Learning graph convolutional network for blind mesh visual quality assessment

ILYASS ABOUELAZIZ, ALADINE CHETOUANI, MOHAMMED EL HASSOUNI, HOCINE CHERIFI, LONGIN JAN LATECKI

1. INTRODUCTION

RECENTLY, the issue of perceptual 3D MVQA has become an essential field of study since 3D models are widely used in a diversity of applications. A 3D object can undergo several geometric transformations resulting in various distortion types that affect the object’s perceived quality. The perceived visual quality can be estimated by human observers (subjective visual quality assessment) or by implementing automatic metrics that try to mimic an ideal human observer (objective visual quality assessment) [20]–[23]. One can classify objective methods according to the availability of the reference object. In Full-Reference methods (FR), the original mesh is known [3]–[7]. In No-Reference (NR) or Blind methods, no information about the reference is available [8]–[12]. Finally, in Reduced-Reference (RR) methods only part of the reference is available (i.e., features extracted from the reference) [13]–[17].

Most quality evaluation methods rely on the quality prediction technical tool and the feature type such as geometric (angles, curvature), topological (mesh connection), spatial (vertices and edges), frequency (Wavelet Transform [66], Discrete cosine transform [67]), spectral (Laplace-Beltrami operator [68]), etc.

One can distinguish two main categories of methods for mesh quality assessment. The first category is based on logistic regression to estimate the quality scores. A plethora of such methods have been proposed in the literature and have shown good performances [18]–[20], [25], [27]–[30]. The second category relies on machine learning to predict quality [31], [32]. More precisely, quality score estimation is tackled into a classification context, or eventually, regression or regression by classification. In a previous work, we have been the first to investigate Convolutional Neural Network (CNN) as one of the most recent and relevant deep learning tools to estimate the perceived visual quality of 3D meshes [33].

Intuitively, to handle simply CNN in our context, 2D images are rendered from multiple views of the 3D mesh. Then, each image is split into small patches, which are learned to a convolutional neural network. As done for 2D images...
In [24], 3D visual saliency is adopted in [62] to select the most relevant regions with high saliency level before using patches from these regions to feed the CNN. This method shows good performances and successfully predicts the perceived visual quality. However, the perceptual mechanisms applied to the images generated from the 3D data make the evaluation viewpoint-dependent. As demonstrated by Rogowitz and Rushmeier [35], the main problem of the image-based metrics is that, in general, the perceived degradation of still images may not be adequate to evaluate the perceived degradation of the equivalent 3D model. Recently, a new blind method using a set of spatial and spectral features has been proposed by Lin et al. [37]. Inspired by Graph Signal Processing (GSP) theory, features are extracted from the Graph Fourier Transform (GFT) derived from the adjacency matrix, and combined with the spatial features. The quality score prediction is estimated by random forest regression. The method achieves good performances and competitive quality scores. Unfortunately, it relies only on spatial and spectral features. It does not incorporate perceptual information, which is an essential feature to assess the mesh visual quality. However, several transforms can be adopted instead of GFT, or to get straight to the point, graph-based deep learning solutions can be a new lead in 3D MVQA.

In this context, we propose here a model-based method using a shallow Graph Convolutional Network (GCN) to process the 3D model itself directly. GCN has been mainly applied for node classification tasks in which the convolution representation vector for a node function is the only feature to classify that node. For the mesh quality assessment task, the 3D model is represented by a graph, and the graph-based convolution vector of nodes can be used to predict the perceived visual quality. Although graph and hypergraph representations have been widely used in image and video processing [38]–[44]. To the best of our knowledge, it is the first work where GCN is used to estimate the quality of 3D meshes.

Our contributions are summarized as follows:

- Development of a model-based method to blindly assess the perceived visual quality of a distorted mesh.
- Use a shallow GCN model that predicts the quality from a graph representation of the 3D mesh and handcrafted features.
- In-depth analysis of the performance contribution of the considered handcrafted features, including visual saliency features.

The remainder of this paper is organized as follows: In Section II, we introduce the required important notions. We give in Section III a detailed description of the proposed method using the GCN. In Section IV, we present the experimental setup. Results of a comparative evaluation are presented in Section V. Finally, we conclude and give some perspectives in Section VI.

II. BACKGROUND

In this section, we provide a brief introduction to the required background including the CNNs and essential notions about graph theory.

A. GRAPH

- **Graphs**: A graph \( G(V, E) \) is defined by a set of vertices \( V = \{v_1, v_2, ..., v_n\} \) and a set of edges \( E \subseteq V \times V \). We denote \( n \) and \( m \) the number of vertices and edges, respectively.

- **Adjacency matrix**: The adjacency matrix \( A \) of size \( n \times n \) representing a graph is defined as \( A_{i,j} = 1 \) if the vertices \( v_i \) and \( v_j \) are connected by an edge (i.e. adjacent vertices), otherwise \( A_{i,j} = 0 \). Each node and edge may have attribute values which are considered as features of the graph. The term attribute value is used instead of label to make the distinction with the concept of labeling in graph-theory. A walk is a sequence of nodes in a graph, in which consecutive nodes are connected by an edge. A path is a walk with distinct nodes. We denote \( d(u,v) \) the distance between \( u \) and \( v \), that is, the length of the shortest path between \( u \) and \( v \). \( N_k(v) \) is the 1-neighborhood of a node, that is, all nodes that are adjacent to \( v \).

- **Graph Laplacian**: The graph Laplacian operator \( \mathcal{L} \) is defined as \( \mathcal{L} = \mathcal{D} - \mathcal{A} \), where \( \mathcal{A} \) is the adjacency matrix and \( \mathcal{D} \) is the degree matrix with \( \mathcal{D}_{i,i} = \sum_j A_{i,j} \). For a graph ascending from a regular mesh, the graph Laplacian corresponds to the standard stencil approximation of the continuous Laplacian. A normalized form of the Laplacian is used and is defined as follows:

\[
\mathcal{L}^{\text{norm}} = D^{-1/2} \mathcal{L} D^{-1/2} = I - D^{-1/2} \mathcal{A} D^{-1/2}
\]

It is noteworthy that the eigenvectors of \( \mathcal{L} \) and \( \mathcal{L}^{\text{norm}} \) are not similar since the two matrices are different.

B. FROM CONVOLUTIONAL NEURAL NETWORKS TO GRAPH CONVOLUTIONAL NETWORKS

Convolutional neural networks offer an efficient architecture to extract significant statistical patterns in large-scale and high-dimensional datasets. The ability of CNNs to learn local stationary structures and compose them to form multi-scale hierarchical patterns has led to breakthroughs in image, video, and sound recognition tasks [48]. Precisely, CNNs extract the local stationarity property of the input data or signals by revealing local features shared across the data domain. These similar features are identified with localized convolutional filters or kernels, which are learned from the data. Convolutional filters are shift- or translation-invariant filters. In other words, they are able to recognize identical features independently of their spatial locations. Localized kernels or compactly supported filters refer to filters that extract local features independently of the input data size, with a support size. The latter can be much smaller than the input size. CNNs have recently attracted the attention of...
many researchers. They have been successfully employed in various computer vision applications allowing them to reach high performances. One of their main advantages over classical neural networks is that they adequately consider the input data’s spatial structure. Moreover, CNNs allow the critical property of weights sharing between the convolutional layers, which restricts the number of parameters to learn. In quality assessment, using CNNs has shown notable improvement in correlation with human judgment. Some datasets can naturally be modeled as scalar data defined on the vertices of graphs [36]. For example, computer networks, transportation (road, rail, airplane) networks, or social networks can be described by graphs, with the vertices corresponding to individual computers, cities, or people, respectively. These data can be structured with graphs, which are universal representations of heterogeneous pairwise relationships. Graphs can encode complex geometric structures and be studied with strong mathematical tools such as spectral graph theory [50]. A generalization of CNNs to graphs is not straightforward as the convolution and pooling operators are only defined for regular grids. It makes the extension a challenging issue, theoretically and implementation-wise. The major bottleneck of generalizing CNNs to graphs is the definition of localized graph filters, which are efficient to evaluate and learn.

### A. LEARNING DATA PREPARATION

The first step of the proposed method relies on preparing the learning data. It consists of a graph represented by an adjacency matrix $A$ of size $n \times n$ and a $n \times p$ feature matrix, with $p$ is the number of features per node.

1) **Graph network representation and adjacency matrix**

The proposed framework’s main idea is to use graphs to represent the 3D model (Triangular mesh). This representation allows us to take advantage of graph properties to manipulate the mesh itself and conceive a model-based method to assess the visual quality blindly. Triangular meshes are a set of nodes connected by edges to form triangular faces. A graph is a data structure consisting of vertices and edges. Each vertex is linked to other vertices by an edge, so-called neighbors. There are different ways to represent a graph. In this work, it is characterized by an adjacency matrix as discussed in Sec. II-A. Fig. 2 depicts the graph representation of a 3D mesh and its corresponding adjacency matrix.

2) **Handcrafted features**

We use three geometric features: curvature, angles, and Laplacian of Gaussian curvature and one perceptual feature: Saliency.

- **Curvature** describes the deviation of the surface from being flat. In this work, we use the maximum curvature amplitude $C_{\text{max}}$ and the minimum curvature amplitude $C_{\text{min}}$. The mean curvature $C_{\text{mean}} = (C_{\text{max}} + C_{\text{min}})/2$ as well as the Gaussian curvature as $C_{\text{gauss}} = C_{\text{max}} \times$...
$C_{\text{min}}$ are also computed. In the next, all the curvature features will be considered as one feature labeled by $C$. Laplacian of Gaussian curvature corresponds to the Laplacian operator applied to the Gaussian curvature field. This feature is labeled here as $\text{LoG}$. Angle is computed for each edge and corresponds to the angle between normal vectors of its adjacent faces. It represents the structural aspect of the mesh. The computed angle values are averaged to obtain a scalar value for each vertex. This feature is labeled here as $A$. Saliency is a perceptual concept that describes the attention of the Human Visual System to some regions due to their specificity’s (curvature, orientation, and so on). In this work, we use the mesh visual saliency method proposed in [61]. This feature is labeled here as $S$.

Fig. 3 shows the used features described above. Mean and Gaussian curvature are depicted separately to give more visibility about the difference between them.

A multi-layer GCN is used with the layer-wise propagation rule defined as:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-1/2} \widetilde{A} \widetilde{D}^{-1/2} H^{(l)} W^{(l)})$$

with

$$\widetilde{A} = A + I_N$$

and

$$\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$$

and $\sigma(.)$ is the activation function defined as $\text{ReLU}(.) = \max(0, .)$. $\widetilde{A}$ is the adjacency matrix of the undirected graph. $I_N$ is the identity matrix. $W^{(l)}$ is a trainable weight matrix. $H^{(l)} \in \mathbb{R}^{N \times D}$ is the matrix of activations in the $l$th layer; $H^{(0)} = X$. This rule is motivated by an approximation presented in [60].

We note that the multiplication with $A$ means that, for every node, we sum up all the feature vectors of all neighboring nodes but not the node itself (unless there are self-loops in the graph). This is fixed by enforcing self-loops in the graph by considering $A$ (adding the identity matrix to $A$).

1) Graph convolution

Let $G = \{V, E\}$ denotes the graph representing the distorted mesh. $v_i \in V$ and $(v_i, v_j) \in E$ are the sets of nodes and edges of $G$, respectively. The graph convolution is the process of filtering a signal $x \in \mathbb{R}^N$ with a filter $f_\theta$ defined as follows:

$$Y = f_\theta \ast x = \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x$$

Using a deep model, the operator $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ may cause numerical instabilities. Thus, a normalization is introduced

$$I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$$

Finally, the generalized graph convolution is defined as follows:

$$Y = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X \Theta$$

where $X \in \mathbb{R}^{N \times P}$ is the feature matrix of size $N \times P$ with $N$ the number of nodes and $P$ the number of features for each node. We denote $F$ the number of filters, $\Theta \in \mathbb{R}^{P \times F}$ is a

![Figure 2](image2.png)

**FIGURE 2.** Graph representation and adjacency matrix from 3D object.

![Figure 3](image3.png)

**FIGURE 3.** Handcrafted features preparation.

### B. GRAPH CONVOLUTIONAL NETWORK

In this section, we provide a detailed description of the GCN implemented in the proposed method $f(X, A)$, where $A$ is an adjacency matrix and $X$ is a feature matrix.
matrix of filter parameters and $Y \in \mathbb{R}^{N \times F}$ is the convolved matrix.

Convolution is a key process in a graph convolutional network, allowing to highlight local receptive features. When applying convolution on graphs, two important steps are considered:

- Determine the neighboring nodes around each given graph node for the convolution process to build locally connected subsets of the global graph. These subsets can be obtained using search strategies. We use the breadth-first search to enlarge the neighborhood nodes on the graph. The breadth-first search crosses an entire level of child nodes first, before passing through the grandchildren nodes. In addition, it goes through a whole level of grandchildren nodes before passing through great grandchildren nodes and so on...

- Arrange the order of execution of the convolution process on each set. Indeed, it is crucial to reasonably sort the nodes in the subsets to ensure that the same rules are used to convolve the elements thus convolution can better activate features for each node. To this end, the L2 similarity method is used. To do so, we use nodes features to compute the similarity between them. The neighbor node with the higher level of similarity is executed first. For example, in the convolution order provided in Fig. 4(c), node 3 is the most similar to 6, then, we find node 9, and finally node 1 is the least similar. We can denote this by: $\text{sim}(3, 6) > \text{sim}(9, 6) > \text{sim}(9, 1)$. Thus, the order of execution in the first subset is 3-9-1. This process is repeated for each subset in the graph. Fig. 4(c) shows the final convolution order for each subset.

![FIGURE 4. Convolution on graph. (a) 3D object represented by a graph; (b) subsets searched by a breadth-first search; (c) Convolution order.](image)

2) Graph coarsening and pooling

The pooling operation requires meaningful neighborhoods on graphs, where similar vertices are clustered together. Doing this for multiple layers is equivalent to a multi-scale clustering of the graph that preserves local geometric structures. There exist many clustering techniques, we are interested here in multilevel clustering algorithms where each level produces a coarser graph. In this work, we make use of the coarsening phase of the Graclus multilevel clustering algorithm [45]. It uses a greedy algorithm to compute successive coarser versions of a given graph.

For each coarsening level Fig. 5(a), the algorithm picks and matches an unmarked vertex $v_i$ with one of its unmarked neighbors $v_j$. These latter are chosen because this maximizes the local normalized cut defined as $A_{ij}(1/d_i + 1/d_j)$. $A$ is the adjacency matrix, $d_i$ and $d_j$ are the weights corresponding to the vertices $v_i$ and $v_j$ respectively. After that, the vertices are marked, and the coarsened weights correspond to the sum of their weights. This operation is performed for all the nodes in the graph. The coarsening operation allows dividing the number of nodes by 2. As a result, the coarsened vertices and the original vertices are not arranged. However, it is possible to arrange the vertices, so the graph pooling operation becomes efficient. We note that, for a pooling of size 4, two coarsening levels of size 2 are needed. Fig. 5 illustrates an example of graph pooling.

After coarsening, each node has either two children, if it was matched at the finer level, or one, if it was not, i.e. the node was a singleton. From the coarsest to finest level, fake nodes, i.e. disconnected nodes, are added to pair with the singletons such that each node has two children. Let us carry out a max-pooling of size 4 on the finest $G_0$ graph given as input of size $n_0 = 12$. Two coarsening levels of size 2 are needed: let Graclus gives $G_1$ of size $n_1 = 6$ and $G_2$ of size $n_2 = 3$ the coarsest graph. Fake nodes (in red) are added to $G_1$ (1 node) and $G_2$ (4 nodes) to pair with the singletons (in yellow), such that each node has exactly two children.

![FIGURE 5. Graph coarsening and pooling.](image)

3) Node classification

In the following, we consider a two-layer GCN for node classification on a graph with a symmetric adjacency matrix $A$. We first calculate $\hat{A} = D^{-1/2} A D^{-1/2}$ in a pre-processing step. The forward model then takes the simple form:

$$Z = f(X, A) = \text{softmax} \left( \hat{A} \text{ReLU} \left( \hat{A} X W^0 \right) W^1 \right)$$  \hspace{1cm} (8)
Here, \( W^{(0)} \in \mathbb{R}^{P \times H} \) is an input-to-hidden weight matrix for a hidden layer with \( H \) feature maps. \( W^{(1)} \in \mathbb{R}^{H \times F} \) is a hidden-to-output weight matrix. The softmax activation function, defined as:

\[
\text{softmax}(x_i) = \frac{1}{z} \exp(x_i)
\]

with \( z = \sum \exp(x_i) \) is applied row-wise. For multiclass classification, we then evaluate the cross-entropy error over all labeled examples:

\[
L = \sum_{l \in Y_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}
\]

where \( Y_L \) is the set of node indices that have labels, and \( F \) is the number of filters. \( Z_{lf} \) is the result of the model for each node \( l \) and each filter \( f \).

The neural network weights \( W^{(0)} \) and \( W^{(1)} \) are trained using gradient descent.

C. TRAINING AND QUALITY SCORE ESTIMATION

To estimate the visual quality of a distorted mesh, we rely on a classification process. We consider five classes based on the subjective Mean Opinion Score (MOS) for each database (very bad, bad, medium, good, and excellent quality).

The training process follows the leave-one-out cross-validation (LOOCV) according to the following process:

- We build a training regression model using all the existing 3D objects in the repository except one object and its distorted versions.
- The excluded subset is then used for the test using the constructed training model.
- The process is repeated for each 3D object in the repository.

It is noteworthy that we generate one training model for each excluded object. In addition, for the cross-dataset evaluation (Section V-C) we train one model using one database and the other databases are used for the test.

The negative log-likelihood is minimized over the training set using the Stochastic Gradient Descent (SGD). The back-propagation algorithm is used to compute the gradients. At the same time, dropout is employed to avoid over-fitting. For the test phase, the excluded objects serve as a test set, and the trained network classifies the given distorted meshes according to their visual quality. Finally, comparisons with the subjective labels are performed using correlation measurements.

IV. EXPERIMENTAL SETUP

In the following, we describe the experimental protocol, including the evaluation criteria. We also report a brief description of the databases.

A. DATASETS

The goal of MVQA is to provide quality predictions correlated with human observer’s opinions. Therefore, a dataset of distorted meshes graded by human observers is needed to evaluate the algorithms. As the meshes geometric aspect significantly influences the evaluation process, care must be taken when choosing the dataset. It must contain meshes that reflect adequate diversity in their content, and generated distortions should reflect a broad range of mesh degradation.

To comply with this argument, four datasets are used in the experiments. These databases, specially designed for quality metrics evaluation, are made of original and distorted mesh. Furthermore, the MOS values are provided for each dataset. Fig. 6 shows the reference objects of the four databases briefly described below.

- LIRIS/EPFL General-Purpose database [46]: This database, created at the Ecole Polytechnique Federale de Lausanne, contains four reference meshes and 84 distorted models (88 models in total). Two types of distortion are applied, smoothing and noise addition, either locally or globally, on the reference mesh.
- LIRIS Masking database [47]: This database, originally from the University of Lyon, consists of four reference meshes and 24 distorted models with local noise addition. This database’s specific objective is to test the ability of MVQA methods in capturing the visual masking effect.
- UWB compression database [2] includes five reference models and 63 distorted models. From Twelve to thirteen distorted versions, obtained by a compression algorithm, are associated with each reference model.
- The IEETA simplification database [59] comprises 35 models: five reference meshes and six simplified models for each reference mesh. The simplified models are obtained using three simplification algorithms with two different vertex reduction ratios.

FIGURE 6. The reference models from: the LIRIS masking database (a), the general-purpose database (b) and the UWB compression database (c) and the IEETA simplification database (d).
Even though some reference objects are the same in some databases, their distortions are completely different regarding the type and the level of distortions, which provides an important diversity to evaluate MVQ assessment methods. For instance, we have two similar objects (Armadillo and Dyno) in the LIRIS masking database Fig. 6 (a) and the general-purpose database Fig. 6 (b). In the first database, distortions are obtained by adding only local noise. However, in the second database, distortions are obtained by smoothing and adding global noise with different levels.

V. EXPERIMENTAL RESULTS

Here, we report the main results on the influence of the mesh features on objective quality effectiveness. Then, we present the results of a comparative evaluation with the state-of-the-art methods using four subjective databases. Finally, a cross-database assessment is performed to test the generalization ability of the proposed method.

A. INFLUENCE OF THE GEOMETRIC ATTRIBUTES

In this section, we evaluate the impact on the performance of the extracted handcrafted features. Several sets used as input to the GCN have been tested. It should be noted that results only take into account three geometric features: curvature (including mean and Gaussian curvature), angles and Laplacian of Gaussian, and one perceptual attribute (saliency), more features have been investigated before finding the best combination. Here, we provide the results for different feature combinations. The features under test are minimum and maximum directions (Umin and Umax), curvature-based (Cmin, Cmax, Cmean, and Cgauss), Normal (Nor), Saliency (S), Angle (An), and Laplacian of Gaussian (LoG).

Table 1 illustrates the main findings. Curvature-based and angle features exhibit good correlation values. Adding the Saliency improves the performance considerably. Adding LoG leads to similar results. Unfortunately, adding the directions (Umin and Umax) and the normal (Nor) does not improve the results. The dihedral angle is the angle between normal vectors of two adjacent faces. Adding the information of normal vectors does not provide added value since it is already included in the dihedral angle. In the same vein, Umin and Umax present only the directions of minimum and maximum curvature which are already included in the curvature feature itself. Experimentally, adding these features leads only to redundant information without any added value and the training time increased remarkably. The best performance is obtained using the four features (C + Ang + LoG + S). Indeed, this leads us to the best correlation scores. Therefore, we only rely on Cmin, Cmax, Cmean, Cgauss, Ang, LoG, and S to construct the features matrix.

B. COMPARISON WITH THE STATE-OF-THE-ART

Here, a comparative analysis is conducted. The proposed method is compared to the state-of-the-art methods including full reference, reduced reference and no reference methods:

- Full reference methods: HD [19], RMS [18], MSDM2 [20], TPDM [25], Chetouani [26].
- Reduced reference methods: 3DWPM1 [56], 3DWPM2 [56], FMPD [28].
- No reference methods: NR-SVR [52], NR-GRNN [51], NR-CNN1 [49], BMQI [57], CNN-BMQA [62], CNNs-CMP [63].
The correlation coefficients values \( r_s \) and \( r_p \) on the LIRIS masking, LIRIS/EPFL General-purpose, UWB compression and the IEETA simplification databases are listed in Tables 2, 3, 4, 5, respectively.

We note that, to compute the correlation coefficients for a given object, the statistical dependence is computed between predicted classes of all its distorted versions and the corresponding ground truth classes. To compute the correlation coefficients for the whole database, the statistical dependence is computed between classes of all objects in the repository and their ground truth classes.

1. **The General-purpose database** (see Table 2) is the largest MVQ database so far. It contains the highest number of distorted meshes than the other databases (i.e., 84 distorted meshes and a variety of distortion types). On this database, the proposed method shows good performance and provides good correlation coefficients \( (r_s = 89.3\% \text{ and } r_p = 88.6\%) \).

2. **On the LIRIS masking database** (see Table 3), the proposed method exhibit high Spearman and Pearson correlation coefficients on the whole corpus \( (r_s = 91.7\% \text{ and } r_p = 90.9\%) \). It outperforms the NR methods \( \text{(BMQI, NRCNN1, and NR-GRNN)} \) and the most effective FR and RR methods.

3. **On the UWB compression database** (see Table 4), the proposed method is the most efficient in terms of \( r_s \) score among the NR methods \( r_s = 90.5\% \). Besides, it provides competitive scores by outperforming many effective FR and RR methods.

4. **On the IEETA simplification database** (see Table 5), the proposed method provides competitive correlation coefficients \( (r_s = 89.9\% \text{ and } r_p = 89.4\%) \). The perceptual methods MSDM2, TPDM, and FMPD are also shown to be effective in this database.

The classical measures, HD and RMS, are widely used to estimate the perceived visual quality. These methods are easy to use since a simple geometric distance permits to predict the quality score. However, they generally fail to assess the perceived quality score. It is the reason why their scores are not high enough on all test databases. Another drawback is that they require the same connectivity on the compared meshes. Including perceptual features increase the performances considerably. For instance, MSDM2 and TPDM incorporate perceptual information (mesh curvature).

Thus, a better prediction is achieved than the geometric measures as reflected by the correlation coefficient scores. FMPD includes a roughness measure which is an essential feature in mesh processing. The scores provided by this method are very competitive. Indeed, they correctly estimate the perceived quality.

Based on these results, the proposed method shows high performance on all subjectively-rated MVQ databases, as proven by its high scores on the individual models and the whole repositories.

The method CNNs-CMP indeed provides the highest scores. However, its main drawback is that it uses 2D images rendered from 3D meshes to feed a deep CNN. Moreover, the performance of this method is obtained using three pre-trained CNNs, which is time-consuming. Indeed, we use in this work the 3D mesh directly without rendering to feed a GCN. To the best of our knowledge, this is the first method that uses a graph network to evaluate the perceived visual quality. The obtained scores are very promising (exceed or close to 90\%), our method outperforms many full reference and reduced reference methods including BMQI, TPDM, 3DWM, and other methods. In addition, It is noteworthy that we used a shallow network to test if graph representations are useful for such an application. The proposed network provides competitive results with the least possible resources.

### C. CROSS DATASET EVALUATION

In this section, we investigate the generalization ability of the proposed method to predict the quality. To do so, we perform a cross-database evaluation by training our network on the General-purpose database and using the other databases for the test. We choose this database for the training process because it contains the highest number of distorted models and a wide variety of distortion types. Table 6 shows the evaluation results. It presents the correlation coefficients of each 3D object in the three tested databases (i.e., LIRIS masking, UWB compression, and IEETA simplification) and the scores for the whole repositories. The cross-dataset evaluation is performed to test the generalization ability of a method. In which a network trained on one corpus would be evaluated on a range of out-of-dataset corpora. Instead of evaluating the performance solely based on one dataset or multiple datasets individually, cross-dataset evaluation enables us to evaluate...
TABLE 2. Correlation coefficients $r_s$ (%) and $r_p$ (%) of different objective methods on the LIRIS/EPFL general-purpose database.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Armadillo</th>
<th>Dyno</th>
<th>Venus</th>
<th>Rocker</th>
<th>All models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Reference</td>
<td>HD [19]</td>
<td>69.5</td>
<td>30.2</td>
<td>30.9</td>
<td>22.6</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>RMS [18]</td>
<td>62.7</td>
<td>32.3</td>
<td>0.3</td>
<td>0.0</td>
<td>90.1</td>
</tr>
<tr>
<td></td>
<td>MSDM2 [20]</td>
<td>81.6</td>
<td>85.3</td>
<td>85.9</td>
<td>85.7</td>
<td>89.3</td>
</tr>
<tr>
<td></td>
<td>TPDM [25]</td>
<td>84.5</td>
<td>78.8</td>
<td>92.2</td>
<td>89.0</td>
<td>90.6</td>
</tr>
<tr>
<td></td>
<td>Chetouani [26]</td>
<td>75.7</td>
<td>86.1</td>
<td>90.6</td>
<td>90.0</td>
<td>94.9</td>
</tr>
<tr>
<td>Reduced Reference</td>
<td>3DWP-M1 [56]</td>
<td>65.8</td>
<td>35.7</td>
<td>62.7</td>
<td>35.7</td>
<td>71.6</td>
</tr>
<tr>
<td></td>
<td>3DWP-M2 [56]</td>
<td>74.1</td>
<td>43.1</td>
<td>52.4</td>
<td>19.9</td>
<td>34.8</td>
</tr>
<tr>
<td></td>
<td>FMPD [28]</td>
<td>75.4</td>
<td>83.2</td>
<td>89.6</td>
<td>88.9</td>
<td>87.5</td>
</tr>
<tr>
<td>No-Reference</td>
<td>NR-SVR [52]</td>
<td>76.8</td>
<td>91.5</td>
<td>78.6</td>
<td>84.1</td>
<td>85.7</td>
</tr>
<tr>
<td></td>
<td>NR-GRNN [51]</td>
<td>87.1</td>
<td>97.3</td>
<td>91.2</td>
<td>94.1</td>
<td>86.3</td>
</tr>
<tr>
<td></td>
<td>NR-CNN1 [49]</td>
<td>87.2</td>
<td>84.3</td>
<td>86.4</td>
<td>86.2</td>
<td>92.2</td>
</tr>
<tr>
<td></td>
<td>BMQI [57]</td>
<td>20.1</td>
<td>83.5</td>
<td>-</td>
<td>-</td>
<td>88.9</td>
</tr>
<tr>
<td></td>
<td>BMQA-GSES [37]</td>
<td>98.7</td>
<td>80.2</td>
<td>99.2</td>
<td>80.4</td>
<td>98.8</td>
</tr>
<tr>
<td></td>
<td>CNN-BMQA [62]</td>
<td>89.8</td>
<td>91.4</td>
<td>91.6</td>
<td>92.2</td>
<td>94.6</td>
</tr>
<tr>
<td></td>
<td>CNNS-CMP [63]</td>
<td>95.8</td>
<td>95.6</td>
<td>93.6</td>
<td>92.9</td>
<td>93.4</td>
</tr>
<tr>
<td></td>
<td>MVQ-GCN (proposed)</td>
<td>91.8</td>
<td>92.5</td>
<td>87.7</td>
<td>84.5</td>
<td>93.7</td>
</tr>
</tbody>
</table>

TABLE 3. Correlation coefficients $r_s$ (%) and $r_p$ (%) of different objective methods on the LIRIS masking database.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Armadillo</th>
<th>Lion</th>
<th>Bimba</th>
<th>Dyno</th>
<th>All models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Reference</td>
<td>HD [19]</td>
<td>48.6</td>
<td>37.7</td>
<td>71.4</td>
<td>25.1</td>
<td>25.7</td>
</tr>
<tr>
<td></td>
<td>RMS [18]</td>
<td>65.6</td>
<td>44.6</td>
<td>71.4</td>
<td>23.8</td>
<td>71.4</td>
</tr>
<tr>
<td></td>
<td>MSDM2 [20]</td>
<td>81.1</td>
<td>88.6</td>
<td>93.5</td>
<td>94.3</td>
<td>96.8</td>
</tr>
<tr>
<td></td>
<td>TPDM [25]</td>
<td>91.4</td>
<td>88.6</td>
<td>88.4</td>
<td>82.9</td>
<td>97.1</td>
</tr>
<tr>
<td></td>
<td>Chetouani [26]</td>
<td>99.0</td>
<td>99.0</td>
<td>83.0</td>
<td>94.0</td>
<td>99.0</td>
</tr>
<tr>
<td>Reduced Reference</td>
<td>3DWP-M1 [56]</td>
<td>58.0</td>
<td>41.8</td>
<td>20.0</td>
<td>9.7</td>
<td>20.0</td>
</tr>
<tr>
<td></td>
<td>3DWP-M2 [56]</td>
<td>48.6</td>
<td>37.9</td>
<td>38.3</td>
<td>22.0</td>
<td>37.1</td>
</tr>
<tr>
<td></td>
<td>FMPD [28]</td>
<td>94.2</td>
<td>88.6</td>
<td>93.5</td>
<td>94.3</td>
<td>98.9</td>
</tr>
<tr>
<td>No-Reference</td>
<td>NR-SVR [52]</td>
<td>89.5</td>
<td>84.7</td>
<td>100</td>
<td>96.3</td>
<td>94.2</td>
</tr>
<tr>
<td></td>
<td>NR-GRNN [51]</td>
<td>82.3</td>
<td>80.5</td>
<td>94.1</td>
<td>97.0</td>
<td>90.2</td>
</tr>
<tr>
<td></td>
<td>NR-CNN1 [49]</td>
<td>95.2</td>
<td>97.6</td>
<td>89.4</td>
<td>91.6</td>
<td>93.4</td>
</tr>
<tr>
<td></td>
<td>BMQI [57]</td>
<td>94.3</td>
<td>94.3</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>CNN-BMQA [62]</td>
<td>92.4</td>
<td>91.7</td>
<td>92.2</td>
<td>93.1</td>
<td>97.9</td>
</tr>
<tr>
<td></td>
<td>CNNS-CMP [63]</td>
<td>96.2</td>
<td>95.5</td>
<td>93.1</td>
<td>92.4</td>
<td>92.5</td>
</tr>
<tr>
<td></td>
<td>MVQ-GCN (proposed)</td>
<td>92.6</td>
<td>91.2</td>
<td>93.5</td>
<td>89.9</td>
<td>88.6</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In this work, we rely on a shallow graph convolutional network to accurately estimate distorted meshes perceived visual quality. The graph consists of two convolutional layers, a pooling layer and a classification layer. The network is fed by a graph represented by an adjacency matrix and a feature matrix containing three geometric (curvature, angles and Laplacian of Gaussian curvature) and a perceptual feature (Saliency). The proposed method successfully predicts distorted meshes visual quality, as proven by the high correlations with human judgment. To study the impact of the network architecture and improve the performances, a possible direction of future work would be testing several models performance from a different point of view.

The results of the masking database in Table. 6 ($r_s = 90.8\%$ and $r_p = 91.2\%$) are close to the results in Table. 3 ($r_s = 91.7\%$ and $r_p = 90.9\%$) and even better in terms of $r_p$. It is due to this database has the same distortion types as the training dataset (general purpose). For the compression and simplification databases that have unknown objects and distortions, it is quite normal to get slightly lower scores compared to Tables. 4 and 5. In terms of generalization ability, the obtained results ensure a good performance.

VOLUME 4, 2016 9

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
A weighted graph could be adopted by including perceptual network architectures (especially more convolutional layers). A weighted graph could be adopted by including perceptual network architectures (especially more convolutional layers).

REFERENCES


network architectures (especially more convolutional layers). A weighted graph could be adopted by including perceptual network architectures to the nodes, such as visual saliency.
Abouelaziz et al.: Learning graph convolutional network for blind mesh visual quality assessment


883

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3094663, IEEE Access

Abouelaziz et al.: Learning graph convolutional network for blind mesh visual quality assessment

ALADINE CETOUANI received his master’s degree in computer science from the University Pierre and Marie Curie, France, in 2005, and his Ph.D. degree in image processing from the University of Paris 13, France, in 2010. From 2010 to 2011, he was a postdoctoral researcher with the L2TI Laboratory, Paris 13 University. He is currently an associate professor with the Laboratory PRISME, Orleans, France. He also received the Habilitation degree from Université d’Orléans in 2020. He serves as a reviewer for major conferences and journals in the field of image analysis and pattern recognition. He co-edited different Special Issues. He is co-author of more than 100 research publications. His present interests are in image quality, perceptual analysis, visual attention, and image processing for cultural heritage using deep learning models.
MOHAMED EL HASSOUNI is a full Professor of computer science at the Mohammed V University in Rabat where he is affiliated to the Faculty of Letters and Human Sciences and the LRIT Laboratory. He received the Ph.D. degree in computer science from the University of Burgundy in 2005. His Ph.D. dissertation concerned the proposition of new method using higher order statistics for video denoising in a severely corrupted environment. He also received the Habilitation degree from Mohammed V University in 2012. He led and participated in several national and international research projects. He supervised more than 15 Ph.D. and Master Students in the area of image video processing. His current research interests include the modeling and analysis of 3D images. He especially, visual quality assessment and textured images modeling. He authored and co-authored numerous original journal and proceedings articles. He serves as a reviewer for major conferences and journals in the field of image analysis and pattern recognition. He was an invited professor at several universities (Bordeaux, Orléans, Dijon, and Konstanz). He is a member of IEEE, IEEE Signal Processing Society, INNS Moroccan chapter and several conferences program committee. Since 2014, he co-founded and co-chaired of Quality of Multimedia Services QUAMUS workshop which is organized in conjunction with the IEEE International Conference on Signal Image Technology and Internet Based Systems (SITEIS). He was also a co-chair of the “Multimedia, Computer Vision and image Processing” Track of AICCSA 2015 and 2016. He was co-chair of the third International Conference on Advanced Technologies in Signal and Image Processing (ATSIIP 2017). He co-chaired the COMBI 2017, ACMIMPH 2018, COMBI 2019 and CIVIA 2020 tracks of ACM-SAC and, co-edited different Special Issues (JEI SPIE; “Perceptually Driven Visual Information Analysis”, MTAP Springer; “Advances in Computational intelligence for Multimodal Biomedical Imaging”, etc.).

HOCINE CHERIFI is a professor of Computer Science at the University of Burgundy, Dijon, France since 1999. He completed his Ph.D. degree at the National Polytechnic Institute, Grenoble in 1984. Prior to moving to Dijon, he held faculty positions at Rouen University and Jean Monnet University, France. He has held visiting positions at Yonsei, Korea, University of Western Australia, Australia, National Pintung University, Taiwan, and Galatasaray University, Turkey. His recent research interests are in computer vision and complex networks. He has published more than 200 scientific papers in international refereed journals and conference proceedings. He held leading positions in more than 15 international conference organization (General chair, Program chair) and he served in more than 100 program committee. He is the founder of the International Conference on Complex Networks their Applications. Currently, he is a member of the Editorial Board of Computational Social Networks, PLOS One, IEEE Access, Journal of Imaging, Complex Systems, Quality and Quantity, and Scientific Reports. He is the Founding Editor-in-Chief of the Applied Network Science journal.

LONGIN JAN LATECKI is a professor of computer science at Temple University, Philadelphia. His main research interests include shape representation and similarity, object detection and recognition in images, robot perception, machine learning, and digital geometry. He received the PhD degree in 1992 and the habilitation in 1996, both from University of Hamburg, Germany. He has published over 225 research papers and books. He is an editorial board member of journals: Pattern Recognition, Computer Vision and Image Understanding, and International Journal of Mathematical Imaging. He received 2010 College of Science and Technology Research Excellence Award, the annual Pattern Recognition Society Award together with Azzriel Rosenfeld for the best article published in the journal Pattern Recognition in 1998. He is also the recipient of the 2000 Olympus Prize, the main annual award, from the German Society for Pattern Recognition (DAGM).