

# Perceptually Improved 3D Object Representation Based on Guided Adaptive Weighting of Feature Channels of a Visual-Attention Model

Ghazal Rouhafzay · Ana-Maria Cretu

Received: 6 February 2018 / Revised: 15 June 2018 / Accepted: 29 June 2018 / Published online: 10 July 2018  
© 3D Research Center, Kwangwoon University and Springer-Verlag GmbH Germany, part of Springer Nature 2018

**Abstract** Real-time interaction in virtual environments composed of numerous objects modeled with a high number of faces remains an important issue in interactive virtual environment applications. A well-established approach to deal with this problem is to simplify small or distant objects where minor details are not informative for users. Several approaches exist in literature to simplify a 3D mesh uniformly. A possible improvement to this approach is to take advantage of a visual attention model to distinguish regions of a model which are considered important from the point of view of the human visual system. These regions can then be preserved during simplification to improve the perceived quality of the model. In the present article, we present an original application of biologically-inspired visual attention for improved perception-based representation of 3D objects. An enhanced visual attention model extracting information about color, intensity, orientation, as in the classical bottom-up visual attention model, but that also considers supplementary features believed to guide the deployment of human visual attention (such as symmetry, curvature, contrast, entropy and edge information), is introduced to identify such salient

regions. Unlike the classical model where these features contribute equally to the identification of salient regions, a novel solution is proposed to adjust their contribution to the visual-attention model based on their compliance with points identified as salient by human subjects. An iterative approach is then proposed to extract salient points from salient regions. Salient points derived from images taken from best viewpoints of a 3D object are then projected to the surface of the object to identify salient vertices which will be preserved in the mesh simplification. The obtained results are compared with existing solutions from the literature to demonstrate the superiority of the proposed approach.

**Keywords** Interest point and salient region detections · Visual attention · Visual perception · 3D mesh · Simplification · Level-of-detail

## 1 Introduction

Recent research on the topic of 3D modeling shows a clear tendency of novel techniques towards user-centeredness. The fact that the final purpose of the model is to be examined (and interacted with) by the users elicited the interest into exploiting and mimicking the human visual capabilities in order to improve the perceived quality of such models in interactive and immersive environments.

---

G. Rouhafzay (✉) · A.-M. Cretu  
Carleton University, 1125 Colonel By Drive, Ottawa,  
ON K1S 5B6, Canada  
e-mail: ghazal.rouhafzay@carleton.ca

A.-M. Cretu  
e-mail: acretu@sce.carleton.ca

Biological and psychological knowledge derived from human visual mechanisms has already been successfully employed in various computational vision systems [1] and in the context of modeling [2]. Due to its capacity to improve the time and complexity of a visual scene understanding by identifying only certain regions of interest for further analysis, visual attention [3] was a technique of choice for many researchers on one side to push the envelope of the current computer vision technology and on the other side to ameliorate the perceived quality of models by concentrating the details in the regions that are subject to observation by a human user. The work in [4] is an example where the salient regions of video frames keep higher quality when encoding. The potential for such computational models of visual attention stems from their power to extract from a complex shape a series of discriminative features. The latter can be successfully used as a basis for classification and recognition, which are two significant problems in the understanding of complex scenes or for creating selectively-densified object models, that are denser in the regions subject to observation, as it is the case of the work proposed in this paper.

There are two types of visual attention models defined in the literature, namely bottom-up and top-down attention models. In the case of bottom-up attention, research has demonstrated the existence of a series of characteristic features (i.e. color, orientation, intensity, edges, etc.) in an image that are believed to capture attention during free viewing conditions, while the user visualizes a scene without looking for a specific object or having a specific interest. On the other hand, top-down attention is engaged once cognitive factors such as knowledge, expectation, current goals come into play. Such factors have an influence on the bottom-up feature and perform a selection of features that better correspond to the visual task.

In bottom-up computational models of visual attention, different modalities (features) which are believed to be effective in guiding visual attention system are each encoded into a distinct feature map referred to as conspicuity map [3, 5–7]. The combination of conspicuity maps of all modalities then yields a final saliency map that encodes salient regions as bright areas against a dark background. It is believed that the top-down influences are used as biasing weights when combining conspicuity maps

such that particular properties of an object become more important and attractive, thus guiding their observation. In the context of this work, we propose to make use of these properties in order to construct perceptually improved 3D object models.

We propose to explore the use of additional features contributing to human visual attention guidance that have not yet been confirmed as useful in the context of 3D object modeling. In particular, to go beyond the classical model of visual attention that is based on color, orientation and intensity [3, 5], to our previous work that included information on the contrast, curvature, symmetry, entropy [6], in this work we aim to include additional edge information to create an enhanced visual attention model. Furthermore, in order to tune our model to the real salient regions identified by human subjects, and thus simulate the role of the top-down influences of visual attention, we propose a solution that makes use of the ground truth points identified by real users in order to compute the contribution weight of each feature. Two different approaches using the structural similarity index and the Euclidean distance respectively are then proposed to determine the weight of each feature. In an attempt to enable the detection of salient regions for new objects, for which the ground truth points provided by users are not available, a machine learning algorithm based on support vector machines (SVM) is trained to learn the position of salient regions on images (2D) according to the extracted features. As saliency computations are applied to images taken from 3D objects, a procedure is proposed to select the best viewpoints of objects from which to capture images. In this procedure, the number of selected view-points is adaptively selected according to the object complexity. Salient points in 2D are identified from salient regions of images using an iterative algorithm. Finally, the 2D salient points extracted from images are projected back on the surface of 3D objects. These points and their immediate neighbors will be preserved while simplifying the object mesh.

The main contributions of the paper are as follows: (1) the construction of a visual attention saliency map from nine different features where the contribution (i.e. weight) of each feature is determined based on feedback from human subjects; (2) a learning algorithm to predict the position of salient points from the extracted features; (3) the determination of best viewpoints of object according to the level of visual

saliency to be employed for salient point detection; (4) determination of number of required view points to capture the entire surface of objects; (5) an iterative technique to identify salient points in order of their saliency from the saliency map; (6) a geometry-based algorithm to determine the coordinates of salient points detected in images on the surface of the 3D object and (7) the automatic determination of the number of neighbors around each salient point to be preserved from the simplification.

## 2 Literature Review

Triangular meshes are widely accepted as one of the best solutions to represent 3D objects in computer graphics and virtual reality applications, due to their fast rendering capability. However, in the case of complex objects, a very large number of triangles is required to achieve an accurate representation. The large number of triangles can inhibit the expected real-time interaction in virtual environments containing several objects. Despite ongoing advances in graphic card performance, this technology still fails in most cases to provide the desired high interaction speed.

Removing unnecessary details from the distant or small objects whose minor details are not remarkably informative is an intelligent solution that can reduce the complexity of the virtual environment and consequently contribute to achieve and guarantee the desired real-time interaction. The idea of decreasing the complexity of 3D objects in computer graphics is referred to as level of detail (LOD) management [8]. In this approach, 3D meshes of the objects are simplified gradually with respect to their distance to the viewer. In discrete LOD methods [8], multiple copies of the same object with different resolutions are created offline, with details uniformly and gradually reduced based on the distance to the viewer. Subsequently, one of the copies is selected for presentation according to the viewer position. A specific data structure consisting of a continuous spectrum of details is used in continuous LOD methods, through which the desired level of detail is extracted at run-time. In continuous level of detail methods, if the appropriate level of detail for the object is selected dynamically, the solution is referred to as view-dependent method. All these methods simplify triangular meshes uniformly without considering the mesh structure which can

degrade the quality of objects especially for low level of details. In 3D graphical models, features characterizing the object can sometimes be very small with respect to the object size (e.g. the tail and the ears of a dog) and uniform simplification of the models can completely remove them.

A possible solution to improve the current algorithms is to modify them such that more details are preserved in regions that are perceptually more important than others. In this approach, an explorative algorithm is first applied to the triangular mesh structure to determine the vertices which are more important in characterizing the object. The neighbourhood of such vertices are then preserved in more detail during mesh simplification.

The existing saliency detection algorithms basically explore the geometrical features of 3D models. A heuristic possible approach to retrieve the salient vertices of the object is to take benefit from human visual attention system such that the entire surface of the object is scanned with a visual attention model to determine perceptually salient regions. This solution can yield interesting results as viewers are human subjects. Authors in [8–11] adopt user input to improve the quality of local details of resulting model, for different resolutions. Alternatively, quality adjustments can be made automatically by exploiting computational models inspired from human visual perception, and in particular human visual attention mechanisms [6, 7, 12]. Taking into consideration the best viewpoints of objects, Rouhafzay and Cretu [7] uses the classical visual attention model in [3] to detect salient regions.

In recent years, several researchers are investigating the computational models of visual attention and their applications in various fields. As briefly mentioned in the introduction, most computational implementations of visual attention are based on bottom-up, scene-driven features [13] which are more distinctive compared to other neighboring features. In the cases where a voluntary choice of the viewer is involved to allocate resources to a subset of the perceptual inputs [14], the solution is referred to as top-down attention models. In this work, we simulate the effect of top-down attention using user inputs on saliency points.

Itti et al. [3] takes advantage of three features namely; orientation, color and intensity to extract salient features. For this purpose, a center-surround antagonism is simulated over multi-level

decomposition of each feature to yield a saliency map which encodes saliency as brighter regions on a black background. Lee et al. [12] applies the same center-surround paradigm to a curvature metric of vertices of a 3D model to compute the saliency (the solution is referred to as *Mesh saliency* in Sect. 4). Some other features which are proved to be effective in visual guidance were tested and validated in a visual attention model for interest (salient) point detection in the context of 3D level-of-detail modeling in [6]. Castellani et al. [15] adopt a perceptually-inspired saliency detector based on Difference-of-Gaussians to find some sparse salient points; subsequently a Hidden Markov Model is employed to describe salient points across different views (this solution is referred to as *Salient points* in Sect. 4). Zhao et al. [16] takes advantage of two perceptual features namely Retinex-based importance Feature and Relative Normal Distance to assign a saliency rank to vertices of 3D objects. In a recent research, Lavoué et al. [17] generate an eye fixation density map for 3D objects by tracking human eye fixation on 3D objects and mapping them onto the surface of 3D shapes.

Other researchers consider the geometric structure of objects to detect salient regions [18–22]. Godil and Wagan [18] apply the Scale Invariant Feature Transform (SIFT) to a 3D voxelized model to detect local saliency (*3D-SIFT* in Sect. 4). A 3D version of Harris corners detector is proposed in [19] (*3D-Harris* in Sect. 4). The authors of [20] consider scale-dependent corners as salient points (this solution is referred to *SD-Corners* in Sect. 4). Local maxima of the Heat Kernel Signature are computed over triangular meshes in [21] to identify salient points (this solution is referred to *HKS* in Sect. 4). Mirloo and Ebrahimezhad [22] proposed a hierarchical solution to detect salient points as vertices with larger average geodesic distances compared to other vertices of the object and equally spread on the object surface.

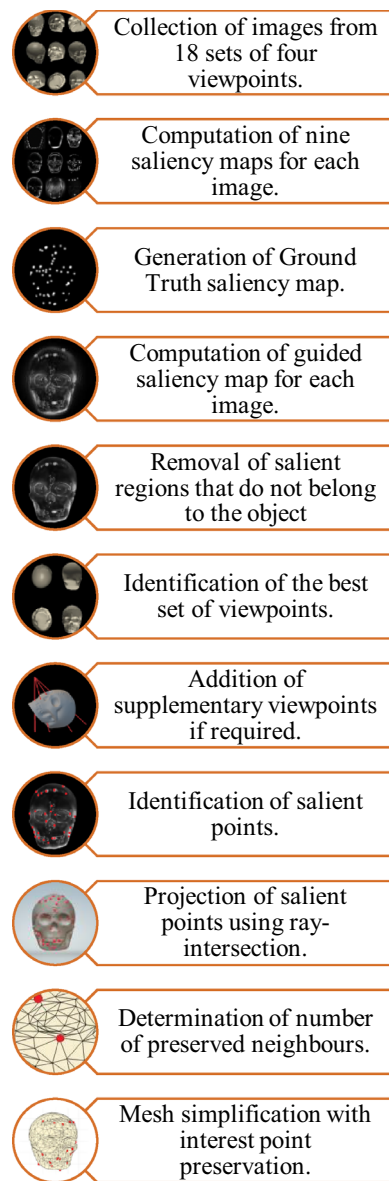
Salient points detected over 3D objects can be further used in a variety of applications. The guidance of mesh simplification process, mesh and shape retrieval [18, 23], matching of objects [15], are some examples. Luebke et al. provide a survey of polygonal mesh simplification methods in [8]. Substandard regions of simplified meshes are retrieved and improved in [11] through weighting and then by applying local refinements to the desired region by users. Kho and Garland [10] apply the quadric-based

simplification algorithm to 3D meshes where the vertices labeled by users as salient are preserved. The same simplification algorithm is applied to 3D models while preserving salient points detected by an enhanced visual attention model in [6]. Song et al. [24] bias the simplification process by amplifying the saliency values in regions of interest. Lee et al. [12] propose a procedure where the QSLim simplification algorithm is modified such that important regions are only removed later during the simplification process. Eye-fixation is another form of saliency according to the human-visual attention system which is used for implementation of a saliency-guided simplification by Howlett et al. [25]. In this article, the simplification process uses salient points extracted iteratively from the saliency map returned by an enhanced-guided computational bottom-up visual attention and is biased by constraining a maximum resolution of the mesh in those regions known to be perceptually salient.

In the present work, the most salient set of viewpoints are selected to determine the visually salient points of 3D models. Another novelty of the proposed approach is the weighted combination of nine modalities of visual attention, where the contribution of each modality is chosen according to the compatibility of each conspicuity map with feedback from users. We propose an innovative solution to build a model which combines the nine modalities based on user feedback. Salient points identified on images are then projected back to the surface of 3D object using an original geometrical approach. The local geometric structure of triangular meshes is also considered to guide the expansion of salient regions.

### 3 Framework for Perceptually Improved 3D Object Representation Based on Enhanced Guided Visual Attention

Figure 1 summarizes the overall framework proposed for the construction of selectively-densified 3D objects characterized by a higher resolution at salient regions (as identified by the visual attention system). In this approach, the triangular mesh of a 3D object with a high number of faces is simplified to a lower number of faces according to a desired Level of Details (LOD). In particular, an enhanced visual attention model in which the contribution weight of each saliency feature



**Fig. 1** Mesh simplification framework

is found by means of salient points identified by human subjects is applied to images captured from 18 sets of viewpoints, each containing 4 perpendicular viewpoints of the object. A number of four viewpoints is chosen because it allows to cover the entire surface of the object in each set. Nine conspicuity maps extracting features that are believed to be relevant for guiding the human visual attention system are computed over the images. In parallel, a saliency map based on feedback provided by human subjects is

generated to guide the contribution weighting of conspicuity maps. The visual saliency level is obtained for each set, and the set with the highest level of saliency is selected for salient point detection. The computation of saliency maps over images taken from virtual objects will detect a certain level of saliency on the background around the object due to the local color and contrast change (i.e. between background and the object). Consequently, we remove all the salient information identified over the background to prevent the selection of salient points which do not belong to the object. An iterative procedure then determines salient points on the corresponding saliency maps from the multiple viewpoints, and the resulting salient points (2D) are projected back on the surface of object using a heuristic approach. Investigating the local geometry of 3D mesh around the salient points, we adaptively determine the number of preserved neighbors for each salient point such that when the neighborhood region around the salient point is less densified, a higher number of neighbor vertices are preserved. Given the appropriate number of faces for each copy of an object within a LOD hierarchy, for which the neural network solution proposed in [6] is used, the QSlim simplification algorithm is adapted to simplify only those faces of the objects that do not contain as vertices the identified interest points and their immediate neighbors.

### 3.1 Computational Visual Attention Model

The classical computational model of visual attention introduced by Itti [3] uses three bottom-up features, namely intensity, color opponency and orientation as the main features guiding human visual attention. The algorithm applies a dyadic Gaussian pyramid to the intensity channel and to four broadly-tuned color channels (red, green, blue and yellow) to produce nine spatial scales for each channel. Then a series of center-surround differences is calculated between the center  $c \in \{2, 3, 4\}$  and surround  $s = c + \delta$  scales, where  $\delta \in \{3, 4\}$ . These center-surround operations are inspired from the fact that human visual system is more sensitive to the center of an image and less to the extremities of the visual field. Six feature maps are constructed. Subsequently, the conspicuity map for each channel is computed as the across-scale addition of normalized feature maps [4]. Similarly, the center-surround operation is applied to four Gabor pyramids

with different angles to extract the orientation feature. The final saliency map (*Smap*) is obtained as:

$$Smap = \frac{1}{3}(C_{color} + C_{orientation} + C_{intensity}) \quad (1)$$

Five other features including contrast, entropy, curvature, symmetry and DKL color channel were added into the classical visual attention algorithm in [6] to enhance the model. The contrast map is computed as the luminance variance over a  $80 \times 80$  pixel neighborhood [26]. To obtain the entropy map, we have first applied a median filter to an image and then computed the local entropy over neighborhoods of  $9 \times 9$  pixels. Symmetry information is extracted by detecting radial and bilateral symmetric points on images as in [27]. The neighborhoods of symmetric points are transposed into bright regions on a dark background and a center-surround operation is applied to create the final symmetry map. The curvature conspicuity map is obtained directly in 3D using the Gaussian-weighted center-surround assessment of surface curvatures suggested by Lee et al. [12]. The Matlab virtual camera is turned around each resulting 3D model (with the curvature information encoded in gray levels, the higher the curvatures, the whiter the corresponding area) to obtain the curvature conspicuity maps. The color opposition model based on Derrington–Krauskopf–Lennie (DKL) color space in [28] is added to provide another color feature that is more attuned to the capabilities of the human visual system.

Moreover, according to the literature, edge extraction is the earliest process in visual object recognition [29], thus in order to further improve the visual attention model presented in [6], we also make use of edge information, as a distinct conspicuity map. For this purpose, we detect object edges using double derivation over the image smoothed by a Gaussian filter (i.e. Laplacian of Gaussian).

Models in the benchmark data set for 3D object interest points [30] are used for testing. Color information is added to the objects as further explained in Sect. 4. Figure 2 illustrates the nine conspicuity maps for the model of skull extracted from the dataset.

### 3.2 Guided Saliency Map Construction

In our previous work and in the vast majority of publications that work with visual attention models, all

conspicuity maps contribute with the same weight to construct the saliency map. Exploring top-down visual attention mechanisms, some authors have proposed other approaches to combine conspicuity maps. Frinotrop [14] suggests determining weights as the ratio of the mean target saliency and the mean background saliency.

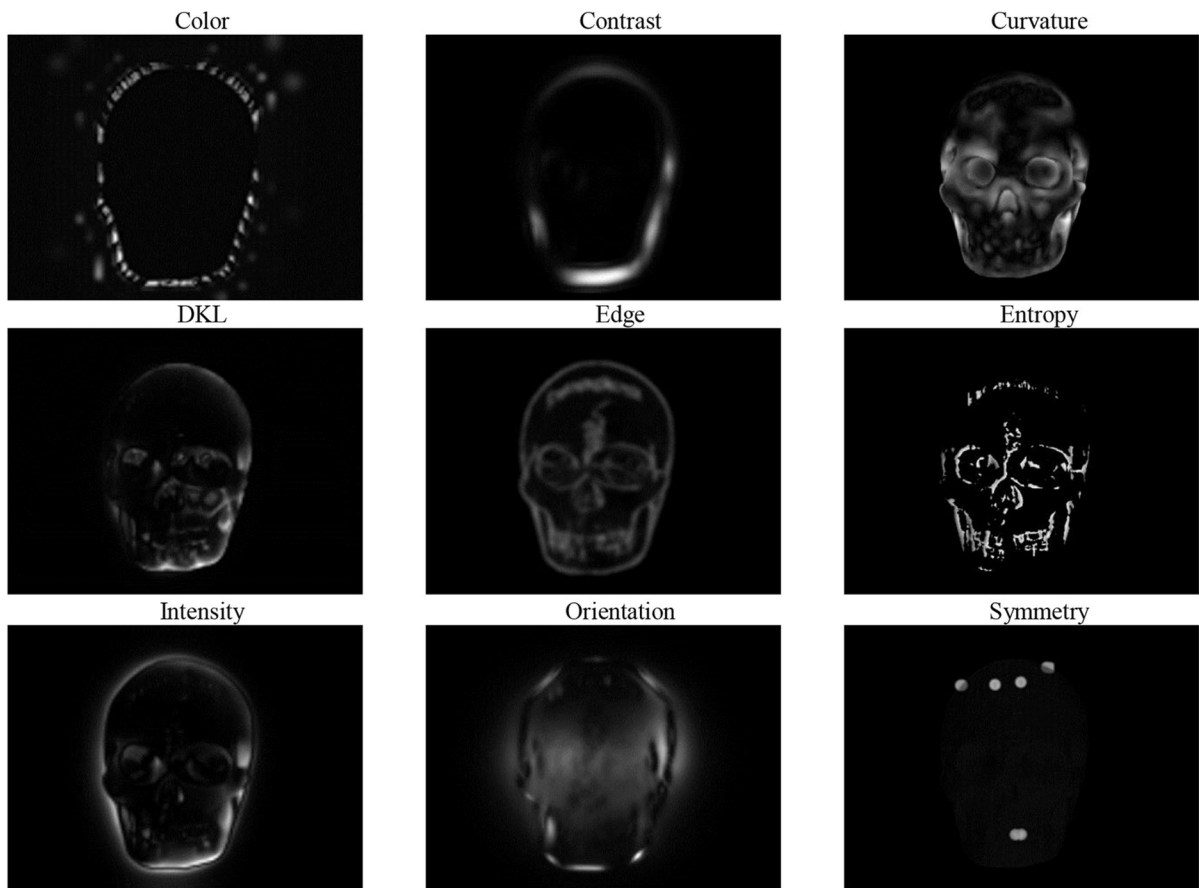
In this work, we aim to determine which characteristics are more effective in guiding human visual system by using two different approaches that capitalize on the points identified as salient by human subjects in [30] (and that we call ground truth points). These methods are explained, and the results are compared in the following sections. In this way, we simulate the influence of top-down visual attention that biases the information derived from the bottom-up features.

#### 3.2.1 Guided Saliency Map Based on Euclidean Distance

The first approach we are using to determine the contribution weight of each conspicuity map, *Cmap*, to the final saliency map, *Smap*, is based on the Euclidean distance between the brightest points of each *Cmap* and the ground truth points. To do this, the  $n$  brightest pixels are found for all *Cmaps*, where  $n$  is the number of visible ground truth points from a given view point. Then, the average pairwise Euclidean distance between the brightest points and the ground truth points is calculated, and the values are normalized between 0 and 1. The highest weight is assigned to the *Cmap* with lowest average Euclidean distance. In this way, we penalize those points that are situated further from the ground truth salient points. The assigned weights according to the average of normalized Euclidean distance (ED) value (1-average Euclidean distance) for a selected viewpoint from model of skull is reported in third column of Table 1.

#### 3.2.2 Guided Saliency Map Based on Similarity Index

A second approach that we have proposed to compute the contribution of each conspicuity map to the final saliency map is based on similarity. In particular, we have generated a saliency map using the ground truth points (that we called the Ground Truth Saliency map, *GTSmap*) and then the similarity between each conspicuity map, *Cmap* and this saliency map is



**Fig. 2** The nine conspicuity maps for the model of skull

**Table 1** SSIM values for each conspicuity map

Conspicuity map	SSIM values	Rank by average ED values
Color	0.1096	0.6417
Contrast	0.7692	0.3941
Curvature	0.7536	0.7929
DKL	0.6389	0.7098
Edge	0.7264	0.5651
Entropy	0.8914	0.7879
Intensity	0.6274	0.4487
Orientation	0.4596	0.6773
Symmetry	0.7235	0.4970

measured to determine which  $Cmap$  will have higher contribution to final saliency map,  $Smap$ .

**3.2.2.1 Ground Truth Saliency Map (GT $Smap$ )** A series of researches from neuropsychology have been

conducted to explore the spatial spread of directed visual attention. Hughes and Zimba [31] revealed a gradient pattern for allocation of attentional resources, where each zone is concentrated in the center of focus and the cortical resource allocation decreases while increasing the eccentricity. They also mention that the size of attention zones can also be adjusted according to circumstances. Hence, taking advantage from the marked locations on the objects as salient by human subjects we attempt to reproduce this phenomenon using an artificial saliency map where the attention zones are determined by a spatial Gaussian kernel of size 50 by 50 pixels with a  $\sigma = 0.15$  centered at interest points. These values are adjusted empirically such that a round region of saliency is shaded around each salient point. We first project the visible ground truth points from each viewpoint to two-dimensional pixel coordinates. Knowing the location of ground truth points on the image, we apply Gaussian kernels

centered at each ground truth point to assign highest intensities to the pixels where the ground truth is projected and lower level to neighborhood pixels, according to the knowledge that the saliency map encodes saliency as bright regions. Finally, any pixel which does not belong to the object surface is set to zero. Figure 3b depicts the obtained ground truth saliency map for the model of skull.

Table 1 compares the weight values obtained by SSIM and average Euclidean distance.

### 3.3 Adaptive Weighting Scheme

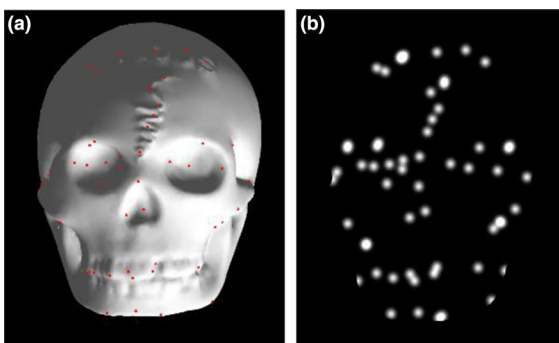
Once the corresponding weight for each conspicuity map is computed using ED or SSIM, it is used as the assigned weight to construct the final (ground truth guided) saliency map as follows:

$$Smap = \frac{w_{col} \cdot C_{col} + w_{con} \cdot C_{con} + w_{curv} \cdot C_{curv} + w_{DKL} \cdot C_{DKL} + w_{edg} \cdot C_{edg} + w_{ent} \cdot C_{ent} + w_{int} \cdot C_{int} + w_{ori} \cdot C_{ori} + w_{sym} \cdot C_{sym}}{\sum W_{Conspicuity\ Maps}} \quad (2)$$

**3.2.2.2 Similarity Measurement** An effective *Cmap* should fulfill two requirements; first, it should have higher intensity in salient areas of the ground truth saliency map; and second, it should not distract the visual attention to unwanted regions. Accordingly, we use the Structural Similarity Index (SSIM) [32], which highlights similarities between the *Cmap* and the Ground Truth Saliency map, *GTSmap*. The SSIM is inspired by biology and measures the similarity between two images by computing contrast, luminance and structural terms. Table 1 provides the average SSIM values of the nine *Cmaps* for 43 models in the dataset [30]. The SSIM is equal to one for two identical images. Consequently, in this particular case, the entropy conspicuity map has the highest similarity value and should be the most prominent conspicuity map in the saliency map. This method is denoted *SSIM* in Sect. 4.

where  $\sum W_{Conspicuity\ Maps}$  is the sum of all weights;  $w_{col}$ ,  $w_{con}$ ,  $w_{curv}$ ,  $w_{DKL}$ ,  $w_{edg}$ ,  $w_{ent}$ ,  $w_{int}$ ,  $w_{ori}$  and  $w_{sym}$  represent the corresponding weight to color, contrast, curvature, DKL, edge, entropy, intensity, orientation and symmetry respectively. Similarly, the conspicuity maps are denoted in order as  $C_{col}$ ,  $C_{con}$ ,  $C_{curv}$  etc.

Figure 4 compares the saliency maps obtained by four different methods namely: Classical Itti [3], VisAttAll (color, contrast, curvature, DKL, entropy, intensity, orientation and symmetry with equal weights) [6], Euclidean Distance based guided VisAttAll (described in Sect. 3.2.1) and Similarity based guided VisAttAll (described in this Sect. 3.2.2). One can notice by comparing the saliency maps without user feedback (Fig. 5b, c) with those obtained when using the interest points selected by users (Fig. 3a) that salient points on forehead of the skull model are only detected by Euclidean Distance and Similarity based guided saliency maps that take advantage of the user feedback to determine contribution weights.

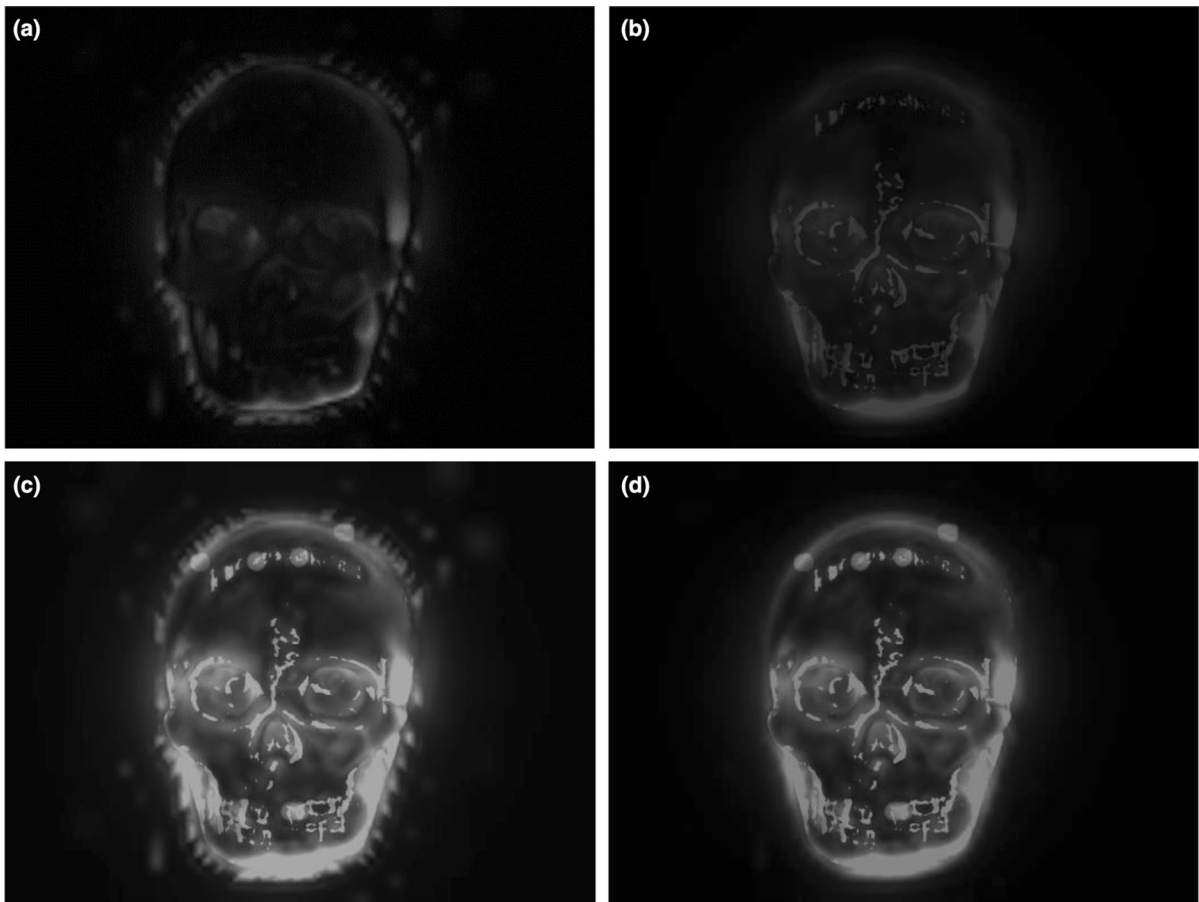


**Fig. 3** a Ground truth points, and b ground truth-based saliency map

### 3.4 Learning Algorithm to Predict Saliency

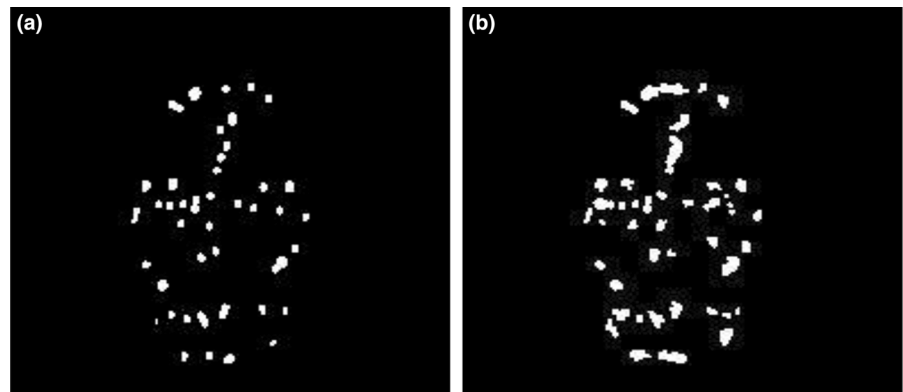
Up to now, two solutions were proposed to adaptively weight the conspicuity maps such that the resulting saliency map is more compatible with the human provided ground truth points. In this section, a support vector machine (SVM) is trained to classify image pixels as salient or non-salient regions using the nine conspicuity maps. The role of this machine learning solution is to build a model to integrate the nine visual attention modalities such that the output reflects





**Fig. 4** **a** Classical Itti, **b** VisAttAll Channels, **c** Guided VisAttAll based on Learning Feature weights and **d** Guided VisAttAll based on similarity

**Fig. 5** **a**  $200 \times 200$  pixels binarized ground truth saliency map ( $200 \times 200$  pixels target). **b** SVM output saliency map



saliencies detected by human subjects and to predict the salient points for objects whose ground truth points are not known a priori.

Each *Cmap* of size  $1200 \times 900$  pixels is resized to a  $200 \times 200$  array to reduce its complexity.

Subsequently, each array is converted to a column of size  $40,000 \times 1$  to form predictor columns. The *GTSmap* is also transformed to a  $40,000 \times 1$  logical array and is used as class labels (i.e. salient and non-salient).

RapidMiner studio is used for training and testing purposes. Considering the fact that data is unbalanced towards the non-salient class, a stratified sampling approach is used to select a balanced subset of data to train the SVM. To adjust feature weights before training the classifier, we exploit the *information gain* that computes how much information about the class membership is gained by knowing each feature.

We have performed several tests for training the SVM, namely: one SVM for all the objects, one SVM for each object as well as one SVM for each viewpoint. The subsampled vectorized feature maps of different viewpoints are concatenated to construct the training/testing dataset. To reduce the computational effort, only 30% of the resulting data is used for network training and validation. This reduced dataset is partitioned for tenfolds cross-validation.

In the case when one single SVM was trained to predict the saliency map for all 43 models, the performance achieved is of  $86.08 \pm 4.15\%$ . Tests were also performed when one SVM was used for each object. In this case, 43 SVMs are trained for the 43 models in the benchmark dataset for 3D object interest points that we are using. The trained support vector machine is then used to predict the saliency maps for each object knowing the conspicuity maps. An accuracy of  $93.18 \pm 5.11\%$  is obtained for all predicted saliency maps. During this series of tests, we have noticed that the determined weight for the symmetry feature by the information gain feature weighting changes a lot for different objects. In those cases where the object extremes have lower deviation from the central of mass, the symmetry feature has zero contribution in guiding the classifier. This is the reason why we are training 43 different SVMs for the 43 models in this work, instead of one single SVM for all objects. At the same time, as revealed in our testing, the performance is better when one SVM is used for each object.

The predicted saliency map is resized back to the initial size for the rest of process. Figure 5 compares the support vector machine output (Fig. 5b) with the binarized *GTSmap* (Fig. 5a) for the skull model. As this method results in larger number of salient points, we evaluated the case when 32 of the salient points with larger distance to each other, are used for mesh simplification process, to keep the number of salient points in the range of the other methods we compare with. It will be demonstrated in Sect. 4 that the

constructed meshes that preserve the salient points obtained by this approach have a superior quality compared to other methods. This method is referred to as *SVM* in Sect. 4.

### 3.5 Adaptive Selection of the Set of Best View Points

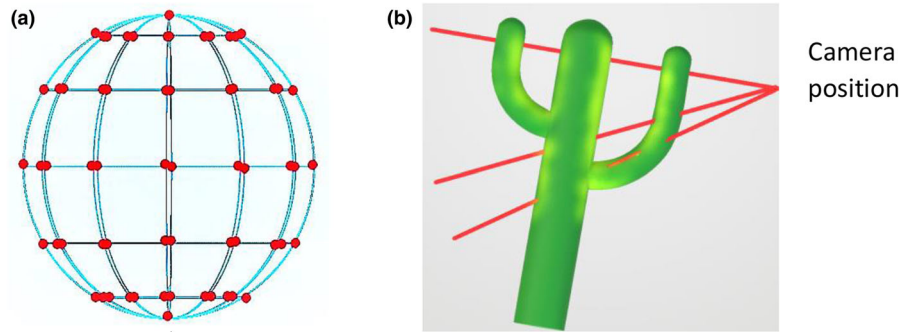
The viewpoints from which we observe an object play a decisive role in identifying its different features. For this purpose, we have proposed a best viewpoint selection algorithm based on the visual-attention model. The algorithm computes the level of saliency for 62 viewpoints (18 sets of viewpoints each containing 4 viewpoints and where 10 viewpoints are shared between multiple sets leading to 62 distinct viewpoints). These viewpoints are depicted as red points in Fig. 6a around the 3D model. The level of saliency for each view point is calculated as:

*Level of Saliency for viewpoints*

$$= \sum_{n=1}^N \sum_{m=1}^M Smap(n, m) \quad (3)$$

Once the level of saliency for all the viewpoints is calculated, to avoid the high computational cost required to scan each object from 62 viewpoints which will result in redundant information, we concentrate the further processing of each model to a set of best view-points, while also ensuring the complete coverage of the object surface. In particular, from the 62 viewpoints, the set of four perpendicular views with the highest level of saliency is selected for the rest of the work. Starting from the identified four perpendicular view-points with the highest level of saliency, we also verify if the whole surface of the object is captured. If the surface of the object is not captured in its entirety due to occlusions by other faces, another viewpoint covering the occluded region is automatically added. To detect occlusions, we use the ray-intersection algorithm presented in [33], which will be adopted as well as a part of our 2D–3D projection algorithm and that is discussed in detail in Sect. 3.9. If any ray starting from the camera position towards the object, intersects the model in more than two faces then there exists an occluded region from that view point. For this purpose, ten rays in random directions are beamed toward each object. If for any of these rays, the number of intersected faces is greater than two,

**Fig. 6** **a** Position of camera for different viewpoints. **b** Occlusion detected from a viewpoint



that indicates the presence of hidden faces which won't be visible to camera when the camera is turned 180° around the object. Thus, in this case, a new view point is added between the initial view point and the next view point (i.e. a deviation of 45° with respect to the occluded viewpoint) to cover the details in the occluded regions. This procedure is repeated for the four perpendicular viewpoints resulting in a maximum of four new viewpoints.

Figure 6b illustrates three rays intersecting the cactus model in 6, 4 and 4 points respectively from top to bottom where the occluded regions are shaded in yellow.

Each collection of four perpendicular viewpoints plus the complementary view-points is referred to as a set of viewpoints over which we estimate the average level of saliency. The set of viewpoints with the highest average level of saliency is considered as the best set of viewpoints and is used for the salient point identification procedure described in the next section.

### 3.6 Salient Point Selection

The brightest pixels on the saliency map are the most salient points. In these saliency maps, all pixels in a salient region have in general close intensity values.

As we don't want to identify all vertices in a region of the 3D mesh as salient, the neighborhood of radius  $r$  around the brightest selected pixels are set to zero [7]. The algorithm repeats iteratively until there is no pixel with an intensity greater than 90% of the maximum intensity. Figure 7 illustrates the salient point selection procedure.

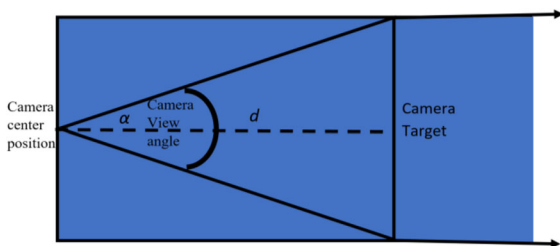
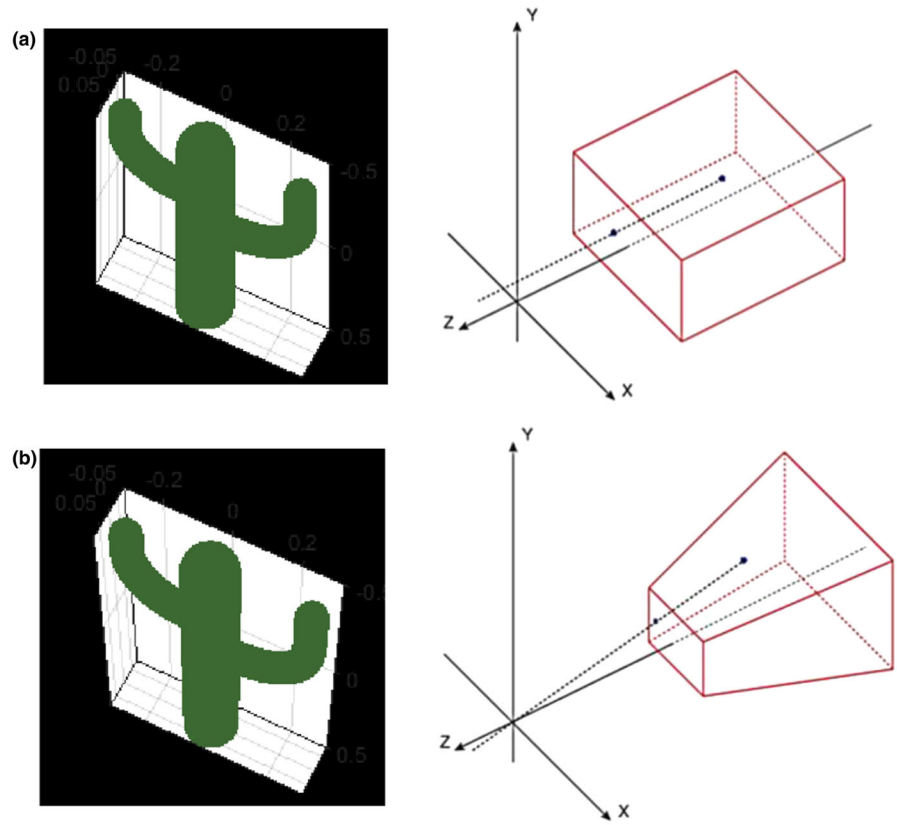
### 3.7 Projection of Detected Points in Pixel Coordinates to 3D World Coordinates

The resulting salient points need to be projected back on the surface of the object. For this purpose, we propose a simple procedure that allows projecting the 2D salient points on images from different viewpoints onto the surface of 3D models using the virtual camera model of Matlab. To simplify the 2D–3D projection, the virtual camera of Matlab is set to display objects using the orthographic projection, as the orthographic rendering gives a clearer measure of distance. In the orthogonal projection, all lines connecting a point on the real object to its corresponding point on the image are parallel. Figure 8 compares the orthogonal and perspective projection system, as well as the rendering result for the model of cactus (Fig. 9).



**Fig. 7** Salient point selection procedure in 2D

**Fig. 8** Comparison between **a** orthogonal and **b** perspective projection



**Fig. 9** Camera view angle for orthogonal projection

The first step in determining the 3D coordinate of a point in pixel coordinate is to determine the number of pixels in image that represent one world unit or the Pixel Per World Unit (*PPWU*). Figure 10 illustrates the geometry of the camera with orthogonal projection, where  $\alpha$  represents the camera view angle and  $d$  represents the distance of camera center to the object. Knowing the camera view angle and the distance  $d$  of camera center to the object, we can find the height value in the real world which is captured on the height of the image. Consequently, *PPWU* can be computed as:

$$PPWU = \frac{\text{Number of rows of the image}}{2 \times d \times \tan \frac{\alpha}{2}} \tag{4}$$

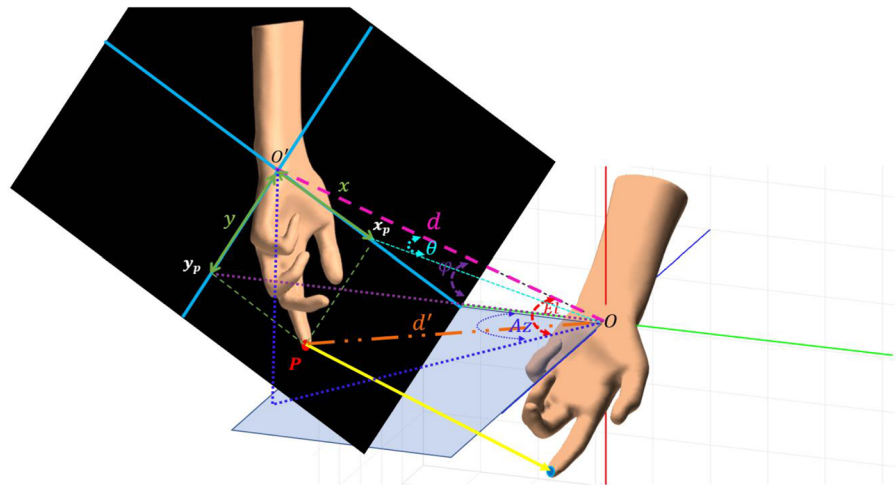
For the experimentations, we have positioned the Matlab virtual camera at distance 10 from the object targeting to the origin with camera view angle of  $6^\circ$ . These camera parameters ensure a complete sight for all objects in dataset while visualizing enough details.

Knowing the Pixel Per World Unit for captured images we can find the real-world coordinate of the point on the image plane using simple geometrical calculations. As depicted in Fig. 10, any arbitrary point on the captured image can be represented as  $P = (x_p, y_p)$ , where  $x_p$  and  $y_p$  are the pixel coordinate of the point in image array. Dividing  $x_p$  and  $y_p$  by the number of pixels presenting a unit of real world (*PPWU*) yields the size of  $x$  and  $y$  line segments on the image plane.

$$x = \frac{x_p}{PPWU}; y = \frac{y_p}{PPWU} \tag{5}$$

The camera is positioned at  $O' = (Az, El)$  where,  $Az$  and  $El$  are the azimuth and elevation angles of the

**Fig. 10** 2D–3D projection geometry



camera respectively and the camera movement is controlled by only changing these two angle values. The camera rotates around the view axis and its up-vector points towards the positive z direction (the angle between the positive z direction and the camera up vector can vary but remains an acute angle). With these assumptions, the spherical coordinate of the arbitrary point  $p$  on image plane can be obtained as:

$$p = (d', EL \pm \phi, Az \pm \theta) \tag{6}$$

Angles  $\theta$  and  $\phi$  express how much the azimuth and elevation angles of the point  $p$  differ from the azimuth and elevation angle of the camera center position. The vector from camera center position to the origin is perpendicular to the image plane, consequently the angles  $\theta$  and  $\phi$  in the right-angle triangles  $OOx_p$  and  $OOy_p$  can be calculated as:

$$\begin{aligned} \theta &= \tan^{-1} \frac{x}{d} \\ \phi &= \tan^{-1} \frac{y}{d} \end{aligned} \tag{7}$$

The distance between the point  $p$  and the origin ( $d'$ ) is:

$$d' = \sqrt{x^2 + y^2 + d^2} \tag{8}$$

The value of  $\theta$  and  $\phi$  can be added or subtracted from  $El$  and  $Az$  according to the location of the point  $p$  on the image plane quadrants.

The spherical coordinate of the point  $p$  is then converted to Cartesian coordinate for further calculations as:

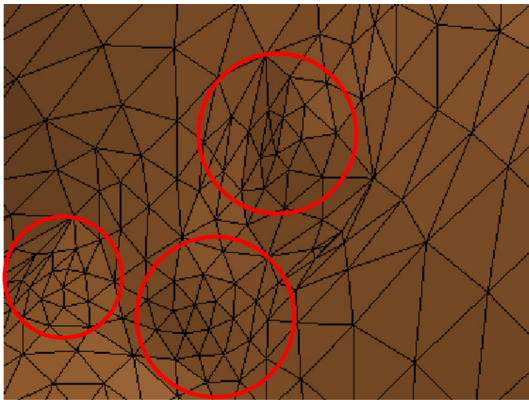
$$\begin{aligned} x &= d' \cos(EL \pm \phi) \cos(AZ \pm \theta) \\ y &= d' \cos(EL \pm \phi) \sin(AZ \pm \theta) \\ z &= d' \sin(EL \pm \phi) \end{aligned} \tag{9}$$

As previously discussed in orthographic projection, any line from a point on the image plane in parallel with camera view axis intersects its corresponding point in real coordinates. We have adapted the ray/triangle intersection model introduced by Moller and Trumbore [33] to find the location of the point  $P$  on the 3D object surface. The algorithm is a fast solution to find the intersection of a ray passing from a desired point and in a desired direction and gives the intersected face. Since for mesh simplification we need to identify salient vertices, the nearest vertex to the centroid of the intersected face is considered as the intersection point.

### 3.8 Adaptive Selection of Preserved Neighborhood

In our previous work [6], three immediate neighbors of all salient vertices were preserved while simplifying 3D meshes. This value was identified by trial and error and consisted in the computation of an error measure for various sizes of neighborhoods. The same size of neighborhood was chosen for all salient points. In this work, we propose an adaptive selection of number of reserved salient points according to the structure of 3D models.

As illustrated in Fig. 11, some regions in 3D models are denser than others. Preserving three



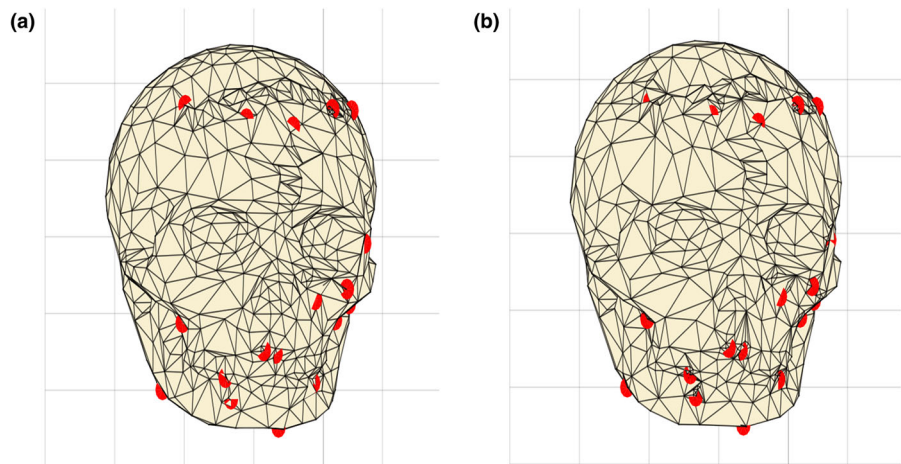
**Fig. 11** Different densities in a mesh structure

neighbors of a salient point in such dense regions degrades the quality of simplified mesh, as the preserved vertices are too close to each other. To deal with this issue, we propose to compute the distances between each salient vertex and all its neighbors and categorize them in three groups, as follows:

$$\begin{cases} \text{if } dist < \frac{1}{3} distmax & NPN = 1 \\ \text{if } \frac{1}{3} distmax < dist < \frac{2}{3} distmax & NPN = 2 \\ \text{if } dist > \frac{2}{3} distmax & NPN = 3 \end{cases} \quad (10)$$

where  $NPN$  is the number of preserved neighbors,  $dist$  is the average distance between a salient point and its immediate neighbors, and  $distmax$  is the maximum average distance from a salient point to its neighboring

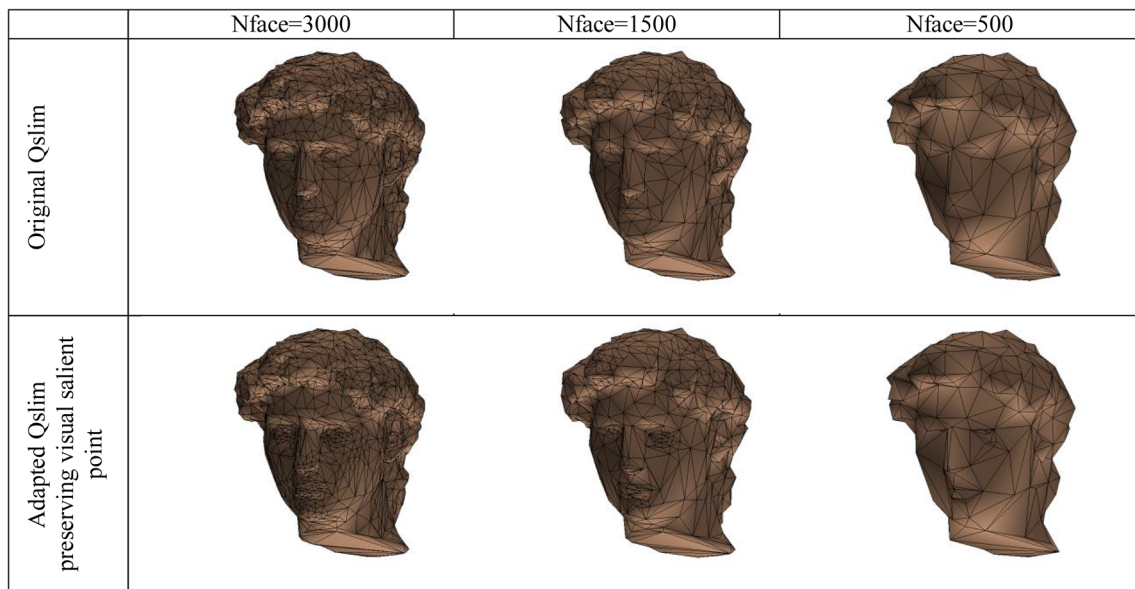
**Fig. 12** Simplified model of skull for **a** adaptive  $NPN$  simplification, **b**  $NPN = 3$  simplification



vertices. Figure 13 compares the model of skull for a simplification to 1500 faces when the  $NPN$  changes adaptively according to mesh density (Fig. 12a) with the case when  $NPN$  is constantly equal to three (Fig. 12b). The simplification algorithm is explained in the following section. One can notice that the adaptive scheme prevents creating unnecessary density in the areas which are already dense. Moreover, the quality of mesh is improved as it contains triangles of roughly equal sizes.

### 3.9 3D Model Simplification

Once the salient vertices of 3D models are determined using the previously explained approaches, the adapted Qslim [34] algorithm is applied to simplify meshes, while preserving the faces whose defining edge points are identified as a salient point or as a point in their immediate neighborhood and the number of preserved neighbors ( $NPN$ ) is determined adaptively according to the mesh density in different regions, as described in Sect. 3.9. Figure 13 compares the simplified models of the bust model by the original Qslim algorithm and the adapted version where salient regions are preserved. One can notice that Qslim algorithm allows a uniform simplification resulting in degradation of face details while in adapted version salient features are preserved even at low resolutions. Constructed meshes using this solution are denoted as *Adaptive ED* (for the method described in Sect. 3.2.1), *Adaptive SSIM* (for the method described in Sect. 3.2.2) and *Adaptive SVM* (for the method described in Sect. 3.4) respectively, in Sect. 4.



**Fig. 13** Simplified model of bust to 3000, 1500 and 500 faces with Qslim algorithm and adaptive Qslim algorithm

## 4 Experimental Results

To evaluate and compare the quality of the constructed meshes using the proposed algorithms, we tested our framework on the dataset for 3D object interest points [30]. As stated before, the dataset contains 43 models. It also contains the interest points identified by several other saliency detectors from the literature (i.e. Mesh saliency, Saliency points, 3D-Harris, 3D-SIFT, SD-Corners and HKS, described in Sect. 2) which allows a direct comparison between different methods and the proposed solution. The dataset provides only the triangular mesh structure of the objects. We have added color, specular, diffuse, reflectance and transparency characteristics to each 3D object using Matlab graphics adjustment to achieve a more realistic illustration and to be able to study the impact of the different color features. One can note that the color conspicuity map gets the least weight in both SSIM approach which can be explained by the fact that all models from the current data set are mono-colored while the human visual attention system is sensitive to color opponency, accordingly the conspicuity map is biased toward the color differences between objects and the background.

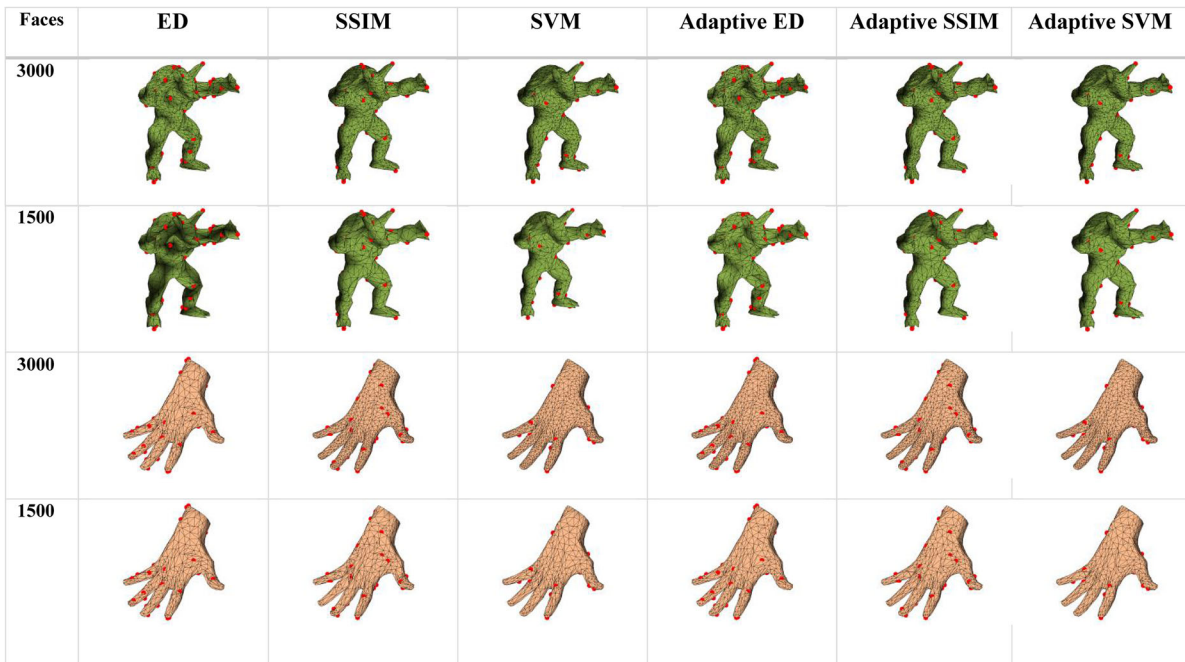
Objects are situated at the center of the coordinates system. For all experiments the camera is positioned at

distance 10 from the origin with camera view angle of  $6^\circ$ , as explained in Sect. 3.6. The procedure explained in Sect. 3 is applied to construct the selectively-densified meshes for all the objects in the dataset. A few simplification results for two objects extracted from the dataset, for 3000 and 1500 faces respectively, are presented in Fig. 14.

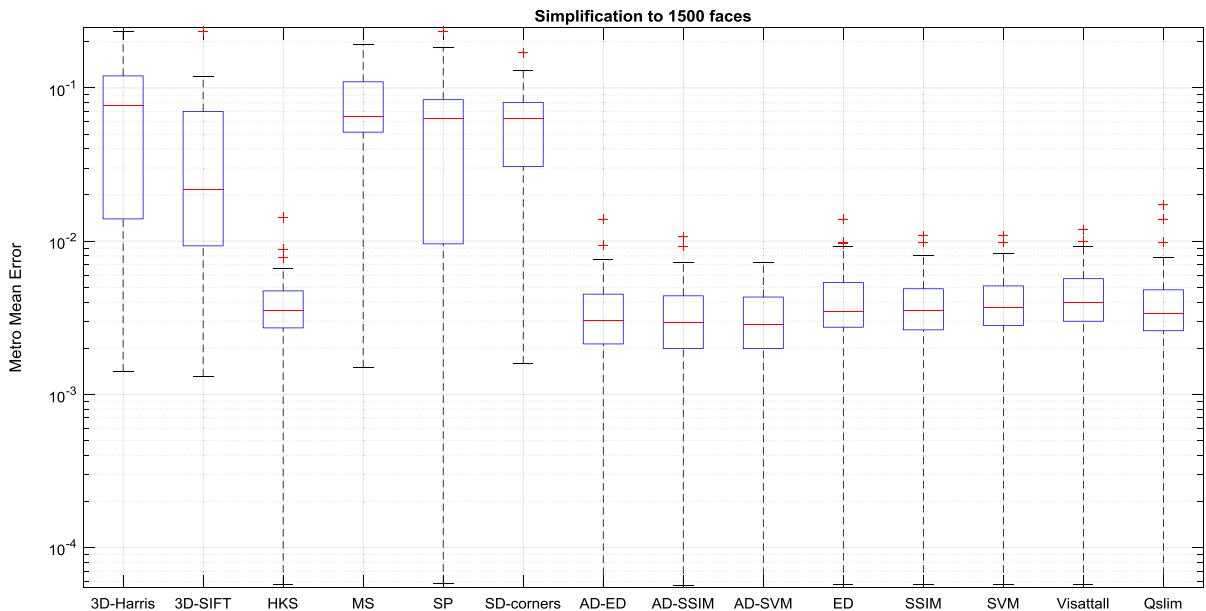
To obtain a quantitative measure of quality of the simplified objects, in the following sections, we computed a series of error measures.

### 4.1 Metro Error

To evaluate the proposed approach for mesh simplification we have adopted the Metro algorithm introduced in [35]. The algorithm measures the distance between a pair of surfaces using a surface sampling approach and returns three measures, namely the maximum, the mean and the root mean square (Hausdorff) distances from which the mean error is selected as the evaluation metric in this section. Figures 15 and 16 compare the mean error metrics for the proposed methods as well as for other saliency detectors from the literature for simplification to 1500 and 3000 faces respectively for the 43 models from the available data set. Error values are reported in a logarithmic scale.



**Fig. 14** Example of constructed meshes with the proposed methods for the models of armadillo and hand

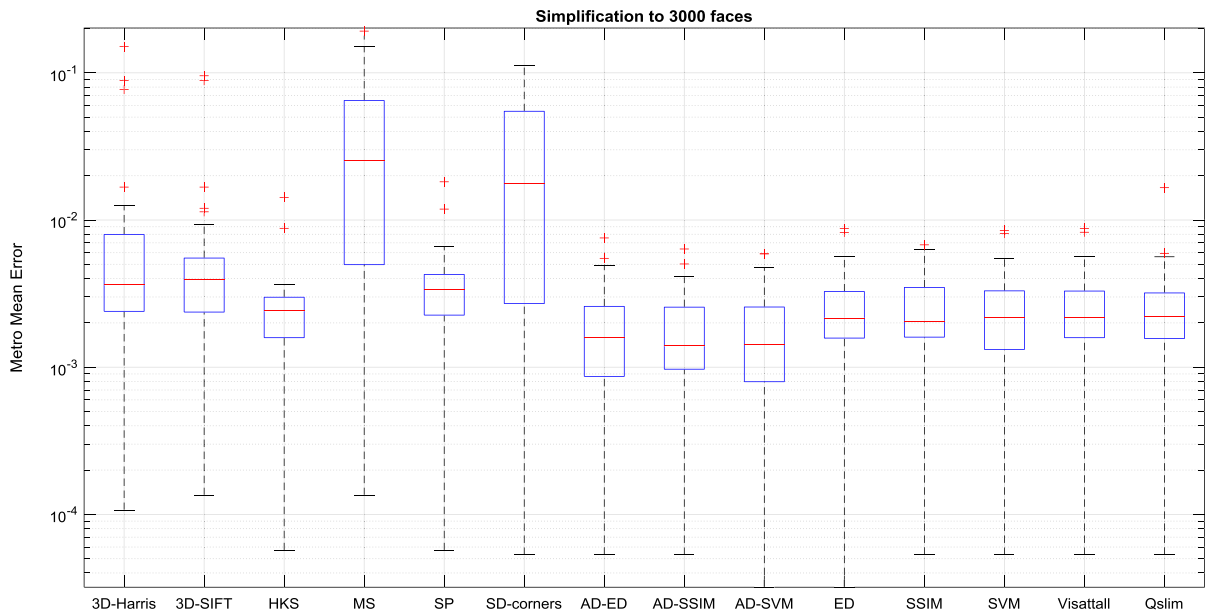


**Fig. 15** Metro mean error for simplification to 1500 faces

The experimental results confirm that Metro mean error for visual attention-based algorithms have a lower error level compared to the case where other saliency detectors are used, especially for lower

resolutions of the simplified mesh. The adaptive selection of the number of preserved neighbors (*Adaptive SSIM, Adaptive ED, Adaptive SVM*), results in a minor decrease in Metro mean error. Using





**Fig. 16** Metro mean error for simplification to 3000 faces

structural similarity measurement to determine the weight of conspicuity maps is slightly more efficient than the weighting procedure based on Euclidean distances. The reason is that the SSIM evaluates all the pixels of conspicuity maps and gives a superior assessment compared to Euclidean distance which only evaluates the position of brightest points on conspicuity maps. The *SVM* approach is distorted through image resize operation, but it still gives the most promising results. One can notice by studying Fig. 16 that simplifying meshes while preserving vertices detected by HKS algorithm also produces high quality models. However the Metro mean error for the proposed adaptive version of visual attention-based methods is slightly lower for models simplified to 1500 faces. Moreover, in the case of simplification to 3000 faces, all visual attention-based methods outperform HKS in terms of Metro mean error.

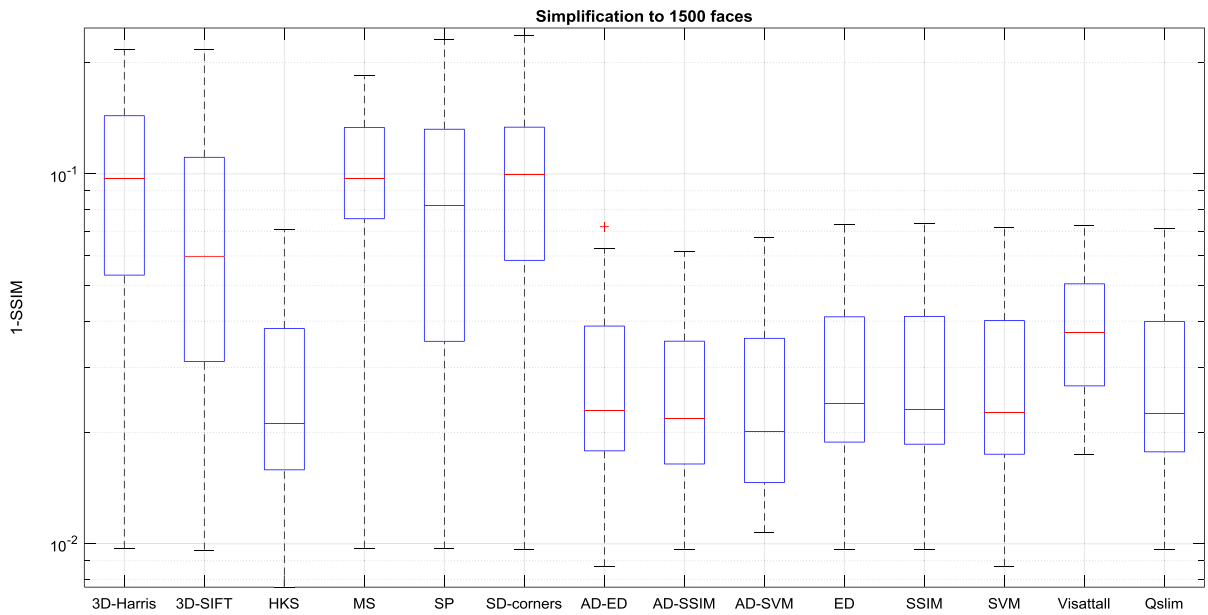
#### 4.2 Perceptual Errors

Since our solution is meant to create perceptually-improved models, in this section we take advantage of bio-inspired error metrics to evaluate the quality of constructed objects. The Structural Similarity Index method that we have already used in Sect. 3.2.1 to determine the contribution weight of each conspicuity

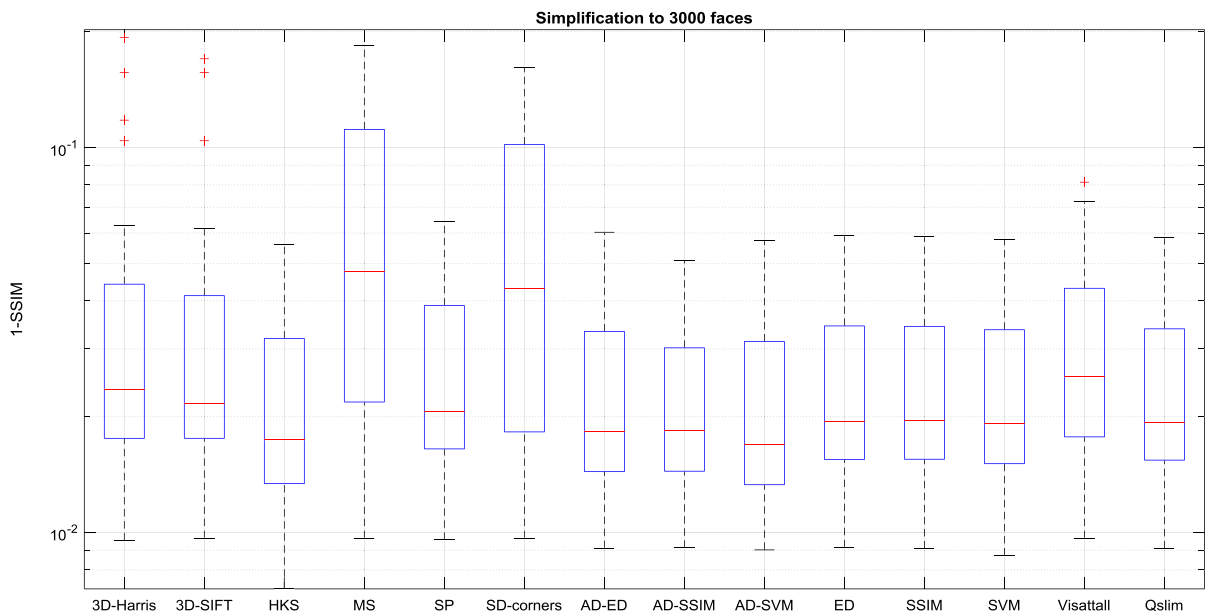
map is adopted once more to measure the similarity between images from the full high-resolution 3D model and the two constructed selectively-densified meshes with lower resolutions of 3000 and 1500 faces respectively. To report the similarity metric in form of error, the SSIM values are subtracted from one, which is the similarity measure for two identical images. The results are compared in Figs. 17 and 18.

The perceptual quality assessment of the simplified objects in most cases confirms the evaluation provided by Metro errors except that the perceptual quality of simplified meshes guided by HKS method is higher than the one achieved by visual attention approaches.

Table 2 summarizes and compares the results obtained by the three proposed salient point identification solutions, namely the *SSIM* based guided saliency map, the *ED* based guided saliency map and the *SVM* based learning approach. The *SSIM* method performs better than *ED* in terms of perceptual errors and both yield lower error rates compared to *Visattall* which assigns equal contribution weight to all features. Similar to the case of Metro errors, the adaptive version of all methods constructs higher quality object meshes. The overall results prove the superiority of the visual attention-based approach, in particular at low resolutions.



**Fig. 17** SSIM error for simplification to 1500 faces



**Fig. 18** SSIM error for simplification to 3000 faces

### 5 Conclusion

In this paper we proposed the use of a computational visual attention model that combines nine different features (which are believed to guide the deployment of visual attention) according to user inputs (in form of

ground-truth points of interest) to selectively simplify 3D object model for LOD representation. Matlab camera is turned around an object of interest to collect images from different viewpoints. Color, contrast, curvature, edge, entropy, intensity, orientation and symmetry features of images are employed to create

**Table 2** Comparison of proposed approaches

Method	Summary of the proposed method	Average Metro error for 43 models		Average SSIM error for 43 models		Advantages
		simplified to:		simplified to:		
		3000	1500	3000	1500	
SSIM	Structural Similarity index between each conspicuity map and the ground truth saliency map is computed as the contribution weight of each feature	0.0018	0.0035	0.0229	0.0258	Superior feature fusion technique compared to ED
ED	Euclidean distances between brightest points on each conspicuity map and the ground truth points are computed as the contribution weight of each feature	0.0019	0.0037	0.0236	0.0285	Better performance compared to the case with equally weighted features (i.e. <i>Visattall</i> )
SVM	The saliency map is predicted using a trained SVM	0.0018	0.0033	0.0221	0.0257	This solution is applicable to predict the salient points for objects whose ground truth points are not known a priori

the saliency map. The structural similarity between each conspicuity map and the ground truth saliency map, and the average Euclidean distance between bright points on the corresponding conspicuity maps and the location of salient points reported by users, are the two approaches developed to determine the contribution of each feature for guiding visual attention. Selected salient points are projected on the surface of 3D model using a geometry-based approach. A solution is proposed to control the expansion of saliency in the neighborhood around detected salient points. The number of preserved neighbors for a salient point located on a densified region of triangular mesh is smaller than the number of neighbors preserved in those cases where the salient point is situated farther than its neighbors.

The experimental results confirmed that visually interest points can be reliable to guide the mesh simplification process for different LOD. The results are also slightly enhanced in comparison with our previous version of visual-attention guided simplified objects.

**Funding** This work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

#### Compliance with Ethical Standards

**Conflict of interest** The authors declare that they have no conflict of interest.

#### References

- Kietzmann, T. C., Lange, S., & Riedmiller, M. (2009). Computational object recognition: A biologically motivated approach. *Biological Cybernetics*, *100*, 59–79.
- Luebke, D., & Hallen, B. (2001). Perceptually driven simplification for interactive rendering. In S. J. Gortler & K. Myszkowski (Eds.), *Rendering techniques*. Eurographics. Vienna: Springer.
- Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254–1259.
- Hadizadeh, H., & Bajic, I. V. (2014). Saliency-aware video compression. *IEEE Transactions on Image Processing*, *23*(1), 19–33.
- Frintrop, S., Rome, E., & Christensen, H. I. (2010). Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception (TAP)*, *7*(1), 6.
- Chagnon-Forget, M., Rouhafzay, G., Cretu, A.-M., & Bouchard, S. (2016). Enhanced visual-attention model for perceptually-improved 3d object modeling in virtual environments. *3D Research*, *7*(4), 1–18.
- Rouhafzay, G., & Cretu, A. -M. (2017). Selectively-densified mesh construction for virtual environments using salient points derived from a computational model of visual attention. In *2017 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA)*, Annecy, 2017 (pp. 99–104).
- Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B., & Huebner, R. (2003). *Level of details for 3D graphics*. Amsterdam: Morgan Kaufmann.
- Pojar, E., & Schmalstieg, D. (2003). User-controlled creation of multiresolution meshes. In *Proceedings of the*

- symposium on Interactive 3D graphics* (pp. 127–130). Monterey, CA.
10. Kho, Y., & Garland, M. (2003). User-guided simplification. In *Proceedings of ACM symposium on interactive 3D graphics* (pp. 123–126).
  11. Ho, T. -C., Lin, Y. -C., Chuang, J. -H., Peng, C. -H. & Cheng, Y. -J. (2006). User-assisted mesh simplification. In *Proceedings of ACM international conference on virtual-reality continuum and its applications* (pp. 59–66).
  12. Lee, C. H., Varshney, A., & Jacobs, D. W. (2005). Mesh saliency. *ACM SIGGRAPH*, 174, 659–666.
  13. Borji, A., & Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 35(1), 185–207.
  14. Frintrap, S. (2006). The visual attention system VOCUS: Top-down extension. In J. G. Carbonell & J. Siekmann (Eds.), *VOCUS: A visual attention system for object detection and goal-directed search. Lecture notes in computer science* (Vol. 3899, pp. 55–86). Berlin: Springer.
  15. Castellani, U., Cristani, M., Fantoni, S., & Murino, V. (2008). Sparse points matching by combining 3D mesh saliency. *Eurographics*, 27, 643–652.
  16. Zhao, Y., Liu, Y., Wang, Y., Wei, B., Yang, J., Zhao, Y., et al. (2016). Region-based saliency estimation for 3D shape analysis and understanding. *Neurocomputing*, 197(2016), 1–13.
  17. Lavoué, G., Cordier, F., Seo, H., & Larabi, M.-C. (2018). Visual attention for rendered 3D shapes. *Computer Graphics Forum*, 37(2), 191–203.
  18. Godil, A., & Wagan, A. I. (2011). Salient local 3D features for 3D shape retrieval. *SPIE 3D Image Processing and Application*, 7864, 78640S.
  19. Sipiran, I., & Bustos, B. (2010). A robust 3D interest points detector based on Harris operator. In *Eurographics 2010 Workshop on 3D Object Retrieval (3DOR'10)* (pp. 7–14).
  20. Novatnak, J., & Nishino, K. (2007). Scale-dependent 3D geometric features. In *IEEE international conference on computer vision* (pp. 1–8).
  21. Sun, J., Ovsjanikov, M., & Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Eurographics symposium on geometry processing* (Vol. 28, pp. 1383–1392).
  22. Mirloo, M., & Ebrahimezhad, H. (2018). Salient point detection in protrusion parts of 3D object robust to isometric variations. *3D Research*, 9, 2.
  23. Alliez, P., Cohen-Steiner, D., Devillers, O., Levy, B., & Desbrun, M. (2003). Anisotropic polygonal remeshing. *ACM Siggraph*, 22(3), 485–493.
  24. Song, R., Liu, Y., Zhao, Y., Martin, R. R., & Rosin, P. L. (2012). Conditional random field-based mesh saliency. In *IEEE international conference on image processing* (pp. 637–640).
  25. Howlett, S., Hammil, J., & O'Sullivan, C. (2005). An experimental approach to predicting saliency for simplified polygonal models. *ACM Transaction on Applied Perception*, 2(3), 1–23.
  26. Harel, J., Koch, C., & Perona, P. (2006). Graph-based visual saliency. In *Proceedings of the neural information processing systems* (pp. 545–552).
  27. Loy, G., & Eklundh, J. -O. (2006). Detecting symmetry and symmetric constellations of features. In *IEEE ECCV* (pp. 508–521).
  28. Derrington, A. M., Krauskopf, J., & Lennie, P. (1984). Chromatic mechanisms in lateral geniculate nucleus of macaque. *The Journal of Physiology*, 357, 241–265.
  29. Wolfe, J. M., & Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5, 1–7.
  30. Dutagaci, H., Cheung, C. -P., Godil, A. (2016) A benchmark for 3D interest points marked by human subjects. <http://www.itl.nist.gov/iad/vug/sharp/benchmark/3DInterestPoint>. Accessed August 1, 2017.
  31. Hughes, H. C., & Zimba, L. D. (1987). Natural boundaries for the spatial spread of directed visual attention. *Neuropsychologia*, 25(1), 5–18.
  32. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.
  33. Moller, T., & Trumbore, B. (1997). Fast, minimum storage ray/triangle intersection. *Journal of Graphics Tools*, 2(1), 21–28.
  34. Garland, M., & Heckbert, P. S. (1997). Surface simplification using quadric error meshes. In *SIGGRAPH '97 proceedings of the 24th annual conference on computer graphics and interactive techniques* (pp. 209–216).
  35. Cignoni, P., Rocchini, C., & Scopigno, R. (1998). Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2), 167–174.