017.8296382

[9] I. Abouelaziz, A. Chetouani, M. E. Hassouni, L. J. Latecki, and H. Cherifi, "3d visual saliency and convolutional neural network for blind mesh quality assessment," *Neural Comput. Appl.*, vol. 32, no. 21, pp. 16 589–16 603, 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04521-1

[10] I. Abouelaziz, A. Chetouani, M. E. Hassouni, and H. Cherifi, "A blind mesh visual quality assessment method based on convolutional neural network," in *3DIPM*. Ingenta, 2018. [Online]. Available: https://doi.org/10.2352/ISSN.2470-1173.2018.18.3DIPM-423

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[12] I. Abouelaziz, A. Chetouani, M. E. Hassouni, L. J. Latecki, and H. Cherifi, "No-reference mesh visual quality assessment via ensemble of convolutional neural networks and compact multi-linear pooling," *Pattern Recognit.*, vol. 100, p. 107174, 2020. [Online]. Available: https://doi.org/10.1016/j.patcog.2019.107174

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, vol. 9, 2010, pp. 249–256.

[14] A. Nouri, C. Charrier, and O. Lézoray, "3d blind mesh quality assessment index," in *3DIPM*, 2017, pp. 9–16. [Online]. Available: https://doi.org/10.2352/ISSN.2470-1173.2017.20.3DIPM-002

[15] Y. Lin, M. Yu, K. Chen, G. Jiang, F. Chen, and Z. Peng, "Blind mesh assessment based on graph spectral entropy and spatial features," *Entropy*, vol. 22, no. 2, p. 190, 2020. [Online]. Available: https://doi.org/10.3390/e22020190

[16] I. Abouelaziz, M. E. Hassouni, and H. Cherifi, "A curvature based method for blind mesh visual quality assessment using a general regression neural network," in *SITIS*, 2016, pp. 793–797. [Online]. Available: https://doi.org/10.1109/SITIS.2016.130

[17] I. Abouelaziz, A. Chetouani, M. E. Hassouni, H. Cherifi, and L. J. Latecki, "Learning graph convolutional network for blind mesh visual quality assessment," *IEEE Access*, vol. 9, pp. 108 200–108 211, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3094663

[18] I. Abouelaziz, M. E. Hassimosouni, and H. Cherifi, "No-reference 3d mesh quality assessment based on dihedral angles model and support vector regression," in *ICISP*, vol. LNCS 9680, 2016, pp. 369–377. [Online]. Available: https://doi.org/10.1007/978-3-319-33618-3_37