The Visual Computer
https://doi.org/10.1007/s00371-018-1592-9

ORIGINAL ARTICLE

A machine learning framework for full-reference 3D shape quality assessment

Zeynep Cipiloglu Yildiz[1] · A. Cengiz Oztireli[2] · Tolga Capin[3]

Abstract
To decide whether the perceived quality of a mesh is influenced by a certain modification such as compression or simplification, a metric for estimating the visual quality of 3D meshes is required. Today, machine learning and deep learning techniques are getting increasingly popular since they present efficient solutions to many complex problems. However, these techniques are not much utilized in the field of 3D shape perception. We propose a novel machine learning-based approach for evaluating the visual quality of 3D static meshes. The novelty of our study lies in incorporating crowdsourcing in a machine learning framework for visual quality evaluation. We deliberate that this is an elegant way since modeling human visual system processes is a tedious task and requires tuning many parameters. We employ crowdsourcing methodology for collecting data of quality evaluations and metric learning for drawing the best parameters that well correlate with the human perception. Experimental validation of the proposed metric reveals a promising correlation between the metric output and human perception. Results of our crowdsourcing experiments are publicly available for the community.

Keywords Visual quality assessment · Mesh quality · Perceptual computer graphics · Crowdsourcing · Metric learning

1 Introduction

Rapid advances in 3D rendering methods and technologies have increased the usage of 3D meshes in mass-market applications. Representing the meshes with high number of vertices provides high-quality visualization, while increasing the computational cost. Nevertheless, visual quality is actually judged by the human perception and there is no need to spend additional cost for the physical realism of the details that cannot be perceived by the observer. Therefore, a perceptual measure for estimating the visual quality of 3D graphical contents is needed.

✉ Zeynep Cipiloglu Yildiz
zeynep.cipiloglu@cbu.edu.tr; zeynepcipil@gmail.com

A. Cengiz Oztireli
cengizo@inf.ethz.ch

Tolga Capin
tolga.capin@tedu.edu.tr

[1] Department of Computer Engineering, Celal Bayar University, Manisa, Turkey

[2] Computer Graphics Lab, ETH Zurich, Zurich, Switzerland

[3] Computer Engineering Department, TED University, 06420 Kolej/Ankara, Turkey

Vast majority of the proposed methods designed for evaluating the visual quality of 3D meshes follow a bottom-up procedure including low-level human visual system (HVS) mechanisms. Yet, developing a bottom-up model for visual quality assessment (VQA) of 3D triangulated meshes is a difficult process. First of all, HVS is not fully explored and all the theoretical models for explaining the visual perception are designed for 2D. As a result, adapting the models originally devised for 2D on 3D realm is really challenging and requires carefully tweaking a number of parameters. Furthermore, adapting 2D metrics on 3D makes the solution view dependent, which is not desirable for 3D models. Therefore, it is required that visual quality metrics for measuring the quality of 3D models should operate directly in 3D. At this point, a machine learning approach could be a neater solution for obtaining a perceptual error metric whose parameters are learned from ratings of the human observers.

Recent advances in machine learning have led to tremendous progress in many fields of computer science. Moreover, the increase in the prevalence of crowdsourcing tools facilitates the data gathering process and leverages the incorporation of human perception into computation. For that reason, the use of crowdsourcing in computer graphics applications,

Published online: 28 August 2018

where visual perception is an important concern, is promoted in recent years [13,14].

In this paper, we propose an objective perceptual distance metric for assessing the global visual quality of 3D static meshes. As an alternative to the classical bottom-up approaches, we suggest a data-driven approach in which a quality metric is directly learned from observer evaluations. The proposed method relies on crowdsourcing and metric learning techniques and well correlates with human perception according to the experimental analysis.

2 Related work

2.1 3D visual quality assessment

Methods for assessing the quality of triangle meshes can be categorized as perceptual and non-perceptual methods. Non-perceptual methods do not take human visual perception into account and propose purely geometric error measures such as Euclidean distance, Hausdorff distance, root-mean-squared error. The most common geometric measure defined for 3D meshes is the Hausdorff distance [7]. On the other hand, perceptual methods aim at measuring the perceived quality of meshes by incorporating HVS mechanisms. Recent works [4,25] review the mesh quality assessment literature. Moreover, Corsini et al. [9] and Lin and Kuo [27] presented recent surveys on perceptual methods for quality assessment.

Curvature and roughness of a surface are widely employed for describing surface quality. $GL1$ [15] and $GL2$ [38] are roughness-based metrics that use Geometric Laplacian of the mesh vertices. Lavoue et al. [24] measured structural similarity between two mesh surfaces by using curvature for extracting structural information. This metric is improved with a multi-scale approach in [22]. Two definitions of surface roughness are utilized for deriving two error metrics called $3DWPM1$ and $3DWPM2$ [8]. Another metric called $FMPD$ is also based on local roughness estimated from Gaussian curvature [43]. Curvature tensor difference of two meshes is used for measuring the visible errors between two meshes [39]. A novel roughness-based perceptual error metric, which incorporates structural similarity, visual masking, and saturation effect, is proposed by Dong et al. [11].

Image-based perceptual metrics operate in 2D image space by using rendered images of the 3D mesh while evaluating the visual quality. These metrics generally employ HVS models such as Contrast Sensitivity Function (CSF), which maps spatial frequency to visual sensitivity. Most common image quality metric is visible difference prediction (VDP) method which produces a 2D local visible distortions map [10]. Similarly, Visual Equivalence Detector method outputs a visual equivalence map which demonstrates the equally perceived regions of two images [34]. A perceptual quality metric based on VDP method is also proposed in [44] for animated triangle meshes.

All of the aforementioned methods are bottom-up which means they are stimulus-driven. Such approaches require applying HVS models and carefully tuning many parameters, which is a difficult process. Alternatively, a machine learning-based approach which is fed by human evaluations could provide a more calibrated perceptual quality metric.

2.2 Machine learning and crowdsourcing for VQA

Machine learning techniques have been successfully employed in 2D image quality assessment, especially for blind VQA where the reference image is not available. Mittal et al. [31] extracted natural scene statistics features from images and learn a mapping between these features and quality scores through a regression model. A similar approach is followed in the study by Saad et al. [35], where DCT domain features are used and a Bayesian learning framework is constructed. A recent work [20] develops a support vector regression (SVR) model for learning the quality of tone-mapped HDR pictures.

There are also several recent attempts which use machine learning techniques in 3D VQA. Lavoue et al. [23] proposed a pioneering work in the sense that it leverages machine learning for mesh quality assessment; by optimizing the weights of several mesh descriptors using multi-linear regression. Abouelaziz et al. [1] used mean curvature values as features of the 3D models and estimate the weights of these features through a general regression neural network. Nouri et al. [33] proposed a 3D blind mesh quality assessment index based on saliency and roughness statistics features, whose weights are learned by a SVR model. A similar SVR model is developed in [5], which uses several VQA metrics such as Hausdorff distance, 3DWPM2 [8], and MSDM [24] as features. Another SVR model employs mesh dihedral angles as features [2].

Despite the success and prevalence of the machine learning techniques in 2D VQA, it is relatively immature in 3D VQA. Thus, we aim to contribute to this field by proposing a full-reference 3D VQA metric based on a machine learning framework. We also strengthen our framework with crowdsourcing tools which facilitate gathering training data.

Crowdsourcing is recently utilized for different purposes such as determining the best viewpoint in 3D scenes [37], semantic editing of 3D models [45], parameter tweaking in visual design explorations such as color correction for images, camera and light control on 3D scenes, shader parameter control, and determining the blendshape weights [18].

Crowdsourcing has also been a common tool for estimating similarity in several applications such as semantic image similarity [16], illustration style for clip arts [12], 3D shape style similarity [29], compatibility for 3D furniture models [28], and style similarity for infographics design [36]. The

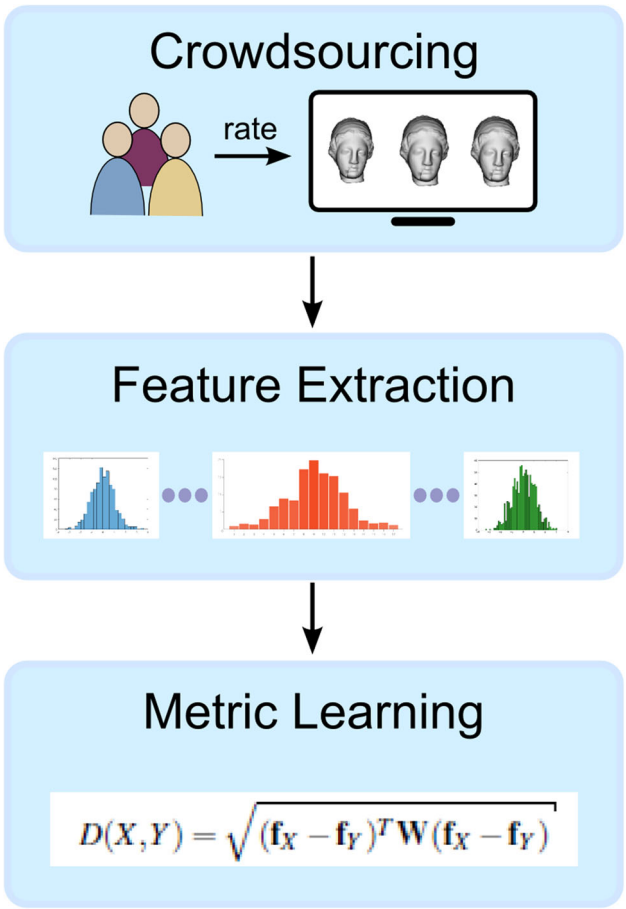

Fig. 1 Overview of the proposed method. Human observer evaluations of 3D meshes are gathered through crowdsourcing. Then important features are extracted from the 3D meshes and lastly, the weights of these features are learned by a metric learning framework

main approach in these studies is that they collect relative comparisons through a crowdsourcing platform; they extract several features for the items whose similarities will be measured; and based on these features, they define a metric whose parameters are learned to maximize the likelihood of observing training data obtained in crowdsourcing. We extend such techniques to 3D shape perception in this study.

3 Approach

Processing pipeline of the proposed method is illustrated in Fig. 1. First of all, using crowdsourcing, we collect comparative evaluations of 3D meshes from human observers. Then we extract several descriptive features from the 3D meshes used in the experiment. Lastly, we define a simple distance function and learn the weights of the extracted features on this function through optimization.

3.1 Crowdsourcing experiment

According to our methodology, we first need to collect user evaluations. The most common way of this process is to utilize online crowdsourcing platforms. We chose *Amazon Mechanical Turk (AMT)*[1] as our crowdsourcing platform due to its prevalence, efficiency, and sound documentation. We benefit from the AMT command line tools,[2] which offer a simple and efficient interface to the AMT library.

Using AMT services, one can easily design and conduct simple user tests each of which is called *Human Intelligence Task (HIT)*. AMT supplies much functionality for performing user experiments which allow displaying images and videos. However, it does not provide built-in functionality to show 3D meshes, which is crucial for our experiments. Therefore, we constructed a framework which can display 3D meshes interactively on the web browser by directing AMT server to external pages running *WebGL*[3] and *Javascript 3D Library*.[4]

3.1.1 Data

We constructed our dataset for training our model, using models from public datasets LIRIS/EPFL general-purpose dataset [24], 3D mesh watermarking benchmark [42], LIRIS masking dataset [21], and 3D mesh animation quality database [40]. Table 1 lists the properties of these meshes, and Fig. 2 displays the reference meshes.

For the meshes *Armadillo, RockerArm, Dinosaur,* and *Venus*, two types of distortion; noise addition and smoothing, were applied with different strengths at four locations: on the whole model, on smooth areas, on rough areas, and on intermediate areas. This dataset also provides mean opinion scores (MOS) and several metric results for the models. For the rest of the meshes, only noise addition is applied at the four locations described above. We selected these distortions since noise addition and smoothing are considered sufficient to reflect many possible distortions in 3D mesh processing methods, thus allowing a general-purpose metric [24].

In our AMT experiments, if the original meshes have high number of vertices ($>$ 40 K), we used 50% simplified versions of the models, with boundary preserving constraint, to prevent possible loading overhead on the client browsers. We applied the *quadric edge collapse decimation* method in MeshLab [6], for simplifying the meshes. All the data used in the experiments are included in the supplemental material.

[1] https://www.mturk.com/mturk/welcome.

[2] https://requester.mturk.com/developer/tools/clt.

[3] http://get.webgl.org/.

[4] http://threejs.org/.

Table 1 Properties of the meshes used in crowdsourcing experiments

Mesh	Reference dataset	#Vertices	#Faces	#Distortions	Distortion types	#Triplets
Armadillo	[24]	20002	40000	21	Noise, smoothing	210
Dinosaur	[24]	21074	42144	21	Noise, smoothing	210
RockerArm	[24]	20088	40176	21	Noise, smoothing	210
Venus	[24]	24834	49664	21	Noise, smoothing	210
Bimba	[21]	8857	17710	12	Noise	66
Bunny	[42]	34835	69666	12	Noise	66
Dragon	[42]	25000	50000	12	Noise	66
Dress	[40]	20772	40570	12	Noise	66
Hand	[42]	36619	72958	12	Noise	66
Rabbit	[42]	35330	70656	12	Noise	66
Ramesses	[42]	30002	60000	12	Noise	66

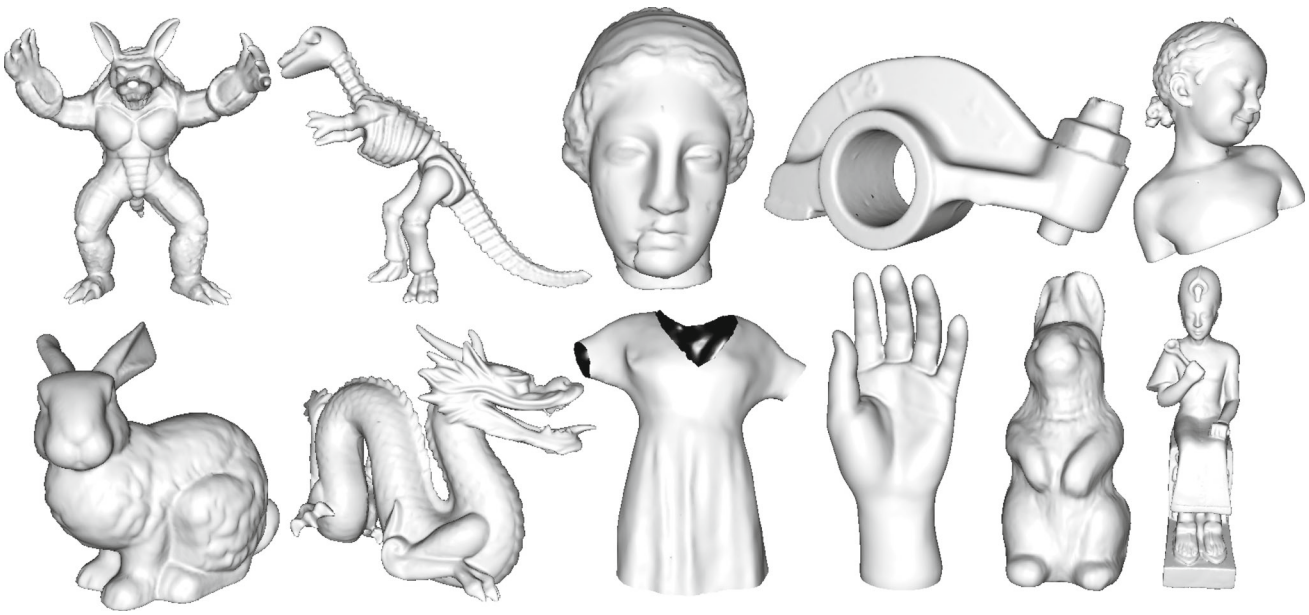

Fig. 2 Meshes used in the AMT experiment

3.1.2 Experiment design

It is known that human observers are better at providing relative comparisons than making absolute judgments. In our experiments, we preferred triplet design in which three meshes from the same object type with different distortions are presented. The task of the viewer is then to select which of the meshes is more similar to the reference mesh (displayed in Panel A), in terms of visual quality (Fig. 3). At the top of the HIT page, we provide a list of guidelines to the subjects explaining that they should consider the spatial distortions in the mesh surface while judging the visual quality. We have opted for a forced choice design with only two options; we have not presented a "None of them" or "Both of them" option since such options are highly abused by lazy turkers.

The user is able to rotate, zoom in/out, and translate the models and the user interaction is simultaneous for three models. Panel A always contains the reference model without distortions. This generates 1302 query triplets in total (Table 1). We asked two comparison questions (triplets) in each HIT, one of which is a control question with an obvious answer. The duration of each HIT, for which we paid $0.03, was approximately 3 minutes. Each triplet was evaluated by at least twenty users, at the end of the experiment.

3.1.3 Reliability check for the crowdsourced data

Data gathered through online crowdsourcing platforms are prone to some reliability concerns; because we do not have full control over the response collection process as it is performed remotely. Thus, in order to assure the reliability of

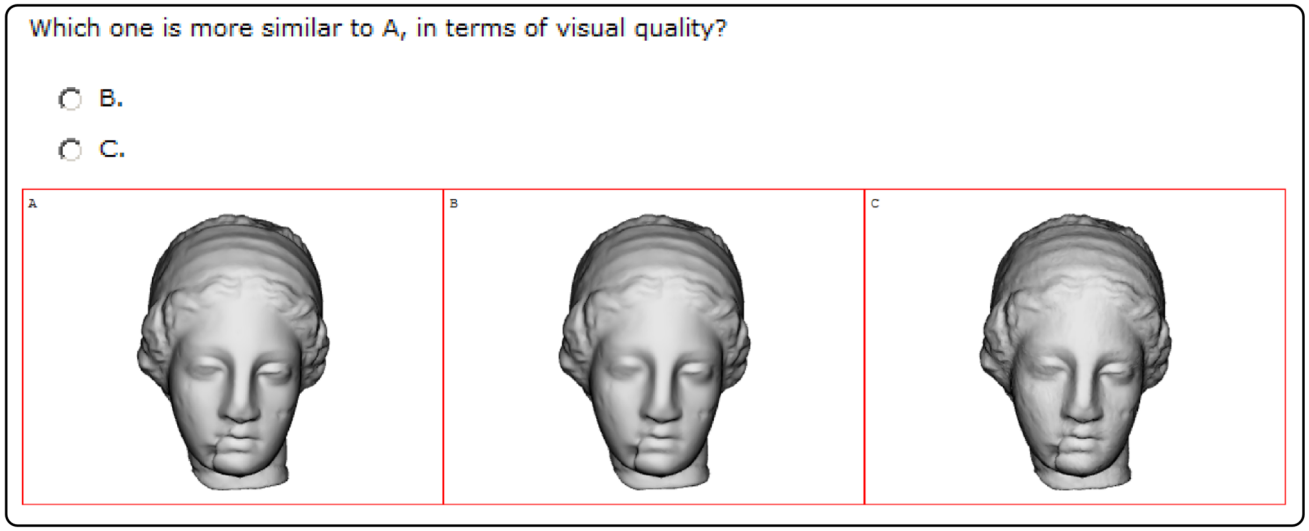

Fig. 3 Screenshot from our AMT experiment

the collected data, we employ several design issues in our AMT experiments, following the suggestions in [13]. Our precautions for the purpose of reliability can be summarized as below:

- First of all, each user has to take a training session with obvious answers, which facilitates the learning of the test procedure for the user. The users are allowed to proceed to the actual test, only if they answer all the training questions correctly. This training session was easily implemented through the "qualification" facility of the AMT library. (See online supplementary material for the details.)
- Secondly, each HIT contains one control question with an obvious answer. If the user fails to answer the control question correctly, that HIT is rejected.
- Lastly, a response time check is performed to identify sloppy participants. In this regard, if the response time for a HIT is shorter than a threshold value, that HIT is also rejected. As the threshold value, we used 15 seconds, which roughly corresponds to the standard deviation.
- If a user has three or more rejected HITs, we regard him as unreliable and do not include his responses in the final dataset.

At the end of the experiment, 22 subjects were blocked among 207 unique subjects participated in the experiment and the ratio of the rejected HITs over the total submitted HITs is about 1.5%.

In addition, before the optimization, we have observed that some of the responses in the collected data are not discriminative in the sense that disagreement between the subjects is high. Such kind of responses do not improve the learning process and introduces computational overhead. Hence, we preprocessed the data collected from the AMT experiment to remove the non-discriminative responses (i.e., 9 of the responses are B and 11 of them are C, or vice versa). About 5% of the tuples were eliminated at the end of this process.

3.2 Feature extraction

The main purpose of this step is to extract several features that describe the geometry of the meshes. We have implemented the following geometric attributes that are widely used for visual quality calculations, in our method. All these attributes are per-vertex. Four moments extracted from the distribution of each per-vertex attribute are used as descriptors. These moments are *mean*, *variance*, *kurtosis*, and *skewness* of the histograms. Stacking all these attributes together generates a feature vector of size 28.

- **Curvatures** Surface curvature is considered to be directly related to the visual quality of the mesh, in the literature. Minimum (κ_1), maximum (κ_2), mean ($(\kappa_1 + \kappa_2)/2$), and Gaussian ($\kappa_1 \times \kappa_2$) curvature fields are estimated as in [3]. According to this definition, curvature tensor T for every vertex v for the neighborhood B, approximated by a geodesic disk around this vertex, is calculated as below.

$$T(v) = \frac{1}{|B|} \sum_{edges\ e} \beta(e)\ |e \cap B|\ {}^{-}e\ {}^{\tau}e \tag{1}$$

where $|B|$ is the surface area over which the tensor is estimated, $\beta(e)$ is the signed angle between the normals of the faces incident to edge e, $|e \cap B|$ is the length of the intersection of edge e with the region B, and ${}^{-}e$ is a unit vector in the same direction with e. Eigendecomposition

of the tensor field T is used to estimate the minimum and maximum curvatures.

- **Shape index** Koenderink and van Doorn [17] state that "all local approximations for which the ratio of the principal curvatures is equal are of the same shape" [17]. Based on this definition, they calculate *shape index* as in Eq. 2, where κ_1 and κ_2 are the minimum and maximum curvatures, respectively.

$$Shape\ Index = 2/\pi \arctan\left[(\kappa_2 + \kappa_1)/(\kappa_2 - \kappa_1)\right] \quad (2)$$

- **Curvedness** In conjunction with the shape index notion, *curvedness* refers to the amount of surface curvature and is defined in Eq. 3.

$$Curvedness = \sqrt{({\kappa_1}^2 + {\kappa_2}^2)/2} \quad (3)$$

- **Surface roughness** Local roughness for each vertex is defined as the absolute value of the Laplacian of the discrete Gaussian curvature [43]. First, mesh Laplacian matrix is calculated as in Eq. 4, with cotangent weights.

$$\begin{aligned} D_{ij} &= \frac{cot(\beta_{ij}) + cot(\beta'_{ij})}{2}, \ for\ j \in N_i^{(V)} \\ D_{ii} &= -\sum_j D_{ij} \end{aligned} \quad (4)$$

where $N_i^{(V)}$ is the one-ring neighborhood of v_i, and β_{ij} and β'_{ij} are the two angles opposite to the edge constructed by v_i and v_j. Then the local roughness at each vertex is defined as in Eq. 5, where GC denotes the discrete Gaussian curvature.

$$LR_i = \left| GC_i + \frac{\sum_{j \in N_i^{(V)}} D_{ij}.GC_j}{D_{ii}} \right| \quad (5)$$

In addition to these features, several other attributes were also included in the initial attempts of our method. However, they are excluded from the final implementation as they do not have significant contribution on the accuracy while amplifying the computational cost. These additional features are mesh saliency values calculated according to the method by Lee et al. [26], largest 10 eigenvalues of the mesh Laplacian operator, and mesh dihedral angles [41].

3.3 Metric learning

Based on the feature vector definition in the previous section and training data gathered through crowdsourcing, we formulate our problem as an instance of metric learning [19]. As a general approach in these studies, an objective function, based on a logistic formulation which expects more noise for relative comparisons with less clear answers, is defined and minimized [12].

More precisely, given two meshes (X and Y) to be compared, let f_X and f_Y be their feature vectors, respectively. We define the weighted Euclidean distance between them as in Eq. 6. Our goal is then to learn the weights on the diagonal of W, in such a way that the likelihood of observing the training data is maximized.

$$D(X, Y) = \sqrt{(f_X - f_Y)^T W (f_X - f_Y)} \quad (6)$$

Given a triplet of meshes $<A, B, C>$, we model the probability that the user selects B as more similar to A than C by a sigmoid function (Eq. 7).

$$P_{BC}^A = \frac{1}{1 + \exp\left(D(A, B) - D(A, C)\right)} \quad (7)$$

Learning is performed by *Maximum A Posteriori (MAP)* estimation which is acquired by minimizing the objective function in Eq. 8, over the set of all training triplets T. The second term in the equation is L_1 regularization term which is added for the purpose of obtaining a sparse feature vector, where w is the diagonal of the weight matrix W.

$$-\sum_T \log\left(P_{BC}^A\right) + \lambda ||w||_1 \quad (8)$$

We solve this nonlinear unconstrained optimization problem by *Sequential Quadratic Programming (SQP)* implementation in Matlab [32], as one of the state-of-the-art numerical solutions for nonlinear optimization. Coefficient λ in Eq. 8 is the regularization weight and experimentally set to 0.1, in our implementation. The optimization procedure is initialized by small random weights.

4 Results

4.1 Implementation of experiments

The results are calculated by leave-one-out cross-validation according to the mesh classes. For instance, all the classes except Armadillo are used for training and the resulting metric is tested on Armadillo class, then the same procedure is applied for Dinosaur class, and so on. It took approximately 10 minutes to converge our optimization procedure, on a 1.8 GHz PC.

The optimization ended up with 10 nonzero weights among 28 features, as listed in Table 2. Too small weights (< 0.1) were also set to 0. In the results, we see that local roughness variation is predominantly high. This is consistent with the findings in the literature which state the importance

Table 2 Learned weights of the feature vector (Only nonzero weights are listed)

Feature	Weight
Minimum curvature variance	2.48
Maximum curvature variance	0.46
Mean curvature mean	1.24
Mean curvature variance	2.16
Shape index mean	4.29
Shape index variance	2.78
Curvedness mean	2.82
Curvedness variance	0.47
Local roughness mean	0.42
Local roughness variance	15.95

of local roughness on perceived mesh quality [21,40]. Mean of shape index is found as the second important feature while curvedness, minimum, maximum, and mean curvatures also contribute to the result. As a remark, this is not the unique solution; therefore, there could be many different settings of the features that produce a similar result.

4.2 Quantitative prediction accuracy

In order to evaluate the success of our data-driven VQA metric, we have calculated *prediction accuracy* which measures how well a distance metric predicts the preferences of the human observers. We compare our metric to several state-of-the-art metrics by computing their prediction accuracy values.

To define the prediction accuracy formally, let tuple t collected from the crowdsourcing experiment, be in the form of $< A, B, C, q >$; where A is the reference mesh, B and C are the distorted test meshes to be compared, and q is the query response as a binary variable with 1 indicating that B was selected as more similar to A and 0 indicating that C was selected as more similar to A. Given the set of testing tuples T, prediction accuracy (PA_d) is computed as the percentage of correct predictions for variable q, when a specific metric d is used as the decision maker (Eq. 9).

$$PA_d = 100 \times \frac{\sum_{t \in T} \delta_{q_t s_t}}{|T|} \tag{9}$$

where δ is the Kronecker delta and s_t is the metric decision for tuple t, determined according to Eq. 10.

$$s_{<A,B,C,q>} = \begin{cases} 0, & d(A,B) > d(A,C) \\ 1, & d(A,B) < d(A,C) \\ 0, & d(A,B) = d(A,C)\ \&\ q = 0 \\ 1, & d(A,B) = d(A,C)\ \&\ q = 1 \end{cases} \tag{10}$$

As stated previously, each tuple is evaluated by at least twelve users in our crowdsourcing experiment. We have determined the binary variable q for a triplet, according to the *majority* response for that triplet, as in other crowdsourcing applications such as [29] and [36].

Table 3 includes the prediction accuracies for our metric and several other state-of-the-art methods. Other than these metrics, we also compared our metric to GL1 [15], GL2 [38], 3DWPM1 [8], and 3DWPM2 [8] metrics for the mesh classes Armadillo, Dinosaur, RockerArm, and Venus since they are given in the original dataset. However, the results of those metrics are very low and their source codes are not available to measure other mesh classes. Therefore, we have not included those results in the table. Still, the most recent metric results are available for all the meshes, giving us the current state-of-the-art results.

In the table, "*Uniform*" is calculated by setting all the weights of the features to 1 in Eq. 6; and "*Learned*" is our metric where the feature weights are learned from the user responses. The table also includes the prediction accuracies calculated only using *local roughness variance* ("*L.R.V.*"), since the weight of this feature is quite high when compared to other features (Table 2). As the results depict, our learned distance yields much better performance than the most recent metrics.

4.3 Validation of the metric

During AMT experiments, we have used meshes with small number of faces in order to prevent overhead on client browsers. Furthermore, our training dataset contains only two types of distortions, noise addition and smoothing, to allow a general-purpose metric (see Sect. 3.1.1). Therefore, it is necessary to validate the generalization of our metric on different distortion types and more complex mesh topologies. To that end, we have calculated the correlation between our metric results and differential mean opinion score (DMOS) values, which correspond to the differences between the MOS values of test and reference meshes, on public datasets. The datasets used for this purpose are original version of LIRIS/EPFL general-purpose dataset [24], UWB mesh compression dataset [41], and 3D Mesh Animation Quality Database (uniform and Gaussian noise) [40].

Table 4 includes average Pearson and Spearman Rank Order Correlation Coefficient (SROCC) for each dataset and metric. These results reveal that our perceptual distance metric is model-free in that it is independent of training dataset and can be used as a general-purpose metric.

4.4 Qualitative results

The results also implicate that our perceptual distance metric well reflects the common perceptual notions. Firstly, *visual*

Table 3 Prediction accuracy of each metric for each mesh (highest values are marked with bold font)

	MSDM2 (%)	FMPD (%)	TPDM (%)	L.R.V. (%)	Uniform (%)	Learned (%)
Armadillo	80	78	84	73	80	**85**
Dinosaur	83	86	**88**	60	47	**88**
RockerArm	84	86	85	71	59	**88**
Venus	86	86	79	53	48	**91**
Bimba	**100**	98	**100**	85	89	**100**
Bunny	95	94	94	79	56	**97**
Dragon	95	94	94	74	83	**98**
Dress	98	**100**	98	74	77	**100**
Hand	89	94	94	69	80	**97**
Rabbit	91	95	91	73	94	**97**
Ramesses	**97**	**97**	90	32	78	**97**
Mean	91	92	91	68	72	**94**

Table 4 Pearson and Spearman correlation coefficients for different metrics and datasets (P denotes Pearson and S denotes Spearman. Dataset 1: LIRIS/EPFL general-purpose original [24], Dataset 2: UWB compression dataset [41], Dataset 3: 3D Mesh Animation Quality Database [40])

	Dataset 1		Dataset 2		Dataset 3	
	P (%)	S (%)	P (%)	S (%)	P (%)	S (%)
MSDM2	84	85	81	55	97	92
FMPD	84	84	76	76	**98**	**93**
TPDM	79	82	90	70	**98**	**93**
Our	**90**	**90**	**91**	**86**	**98**	**93**

masking is a well-known perceptual issue which refers to the fact that perception of a target stimulus is affected by the presence of a masking stimulus. In the field of mesh quality evaluation, the consequence of visual masking effect is that distortions on the rough regions of a mesh are less likely to be perceived than the distortions on smooth regions.

In Fig. 4, two distorted versions are presented for two models: Venus and Armadillo. Although the same amount of noise is introduced by both distortions, their DMOS values are quite different. The first version of noise addition is applied on the whole mesh uniformly, while the second distortion is applied only on the rough regions of the meshes. As a result of the visual masking effect, the perception of quality degradations is less likely in rough regions. This can be directly seen in the DMOS as well as our metric scores.

Another property that affects visual quality perception is the structure of the models; smoothing and blurring may degrade the structure of 3D models. In Fig. 5, a reference mesh and three distorted meshes that are smoothed with Taubin smoothing algorithm in different amounts are portrayed. For the smoothed meshes in medium and high levels,

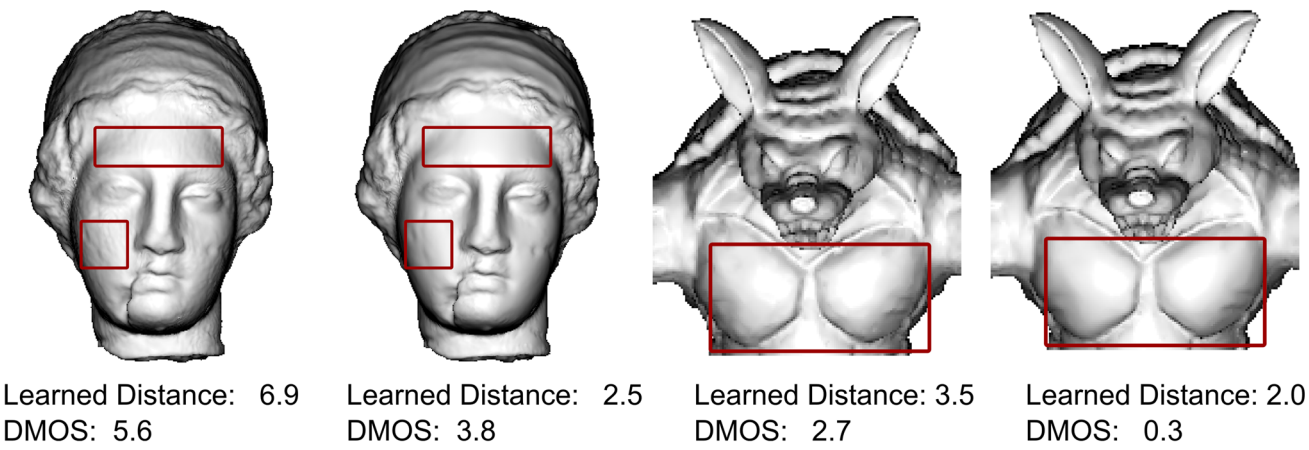

Fig. 4 Visual masking effect. Same amount of noise is applied on both meshes, but it is applied uniformly on the first mesh and on rough regions in the second mesh. The effect is visible in smooth regions, marked with borders. (Learned distance is directly correlated with the DMOS score, with a lower distance corresponding to a lower DMOS for each mesh. DMOS scores are obtained from the original datasets given in Table 1)

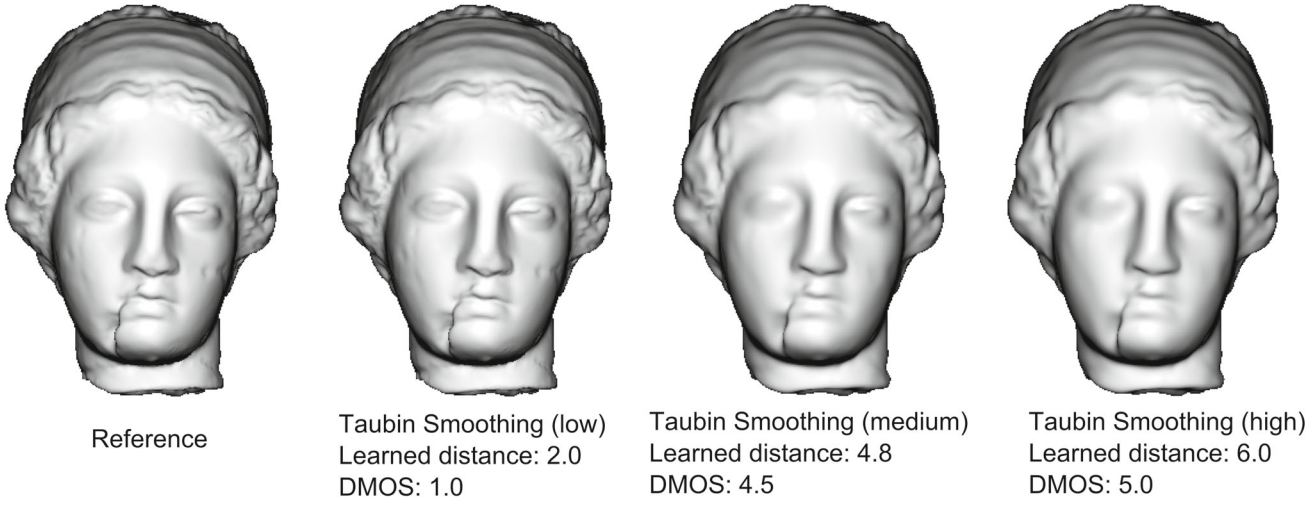

Fig. 5 Effect of the structural changes on perceived quality. (Learned distance is directly correlated with the DMOS score, with a lower distance corresponding to a lower DMOS for each mesh. DMOS scores are obtained from the original datasets given in Table 1.)

both our perceptual distance and DMOS values are quite high; indicating a low perceived quality. The reason is that smoothing fades away some structural properties of the mesh such as eye boundaries. This property is also well captured in our distance metric.

4.5 Failure cases

We have also examined the failure cases where our metric results contradict with DMOS values. Our metric generally fails when compared meshes are very similar and their distances are close. Figure 6 displays several examples for typical failures of our metric. When we investigate these cases deeply, we see that either perceptual distances or DMOS values are quite close.

In the comparison triplet of Armadillo (the first row of Fig. 6), the second mesh is smoothed in medium amount and the third mesh is smoothed in high amount. The second row of the figure shows Dinosaur triplet, where rough regions are smoothed in the second mesh and intermediate regions are smoothed in the third mesh. In both of these examples, DMOS values and our metric results produce opposite rankings of the mesh distances, although they are very close. Our metric fails in similar cases since it cannot properly sort medium and high distortion amounts, also rough and intermediate regions.

The third row in Fig. 6 includes another failure case for the RockerArm mesh. In this example, the second mesh has the same amount of noise with the third mesh; but the noise is added on the rough regions of the second mesh, while it is added only on the smooth regions in the third mesh. The same situation holds on for the Venus triplet in the last row. Our metric finds the meshes with noise in smooth regions more distant to the reference meshes. Actually, this is more suitable to the visual masking effect since noise in smooth regions is more perceptible.

5 Application of the metric

To show the effectiveness of our perceptual distance metric, we have also applied it on vertex coordinate quantization, which is widely employed in mesh compression algorithms [30,38]. Here, our aim is to find the optimum quantization level in bits per coordinate (bpc) that enables the maximum possible compression rate without introducing any visible artifact. In view of that we have quantized several meshes with bpc values from 7 to 12 and measured the perceptual distances of the resulting meshes to the undistorted meshes.

Figure 7 displays the results of these measurements. The optimum quantization level for each mesh can be determined easily by inspecting this plot. For each mesh, perceptual distance becomes constant and almost zero after a specific bpc value. A threshold distance value (1 for our metric) can be set to determine this region, following a similar approach to recent studies [39,43]. According to this thresholding, the optimum quantization levels are determined as 9, 10, 10 bpc for Armadillo, Dress, and RockerArm meshes, respectively. This simple approach is quite effective as illustrated in Fig. 8.

6 Conclusions

Main contribution of this study is a method for estimating the perceived quality of a static mesh using a machine learning pipeline, in which crowdsourced data is used while learning the parameters of a distance metric that best fits the human perception. To the best of our knowledge, this is the first

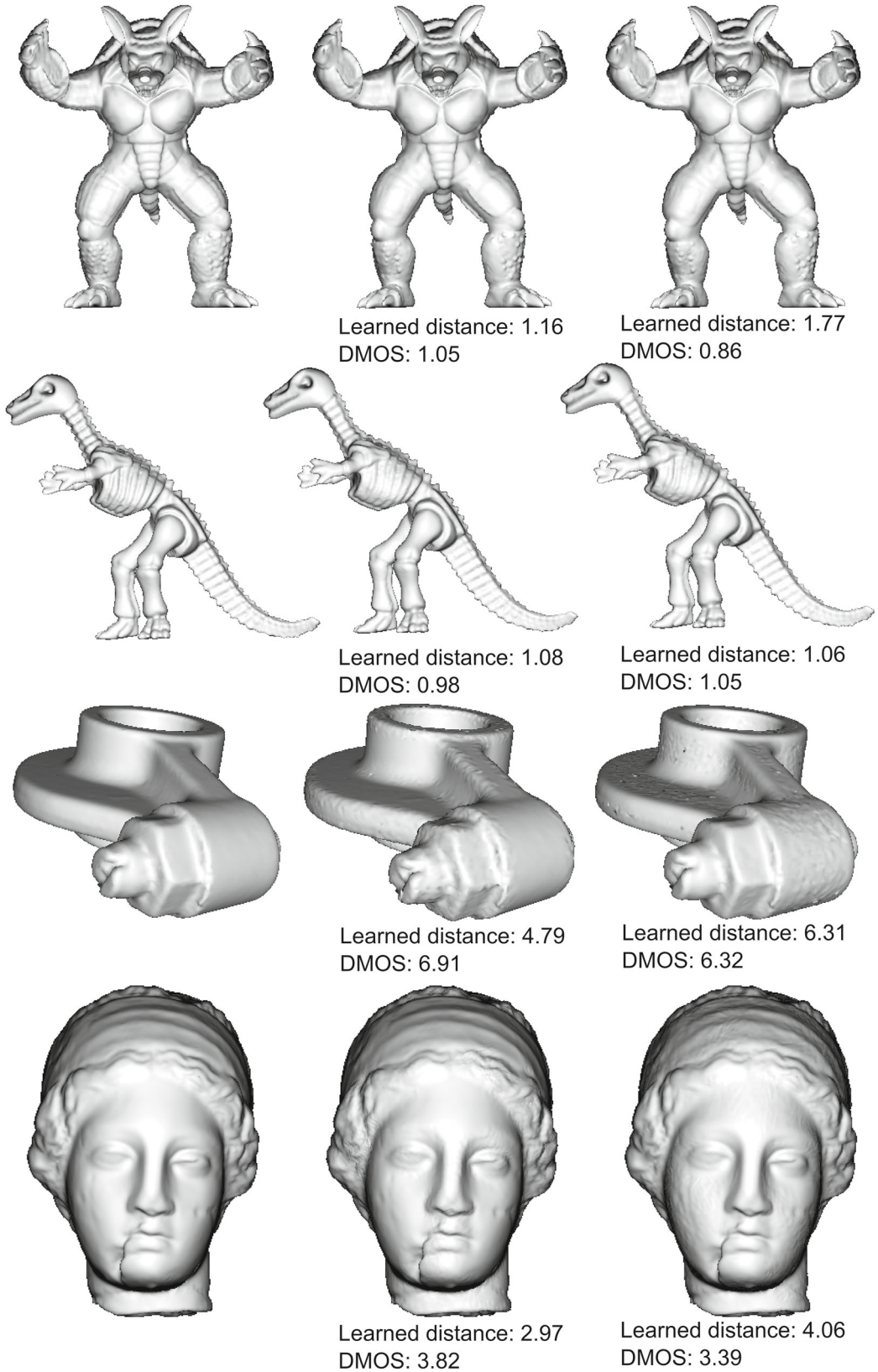

Fig. 6 Typical examples for the failure cases of our metric. (First column includes reference meshes for each row, second and third columns include compared meshes along with their learned distances and DMOS values

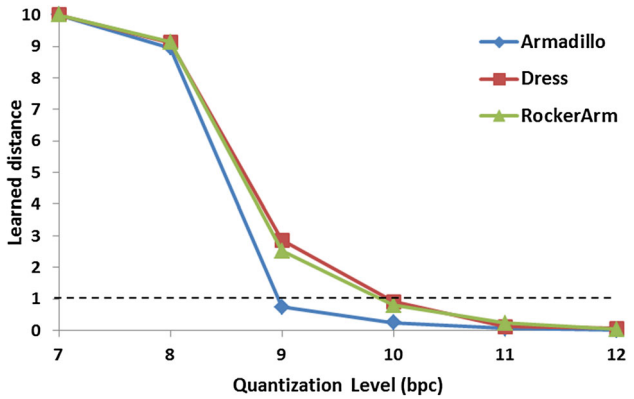

Fig. 7 Plot of learned distance vs. quantization level (in bits per coordinate, bpc) of three meshes. Dashed line shows the threshold distance value for selecting the optimum quantization level

attempt for a 3D VQA metric that utilizes crowdsourcing tools in a machine learning pipeline. Experimental results show that our metric is model-free and outperforms the baseline metrics. In addition, feature vector calculation makes the metric independent of the mesh topology and allows distance computation between meshes with different vertex count and connectivity.

Limitations and future work

Despite the efficiency of the proposed method, there are several limitations. We have already explained the reliability concerns and our preventive actions. Although we believe that these precautions minimize the bias in the collected data, they may not be sufficient. For instance, we can foresee that a diverse range of viewing parameters and display properties are used in the experiments by the subjects. Since these parameters have significant impact on the perception of visual quality, we are planning to expand this method by applying a multi-scale approach in which features are extracted for several simplification levels of the original mesh. This will improve the robustness of the algorithm by incorporating different levels of detail.

In addition, feature extraction is an important step and may influence the accuracy of the metric rigorously. Thus, new features should be investigated and their effects on the accuracy should be experimented. We also intend to exploit deep learning methods to automatically extract mesh descrip-

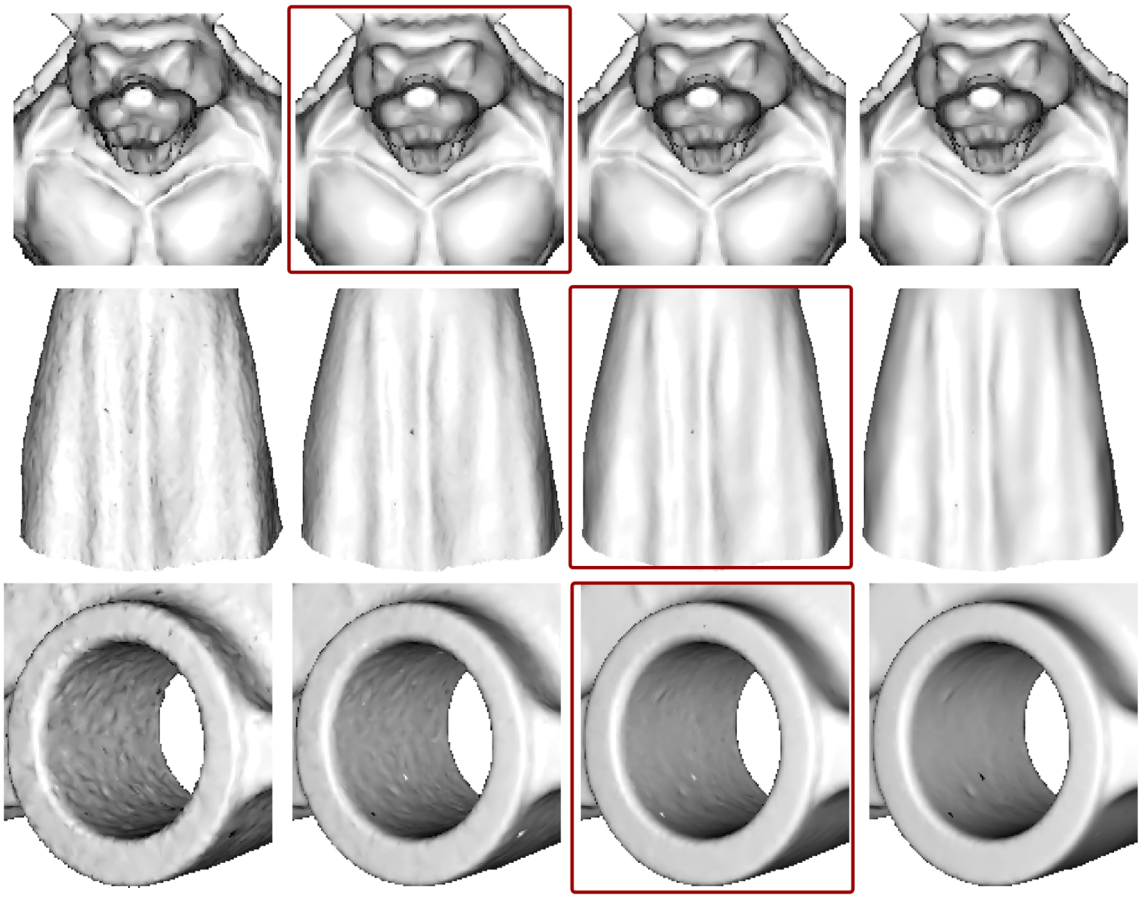

Fig. 8 Quantized meshes with different bpc values (first three columns are 8, 9, 10 bpc, respectively). The last column is the original mesh. Optimum quantization level for each mesh is marked with borders

tors, instead of using the manually extracted features. Lastly, the model should be trained by a more diverse set of meshes conveying several distortion types, with more participants.

As the initial attempt for a crowdsourcing-based VQA method for 3D meshes, our metric handles the global visual quality of static meshes only, in a full-reference scenario. Nevertheless, we have plans to extend this idea for evaluating the local visibility of distortions and also considering animated meshes. A similar approach can be devised even for no-reference quality assessment.

7 Supplemental material

Supplementary material consisting of meshes used in this study, comparison tuples obtained from crowdsourcing experiment, and resulting feature weights can be downloaded via the following link:
https://www.dropbox.com/s/m3bnb93vun91763/Learning.VQA.zip?dl=0.

Acknowledgements This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK). Also, we would like to thank Yeojin Yun for her kind help in setting up the crowdsourcing platform.

References

1. Abouelaziz, I., El Hassouni, M., Cherifi, H.: No-reference 3D mesh quality assessment based on dihedral angles model and support vector regression. In: International Conference on Image and Signal Processing, pp. 369–377. Springer (2016)
2. Abouelaziz, I., El Hassouni, M., Cherifi, H.: Blind 3D mesh visual quality assessment using support vector regression. Multimedia Tools and Applications pp. 1–22 (2018)
3. Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., Desbrun, M.: Anisotropic polygonal remeshing. ACM Trans. Graph. **22**, 485–493 (2003)
4. Bulbul, A., Capin, T., Lavoué, G., Preda, M.: Assessing visual quality of 3-d polygonal models. IEEE Signal Processing Mag. **28**(6), 80–90 (2011)
5. Chetouani, A.: A 3D mesh quality metric based on features fusion. Electron. Imaging **2017**(20), 4–8 (2017)
6. Cignoni, P., Corsini, M., Ranzuglia, G.: Meshlab: an open-source 3D mesh processing system. ERCIM News **73**, 45–46 (2008)
7. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: measuring error on simplified surfaces. In: Computer Graphics Forum, vol. 17, pp. 167–174. Wiley Online Library (1998)
8. Corsini, M., Gelasca, E., Ebrahimi, T., Barni, M.: Watermarked 3-D mesh quality assessment. IEEE Trans. Multimedia **9**(2), 247–256 (2007)
9. Corsini, M., Larabi, M.C., Lavoué, G., Petřík, O., Váša, L., Wang, K.: Perceptual metrics for static and dynamic triangle meshes. Comput. Graph. Forum **32**, 101–125 (2013)
10. Daly, S.J.: Visible differences predictor: an algorithm for the assessment of image fidelity. In: SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology, pp. 2–15. International Society for Optics and Photonics (1992)
11. Dong, L., Fang, Y., Lin, W., Seah, H.S.: Perceptual quality assessment for 3D triangle mesh based on curvature. IEEE Trans. Multimedia **17**(12), 2174–2184 (2015)
12. Garces, E., Agarwala, A., Gutierrez, D., Hertzmann, A.: A similarity measure for illustration style. ACM Trans. Graph. **33**(4), 93 (2014)
13. Gingold, Y., Shamir, A., Cohen-Or, D.: Micro perceptual human computation for visual tasks. ACM Trans. Graph. **31**(5), 119 (2012)
14. Heer, J., Bostock, M.: Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 203–212. ACM (2010)
15. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 279–286. ACM Press/Addison-Wesley Publishing Co. (2000)
16. Kleiman, Y., Goldberg, G., Amsterdamer, Y., Cohen-Or, D.: Toward semantic image similarity from crowdsourced clustering. Vis. Comput. **32**(6–8), 1045–1055 (2016)
17. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. Image Vis. Comput. **10**(8), 557–564 (1992)
18. Koyama, Y., Sakamoto, D., Igarashi, T.: Crowd-powered parameter analysis for visual design exploration. In: Proceedings of the 27th annual ACM symposium on User interface software and technology, pp. 65–74. ACM (2014)
19. Kulis, B.: Metric learning: a survey. Found. Trends Mach. Learn. **5**(4), 287–364 (2012)
20. Kundu, D., Ghadiyaram, D., Bovik, A.C., Evans, B.L.: No-reference quality assessment of tone-mapped hdr pictures. IEEE Trans. Image Process. **26**(6), 2957–2971 (2017)
21. Lavoué, G.: A local roughness measure for 3D meshes and its application to visual masking. ACM Trans. Appl. Percept. **5**(4), 21 (2009)
22. Lavoué, G.: A multiscale metric for 3D mesh visual quality assessment. Comput. Graph. Forum **30**, 1427–1437 (2011)
23. Lavoué, G., Cheng, I., Basu, A.: Perceptual quality metrics for 3D meshes: towards an optimal multi-attribute computational model. In: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, pp. 3271–3276. IEEE (2013)
24. Lavoué, G., Gelasca, E.D., Dupont, F., Baskurt, A., Ebrahimi, T.: Perceptually driven 3D distance metrics with application to watermarking. In: Optics & Photonics, pp. 63,120L–63,120L. International Society for Optics and Photonics (2006)
25. Lavoué, G., Mantiuk, R.: Quality assessment in computer graphics. In: Visual Signal Quality Assessment, pp. 243–286. Springer (2015)
26. Lee, C., Varshney, A., Jacobs, D.: Mesh saliency. In: ACM SIGGRAPH 2005 Papers, pp. 659–666. ACM (2005)
27. Lin, W., Kuo, C.C.J.: Perceptual visual quality metrics: a survey. J. Vis. Commun. Image Represent. **22**(4), 297–312 (2011)
28. Liu, T., Hertzmann, A., Li, W., Funkhouser, T.: Style compatibility for 3D furniture models. ACM Trans. Graph. **34**(4), 85 (2015)
29. Lun, Z., Kalogerakis, E., Sheffer, A.: Elements of style: learning perceptual shape style similarity. ACM Trans. Graph. **34**(4), 84 (2015)
30. Maglo, A., Lavoué, G., Dupont, F., Hudelot, C.: 3D mesh compression: survey, comparisons, and emerging trends. ACM Comput. Surv. **47**(3), 44 (2015)
31. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Trans. Image Process. **21**(12), 4695–4708 (2012)
32. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)
33. Nouri, A., Charrier, C., Lézoray, O.: 3D blind mesh quality assessment index. Electron. Imaging **2017**(20), 9–26 (2017)

34. Ramanarayanan, G., Ferwerda, J., Walter, B., Bala, K.: Visual equivalence: towards a new standard for image fidelity. In: ACM SIGGRAPH 2007 papers, SIGGRAPH '07. ACM, New York (2007)
35. Saad, M.A., Bovik, A.C., Charrier, C.: Blind image quality assessment: a natural scene statistics approach in the dct domain. IEEE Trans. Image Process. **21**(8), 3339–3352 (2012)
36. Saleh, B., Dontcheva, M., Hertzmann, A., Liu, Z.: Learning style similarity for searching infographics. In: Proceedings of the 41st Graphics Interface Conference, pp. 59–64. Canadian Information Processing Society (2015)
37. Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A.: Perceptual models of viewpoint preference. ACM Trans. Graph. **30**(5), 109 (2011)
38. Sorkine, O., Cohen-Or, D., Toledo, S.: High-pass quantization for mesh encoding. In: Symposium on Geometry Processing, vol. 42 (2003)
39. Torkhani, F., Wang, K., Chassery, J.M.: A curvature-tensor-based perceptual quality metric for 3D triangular meshes. Mach. Graph. Vis. **23**(1–2), 59–82 (2014)
40. Torkhani, F., Wang, K., Chassery, J.M.: Perceptual quality assessment of 3D dynamic meshes: subjective and objective studies. Signal Process. Image Commun. **31**, 185–204 (2015). https://doi.org/10.1016/j.image.2014.12.008
41. Váša, L., Rus, J.: Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes. Comput. Graph. Forum **31**, 1715–1724 (2012)
42. Wang, K., Lavoué, G., Denis, F., Baskurt, A., He, X.: A benchmark for 3D mesh watermarking. In: Proc. of the IEEE International Conference on Shape Modeling and Applications, pp. 231–235 (2010)
43. Wang, K., Torkhani, F., Montanvert, A.: A fast roughness-based approach to the assessment of 3D mesh visual quality. Comput. Graph. **36**(7), 808–818 (2012)
44. Yildiz, Z.C., Capin, T.: A perceptual quality metric for dynamic triangle meshes. EURASIP J. Image Video Process. **2017**(1), 12 (2017). https://doi.org/10.1186/s13640-016-0157-y
45. Yumer, M.E., Chaudhuri, S., Hodgins, J.K., Kara, L.B.: Semantic shape editing using deformation handles. ACM Trans. Graph. **34**(4), 86 (2015)

Zeynep Cipiloglu Yildiz received her B.S., M.S., and Ph.D. degrees in Computer Engineering from Bilkent University in 2007, 2010, and 2016, respectively. Currently, she is an assistant professor in the Department of Computer Engineering, Celal Bayar University, Manisa, Turkey. She worked as a researcher in the 3D-Phone project which is funded by the Seventh Framework Program of the European Union. Her research interests are computer graphics, vision, visual perception, and 3D user interfaces.

A. Cengiz Oztireli is currently a Research Scientist at Disney Research Zürich. His research interests are in computer graphics, vision, and machine learning. With his collaborators from academia and industry, he has been publishing works in international journals and conferences. He obtained his M.S. and Ph.D. degrees in computer science from ETH Zurich (jointly funded by the Swiss National Science Foundation), and completed a double major in computer engineering and electronics engineering at Koc University (valedictorian). He has been honored with several awards including the Eurographics Best Ph.D. Thesis Award and Fulbright Science and Technology Award.

Tolga Capin received his B.S. and M.S. degrees in Computer Engineering from Bilkent University in 1991 and 1993, and Ph.D. degree in Computer Sciences from Ecole Polytechnique Federale de Lausanne (EPFL) in 1998. He is an associate professor in the Computer Engineering Department, TED University. Before joining TED University, he worked at Nokia Research Center U.S.A. and worked on various projects related to the fields of mobile graphics, mobile interaction, and augmented reality between 2000 and 2006. He worked as a member of staff at Computer Engineering Department of Bilkent University between 2006 and 2014. He has been the coordinator of the FP7 3DPHONE project. He received a service award from ISO for his contributions to the ISO-MPEG standard between 1991 and 1998. In 2014, he received an "Outstanding Specification Lead" award for his contribution to mobile Java standards.