

Enhancement of perceptual FR quality metrics for geometric distortion in 3d static models

Nessrine ELOUUMI*

*SETIT, ISBS, University of Sfax,
Sfax, BP 1175 - 3000 Sfax, Tunisia
ellouminessrine@gmail.com*

Habiba LOUKIL

*ISGI, University of Sfax,
Sfax, BP 1164 - 3021 Sfax, Tunisia
habiba.loukil@isgis.usf.tn*

Johan DEBAYLE

*SPIN, ENSM-SE, 158 cours Fauriel, SAINT-ETIENNE,
CS 62362 42023, Saint-Étienne CEDEX 2, FRANCE
debayle@emse.fr*

Med Salim BOUHLEL

*SETIT, ISBS, University of Sfax,
Sfax, BP 1175 - 3000 Sfax, Tunisia
medsalim.bouhlel@isbs.usf.tn*

With the blooming growth of data, transmission, and storage capabilities through multimedia applications, it is common to exchange Multimedia content, including pictures, films, and 3D models, which is processed and displayed online. However, to opt for an efficient processing algorithm, the quality evaluation of the processed data is recommended. Since multimedia representations are evaluated by a human observer, it is essential to research and identify the structures of the visual material to design an efficient evaluation system. Due to the intricacy of the 3D data representation via meshes, this process is more difficult for three-dimensional representation. For the representation of 3D objects and scenes, these 3D meshes are non-uniformly discrete bundles. These 3D models can undergo a variety of modifications to be used in multimedia applications. 3D models, which are made up of a substantial number of vertices and related faces, must be rendered in real-time for use in multimedia systems. In fact, complicated 3D models must be processed quickly and optimized for rendering in real-time applications. Applying modifications, which include things like compression, simplification, and watermarking, distort the 3D models' geometric form. The primary goal of this work was to provide an objective metric that correlates with the human visual system (HVS) and allows for the measurement of the quality of static 3D models regardless of the type of processing used.

Keywords: Perceptual Quality metric, 3D static models, 3D objective evaluation, Full-reference metrics, geometric distortion

1. Introduction

A three-dimensional (3D) representation of an object created on a computer using modeling software or a 3D scanner is referred to as a 3D model. Two methods exist to model a real object, either through a volume mesh, which is a generalization of surface meshes in three dimensions. (the surface meshes become volume meshes, which are assembled to form the total volume of the model); or through a surface mesh which is composed of vertices set connected by edges. The surface can be triangular, quadrilateral, or any polygon (1). The surface mesh is used for applications that do not require the internal content information of the models, for example, augmented/virtual reality applications.

1.1. *3D model representation*

A 3D model is composed of geometric data collection, representing the shape of the model, with other data that define the visual appearance of the model, such as surface properties, and mesh. The modeling of 3D objects can be carried out through several 3D representations.

The most simple and effective representation of a 3D object is a point cloud. It is composed of a finite set of points in a three-dimensional space with Cartesian coordinates. These points correspond to the shape of a certain geometry, but they are not connected to each other. The acquisition tool for this type of representation can be a 3D laser scanner, a manual digitizing or a reconstruction based on the sampling of the model. This representation can then be meshed to obtain a triangulated surface.

Additionally, Voxelization is a simple method for the volumetric representation of a model in three dimensions. The 3D model is composed of volumetric elements called voxels (volumetric pixels) to describe the model geometry. The main drawback of this approach is that it necessitates high resolution to define contours well. Therefore, it is necessary to reduce the voxel size. In fact, there is a noticeable rise in the number of voxels. This has the effect of increasing the file size, and therefore compression approaches become necessary.

Furthermore, the Non-Uniform Rational Basis-Spline (NURBS) representation (2), stands for the most widely used parametric tool for modeling surfaces. These surfaces are interpolation curves influenced by control points that have a weight. The curve follows the points in such a way that increasing the weight of a point will pull the curve closer to that point. This representation allows the creation of curves and smooth surfaces that are characterized by sampling ease. This type of modeling is mainly used in CAD software to represent industrial products in 3D.

Notably, the 3D surfaces are represented by meshes, which can be polygonal or triangular. The polygonal mesh is a set of polygonal facets that attempts to accurately approximate an actual three-dimensional object (3). Vertices, edges, and facets are the three different combinatorial components of this object's model. A group of points known as vertices in a three-dimensional space (x, y, z) create

a three-dimensional model. These are the points where polygons' edges or lines converge to produce the 3D representation. An edge is an extremity of the polygon connecting two vertices. A face is a 2D part of a polygon bounded by vertices and edges. The orientation of this model, which is determined by three rotational angles with respect to the X, Y, and Z axes, is what makes it unique.

The triangular mesh is one of the most widely used mathematical models owing to its simplicity and efficiency. The triangle represents the elementary geometric primitive; any polygonal mesh can then be represented following a triangulation of each facet into a triangular mesh without any change in geometry or topology. The triangulation can be generated using one of three methods, including Delaney-type methods and spatial decomposition methods. The triangle corresponds to the standard display unit of graphical accelerator cards dedicated to the management of three-dimensional scenes; independently of its representation, a mesh is triangulated during displays. The surface representation in triangular form allows compressing and visualizing 3D models efficiently. The most popular 3D computer-aided design (CAD) software uses the triangular mesh representation to model 3D object. In this context, Multimedia applications require the exchange of 3D models over the Internet to view and process them on computers.

These three-dimensional representations can be manipulated using a variety of techniques (4). Watermarking algorithms, for instance, can be applied to the 3D content to ensure their security. Compression can also be used to transport 3D models across a network with limited capacity or to minimize the size of the 3D model's file. (5). Other operations can be applied on 3D models, among which we mention: remeshing, reconstruction or simplification (6). These algorithms present an unavoidable alteration in the 3d models geometry. As a result, it's critical to quantify and assess how these deformations affect the final quality of 3D models.

1.2. Contributions

In general, 3D multimedia applications require a high level of detail, such as in surgery medical applications (7). Within this framework, telemedicine covers different medical practices, such as remote observation or data sharing (medical imaging, patient reports, etc...). Despite how specific a field of medicine is, it can use telemedicine. (8); for instance, teleconsultation, radiology, and medical hotline. Telemedicine has several benefits, including the development of home care, increased patient control and the prevention of complications, movement restrictions (particularly for old or disabled patients), and the facilitation of general practitioners and specialists consulting with one another. Additionally, 3D models are becoming a more widely used kind of media, particularly in gaming consoles and 3D TVs. By engaging with 3D environments, these products create new options for improving user experience.

Other multimedia applications require real-time rendering of 3D models. These models are made up of an important amount of linked faces and vertices. These

vertices and faces allow a more detailed and accurate representation of 3D objects. Indeed, real-time applications necessitate fast processing and optimization of complex 3D models. This needs applying treatments on 3D models to be displayed for a final user. It is noteworthy, at this stage of analysis, to point out that such operations as simplification, compression, and watermarking, occur distortions on the geometric shape of the 3D models.

In this regard, numerous studies have focused on developing subjective and objective metrics to evaluate the visual quality of distorted models. Subjective measures are assessments and conclusions reached by human observers in light of their impressions and past encounters. They frequently involve human assessors who look at or play around with 2D or 3D models and give feedback based on their tastes or standards. Individual preferences, biases, and individual assessor differences can all have an impact on subjective measures. Lighting, viewing angles, and context are a few examples of specific visualizing variables that might affect how subjectively we judge quality.

On the other hand, objective metrics, are quantifiable and are based on computational or mathematical models. These measurements are typically generated by computer programs or algorithms, which can analyze various aspects of 3D models or 2D images. Objective measurements aim to provide consistent and reproducible results, reducing the influence of human subjectivity and variability.

The basic objective of this work is to evaluate the perceptual quality of static 3D models using 3D objective metrics. This study examines the effectiveness of several 3D model processing strategies, such as compression, watermarking, and simplification. In this context, we want to improve that objective metric which correlates with the human visual system HVS, allowing us to predict the quality of static 3D models independently of the type of processing applied to the model. These measurements enable us to forecast the quality of the 3D static models, regardless of the modeling technique used.

The main objective of this research work is to evaluate the presented metrics, namely 3DrwPSNR and LWRMS, for measuring geometric distortion introduced by the addition of noise (both uniform and Gaussian) on a dataset encompassing a diverse set of 3D static models. These models exhibit variations in properties, such as mesh complexity, number of 3D points, and number of faces. The evaluation process involves a comprehensive analysis of how the metrics perform across these variations. In this paper, we shall detail two quality metrics which consist of estimating the perceptual quality of static 3D deformed models by integrating the properties of HVS.

2. State of the art

The majority of 3D multimedia applications require optimization treatments on the 3D models either to protect the 3D content, to visualize, or to reduce the size of 3D models. These treatments trigger modifications to the geometry of the processed

3D model. These distortions can be categorized into two groups, namely geometric distortions and topological or connectivity distortions. Geometric distortions only modify the part of the mesh geometry, namely the vertices coordinates. The quantity of vertices and their adjacency between vertices are always kept unchanged. Affine transformations, noise addition, smoothing and quantization of uniform coordinates are considered as geometric distortions.

In most cases, affine transformations are considered a common geometric operation on a 3D mesh rather than a malicious distortion. Computer graphics frequently employ these basic geometric adjustments to insert a 3D model into a scene.

In fact, a virtual scene is usually composed of 3D models that need to be rotated, scaled and translated to get the final result. These transformations are considered as usual modifications of the model which include translations, rotations, uniform scaling and any combination of these operations.

Furthermore, noise addition is a geometric distortion that consists of adding perturbations to the digital media to compromise watermarking. This is one of the classical distortions of digital watermarking which results in the addition of random noise on each component (x, y, z) of vertices. This method consists in perturbing the position of the mesh vertices by adding random displacement vectors. If the maximum modulus of these vectors is lower than the dimensions of the mesh, the model shape is preserved. The watermarking may be impacted by noise introduced to the geometry, assuming that the watermarking is encoded by modest changes to the geometric entities used to construct the 3D model, such as slightly shifting the mesh vertices' positions. Such noise can be added intentionally to undermine the watermarking or as part of common 3D model modifications like format conversion or lossy compression.

In addition, watermarking consists of inserting a digital signature in a host document. This insertion is carried out by combining certain components of the model with the signature elements. Watermarking algorithms differ in their methods of insertion, extraction, or even in the type of information. The watermarking algorithm relies on the choice of watermarking primitives and the choice of the insertion domain. The selection of watermarking primitives relies on the selection of a set of properties of the 3D model that will carry the information to be inserted.

Besides, smoothing is a technique that removes noise on a surface to give it a smoother appearance. Mesh smoothing is a geometric operation that is usually performed on meshes obtained through a 3D scan (9). In actuality, the approximation errors included during the surface reconstruction process by 3D scanners typically result in noisy surfaces. While maintaining the structure of the mesh, smoothing enhances its quality. The mesh geometry's high-frequency information is reduced by the smoothing procedure.

Moreover, compression processing is a geometric deformation of a 3D mesh. It is most often achieved by quantifying the coordinates of the vertices (10). By reducing the least significant bits, it is possible to encode the original vertex coordinates, which are represented by 32-bit floating point integers, with fewer bits.

In the same regards, connectivity distortions consist mainly of surface simplification, subdivision and cut distortion. This type of distortion begets a significant deterioration to the mesh. Some combinatorial elements (vertices, edges and facets) of the mesh can be removed while some new elements can be inserted. In general, connectivity distortions are much more difficult to handle than geometric distortions.

The goal of mesh simplification (11) is to decrease the number of vertices and triangles in a polygonal mesh model while keeping its structure. Typically, a simplification operation is applied several times to obtain the simplified mesh. It is often used to convey a lower level of resolution of the 3D mesh or to optimize a model by eliminating most of the non-salient triangles. The simplification consists in the elimination of some vertices, with a successive triangulation of the produced holes. This kind of treatment can be considered as a non-meaningful distortion, when it is applied to optimize the mesh, i.e., to eliminate oversampling of the model surface. Indeed, mesh simplification's goal is to display the 3D polygonal mesh with fewer triangles while maintaining the same shape, which speeds up the manipulation of 3D objects. Unlike simplification, subdivision adds edges and vertices to the 3D model to improve its perceptual quality through an iterative process that smoothens the mesh while increasing its density (12).

In 3D computer graphics, Subdivision is a technique used to portray a smooth surface by specifying a coarse linear polygonal mesh. The boundary of a recursive procedure that divides each polygonal face into smaller faces that more closely resemble the smooth surface can be used to compute the smooth surface from the coarse mesh. The result of this technique is a mesh that is more refined and has more polygonal faces than the initial mesh.

As with 2D images, it is possible to remove some geometry parts from a 3D model. Cutting distortion reduces the number of 3D point and face in the model by cutting it into several sections. The geometry and connectivity of the section remain constant after the cut. In some cases, this may result in an object that makes no sense, but in other cases, the remaining part might still be of some value. Indeed, cutting often affects the structure of the 3D model.

In this line, multiple studies have been devoted to remeshing, watermarking, simplification, and compression of geometric data of static 3D models (13). These experiments were carried out to enhance the data transfer and rendering processes across various mediums. The 3D models' contents are altered by these processes in a variety of ways, which causes them to be seen incorrectly (14). Therefore, it is crucial to look at the perceptual quality of visual data to assess new transmission or processing methods (such as compression or watermarking). Two methods can be used to assess the quality of distorted 3D models. The first technique uses measurements made by observers (subjective measurements), whereas the second uses computational methods to implement objective measures.

Subjective measurements are rarely employed, since each observer's concept of quality may vary based on personal preferences. To accurately assess the percep-

tual quality, subjective measurements need a precise setting (light, screen, etc.). Contrarily, objective metrics enable combining the behavior of the human visual system to assess the quality as it is seen by observers through developed algorithms that integrate the HVS's characteristics.

The basic aim of objective measures is to estimate the quality of a 3D model and its distorted version by taking into account the characteristics of human perception. To predict the quality of deformed models, three types of metrics are identified (15):

Full Reference Metrics: These metrics compare the deformed version of an object to the original model, where the original model is fully available. The goal is to quantify the differences between these two models accurately.

Reduced Reference Metrics: When the original model is not fully available but is represented by a vector with a reduced set of attributes. They assess the differences between this attribute vector and the vector derived from the deformed version.

No Reference Metrics: In scenarios where the original object is entirely unavailable, this type of metric is used. It either considers potential distortion types or operates without making assumptions about possible distortions. No-reference metrics leverage the understanding of the human visual system to evaluate quality.

Each type of these metrics serves a specific purpose in assessing the quality of objects and models. Objective measures are divided into two types: geometric metrics and metrics integrating visual perception properties (16). In this work, we present two objective metrics which investigate the HVS characteristic. The first commonly developed metrics are RMS and Hd.

RMS refers to a mathematical measure used to calculate the distance between two surfaces in 3D space (17). The distance $d(p, S')$ measures the distance between a point P belonging to the surface S and P' of the surface S' .

The Hausdorff distance concerns a geometric distance measure between two surfaces, (18). It is determined by calculating the minimum Euclidean distance between a point p on a continuous surface S and a corresponding point on another surface S' . Geometric metrics, being grounded solely in mathematical equations, do not align with human vision and perception.

Full-reference perceptual metrics are the most used to assess the quality of 3D models after processing since the reference model is usually fully available after processing. Karni and Gotsman set forward a metric named Geometric Laplacian (GL) based on roughness to evaluate their compressing approach of 3D models. One limitation of this measure resides in the fact that the models being compared must have the same connectivity. Sorkine et al elaborated a different version of GL1 called GL2 that introduces a small modification of the α value ($\alpha = 0.15$) (19).

In 2009, Lavou'e et al. introduced a perceptual metric known as the Structural Mesh Distortion Measure (MSDM) (20). This metric drew inspiration from the 2D image quality assessment measure known as SSIM (Structural Similarity Index), developed by Wang et al. (21). MSDM, akin to SSIM, employs differences in mean curvature amplitudes to evaluate perceptual quality between two 3D meshes. Subse-

quently, in 2011, Lavou'e et al. enhanced the MSDM metric by introducing MSDM2 (22). Both MSDM and MSDM2 rely on statistics related to curvature magnitude, allowing them to take into consideration the structural characteristics of the model.

In 2012, Vasa and Rus introduced a perceptual metric known as the Dihedral Angle Mesh Error (DAME) (23), designed to assess the perceptual quality of a deformed 3D model in comparison to its original counterpart. DAME's core principle involves calculating dihedral angles between pairs of neighboring triangles within the processed mesh.

Moving to 2014, Torkhani et al. presented a novel metric called the Tensor-based Perceptual Distance Measure (TPDM) (24). Unlike MSDM and MSDM2, TPDM not only evaluates the visual quality of a mesh based on curvature statistics but also incorporates complete curvature tensor information, including directionality. TPDM also addresses the visual masking effect by incorporating a weighted factor in the calculation of perceptually oriented curvature tensor distances.

Dong et al. (25) proposed an approach based on visual masking effects, saturation, and curvature intensity to assess the visual quality of 3D meshes.

In the realm of visual metrics for full-reference 3D meshes that incorporate 3D visual saliency, a metric was introduced that employs a spatial pooling strategy, visual salience weighting, statistical descriptors from a distortion map (26), and support vector regression. More recently, Lin et al. (27) presented a blind mesh quality assessment metric grounded in graph spectral entropy and spatial features, utilizing random forest regression to predict quality scores.

In this paper, we detail two full references of objective perceptual measures that are based on the geometric information of 3D mesh. These metrics incorporate the visual properties to estimate the deformation quantity of the 3D processed model.

3. Quality evaluation metrics

Evaluating the quality of 3D deformed models as perceived by human observers is a fundamental task, as the primary audience for most visual data consists of human observers. Perceptual quality can be assessed through either subjective measures, where assessments are made by human observers or objective measures derived from algorithmic processing.

For assessing the visual quality of 3D models, objective measures find significant utility in various image-based application domains, including reconstruction and medical image processing. In the context of 3D imaging, the examination of degraded 3D models often hinges on discerning differences in geometric information between the original model and its deformed version.

Within the existing metrics of literature, full reference metrics have emerged as the most prevalent approach for predicting the quality of 3D deformed models following processing. In this context, we introduce two full-reference objective metrics designed to accurately assess and evaluate the perceptual quality of 3D models.

3.1. 3D relative weighted PSNR (3DrwPSNR).

This metric is a full-reference image-based metric. 3DrwPSNR measures the perceptual quality of a distorted 3D model compared to its original version (28). This metric is engineered to encompass the visual characteristics of the Human Visual System (HVS) to effectively identify distortions within a 3D model. The nomenclature 'three-dimensional weighted relative PSNR' (3DrwPSNR) derives its essence from the evaluation of the relative positional disparities between two vertices within a 3D space. Within this framework, we have successfully integrated two crucial visual attributes, visual relativity and selectivity.

In the context of 3D geometry, it's essential to recognize that differences in the positions of an original vertex and its distorted version may not yield equivalent visual errors when compared to another vertex within the same model, even with identical positional differences.

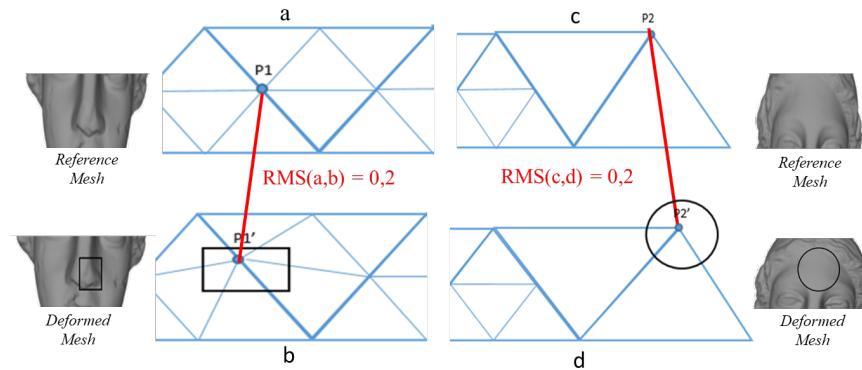


Fig. 1. Different deformed vertices. Original vertex (P1, P2), deformed version (P1', P2'), First deformation is located in a rough area (a, b), Second deformation is located in the smooth area (c, d).

The illustrated example in Figure 1 serves to elucidate a crucial observation. In this context, the numerical disparity between two positions, exemplified by vertex P1 and P1' (Figure 1.a and 1.b, enclosed within the square area), might bear an identical numeric value to that between another pair of vertices within the same 3D model (equal to 0.2), represented by positions P2 and P2' (as depicted in Figure 1, confined within the circular area).

3.1.1. Visual characteristics

3DrwPSNR metric incorporates the visual characteristics of the HVS through the calculation of the relative difference between the vertices distance of two 3D models (reference and deformed). Selective perception is the process by which humans

select, categorize, and analyze stimuli from the environment. Multiple researchers in the field of visual perception have demonstrated that perceptual attention is selective (29).

In 2D and 3D imaging, the selectivity of the human sensorial system is proved by Weber Law. Weber's Law, which describes the relationship between the magnitude of a stimulus and the perceived difference in that stimulus, can also be applied in the context of 3D vision, where it helps us understand how humans perceive differences in depth, size, and spatial relationships in three-dimensional environments. Weber's Law helps explain how sensitive we are to changes in the degree of binocular disparity, which in turn affects our perception of depth. For example, when viewing a 3D movie or a virtual reality (VR) environment, slight variations in the separation between the images presented to each eye can influence our perception of the models' depth. Weber's Law can be used to determine the just-noticeable difference (JND) in binocular disparity and, by extension, the smallest detectable change in depth perception. In 3D visualization and computer graphics, models often need to be scaled to create a sense of depth and perspective.

Weber's Law can be applied to determine how much a model size can change while maintaining the perception of its relative depth. By understanding the Weber fraction for size perception, designers and developers can adjust the size of models in 3D scenes to ensure that changes in size are perceivable and maintain the desired visual depth. Weber's Law can help designers and developers make informed decisions about the level of detail, and spatial scaling, needed to create a convincing and comfortable 3D environment. In this work, we are based on Weber's law to calculate the relative distance between the 3D points.

Therefore, visual selectivity is a significant human visual system property. This metric integrates this property through the computation of the distance's variance between the 3D model vertices to detect the detail level of the distorted models. In the 3D domain, distortions affect the rough regions that represent the areas of detail (the borders and the specified parts). As a result, distortions in these areas are more visible to the human visual system 2.

This phenomenon is related to the visual selectivity of the human eye. This property is modulated by the distance variance calculation between the deformed model vertices to estimate their detail level. Notably, models with a low variance value have a high level of detail (small mesh and rough area). Given a series of a Cartesian distance between each point and all the other 3D points of the distorted model (x_1, x_2, \dots, x_n), their mean distance is calculated using the following equation :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^n x_i \quad (1)$$

The variance is expressed through the average of the squared deviations from this mean :

$$V(X) = \frac{1}{N} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2)$$

where x_i represents the mean distance between a given point and all other points of the deformed model and \bar{x} indicates the mean of all calculated distances x_i .

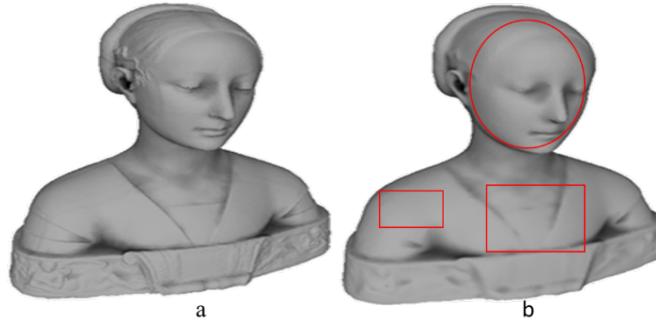


Fig. 2. 3D model with deformation in different area (a) Original model, (b) Distorted model

Figure 2 depicts that the degradation is more visible in areas with low variance (Figure 2.b circular surface) and less visible in areas with high variance (Figure 2.a square surface).

3.1.2. 3DrwPSNR Pipeline

The full reference perceptual metrics 3DrwPSNR can be utilized to assess the visual quality of 3D models without taking into account the connection of the 3D mesh. To measure deviations that are significant to human perception, this metric incorporates the HVS's visual characteristics. The fundamental concept behind this metric is that the separation between two vertices A and B is determined by the relative difference in where they are located in 3D space (28). Calculating the difference between the original model and the distorted version of a 3D model is necessary to determine the degree of distortion. The 3D models are taken into account by the 3DrwPSNR metric as a coordinate matrix. Figure 3 illustrates how this metric works.

First, a matching operation is defined between each vertex of the original model and its distorted version. This step consists of extracting the geometrical information of the 3D models. Then, the relative difference for each reference vertex and its new position in the deformed version is computed. At the same time, the variance of the distances between all points of the model to detect the level of detail of the model is estimated. Then, the global error is calculated using the weighted relative root-mean-square error (3DrwMSE) equation. Finally, the final value is quantified

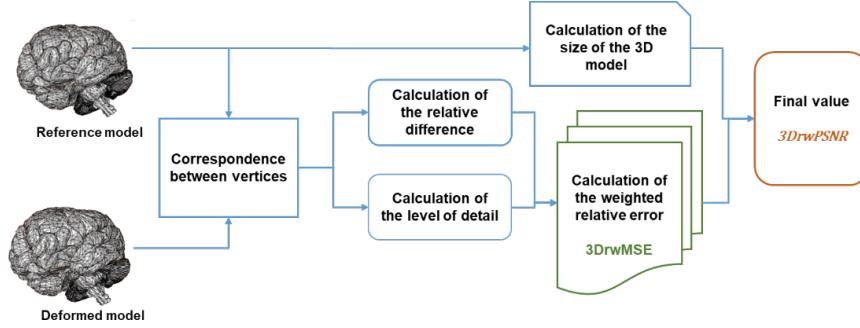


Fig. 3. Overview of the "3DrwPSNR" Metric's Evaluation Pipeline

through a logarithmic function between the maximum distance in the model and the relative weighted mean squared error.

To determine the final 3DrwPSNR value, the 3DrwMSE (relative weighted mean squared error) of the differences between the reference model points X and its deformed version Y is specified using the new definition of the 3DrwMSE:

$$3DrwMSE(X, Y) = \frac{1}{M} \sum_{j=1}^M \left(\frac{2 * \frac{(x_j - y_j)}{(x_j + y_j)}^2}{1 + var(y_j)} \right)^2 \quad (3)$$

where M is the number of 3D point, x and y are the reference model and the deformed model respectively. x_j represents a vector that contains the coordinates of the original 3D point. Typically, this notation denotes a point as (x, y, z) , where x represents the x-coordinate, y represents the y-coordinate, and z represents the z-coordinate of the point within the 3D space. y_j represents a vector that contains the coordinates of the distorted 3D point. Similar to the notation for the original point, it is often denoted as (x, y, z) , where x represents the x-coordinate, y represents the y-coordinate, and z represents the z-coordinate of the distorted point within the 3D space.

$var(y_j)$ is the distance variance between all vertices of the deformed model, the quotient refers to the relative difference between both positions of the same vertices, namely x_j the reference model vertices and the y_j the deformed model vertices. The value of 3DrwPSNR is calculated using the following equation:

$$3DrwPSNR = 10 * \log_{10} \frac{dx_{max}^2}{3DrwMSE} \quad (4)$$

The presented metric measures the weighted relative error between the reference model, x , and the deformed model y , where dx_{max}^2 corresponds to the maximum distance between all points of the reference model.

3.2. Locally weighted root-mean-square error (LWRMS)

The second measure that we present in this paper is denoted LWRMS (locally weighted root-mean-square error). This metric stands for a full reference model-based metric that investigates the HVS properties in order to estimate the visual quality of a deformed 3D model as perceived by the human observer. LWRMS is based on a known fundamental mathematic measure called RMS "root-mean-square error" which measures the deformation of a 3D model in a purely mathematical way.

LWRMS measures the visual quality of a distorted 3D model compared to its original version, considering the connectivity constraint between meshes(30). Furthermore, several studies emphasized that the distortion quantity of a 3D deformed model depends on the level of detail and the rendering size of the 3D shape. This specificity is an intrinsic property of visual perception. If the distortion is located in an area that contains a high level of detail, the distortion is more significant for HVS. In this regard, LWRMS metric incorporates the human visual system properties to measure the perceptual quality of different geometric distortions that can affect the shape of 3D models.

3.2.1. LWRMS pipeline

The majority of applications in the 3D field invest the fundamental measure denoted RMS to assess the visual quality between 3D models referring to its simplicity and speed, though RMS does not correlate with subjective scores since it is a mathematical metric. Therefore, To achieve agreement between the objective measurements acquired by the presented metric and the subjective scores, we apply the principle of RMS distance between two 3D vertices in space while combining the principles of visual relativity and selectivity. To explore the properties of HVS, we calculated the distance between a vertex and its neighbors to measure the distortion area level of detail and to determine the invariance to scale changes in order to normalize the error to the size of model figure 4.

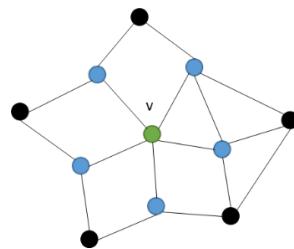


Fig. 4. Visualization of Vertex (v) and Its Neighboring Vertices

The LWRMS value is calculated in four steps : First, a matching operation is defined between each vertex of the reference object and its deformed version.

Second, the local distance between each point and its neighbor is calculated to detect the detail areas. Meanwhile, the 3D model size is estimated. Subsequently, the error between the points weighed by the level of detail of the selected area is specified. Finally, the final score is calculated through the average error of the distances normalized by the size of model 5.

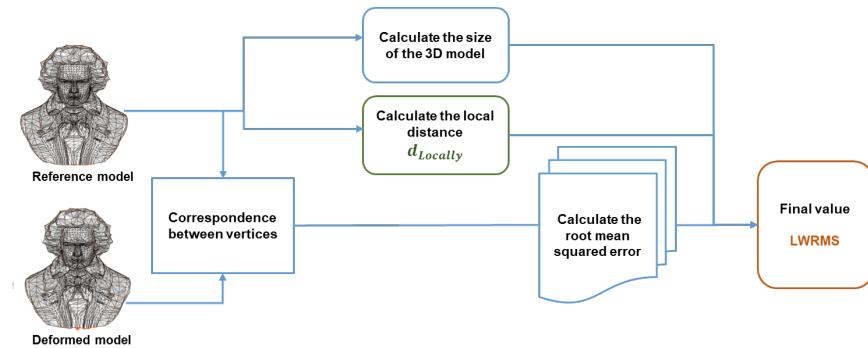


Fig. 5. Step-by-Step Pipeline of the "LWRMS" Metric

The LWRMS metric is defined as:

$$\text{LWRMS}(A, B) = \frac{1}{D_{\max}(A)} * \frac{1}{M} \sum_{i=1}^M \frac{\|P_A^i - P_B^i\|_2^2}{d_{Locally}(P_A^i)} \quad (5)$$

where A refers to the reference model, B is its distorted version, $D_{\max}(A)$ represents the higher distance between two points in the reference model and M indicates the number of vertices in the 3D model.

$\|\cdot\|$ expresses the Euclidean norm, P_i designates the coordinates of the i -th vertex and $d_{Locally}(P_A^i)$ corresponds to the distance between the selected 3D point i and its neighbors, which is calculated as follows:

$$d_{Locally}(P_A^i) = \frac{1}{N} \sum_{\substack{i=1 \\ j \in vois_i}}^N \|P_A^i - P_A^j\|_2 \quad (6)$$

where N is the neighbor number of the selected vertex i
 $vois_i$ contains all neighboring vertices of the selected vertex, i .

The index j represents 3D points belonging to the set of $vois_i$ such that P_i and P_j are linked.

3.3. Visual properties

Mathematical metrics that take into account the characteristics of the human visual system might improve the evaluation of the visual quality of 3D models. Vertices and faces are the two main components that define these models. The number of faces increases as the number of vertices increases.

Vertex positions typically alter as a result of the degradation that affects the 3D model. A local deformation is more noticeable to the human observer in an area with a high level of detail (near vertices and small faces) than in a low level of detail.

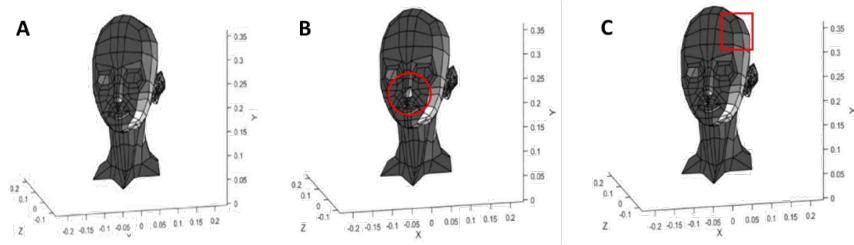


Fig. 6. Deformation applied on the Head model in different areas. (a) Reference model, (b) Distortion in a high detail area, (c) Distortion in a low detail area

Figure 6.A depicts the reference model, 6. B portrays the deformed model with distortion in a high detail area and, 6.C presents the deformed model with distortion in an area with low level of detail.

The following values are obtained by the developed metric LWRMS and the RMS measure; LWRMS (A, B) = 4.35 > LWRMS (A, C) = 1.30, RMS (A, B) = 0.0173 RMS (A, C) = 0.0173. This suggests that the LWRMS metric is sensitive to the level of detail. Figure 6 illustrates the same quantity of deformation located in different detail areas. We, therefore, obtain the same error value with the RMS, but the LWRMS value is different. LWRMS provides a better forecast of the distortion quantity by taking into account the difference between two 3D models based on their level of detail. The level of detail is modeled with the average of the distance measured locally between each 3D vertex and its neighboring vertices. Thus, a low value implies that the selected point is located in a detailed area and that this distortion will be more significant for the human observer. On the other hand, different-sized screens can be used to see three-dimensional models. Additionally, once 3D models are built, their look is influenced by both the model's size and geometry in addition to the model's geometry. A 3 mm deformation, for instance, does not have the same perceptual importance in a 3D model with an overall size of 10 mm as it does in a model with an overall size of 100 mm. In fact, a measure of perceptual quality must take into account the 3D model's size because it affects how humans see it. (Figure 6).

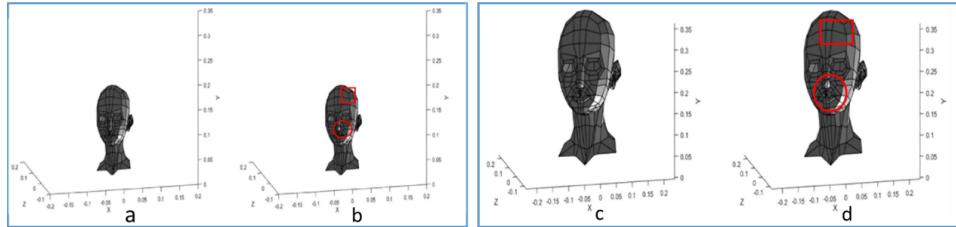


Fig. 7. Head model displayed with different scales. (a) Original model (small scale), (b) Distorted model (small scale) (c), Original model (large scale), (d) Distorted model (large scale)

Figures 7(a) and 7(c) highlight the reference model at different scales, 7(b) and 7(d) depict a distorted version with another scale (homothety ratio: 0.5).

Applying the LWRMS and RMS measures, the following results are obtained: LWRMS (a, b) = 1.3 LWRMS (c, d) = 1.3 RMS(a, b) = 0.0173 RMS(c, d) = 0.0087. These findings suggest that the LWRMS metric accounts for the distorted areas' size level.

The quality assessment metric estimates the visual quality of a 3D model as it is rendered on the screen (Figure 7), regardless of its scale. The LWRMS metric integrates this property by computing the maximum distance between the pair's vertex of the original 3D model and its deformed version.

4. Experimental study

To assess the efficiency of the presented metrics, two distortions are applied (uniform noise, and Gaussian noise) on a set of 3D models that have different characteristics: mesh topology, points number, and face number. These models are used to test different types of 3D graphics treatments. Table 1 foregrounds the description of each model of the studied corpus.

The different distortions are applied to the models using the "MEPP" platform. MEPP (MEsh Processing Platform) is an environment dedicated to the processing and visualization of three-dimensional meshes (31). This platform is intended for engineers and researchers. It offers users a very simple handling to process 3D meshes. The first distortion consists of applying a uniform noise with three strengths. The second distortion consists of applying a Gaussian noise with three strengths.

4.1. First distortion by applying a uniform noise

Uniform noise is a type of noise that has a constant amplitude over a specified range. It is often used in various fields, including signal processing and image processing, to simulate random disturbances or errors.

To generate uniform noise with different strengths or variances, the width of the range from the sample random values should be adjusted. In a uniform noise signal,

Table 1. 3D models description of the studied corpus

Model name	Application field	Acquisition	Number of vertices	Number of face	Mesh topology
Stanford Bunny	3D image synthesis	3D laser scan of a ceramic rabbit figure	2503	4968	triangular and irregular
Head	3D image synthesis	CAD software	475	442	polygonal and regular
Hand	3D medical image synthesis	3D scan of a human hand	41122	41120	polygonal and regular
Skull	3D medical image synthesis	3D scan of a human skeleton	40062	39288	polygonal and regular
Brain	3D medical image synthesis	3D scan of the human brain	18844	36752	triangular and irregular
Duck	animation, games and VR/AR application	CAD software	8590	8588	polygonal and regular
Beethoven	animation, games and VR/AR application	3D laser scan of a Ludwig van Beethoven figure	2655	8588	triangular and irregular

a wider range results in a larger variance (greater strength), while a narrower range results in a smaller variance (lesser strength).

To calculate uniform noise with different strengths, we start by defining a range for the uniform noise. This range is determined by a minimum (a) and a maximum (b) value. The width of this range will control the strength of the noise. Then, choose a random number generator that can generate random values uniformly distributed within the specified range. Next, a generation of a random value within the specified range is needed. The width of the range ($b - a$) controls the strength of the noise. A wider range will result in stronger noise, while a narrower range will result in weaker noise. If the random values generated by the random number generator are in a different range or have a different mean than the desire, the values can be scaled to match the desired value using the following equation:

$$\text{Scaled Value} = \frac{(G_v - a) \cdot (D_r)}{b - a} + D_m \quad (7)$$

Where G_v presents the random value generated by the random number generator. a is the minimum value of the original range, b is the maximum value of the original range, D_r is the desired range of the uniform noise and D_m is the minimum value of the desired range. To change the strength of the uniform noise, modify the values of a and b . A wider range (greater difference between a and b) will result in stronger noise, while a narrower range will result in weaker noise. The first distortion is applied to the original version of each model by adding the uniform noise distortion with three strengths (strength = 0.0050, strength = 0.0100 and strength = 0.0150). The results of the uniform noise application are portrayed from figure 8 to figure 14.

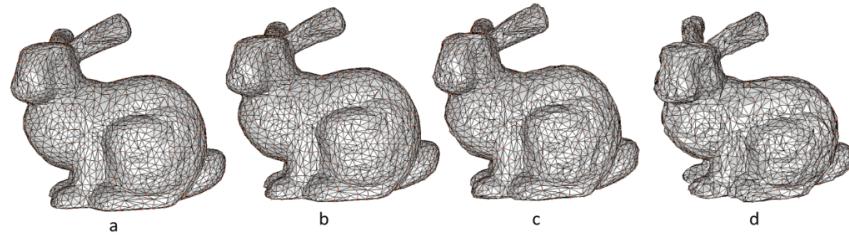


Fig. 8. "Bunny" model after "uniform noise" distortion

Figure 8 illustrates the "Bunny" model under various conditions: (a) Reference model, (b) "Bunny" deformed model with uniform noise strength = 0.0050, (c) "Bunny" deformed model with uniform noise strength = 0.0100, (d) "Bunny" deformed model with uniform noise strength = 0.0150

Table 2. Quantitative results after applying uniform noise to the "Bunny" model

	uniform noise strength = 0.0050	uniform noise strength = 0.0100	uniform noise strength = 0.0150
3DrwPSNR	0,805	22,964	18,197
LWRMS	0,069	0,282	0,6265
MSDM	0,34	0,50	0,61

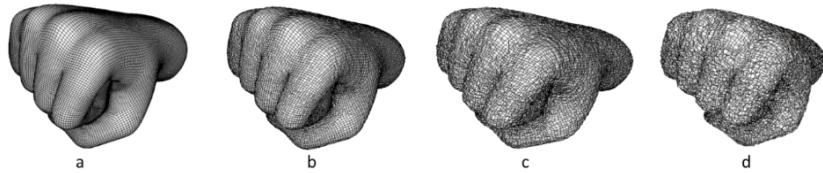


Fig. 9. "Hand" model after "uniform noise" distortion

Figure 9 illustrates respectively the "Hand" model in terms of (a) original model, (b) "Hand" distorted model applying uniform noise strength = 0.0050, (c) the distorted version of "Hand" model applying uniform noise strength = 0.0100, and (d) distorted "Hand" model applying the uniform noise strength = 0.0150.

Table 3. Quantitative results after applying uniform noise to the "Hand" model

	uniform noise strength = 0.0050	uniform noise strength = 0.0100	uniform noise strength = 0.0150
3DrwPSNR	104,29	98,22	94,61
LWRMS	0,0004	0,0014	0,0032
MSDM	0,73	0,80	0,82

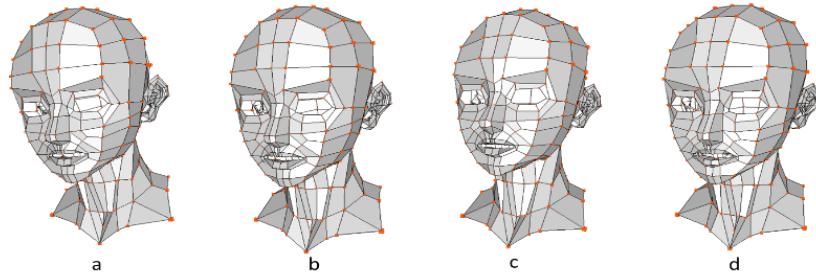


Fig. 10. "Head" model after "uniform noise" distortion

Figure 10 present respectively the "Head" model in terms of (a) original model, (b) "Head" model distorted with uniform noise strength = 0.0050, (c) "Head" model distorted with uniform noise strength = 0.0100 and (d) "Head" model distorted with uniform noise strength = 0.0150

Table 4. Quantitative results after applying the uniform noise on the "Head" model

	uniform noise strength=	uniform noise strength=	uniform noise strength=
	0.0050	0.0100	0.0150
3DrwPSNR	11,76	11,95	10,21
LWRMS	0,053	0,222	0,502
MSDM	0,18	0,27	0,35

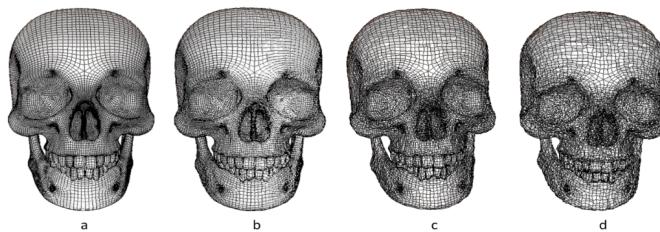


Fig. 11. "Skull" model after "uniform noise" distortion

Figure 11 presents respectively the "Skull" model in terms of (a) Reference "Skull" model, (b) "Skull" distorted model by a uniform noise strength = 0.0050, (c) "Skull" distorted model by a uniform noise strength = 0.0100, and (d) "Skull" distorted model by a uniform noise strength = 0.0150.

Table 5. Quantitative results after applying uniform noise to the "Skull" model

	uniform noise strength=	uniform noise strength=	uniform noise strength=
	0.0050	0.0100	0.0150
3DrwPSNR	60,93	63,15	64,61
LWRMS	0,0003	0,0012	0,0028
MSDM	0,55	0,69	0,74

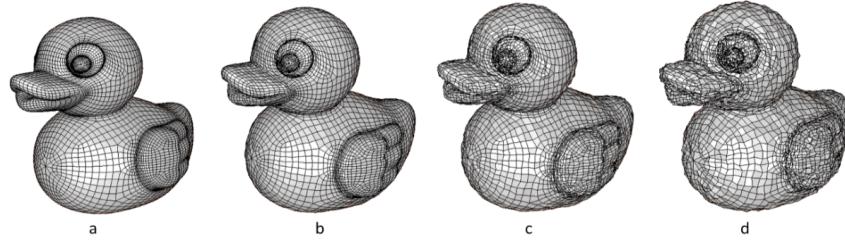


Fig. 12. Duck model after the "uniform noise" distortion

Figure 12 presents the "Duck" model, respectively, (a) Reference "Duck" model, (b) "Duck" distorted model by uniform noise strength = 0.0050, (c) "Duck" distorted model by uniform noise strength = 0.0100 and (d) "Duck" distorted model by uniform noise strength = 0.0150.

Table 6. Quantitative results after applying uniform noise on the "Duck" model

	uniform noise strength=	uniform noise strength=	uniform noise strength=
	0.0050	0.0100	0.0150
3DrwPSNR	34,74	35,84	34,85
LWRMS	0,0002	0,0007	0,0015
MSDM	0,57	0,69	0,73

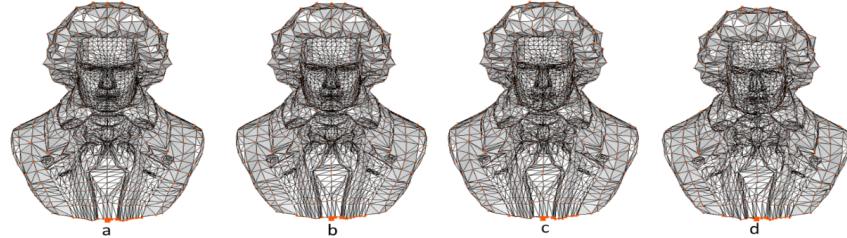


Fig. 13. "Beethoven" model after the "uniform noise" distortion

Figure 13 incorporates respectively the "Beethoven" model in terms of (a) "Beethoven" reference model, (b) "Beethoven" distorted model by a uniform noise strength = 0.0050, (c) "Beethoven" distorted model by a uniform noise strength = 0.0100 and (d) "Beethoven" distorted model by a uniform noise strength = 0.0150

Table 7. Quantitative results after applying the uniform noise on the "Beethoven" model

	uniform noise strength=	uniform noise strength=	uniform noise strength=
	0.0050	0.0100	0.0150
3DrwPSNR	50,97	30,08	44,82
LWRMS	0,0608	0,2469	0,5346
MSDM	0,25	0,40	0,49

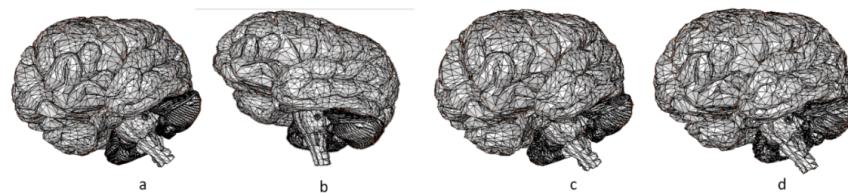


Fig. 14. Brain" model after "uniform noise" distortion

Figure 14 integrates, respectively, the "Brain" model in terms of (a) "Brain" reference model, (b) "Brain" distorted model by uniform noise strength = 0.0050, (c) "Brain" distorted model by uniform noise strength = 0.0100 and (d) "Brain" distorted model by uniform noise strength = 0.0150.

The obtained results through the developed metrics "3DrwPSNR and LWRMS" detailed in tables 2,3,4,5,6,7,8 are indicative that each time we increase the intensity

Table 8. Quantitative results after applying the uniform noise on the "Brain" model

	uniform noise strength= 0.0050	uniform noise strength= 0.0100	uniform noise strength= 0.0150
3DrwPSNR	29,85	7,17	33,32
LWRMS	0.0930	0.3717	0.8434
MSDM	0,53	0,60	0,63

of noise, the quality of the model degrades in a relative way. Additionally, the human eye visually detects this degradation relative to the amount of deformation.

The 3DrwPSNR metric yields good results on the Bunny model (uniform noise strength = 0.0050, 3DrwPSNR = 0.80) and (uniform noise strength = 0.0150, 3DrwPSNR= 18.19). The value of 3DrwPSNR decreases as the quality degrades, which is visually proven by the obtained results.

According to the second metric LWRMS (uniform noise strength =0.0050, LWRMS = 0.0690) and (uniform noise strength = 0.0150, LWRMS = 0.6265), the value of LWRMS increases when the quality degrades. This is relative to the average error calculated by our metric. In addition, with the models "Hand, Head and Beethoven", the values of the 3DrwPSNR and LWRMS metrics offer satisfactory results when compared to human vision in figures 9, 10, 13.

For the "Skull and Duck" models, LWRMS provides good results compared to the 3DrwPSNR metric: for the "Skull" model (uniform noise strength = 0.0050, LWRMS = 0.0003, 3DrwPSNR = 60.9321) and (uniform noise strength = 0. 0150, LWRMS = 0.0028, 3DrwPSNR= 64.6139), on the "Duck" model (uniform noise strength = 0.0050, LWRMS = 0.0002, 3DrwPSNR= 34.7414) and (uniform noise = 0.0150, LWRMS = 0.0015, 3DrwPSNR = 34.8529). This refers basically to the consideration of connectivity between meshes.

4.2. Second distortion by applying a Gaussian noise:

In order to apply Gaussian noise to a 3D model, we have added noise to all the coordinates of the 3D vertices. For a single vertex at coordinates (x, y, z), the Gaussian noise is added as follows:

$$G_X = X + \epsilon x, \quad G_Y = Y + \epsilon y, \quad G_Z = Z + \epsilon z \quad (8)$$

Where: G_X , G_Y , and G_Z are the new coordinates of the vertex with noise applied.

X , Y , and Z are the original coordinates of the vertex.

ϵ_x , ϵ_y , and ϵ_z are random values drawn from a Gaussian distribution with mean μ and standard deviation σ . These values represent the noise added to each coordinate.

Adding a "strength" parameter to Gaussian noise means that the intensity of the noise can be controlled. In this context, the standard deviation (σ) of the Gaussian distribution can be scaled to control the strength of the noise. If we want to add a precise noise quantity "strength", the standard deviation (σ) of the Gaussian distribution is adjusted as follows:

$$\sigma = \text{base_stddev} \cdot \text{noise_strength} \quad (9)$$

Where: σ is the modified standard deviation used to control the strength of the noise. base_stddev is the base standard deviation for the Gaussian distribution, which determines the noise's nature. noise_strength is the parameter that controls how strong or weak the noise is. A value of 1 would represent the base noise strength, while values greater than 1 would increase the strength, and values less than 1 would decrease it. The second treatment is applied to the original version of each model by adding the Gaussian noise distortion with three strengths (strength = 0.0050, strength = 0.0100 and strength = 0.0150). The results of the Gaussian noise application are plotted from figure 15 to figure 21.

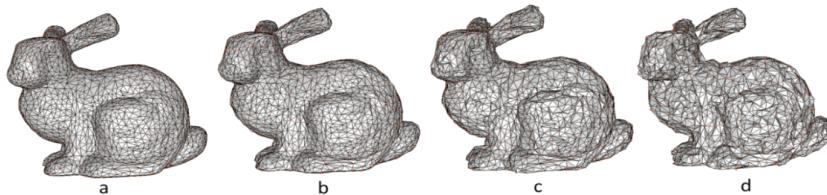


Fig. 15. "Bunny" model after the "Gaussian noise" distortion

Figure 15 clarifies respectively the "Bunny" model in terms of (a) Reference "Bunny" model, (b) "Bunny" distorted model by Gaussian noise strength = 0.0050, (c) "Bunny" distorted model by Gaussian noise strength = 0.0100 and (d) "Bunny" distorted model by Gaussian noise strength= 0.0150.

Table 9. Quantitative results after application of Gaussian noise on the "Bunny" model

	Gaussian noise strength = 0.0050	Gaussian noise strength = 0.0100	Gaussian noise strength = 0.0150
3DrwPSNR	18,003	9,297	18,360
LWRMS	0,0002	0,0009	0,0020
MSDM	0,46	0,65	0,72

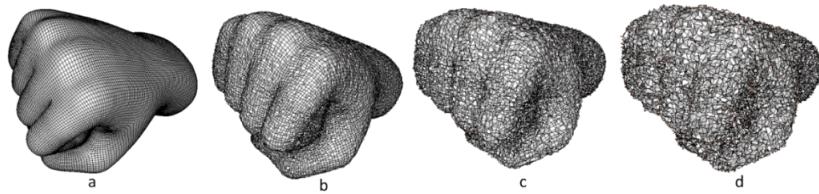


Fig. 16. "Hand" model after the "Gaussian noise" distortion

Figure 16 elucidates, respectively, the "Hand" model in terms of (a) Reference model "Hand", (b) "Hand" distorted model by Gaussian noise strength = 0.0050, (c) "Hand" distorted model by Gaussian noise strength = 0.0100 and (d) "Hand" distorted model by Gaussian noise strength = 0.0150.

Table 10. Quantitative results after application of Gaussian noise on the "Hand" model

	Gaussian noise strength = 0.0050	Gaussian noise strength = 0.0100	Gaussian noise strength = 0.0150
3DrwPSNR	99,3308	93,4383	89,8874
LWRMS	0,0011	0,0043	0,0097
MSDM	0,79	0,83	0,83

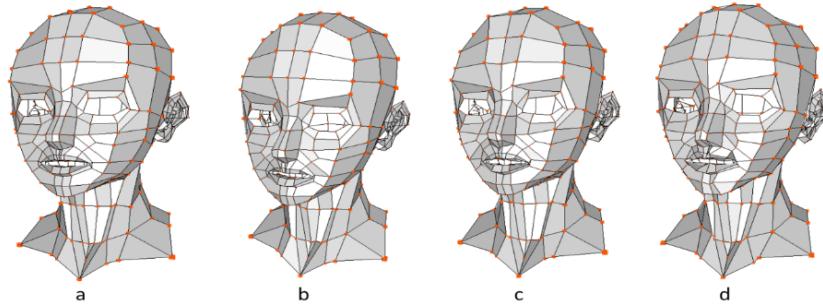


Fig. 17. "Head" model after the "Gaussian noise" distortion

Figure 17 reveals, respectively, the "Head" model in terms of (a) Reference model, (b) "Head" distorted model by Gaussian noise strength = 0.0050, (c) "Head" distorted model by Gaussian noise strength = 0.0100 and (d) "Head" distorted model by Gaussian noise strength = 0.0150.

Table 11. Quantitative results after application of Gaussian noise on the "Head" model

	uniform Gaussian	noise =	uniform Gaussian	noise =	uniform Gaussian	noise =
	0.0050		0.0100		0.0150	
3DrwPSNR	9,4682		6,4180		13,1434	
LWRMS	0.0002		0.0007		0.0016	
MSDM	0,24		0,42		0,50	

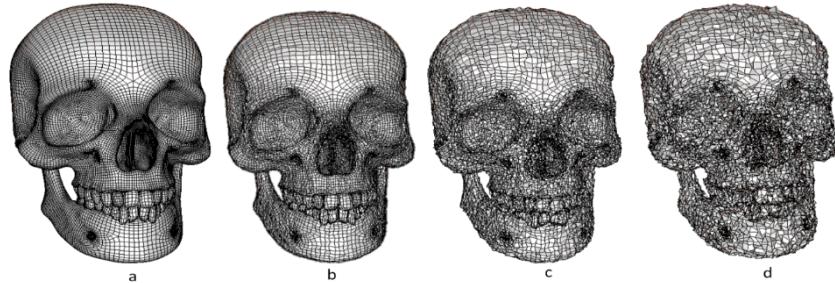


Fig. 18. "Skull" model after the "Gaussian noise" distortion

Figure 18 demonstrates respectively the "Skull" model in terms of (a) Reference "Skull" model, (b) "Skull" distorted model by Gaussian noise strength = 0.0050, (c) "Skull" distorted model by Gaussian noise strength = 0.0100 and (d) "Skull" distorted model by Gaussian noise strength = 0.0150.

Table 12. Quantitative results after application of Gaussian noise on the "Skull" model

	Gaussian noise strength = 0.0050	Gaussian noise strength = 0.0100	Gaussian noise strength = 0.0150
3DrwPSNR	60,7247	37,2414	54,4358
LWRMS	0,0011	0,0045	0,0102
MSDM	0,24	0,75	0,77

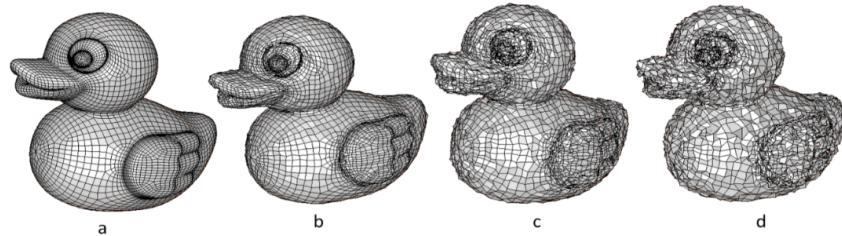


Fig. 19. "Duck" model after the "Gaussian noise" distortion

Figure 19 identifies, respectively, the "Duck" model in terms of (a) Reference "Duck" model, (b) "Duck" distorted model by Gaussian noise strength = 0.0050, (c) "Duck" distorted model by Gaussian noise strength = 0.0100 and (d) "Duck" model distorted by Gaussian noise strength = 0.0150.

Table 13. Quantitative results after application of Gaussian noise on the "Duck" model

	Gaussian noise strength = 0.0050	Gaussian noise strength = 0.0100	Gaussian noise strength = 0.0150
3DrwPSNR	25,1082	30,7535	27,6421
LWRMS	0,0005	0,0020	0,0045
MSDM	0,67	0,75	0,77

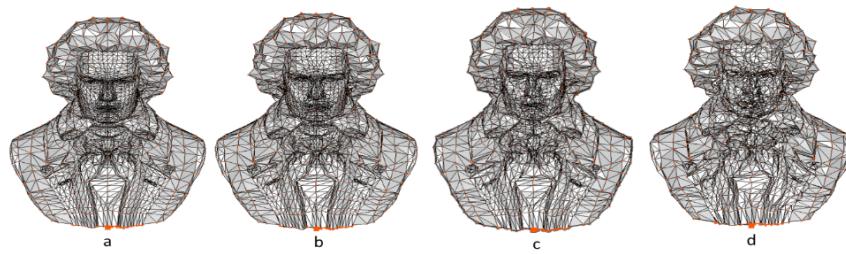


Fig. 20. "Beethoven" model after the "Gaussian noise" distortion

Figure 20 introduces, respectively, the "Beethoven" model in terms of (a) reference model, (b) "Beethoven" distorted model by a Gaussian noise strength = 0.0050, (c) the "Beethoven" distorted model by a Gaussian noise strength = 0.0100 and (d) the "Beethoven" distorted model by a Gaussian noise strength = 0.0150.

Table 14. Quantitative results after application of Gaussian noise on the "Beethoven" model

	Gaussian noise strength = 0.0050	Gaussian noise strength = 0.0100	Gaussian noise strength = 0.0150
3DrwPSNR	42,0311	31,9184	40,6843
LWRMS	0,0002	0,0007	0,0016
MSDM	0,36	0,5	0,58

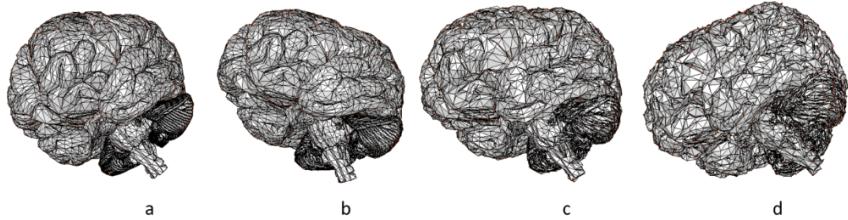


Fig. 21. "Brain" model after the "Gaussian noise" distortion

Figure 21 represents, respectively, the "Brain" model in terms of (a) the reference "Brain" model, (b) the "Brain" distorted model by Gaussian noise strength = 0.0050, (c) the "Brain" distorted model by Gaussian noise strength = 0.0100, (d) the "Brain" distorted model by Gaussian noise strength = 0.0150.

Table 15. Quantitative results after application of Gaussian noise on the "Brain" model

	Gaussian noise strength = 0.0050	Gaussian noise strength = 0.0100	Gaussian noise strength = 0.0150
3DrwPSNR	40,9506	30,2151	27,8255
LWRMS	0,0003	0,0011	0,0026
MSDM	0,59	0,64	0,67

Examining the obtained results by our developed metrics, "3DrwPSNR and LWRMS" detailed in tables 9,10,11,12,13,14 in addition to 15 several observations. First, we notice that the LWRMS metric performs better on all models compared to the 3DrwPSNR metric.

Both 3DrwPSNR and LWRMS metrics provide a good quality estimation on the majority of models: on the "Bunny" model (Gaussian noise strength= 0.0050, 3DrwPSNR = 18.0031, LWRMS= 0.0002) and (uniform noise strength = 0. 0150, 3DrwPSNR= 18.3606, LWRMS= 0.002); on the "Hand" model (Gaussian noise strength = 0.0050, 3DrwPSNR = 99.3308, LWRMS= 0.0011) and (uniform noise strength = 0.0150, 3DrwPSNR= 89.8874, LWRMS= 0.0097); on the "Head" model (Gaussian noise strength = 0. 0050, 3DrwPSNR = 9.4682, LWRMS = 0.0002) and (uniform noise strength = 0.0150, 3DrwPSNR= 13.1434, LWRMS= 0.0016); on the "Skull" model (Gaussian noise strength = 0.0050, 3DrwPSNR = 60.7247, LWRMS= 0.0011) and (uniform noise strength = 0. 0150, 3DrwPSNR= 54.4358, LWRMS= 0.0102); and on the "Beethoven" model (Gaussian noise strength = 0.0050, 3DrwPSNR = 42.0311, LWRMS= 0.0002) and (uniform noise strength = 0.0150, 3DrwPSNR= 40.6843, LWRMS= 0.0016). The results obtained in the above-plotted tables are visually proven in figures 15, 16,17,18,19,20 and 21. Indeed, the "Duck"

model LWRMS reveals better results compared to the 3DrwPSNR metric on the "Duck" model (uniform noise strength = 0.0050, LWRMS = 0.0005, 3DrwPSNR= 25.1082) and (uniform noise strength = 0.0150, LWRMS = 0.0045, 3DrwPSNR = 27.6421). The obtained values by the presented metrics indicate that these metrics correlate well with human vision.

5. Computational costs analysis

In this section, we provide a comprehensive comparison of the execution time for both Gaussian and uniform noise across all 3D models in our study using the presented metrics (3DrwPSNR and LWRMS). The presented values in this section are calculated between each 3D reference model and their three deformed versions. The following table (16) shows the results of various experiments with different types of noise applied to different 3D models. The computational cost is measured in seconds.

Table 16. Computational cost Results

Model	Noise Type	LWRMS	3DrwPSNR
bunny	Gaussian noise	9.879	7.946
	Uniform noise	37.999	2.480
hand	Gaussian noise	61.022	7.873
	Uniform noise	59.611	46.294
head	Gaussian noise	8.065	1.233
	Uniform noise	5.041	0.551
skull origin	Gaussian noise	55.445	30.907
	Uniform noise	57.279	44.279
Duck	Gaussian noise	20.707	14.039
	Uniform noise	23.990	7.326
Beethoven	Gaussian noise	9.527	2.437
	Uniform noise	9.496	2.418
brain	Gaussian noise	53.162	20.559
	Uniform noise	31.959	20.581

Regarding Table 16 we observed a notable difference in execution time between the LWRMS and 3DrwPSNR metrics. In general, the execution time of the 3DrwPSNR metric was consistently lower than that of the LWRMS. This disparity can be attributed to the inherent simplicity of the 3DrwPSNR metric. The 3DrwPSNR metric requires less computational effort, making it a faster option for evaluating the 3D model's quality.

Despite the faster execution time of 3DrwPSNR, the LWRMS metric presented competitive results. The relatively higher execution time of LWRMS is primarily

due to the complex process of extracting the neighbors of each vertex. For each point within a 3D model, a neighbor graph is meticulously extracted, involving numerous calculations and data manipulations. Consequently, this process demands more computational resources and time.

The choice of the optimal metric also hinges on specific characteristics of the 3D models, including the number of vertices and the mesh topology. These observations led us to conclude that 3D models with a triangular mesh structure tend to exhibit lower execution times when either metric is employed. This is primarily because the calculation of geometric information is more straightforward in such models, given their well-structured triangular connectivity. Conversely, for 3D models with alternative mesh topologies, such as irregular or non-triangular structures, an additional discretization step is often necessary. This step aims to simplify the calculation process by converting complex topologies into triangular forms, a transformation that demands additional computation time. This insight explains why models with triangular meshes typically achieve a more economical computational cost. From this experiment, researchers should carefully evaluate these factors when choosing between these metrics for their specific applications. The execution time represents an advantage to using 3DrwPSNR as a 3D evaluation metric.

6. Benchmark databases used in mesh quality assessment

To evaluate the mesh quality assessment approaches, there are many different databases, such as the LIRIS/EPFL database. This is a noteworthy benchmark database used in the assessment of mesh quality for 3D models. Developed collaboratively by the Laboratory of Computer Science, Image, and Information Systems (LIRIS) at the Swiss Federal Institute of Technology in Lausanne (EPFL), this database offers a comprehensive collection of 3D meshes that serve as valuable resources for evaluating mesh generation and processing algorithms. The LIRIS/EPFL database includes a diverse range of 3D models, encompassing various levels of complexity and geometry types, which make it an essential tool for researchers and practitioners in computer graphics, computational geometry, and related fields. Researchers can use this database to test and validate their mesh quality assessment methods, ensuring that 3D models are of high quality and suitable for a wide range of applications, from computer-aided design to virtual reality and simulation.

In addition to the LIRIS/EPFL database, there are several other notable benchmark databases used for assessing mesh quality in the realm of 3D modeling and computational geometry. These databases include the Stanford 3D Scanning Repository, a widely recognized resource containing a vast array of 3D models captured from real-world objects and scenes. Similarly, the ModelNet dataset offers a diverse collection of 3D CAD models, making it a valuable asset for evaluating mesh generation techniques in the context of computer-aided design. Moreover, the BRep-DB, which focuses on boundary representation models, provides a unique set of chal-

lenges for mesh quality assessment. The difference between these datasets and the LIRIS/EPFL dataset is the availability of the subjective measures value for each deformed model, which is essential in the process of objective metrics validation. These databases play a crucial role in advancing the quality and reliability of 3D meshes, ensuring their suitability for applications ranging from virtual reality and video games to finite element analysis and medical imaging. Researchers and engineers leverage these resources to benchmark their algorithms, ultimately contributing to the enhancement of 3D modeling and simulation technologies.

7. Comparative study

To validate the developed metrics, the general-purpose database developed by LIRIS/EPFL was used. We infer that the investigation of the visual properties of the HVS in the objective measures has a very important role in terms of yielding results that correlate well with the subjective measures. In this work, we presented two objective metrics; the first one is based on the principle of the 2D perceptual quality metric "rwPSNR" which exploits the relativity and visual selectivity of the HVS. It allows the detection of the amount of distortion by referring to the original model. The distortion calculation is carried out vertex by vertex without taking into account the connectivity between the meshes. The second metric is based on the principle of the "RMS" measure, which calculates the quantity of deformation of a model in a mathematical way taking into account the connectivity between the meshes. The main contribution of this metric lies in the integration of such visual properties as the sensitivity to detail and the scale change. In this paper, we have assessed the correlation between the subjective measures provided by LIRIS/EPFL and the quality values obtained through the presented metrics, employing both linear correlation (Pearson correlation) and non-linear correlation (Spearman correlation) techniques (see Table 17).

- Pearson Correlation also known as the linear correlation coefficient, measures the linear relationship between two continuous variables. It provides a value between -1 and 1, where -1 indicates a perfect negative linear relationship, 1 indicates a perfect positive linear relationship, and 0 suggests no linear relationship. Pearson correlation assumes that the data is normally distributed and that there is a linear association between the variables.
- Spearman Correlation (Non-linear Correlation), assesses the strength and direction of a monotonic relationship between two variables. Monotonic relationships need not be linear; they can be any consistently increasing or decreasing pattern. Unlike Pearson correlation, Spearman correlation does not assume that the data is normally distributed or linearly related. It assigns ranks to the data points and calculates the correlation based on these ranks. It provides a value between -1 and 1, where -1 indicates a perfect negative monotonic relationship, 1 indicates a perfect positive

monotonic relationship, and 0 suggests no monotonic relationship.

Table 17 summarizes the linear and non-linear correlation results of the existing metrics compared to the presented ones (3DrwPSNR and LWRMS). In this table, we shall compare the presented metrics with the existing ones, relying on the results found in the literature.

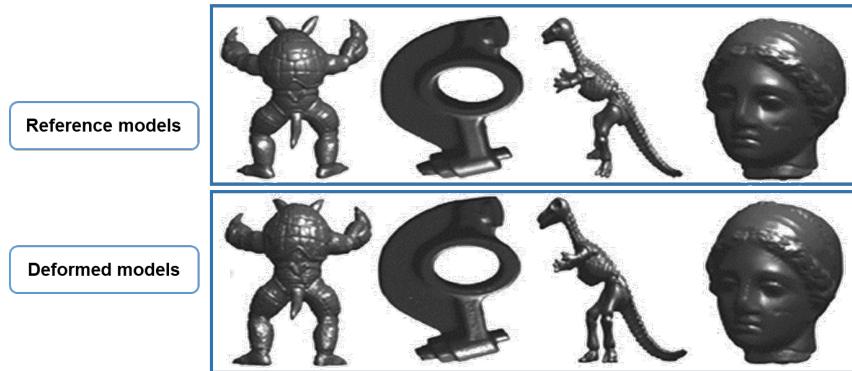


Fig. 22. LIRIS / EPFL general purpose database models; Reference and Deformed models

The LIRIS/EPFL general purpose database use created in 2006 contains the values of subjective values created by 12 observers at EPFL and LIRIS, University of Lyon (32). This database contains 88 models, including 4 reference models and 84 deformed models. On each reference model, two types of treatments (uniform noise and smoothing) were applied with different strengths and at three locations (on all models, on the smooth areas, on the rough areas) Figure 22.

Table 17. Spearman (Rs) and Pearson (Rp) correlation values (%) between subjective and objective scores from the LIRIS/EPFL general purpose database.

	Armadillo		Dinosaur		Venus		RockerArm		all Corpus	
	Rs	Rp								
Hausdorff	69,5	30,2	30,9	22,6	1,6	0,8	18,1	5,5	13,8	1,3
RMS	62,7	32,2	0,3	0,0	90,1	77,3	7,3	3,0	26,8	7,9
MSDM	84,8	70,0	73,0	56,8	87,6	72,3	89,8	75,0	73,9	56,4
MSDM 2	81,6	72,8	85,9	73,5	89,3	76,5	89,6	76,1	80,4	66,2
SMQI	77,5	74,3	84,8	80,3	91,6	90,0	91,8	92,6	84,6	84,3
3DrwPSNR	80,2	79,9	87,5	89,0	85,0	77,4	89,8	86,8	85,6	83,3
LWRMS	80,8	87,4	88,8	87,8	85,1	78,8	90,9	86,1	86,4	85,1

Examining the obtained results, several observations can be drawn: First, the

correlation scores of the geometric measures (Hausdorff, RMS) provide low correlations. This is a reference to the fact that because these measures do not account for the visual qualities, it is exceedingly challenging for them to accurately gather the visual effects resulting from various types of distortion (noise and smoothing) given to the 3D reference models. Particularly in terms of linear correlation, which indicates the strength of the association, the developed approaches show a significant improvement. On all models of the general purpose database created by LIRIS/EPFL, 3DrwPSNR and LWRMS also showed competitive results, with non-linear correlation (R_s) values of 85.6 percent and 86.4 percent, respectively, and linear correlation (R_p) values of 83.3 percent and 85.1 percent, respectively. This confirms the strong correlation between the subjective measurements in this database and the objective measurements produced by the presented metrics. The acquired results support the claim that, as compared to 3DrwPSNR, the LWRMS metric has a greater correlation with subjective outcomes. This could be attributed to the local level of detail calculation taking into account the connectivity between the vertices.

8. Conclusion

The distortion of a 3D deformed model is measured using a variety of techniques for measuring the visual quality of static 3D models. There are two distinct groups of 3D measurements: geometric approaches and HVS metrics, which take into account the characteristics of the human visual system. Full-reference measurements, No-reference metrics, and reduced-reference metrics are the three categories into which these techniques are separated. This study examines various techniques for improving the perceptual quality of static 3D models and 2D photographs. Notably, it creates two perceptual metrics that account for the limits of the human visual system in order to forecast the quality of a deformed static 3D model.

This work's main goal is to present a method for assessing the perceptual quality of static 3D models that correlates with human vision to assess the deterioration of a challenging distorted 3D model. Two measures that integrate HVS characteristics to gauge the degree of distortion in a distorted model were highlighted.

The first metric, 3DrwPSNR, is an image-based metric that is inspired by the 2DrwPSNR metric. We invested the properties used in the 2D rwPSNR metric, such as visual relativity and selectivity, by adapting them with the notions of 3D geometry. To achieve this migration, we combined the relativity principle of the Weber-Fechner law stating that the sensation of the eye is logarithmic as well as the variance function to integrate the visual selectivity.

The second metric is a model-based metric which is based on the RMS measure. RMS computes the deformation of a 3D model using purely mathematical equations. To integrate the HVS characteristic, the distance between a vertex and its neighbors was computed in order to detect the level of detail of the distortion area and determine the invariance to scale changes by normalizing the error with

respect to the size of the model.

The obtained results were compared with those of state-of-the-art methodologies in order to evaluate the performance of the offered measures. The general-purpose database from LIRIS/EPFL was used for the comparison. The effectiveness of the results was assessed using two correlation coefficients, namely Pearson's correlation coefficients for linear correlation and Spearman's correlation coefficients for non-linear correlation, between the values of objective measures and subjective measures. We demonstrated that the correlation values obtained by the developed metrics have a better correlation compared to the existing metrics using the same comparison parameters.

In the next research, we shall improve the highlighted algorithms by taking into account topological distortions. Our research can be extended further by integrating the improved metrics into modeling/visualization and 3D printing platforms. Further fruitful lines of investigation can be provided by elaborating objective perceptual measures to assess the quality of animated 3D models. Another promising future research direction involves equally applying the developed metrics in the field of Virtual/Augmented reality.

References

1. W. Barbier and J. Dupuy, "Htex: Per-halfedge texturing for arbitrary mesh topologies," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 5, no. 3, pp. 1–14, 2022.
2. Y. Audoux, M. Montemurro, and J. Pailhes, "A surrogate model based on non-uniform rational b-splines hypersurfaces," *Procedia CIRP*, vol. 70, pp. 463–468, 2018.
3. E. Artioli, L. Beirão da Veiga, C. Lovadina, and E. Sacco, "Arbitrary order 2d virtual elements for polygonal meshes: part i, elastic problem," *Computational Mechanics*, vol. 60, no. 3, pp. 355–377, 2017.
4. A. T. Chan, A. Gamino, F. C. Harris, and S. Dascalu, "Integration of assistive technologies into 3d simulations: an exploratory study," in *Information Technology: New Generations*, pp. 425–437, Springer, 2016.
5. D. Qin, L. Sang, Z. Zhang, S. Lai, and Y. Zhao, "Compression performance and deformation behavior of 3d-printed pla-based lattice structures," *Polymers*, vol. 14, no. 5, p. 1062, 2022.
6. E. Wood, T. Baltrušaitis, C. Hewitt, M. Johnson, J. Shen, N. Milosavljević, D. Wilde, S. Garbin, T. Sharp, I. Stojiljković, *et al.*, "3d face reconstruction with dense landmarks," in *European Conference on Computer Vision*, pp. 160–177, Springer, 2022.
7. S. F. McLean, "Case-based learning and its application in medical and health-care fields: a review of worldwide literature," *Journal of medical education and curricular development*, vol. 3, pp. JMECD–S20377, 2016.
8. E. Padovan, G. Marullo, L. Tanzi, P. Piazzolla, S. Moos, F. Porpiglia, and

- E. Vezzetti, “A deep learning framework for real-time 3d model registration in robot-assisted laparoscopic surgery,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 18, no. 3, p. e2387, 2022.
9. A. Agathos, P. Azariadis, and S. Kyratzi, “Elliptic gabriel taubin smoothing of point clouds,” *Computers & Graphics*, vol. 106, pp. 20–32, 2022.
 10. D. Thanou, P. A. Chou, and P. Frossard, “Graph-based compression of dynamic 3d point cloud sequences,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016.
 11. J. She, X. Gu, J. Tan, M. Tong, and C. Wang, “An appearance-preserving simplification method for complex 3d building models,” *Transactions in GIS*, vol. 23, no. 2, pp. 275–293, 2019.
 12. S. Mansouri and H. Ebrahimnezhad, “Segmentation-based semi-regular remeshing of 3d models using curvature-adapted subdivision surface fitting,” *Journal of Visualization*, vol. 19, no. 1, pp. 141–155, 2016.
 13. W. L. Theo, J. S. Lim, W. S. Ho, H. Hashim, and C. T. Lee, “Review of distributed generation (dg) system planning and optimisation techniques: Comparison of numerical and mathematical modelling methods,” *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 531–573, 2017.
 14. S. Lee, J.-B. Jeong, and E.-S. Ryu, “Group-based adaptive rendering system for 6dof immersive video streaming,” *IEEE Access*, vol. 10, pp. 102691–102700, 2022.
 15. K. Gu, V. Jakhetiya, J.-F. Qiao, X. Li, W. Lin, and D. Thalmann, “Model-based referenceless quality metric of 3d synthesized images using local image description,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 394–405, 2017.
 16. C. Liu, Z. Wang, E. J. C. Naçpil, W. Hou, and R. Zheng, “Analysis of visual risk perception model for braking control behaviour of human drivers: A literature review,” *IET Intelligent Transport Systems*, vol. 16, no. 6, pp. 711–724, 2022.
 17. H. Garg, “A comprehensive study of watermarking schemes for 3d polygon mesh objects,” *International Journal of Information and Computer Security*, vol. 19, no. 1-2, pp. 48–72, 2022.
 18. S. Haddad and A. Halder, “Hausdorff distance between norm balls and their linear maps,” *arXiv preprint arXiv:2206.12012*, 2022.
 19. I. Abouelaziz, A. Chetouani, M. El Hassouni, L. J. Latecki, and H. Cherifi, “3d visual saliency and convolutional neural network for blind mesh quality assessment,” *Neural Computing and Applications*, vol. 32, no. 21, pp. 16589–16603, 2020.
 20. G. Lavoué and M. Corsini, “A comparison of perceptually-based metrics for objective evaluation of geometry processing,” *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 636–649, 2010.
 21. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

22. M. Botsch and S. Schaefer, “A multiscale metric for 3d mesh visual quality assessment,” 2011.
23. L. Váša and J. Rus, “Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes,” in *Computer graphics forum*, vol. 31, pp. 1715–1724, Wiley Online Library, 2012.
24. F. Torkhani, K. Wang, and J.-M. Chassery, “Perceptual quality assessment of 3d dynamic meshes: Subjective and objective studies,” *Signal Processing: Image Communication*, vol. 31, pp. 185–204, 2015.
25. L. Dong, Y. Fang, W. Lin, and H. S. Seah, “Perceptual quality assessment for 3d triangle mesh based on curvature,” *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2174–2184, 2015.
26. A. Nouri, C. Charrier, and O. Lézoray, “A genetically based combination of visual saliency and roughness for fr 3d mesh quality assessment: A statistical study,” *The Computer Journal*, vol. 65, no. 3, pp. 606–620, 2022.
27. 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.
28. N. Elloumi, H. Loukil, and B. Med Salim, “Full-reference objective quality metric for three-dimensional deformed models,” *International Journal of Image and Graphics*, 2022.
29. G. Wendt and F. Faul, “Binocular luster—a review,” *Vision Research*, vol. 194, p. 108008, 2022.
30. N. Elloumi, J. Debayle, H. Loukil, and M. S. Bouhlel, “A locally weighted metric for measuring the perceptual quality of 3d objects,” in *International Conference on Intelligent Systems Design and Applications*, pp. 128–139, Springer, 2020.
31. V. Vidal, E. Lombardi, M. Tola, F. Dupont, and G. Lavoué, “Mepp2: a generic platform for processing 3d meshes and point clouds,” in *EUROGRAPHICS 2020 (Short Paper)*, 2020.
32. 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, pp. 150–161, Spie, 2006.