

Selectively densified 3D object modeling based on regions of interest detection using neural gas networks

Ana-Maria Cretu^{1,2} · Maude Chagnon-Forget¹ · Pierre Payeur²

© Springer-Verlag Berlin Heidelberg 2016

Abstract The paper discusses automated solutions for 3D object modeling at multiple resolutions in the context of virtual reality. An original solution, based on an unsupervised neural network, is proposed to guide the creation of selectively densified meshes. A neural gas network, applied over a sparse density object mesh, adapts its nodes during training to capture the embedded shape of the object. Regions of interest are then identified as areas with higher density of nodes in the adapted neural gas map. Meshes at different level of detail for an object, which preserve these regions of interest, are constructed by adapting a classical simplification algorithm, the QSlm. The simplification process will therefore only affect the regions of lower interest, ensuring that the characteristics of an object are preserved even at lower resolutions. Various interest point detectors are incorporated in selectively densified meshes in a similar manner to enable the comparison with the proposed neural gas approach. A novel solution based on learning is proposed to select the number of faces for the discrete models of an object at different resolutions. Finally, selectively densified object meshes are incorporated in a discrete level-of-detail method for presentation in virtual reality applications.

Keywords Neural gas · 3D object models · Interest point detectors · Mesh simplification

1 Introduction

The current generation of graphics software packages offers enhanced capabilities to create extremely detailed and realistic 3D models associated with a high count of polygons. Similarly, the new generation of 3D data acquisition devices is able to collect large amounts of data in a relatively limited time. However, the increased complexity and size of datasets and models frequently represent a challenge for further processing, transformation and visualization at a reasonable computational cost, particularly in interactive applications where users expect to see results in real time. This explains the interest of researchers into identifying appropriate procedures that minimize the size and complexity of models by retaining only the most important characteristics of objects to be incorporated in interactive environments.

One approach to address the real-time constraint in computer graphics applications is to manage the level of detail (LOD) of an object. The three major categories of methods for managing the LOD of an object are: discrete, continuous and view-dependent methods (Luebke et al. 2003). Among these, the most frequently used is the discrete LOD method. Multiple copies of different resolutions of the same object, in which details are uniformly reduced, are created off-line, during a preprocessing step. Closer objects use a denser mesh associated with a higher resolution, while distant ones use a coarser resolution, leading to a higher rendering speed. The method is simple, but it generally takes long to prepare the multiple copies. Continuous LOD methods use instead a data structure, encoding a continuous spectrum of details, and the desired level is extracted from the structure at run-

Communicated by V. Loia.

✉ Ana-Maria Cretu
ana-maria.cretu@uqo.ca

Maude Chagnon-Forget
cham140@uqo.ca

Pierre Payeur
ppayeur@uottawa.ca

¹ Université du Québec en Outaouais, 101 Saint-Jean-Bosco, Gatineau, QC J8X 3X7, Canada

² University of Ottawa, 800 King Edward, Ottawa, ON K1N 6N5, Canada

time. This encoding is advantageous because it provides interruptible loading and progressive rendering properties. View-dependent methods extend continuous LOD methods by dynamically selecting the most appropriate level of detail for the current view of an object. Parts of the object that are closer exhibit higher detail than distant parts, resulting in a better fidelity but also in higher memory use and higher computation time. All three methods succeed in general to provide appealing results. However, due to the fact that they simplify the object geometry uniformly, they generally perform poorly at very low levels of detail. Some localized areas or features, while they could be visually more important than others, can be completely removed during the simplification process (Luebke et al. 2003). To address this issue, some researchers propose the intervention of the user to guide the desired quality of an object model. The user can therefore select the areas of lower quality that are then subject to local improvements (Kho and Garland 2003; Ho et al. 2006; Pojar and Schmalstieg 2003; Song et al. 2012). Only a few solutions were proposed to automatically guide the local mesh density of a model (Lee et al. 2005).

In this paper, regions of interest on the surface of an object are used as an automatic mean to manage the level of detail. The research work presented expands on the previously proposed method by the authors for the creation of selectively densified object meshes (Monette-Thériault et al. 2014). The regions of interest are detected by automatically finding higher density areas in the adaptation map of a neural gas network. A classical mesh simplification algorithm, namely QSLim (Garland and Heckbert 1997), is then adapted to only simplify the polygons that do not belong to regions of interest. These regions contain points of interest and points in the n -nearest neighborhood of points of interest. This selective simplification process allows the retention of the characteristics in the simplified model even at lower resolutions. A series of interest point detectors from the literature is integrated in a similar fashion with the QSLim algorithm to enable a comprehensive comparison with the proposed approach. Finally the proposed solution is incorporated in a discrete LOD scheme to be used in virtual reality applications. A novel solution is proposed to learn the mapping between the number of faces to be used for each of the multiple copies at different resolutions and an initial mesh at full resolution. The role of the latter is to eliminate the need for the user to provide the appropriate number of faces for each of the discrete copies of an object.

2 Literature review

There has been a lot of interest in the research community for the detection of points of interest. A first category of solutions for region of interest detection is capitalizing on

saliency, or the property of a certain region (or point) to be different from its surroundings (its neighborhood). Lee et al. (2005) compute mesh saliency using the curvature map, a mapping from each vertex to its mean curvature, based on the observation that changes in curvature identify regions that are different from their surroundings, and are therefore of interest. The approach is further improved in Liu et al. (2007) that combines the mesh saliency with Morse theory. Another improvement of Lee et al. (2005) is proposed in Song et al. (2012) that incorporates multiscale information in a conditional random field framework to impose consistency constraints between neighboring points. Castellani et al. (2008) re-mesh an object mesh at different levels of decimation and apply a Difference-of-Gaussian (DoG) operator to identify the points of interest over its surface. An adaptive inhibition process chooses only the vertices characterized by values larger than 85 % of their mean neighborhood values, and from these, only vertices that are local maxima and have a value higher than 30 % of the global maximum are retained as points of interest. In Wu et al. (2013), the mesh saliency is computed considering both the local contrast and global rarity, the latter being defined using the global saliency of each vertex based on its contrast with respect to all other vertices. Yang et al. (2009) calculate vertex saliency using its distance with respect to its neighborhood and include it in face saliency computation. Object regions are then ranked based on their face saliency and the ones with higher values are retained as regions of interest. In Dugatuci (2012), some of the previously discussed methods for interest point detection are compared with human-generated ground truth. Zhao et al. (2013a) compute the salient regions over a mesh by first diffusing with a random center surround operator the shape index field (calculated using the maximum and minimum curvatures at each vertex) and enhancing it using Retinex theory, a bilateral filtering based on the photometric spread. The results are segmented and interest points are either sampled using a stratified point sampling approach (Zhao et al. 2012) or using the entropy of saliency values (Zhao et al. 2013b) from each segment. The authors of Song et al. (2014) combine local multiscale saliency with global spectral response to obtain 3D mesh saliency. No interest points are identified, the saliency being employed for mesh segmentation and scan integration. In Gal and Cohen-Or (2006), salient geometric features based on curvature are obtained by first approximating locally a surface at each vertex by an analytic quadric patch, by selecting a representative point on each patch and by calculating analytically the differential properties (descriptors) at that point. Salient features are obtained by clustering sets of descriptors that have a high curvature with respect to their neighborhood and a high variance of curvature values. Feixas et al. (2009) compute view-based mesh saliency using mutual information between polygons. The saliency of a polygon is defined as the average dissimilarity, measured

in terms of Jensen–Shannon divergence, between itself and its neighbors. [Leifman et al. \(2012\)](#) detect regions of interest over a triangulated surface by searching regions that are distinct using three criteria: local vertex distinctiveness, shape extremities and patch association. A method is then proposed to select viewpoints based on these salient regions. A series of ten local descriptors, namely the first and second principal curvature, the Gaussian curvature, the mean curvature, the shape index, the log curvedness, the distance to local plane, the local volume, the spin image histogram and the spherical histogram, is used in [Creusot et al. \(2013\)](#) to find keypoints over the surface of 3D human faces.

Another category of interest point detectors is related to the identification of corners. In [Sipiran and Bustos \(2010\)](#), a Harris corner detector is proposed for 3D objects. In the method proposed by [Novatnack and Nishino \(2007\)](#), the surface of a mesh model is initially parameterized on a 2D plane, to create a distortion map that encodes the relative change in the edge length. A normal map is then constructed by interpolating the normals at each vertex, and a Gaussian filter, modified to take into account the distortion, is applied to the normal map obtained at different scales to allow for the detection of corners. The corners detected at several scales are then projected back on the 3D surface. In [Zaharescu et al. \(2009\)](#), corners are identified as extrema in across scales DoGs, further processed with a Hessian operator to eliminate non-stable responses. Other solutions capitalize on minimum and maximum values for the detection of points of interest. The solution proposed by [Godil and Wagan \(2011\)](#) uses a voxel grid and capitalizes on the SIFT algorithm. 3D Gaussian filters are applied at increasingly large scales and a DoG operator is used to identify minimum and maximum points, respectively. From these, only those located on the surface are retained. [Sun et al. \(2009\)](#) identify points of interest by finding local maxima of the heat kernel signature (HKS) computed over a triangular mesh. The information about the neighborhood of a point on a shape is calculated by recording the dissipation of heat from the point onto the rest of the shape over time. The authors of [Zou et al. \(2008\)](#) detect salient keypoints as local extrema of the DoG function defined over a curved surface in geodesic scale space.

The detection of points of interest plays a key role for various applications including: matching of objects and shapes ([Lee et al. 2005](#); [Gal and Cohen-Or 2006](#); [Zaharescu et al. 2009](#); [Zou et al. 2008](#)), mesh and shape retrieval ([Yang et al. 2009](#); [Godil and Wagan 2011](#)), mesh analysis ([Zhao et al. 2013b](#)), mesh segmentation ([Zhao et al. 2013b](#); [Song et al. 2014](#)), adaptive scanning of 3D objects ([Payeur et al. 2013](#)), scan integration ([Song et al. 2014](#)), and guiding mesh simplification ([Song et al. 2012, 2014](#); [Lee et al. 2005](#); [Monette-Thériault et al. 2014](#)). Finally a general evaluation of various 3D keypoint detectors in terms of repeatability, distinctiveness and computational efficiency for object detec-

tion, recognition and registration is presented in [Yu et al. \(2013\)](#) and for feature-based matching in [Tombari et al. \(2013\)](#).

As the interest in this paper is to guide the mesh simplification process using the points of interest, a few solutions are discussed that allow for user guidance to only impact certain regions of an object instead of its whole surface. It is worth mentioning that in most of the papers from the literature that consider this form of region-based guidance, the simplification algorithm of choice is QSlm proposed by [Garland and Heckbert \(1997\)](#). The main reason is the fact that the algorithm is known to ensure one of the best balances between fidelity, speed and robustness among all the simplification algorithms ([Luebke 2001](#)). A user-guided version of the QSlm algorithm is proposed in [Kho and Garland \(2003\)](#). The importance of different regions is controlled selectively by the user, while the algorithm ensures the preservation of features in those regions by local adaptive weighting and by imposing boundary constraints on contours, planes and points. In [Lee et al. \(2005\)](#), the order of simplification contractions of the QSlm algorithm is guided using a weight map that is obtained by thresholding the mesh saliency map, therefore making the weighting stronger in salient regions. The vertex pairs to be repeatedly contracted are ordered by increasing quadric error (collapsing cost) as in [Garland and Heckbert \(1997\)](#). [Song et al. \(2012\)](#) weigh the simplification process by amplifying the saliency values of the points situated in the regions of interest. The solution proposed by [Pojar and Schmalstieg \(2003\)](#) enables the user to control the simplification of a mesh by painting the regions that are considered important by the user in a Maya plug-in. Their algorithm weighs the collapsing cost, but also takes into account appearance attributes. In [Ho et al. \(2006\)](#), the user can improve regions that are considered unsatisfactory in a two-step procedure: in a first step the desired regions are weighted, while in the second step, a local refinement occurs. Most of these solutions count on the user guidance for the local improvement of an object mesh.

3 Proposed approach for regions of interest detection and their integration in a selectively densified 3D object model

In this paper, an original automated method is proposed to create selectively densified 3D object models based on interest regions over their surface. A neural gas network is initially employed to determine the regions of interest, using a sparse point cloud representing the uniformly down-sampled vertices of an object mesh. The identified regions are then incorporated in an adapted version of the QSlm simplification algorithm to create selectively densified meshes that encode the shape of 3D objects while preserving their

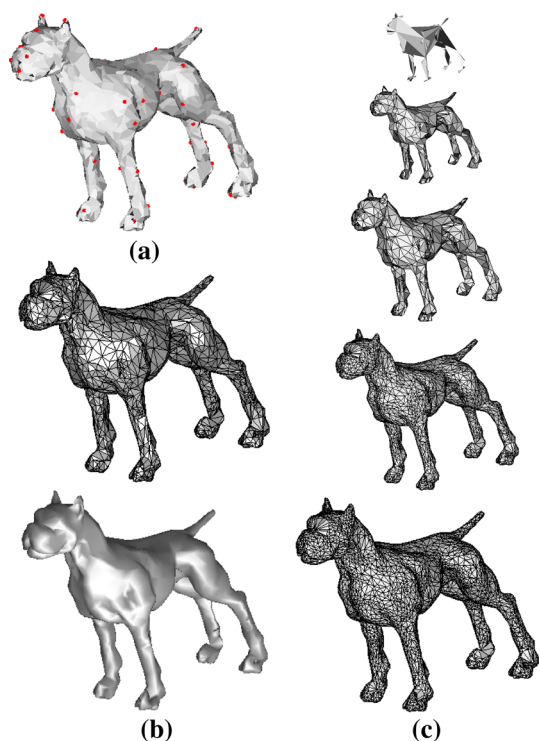


Fig. 1 **a** Interest points detected by the proposed method, **b** selectively densified mesh with interest region preservation for a number of faces in the simplified mesh equal to half of the faces in the initial mesh, without and with material properties, **c** discrete levels of details with region of interest preservation for a dog model

characteristics. Using a discrete LOD method inspiration, multiple copies of an object are created during an off-line pre-processing step. The details of object are uniformly reduced in the areas that are not detected as regions of interest, but are preserved for the interest points and their immediate n -neighborhood. Distant objects use a coarser resolution, therefore a smaller number of polygons, but ensure that the characteristics of an object are preserved. An illustration of the approach is shown in Fig. 1 for a dog model. Figure 1a illustrates identified interest points, shown in red, using the proposed method over the object surface, while Fig. 1b shows the selectively densified model obtained for half of the number of faces in the initial model as a mesh and also using a gray material.

The discrete level-of-detail copies for the same model are shown in Fig. 1c. One can notice that even at very low level of detail, as in the upper part of Fig. 1c, the most important features of the object (i.e. tail, ears, etc) are preserved.

3.1 Neural gas networks

In general, the purpose of neural gas networks is to cluster multi-dimensional vectors. Such a network consists of nodes (or cluster centers), which independently move over

the data space during the adaptation process. The adaptation algorithm starts by initializing the set of network nodes with a predefined number of nodes whose corresponding reference vectors are chosen randomly according to a probability density function or from a finite set (Martinetz et al. 1993; Cretu 2009). Each node has an associated reference vector that indicates its position in the input space. At each training step, the winning neuron that best matches an input vector is identified using the minimum Euclidean distance criterion. The neurons to be adapted in the learning procedure are selected according to the rank they have in an ordered list of distances between their weights and the input vector. A full description of the neural gas algorithm is available in Martinetz et al. (1993). In the context of this work, the neural gas is employed to model a decimated point cloud representing a 3D object. This decimated point cloud is obtained by uniformly reducing the initial point cloud to 10% of its size. Starting from the resulting sparse point cloud, a neural gas network adapts a predefined number of nodes to the shape of the object contained in the point cloud. During this adaptation process, the nodes converge toward regions where more pronounced depth variations are present and therefore where local features are situated.

The algorithm starts by initializing the set S of network neurons to contain N units c_i with the corresponding reference vectors w_{ci} having very small values, of the order of 10^{-5} . Input vectors x are chosen randomly from a finite training data set that in the current case contains the X, Y, Z coordinates of vertices in the sparse point cloud, $x = [XYZ]$. The nodes to be adapted in the learning procedure are selected according to their rank in an ordered list of distances between their weights and the input vector. Each time a new input vector x is presented to the network, a list of neighborhood ranked indices is built (j_0, \dots, j_{N-1}) , where w_{j_0} is the weight of the closest neuron to x , w_{j_1} the weight of the second closest neuron, and w_{j_k} is the reference vector such that k vectors w_i exist with: $\|x - w_i\| \leq \|x - w_{j_k}\|$, $k = 0, \dots, N - 1$. In each training step, the best matching neuron $s(x)$ at time t is computed using the minimum Euclidean distance. All neurons are then ordered and based on the rank, a given number of nodes is adapted according to:

$$w_j(t+1) = w_j(t) + \alpha(t)h_\lambda(k_j(x, w_j))[x(t) - w_j(t)] \quad (1)$$

where $\alpha(t) \in [0, 1]$ describes the overall extent of the modification, and h_λ is 1 for $k_j(x, w_j) = 0$ and decays to zero for higher values according to Eq. (2):

$$h_\lambda(k_j(x, w_j)) = \exp(-k_j(x, w_j)/\lambda(t)) \quad (2)$$

As in Martinetz et al. (1993), $k_j(x, w_j)$ is a function that represents the ranking of each weight vector w_j . If j is

the closest to input x then $k = 0$, for the second closest $k = 1$ and so on. The learning rule is local, meaning that the operation at neuron level only requires the knowledge of the weight associated with that neuron and the input vector. The network minimizes the following cost (error) function:

$$E(w, \lambda) = \frac{1}{2C(\lambda)} \sum_{j=1}^N \int P(x(n)) h_{\lambda}(k_j(x, w)) \cdot (x - w_j)^2 \quad (3)$$

where $P(x)$ is the distribution of input vectors $x(n)$ and

$$C(\lambda) = \sum_{k=0}^{N-1} h_{\lambda}(k)$$

The parameter λ can be seen as a temperature factor. At very low temperatures only the winning node produces a cost. As the temperature is increasing, only the nodes close to the input contribute significantly to the error, while at high temperatures many of the nodes in the network add to the cost of the network. The corresponding update rule that is applied to all nodes reduces the cost function in Eq. (3).

The learning rate $\alpha(t)$ and the function $\lambda(t)$ are both time-dependent. To ensure that the algorithm converges, it is important that these parameters are decreased slowly during the learning process. The following time dependencies are used:

$$\alpha(t) = \alpha_o(\alpha_T/\alpha_o)^{t/T}, \quad \lambda(t) = \lambda_o(\lambda_T/\lambda_o)^{t/T} \quad (4)$$

where the constants α_o and λ_o are the initial values for $\alpha(t)$ and $\lambda(t)$, α_T and λ_T are the final values, t is the time step and T the training length. For the time-dependent variables, initial and final values have to be chosen. The algorithm continues to generate random input signals x while $t < T$.

The neural gas network converges quickly, reaches after convergence a lower distortion error and better captures finer details than other self-organizing architectures (Cretu 2009). However, its main limitation is related to the fact that the initial number of nodes of the neural gas map needs to be decided prior to adaptation (tuned by the user) and that impacts the accuracy of the results. After multiple trials, a series of guidelines have been developed to choose an appropriate map size according to the size of the initial scan (Cretu 2009), as shown in Table 1.

Nevertheless, the desired accuracy of the model can vary slightly with the characteristics of the objects considered, such as the size of the object, and the number and the size of the features. Therefore, to ensure that the best map is chosen, multiple neural gas map sizes are tested and the best is chosen according to an error measure, as further described in Sect. 3.4. The other parameters are set as follows: $\alpha_o = 0.5$ and λ_o

Table 1 Neural gas map size with respect to the initial model

Size of initial model (number of points, N)	Approximate neural gas map size
2000–3000	60–100% N
3000–4000	50–90% N
4000–5000	30–70% N
5000–6000	20–60% N
6000–7000	15–50% N
7000–15,000	10–40% N
Over 15,000	5–10% N

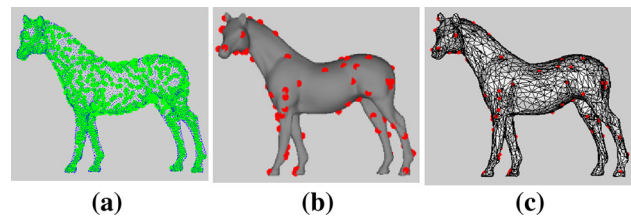


Fig. 2 **a** Neural gas map, **b** points of interest detected in the neural gas map and **c** selectively densified 3D horse model

is set equal to half the number of neurons in the initial map. The network is trained for 15 epochs.

3.2 Region of interest detection based on neural gas

The regions of interest are then identified by detecting areas with a higher density of nodes in the adaptation map of the neural gas. A Delaunay triangulation is first applied to the neural gas output, knowing that areas of high density of nodes are represented by smaller triangles in the resulting mesh. All the edges of triangles larger than a threshold are removed from the tessellation. The threshold is selected empirically as half of the mean value of the length of vertices between every pair of nodes for every triangle in the tessellation. This process ensures the identification of points that are close to one another and, therefore, of areas that contain features. This approach is used to identify regions (and implicitly points) of interest over the surface of a 3D object. In particular, points of interest are those that represent vertices in the remaining tessellation. Figure 2 illustrates the detected points of interest for a horse model.

Figure 2a shows with green points the neural gas map for a size of 1500 nodes in the map, selected according to Table 1, while Fig. 2b illustrates the detected points of interest, in red, using the procedure described above. Figure 2c shows the selectively densified object mesh that preserves the regions around these interest points.

3.3 Incorporation of regions of interest in the selectively densified 3D model

The simplification algorithm proposed in Garland and Heckbert (1997), QSLim, is chosen and adapted into an automated solution to incorporate the detected areas of interest and create a selectively simplified mesh M_s . The QSLim algorithm can be briefly described as follows: given a triangular mesh $M(V, F)$, where V is a set of vertices v_i and F is a set of faces, or more precisely a set of triplets $\{v_j, v_k, v_l\}$ that define three vertices that each create a face in the mesh, in counter-clockwise order, it simplifies a mesh by repeated edge contractions (collapses) based on a quadric error metric computed of each vertex of the mesh. This metric is a 4×4 matrix that represents the sum of squared distances from the vertex to the planes of neighboring triangles. If its value is large, the corresponding vertex could be considered a distinctive feature over the surface of the object, and therefore will be retained longer in the mesh. Otherwise, it will be removed earlier. The same metric is used to calculate the cost of a contraction and the optimal position for the unified vertex resulting after an edge removal. In this way, each edge is associated to a contraction cost and all the edges are stored in an ordered list of costs. The idea of the algorithm is then to remove at each step the edge with the lowest cost, update the neighborhood as a result of the contraction and update the costs of edges connected to the unified vertex. The most important parameter to control is the final resolution of the

mesh, or number of faces, f_i , to be retained after the simplification. Most solutions available in the literature weigh stronger the regions of interest after the creation of a simplified mesh (Ho et al. 2006; Song et al. 2012; Lee et al. 2005) and adjust their cost to delay their simplification. Instead, in this work, the simplification process is not allowed to affect regions of interest at all. In particular, the faces of the mesh that contain points of interest and their immediate neighboring faces (n -nearest neighbors) are eliminated from the list of faces to be simplified by the QSLim algorithm. Because the number of faces is fixed at the beginning of the algorithm, the faces will be distributed over the object model in a selective manner, with a higher density in the regions of interest.

The choice of the neighborhood size, n , is further discussed in the experimental results section. An improvement with respect to the previous version in Monette-Thériault et al. (2014) is the use of a different algorithm for the selection of the nearest neighborhood. In Monette-Thériault et al. (2014) a version of the quick hull method was used for the detection of neighbors of interest points. In this work, a k -nearest algorithm is used as a basis for the identification of regions of interest around points of interest. It is faster to compute (improvement of about 25% in computation time) and leads to a slightly better quality of the simplified mesh (i.e. lower error with about 5%).

The algorithm in pseudo-code for our approach is the following:

```

Step 1:
// interest points detection for an object O represented by a mesh M of size N
Obtain sparse 3D point cloud representing an object in a mesh M by uniform decimation:
O =  $\{P_1, \dots, P_i, \dots, P_{NS}\}$ ,  $P_i = [X_i, Y_i, Z_i] \in \mathbf{M}$ ,  $i=1 \dots NS$ ,  $NS < N$ ;
Initialize neural gas network;
Apply neural gas adaptation over O to obtain the neural gas map:
ONG =  $\{P_1, \dots, P_j, \dots, P_{ND}\}$  |  $P_j = [X_j, Y_j, Z_j]$ ,  $j=1 \dots ND$ ,  $ND < NS/2$ ;
Apply Delaunay on the resulting map ONG to obtain:
D =  $\{P_1, \dots, P_j, \dots, P_{ND}\}$  |  $P_j = [X_j, Y_j, Z_j]$ ,  $j=1 \dots ND$ 
Threshold D to identify the interest points P for the object;
P =  $\{P_1, \dots, P_k, \dots, P_{NP}\}$  |  $P_k = [X_k, Y_k, Z_k]$ ,  $k=1 \dots NP$ ,  $NP < ND$ 
Step 2:
// simplification for a mesh M to obtain selectively-densified mesh Ms
Calculate PN = the  $n$ -nearest neighborhood of points P in M,  $n=1, 2, \dots$ 
Recuperate edges on the mesh that do not connect to points of interest or
neighbours of points of interest:
ED =  $\{e(u,v) \mid e(u,v) \in \mathbf{M}, e(u,v) \notin \mathbf{PN}\}$ 
Initialize Ms=ED
// apply QSLim
Compute quadric error at each vertex of Ms
Determine contraction cost at each edge  $e(u,v)$  in Ms
Create ordered list of edges based on cost
Step 3:
Remove the edge  $e(u,v)$  with lowest cost
Use quadric error to choose the optimal contraction target
Contract  $u$  and  $v$  and recalculate costs for the adjacent edges in Ms
While desired resolution (number of faces) not reached repeat Step 3

```

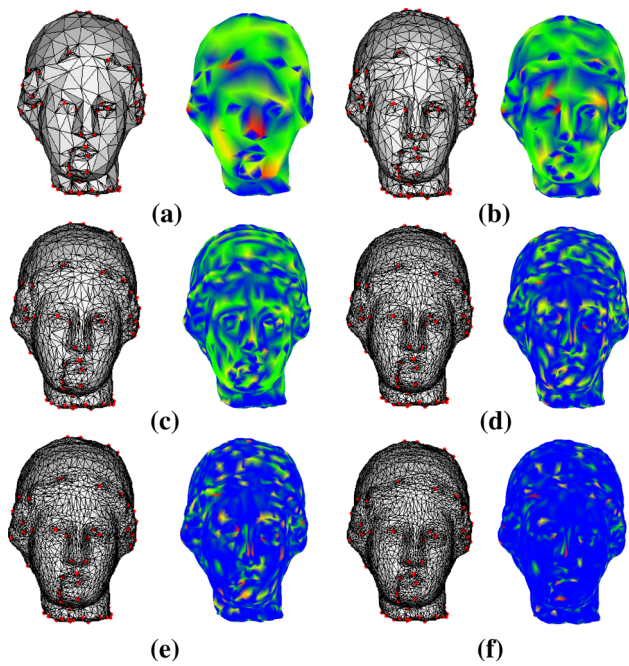


Fig. 3 Proposed method with regions of interest preservation for the Igea model with interest points shown in red, and selectively densified models with **a** 2500, **b** 3500, **c** 4500, **d** 5500, **e** 8500 and **f** 10,000 faces in the simplified mesh and their corresponding error distribution and magnitude, from blue (perfect match) to red (large error) (color figure online)

P, the set of interest points, can be obtained using any interest point detector, for example any of the detectors that are used for comparison in Sect. 4. Because the proposed region of interest detector in Sect. 3.1 runs on Matlab, the Matlab wrapper of QSlim is adapted from Peyre (2007) and the same platform is used to implement the proposed solution.

3.4 Comparison of simplified meshes

The Metro software (Cignoni et al. 1998) is the solution chosen to enable the comparison of different selectively simplified meshes. To measure the output quality of simplified meshes, Metro uses a point-to-surface distance metric. It takes as input a pair of surfaces, meshes of objects in our case, and outputs the maximum distance, as well as the mean and variance from the first mesh to the second mesh. Using the initial mesh as a reference, the reported distance can be viewed as a measure of error with respect to this mesh. It therefore allows for a quantitative comparison of different simplified meshes. The Cloud Compare tool (<http://www.danielgm.net/cc/>) is employed for a visual comparison of error distribution and magnitude over the simplified mesh with respect to the initial mesh.

3.5 Multi-resolution object modeling

Taking inspiration from discrete level-of-detail methods, an original solution is proposed to create automatically the multiple copies of an object at different resolutions. For this purpose, the simplification algorithm with region of interest preservation is applied for an increasing number of faces, varying from 1500 up to the entire number of faces in the initial mesh of an object. A neural network is trained to learn the mapping between the initial mesh, at full resolution and the number of faces for each of the multiple copies at different resolutions. This replaces the need for a user to estimate by trial and error the number of faces that would be required for an object. A two-layer feed-forward architecture with 40 neurons in the hidden layer is employed for this purpose. The

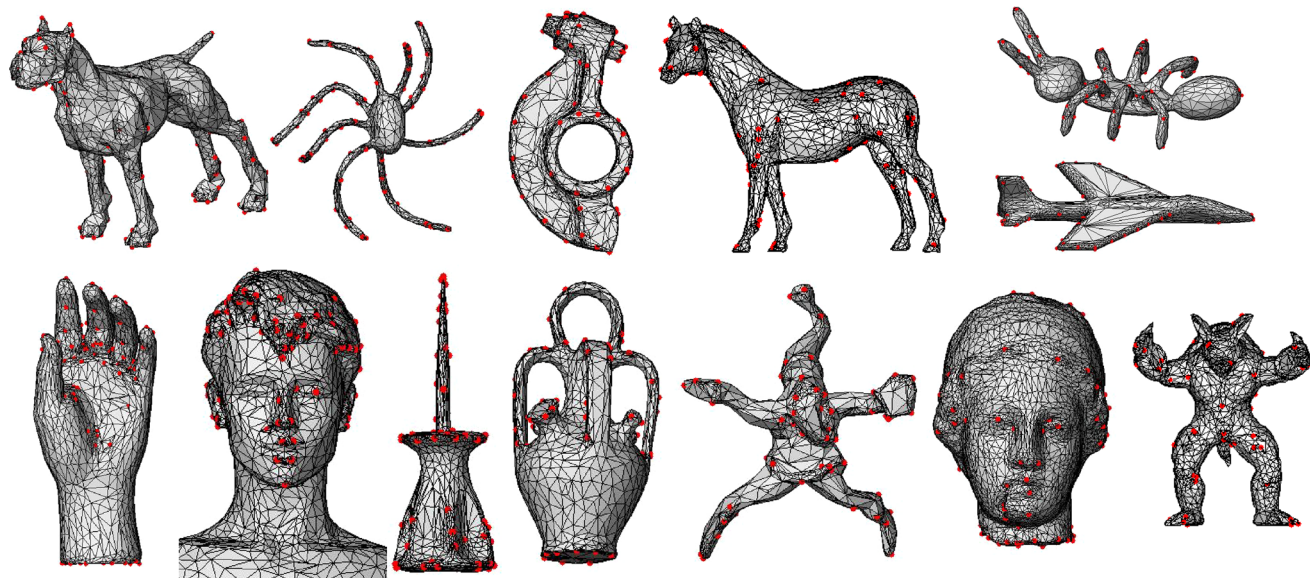


Fig. 4 Selectively densified models obtained using the proposed approach for a sample of objects extracted from the dataset

number of neurons is empirically determined. To perform the training of this network, among the multiple copies of simplified meshes within a given range of resolution, the one that obtains the lowest error (i.e. lowest maximum distance) within the range is chosen as a sample for training. The range of resolutions is calculated by dividing the interval between 1500 faces and three-quarter of the number of faces in the entire mesh in the desired number of copies of different resolutions that one wants to obtain. When provided, at run-time, with the mesh of an unknown object, the trained network outputs the number of faces for each of the copies, from coarse to detailed. The simplification algorithm with regions of interest preservation is then applied to create the discrete level-of-detail copies, constraining the simplification to the identified number of faces.

4 Experimental results

To evaluate the proposed framework, the approach is tested over all the objects from the dataset associated with the benchmark for 3D interest point detection algorithms (<http://www.itl.nist.gov/iad/vug/sharp/benchmark/3DInterestPoint/>). This dataset is chosen as it allows a direct comparison of the neural gas-based detection of regions of interest for the same objects with other interest point detectors identified in the benchmark, namely mesh saliency (MS) (Lee et al. 2005), salient points (SP) (Castellani et al. 2008), 3D-SIFT (3DS) (Godil and Wagan 2011), 3D Harris (3DH) (Zhao et al. 2012), scale-dependent corners (SDC) (Novatnack and Nishino 2007) and HKS (Sun et al. 2009). Figures 3 and 4 illustrate samples of results over the dataset obtained with the neural gas approach. Figure 3a–f illustrates the results obtained when the number of faces is gradually increased, starting from coarse to detailed, as a mesh with the detected interest points marked in red and as a model containing the error distribution and magnitude over the object surface. The latter models show the errors in green (smaller error), yellow (medium error) and red (large error), while the regions in blue show a perfect match to the initial mesh. One can notice that even at very low number of faces, such as in Fig. 3a, the regions containing areas of interest (i.e. eyes, mouth) are very well preserved. Figure 4 presents selectively densified models, obtained using the proposed approach, for several objects in the dataset, for a number of faces in the simplified mesh equal to half of the number of faces in the initial mesh and for a preserved neighborhood $n = 3$ around each interest point.

In both of these figures, it can be noticed that the simplified mesh has a higher density in the regions of interest.

Figure 5 compares the proposed approach with regions of interest preservation (Fig. 5d and enlarged in 5e) with the case where the QSlm is not constrained in the regions of interest

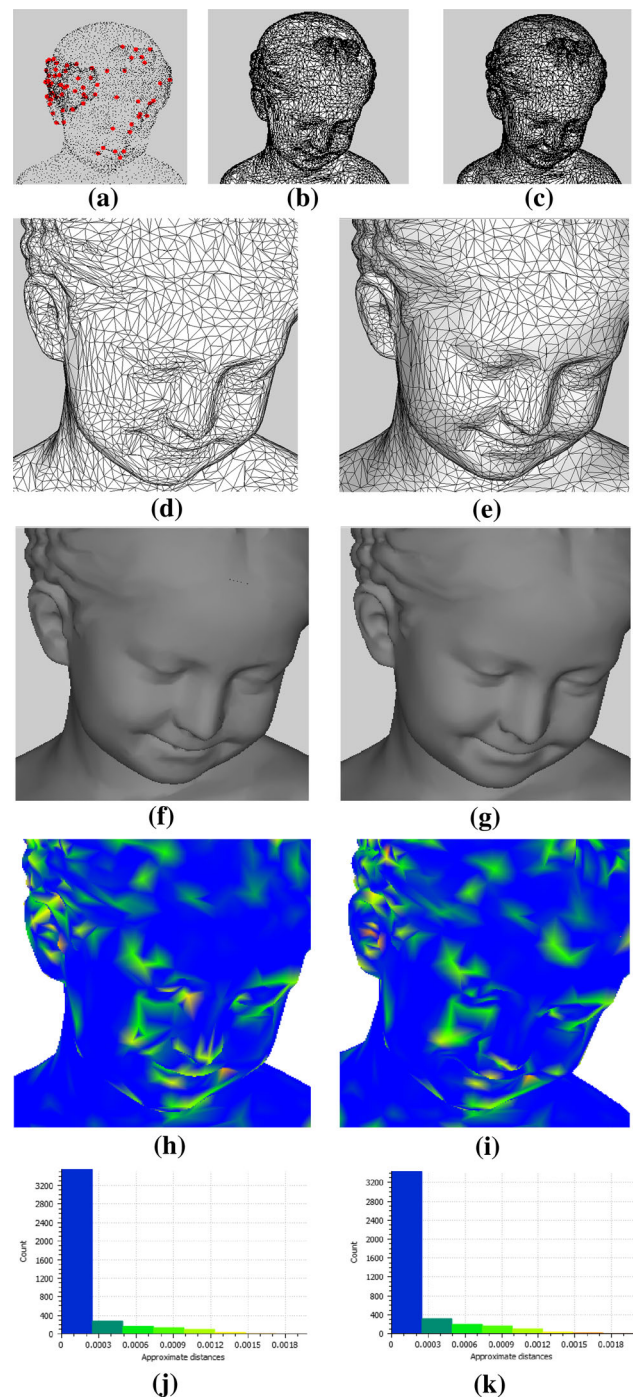
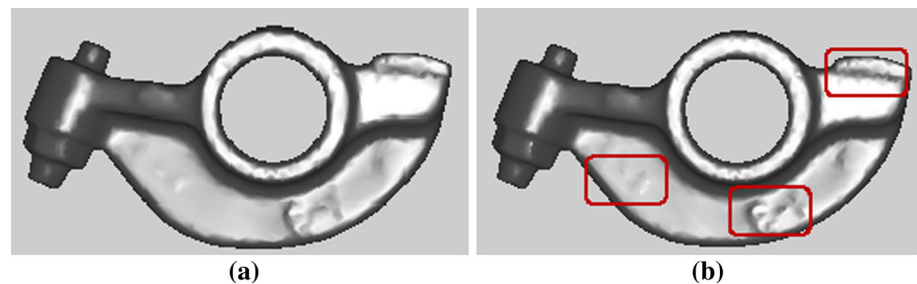


Fig. 5 QSlm without (*left column*) and proposed solution with region of interest preservation (*right column*) for the girl model: **a** identified interest points shown in *red*, **b–e** simplified mesh and details for the same number of faces, **f, g** textured meshes, **h, i** error distribution and **j, k** error magnitude (color figure online)

(Fig. 5b, c) for the same number of faces, that is 15,000 faces in the simplified mesh. Because in general, objects in virtual environments are presented as colored and textured meshes, in Fig. 5f and 5g a comparison is illustrated between the

Fig. 6 Comparison for the same number of faces (half of the initial model) of **a** standard QSlim simplification with **b** the proposed approach (color figure online)



QSlim simplified mesh and the selectively densified mesh using a dull gray material and Gouraud lighting.

Figure 5h–k compares the error distribution and magnitude for the two solutions. While the overall error is slightly higher for the proposed method, as it can be observed by comparing Fig. 5k with Fig. 5j, the error is not located in the same regions. In particular, for the proposed solution, the error is lower (i.e. more blue) in the areas of interest, namely around the nose, the eyes and the mouth, as illustrated in Fig. 5i versus Fig. 5h.

Figure 6 illustrates the same comparison between the standard simplification and the proposed solution for a shiny material. The fine details are more visible in Fig. 5g versus Fig. 5f and Fig. 6b (in particular the areas marked in red) versus Fig. 6a that present the results obtained by the proposed interest region preservation method.

The identification of interest points using neural gas takes on average 10s for each 10,000 vertices of an initial model for a Matlab platform, running on a Windows machine with an Intel Pentium CPU at 4.5 GHz and 2 GB of RAM. An additional 0.06s per each batch of 10,000 faces of the initial model is required for the adapted QSlim algorithm. Figure 7 compares visually the simplified meshes for several interest point detectors, namely SS (standard simplification without region of interest preservation), MS, SP, 3DH, 3DS, SDC, HKS, and the proposed solution using neural gas, NG, for a neighborhood $n = 3$ and a number of faces in the simplified mesh equal to half of the number of faces in the initial mesh. It is worth mentioning that the series of objects presented in the figure is selected such that the final shape of the object remains undistorted.

To allow for a quantitative comparison of performance, the simplified meshes obtained by the different interest point detectors are evaluated using the error measures detailed Sect. 3.4. To perform this comparison, the interest points provided in <http://www.itl.nist.gov/iad/vug/sharp/benchmark/3DInterestPoint/> for the interest point detection methods are incorporated in the adapted QSlim algorithm in a similar manner to the one proposed for our approach. The error measures are computed in each case with respect to the full size mesh.

A slightly higher error is in general expected with the preservation of regions of interest, because the triangles of the mesh are redistributed to better cover these regions and

therefore resulting in a less detailed coverage of regions that are less important, as explained above. As the error is calculated as an average over all triangles of the mesh, this fact will lead to slightly higher errors for methods that include region preservation with respect to a standard simplification (SS).

Table 2 summarizes the number of interest points achieved by each method, the average error measures computed over the objects in the dataset and as well as the percentage of distorted objects with respect to number of objects in the dataset. The error measures are computed for a number of faces equaling one-fourth (1/4), one-half (1/2), three-quarter (3/4) and the entire (1) number of faces in the initial dataset, respectively, and for a size of neighborhood, $n = 3$. It can be noticed that interest point detectors such as SDC and MS are the most affected by the distortion phenomenon, and that the phenomenon is amplified with an increase in the neighborhood size (decrease in the number of faces in the simplified mesh). When an interest point detection method returns many interest points, and therefore constrains a large number of triangles to be fixed and not affected by the simplification algorithm, the number of remaining triangles may be not large enough to preserve the shape of the object in the areas that are considered less important. The proposed method, NG, is less affected by distortion when compared with most of other interest point detectors. It is surpassed only by the standard simplification (SS) and the HKS detector that returns a very low number of interest points. This implies that the proposed method can overall achieve lower resolution copies of a model and higher compression rates than the other interest point detectors.

In order not to bias the reported error values, the distorted objects are removed when computing the average error reported in the table. The error value is therefore computed as an average over all the objects in the database whose simplified mesh is undistorted. The columns 3–5 of Table 2 display the average error measures over all objects, namely the maximum, mean and variance, respectively. One can notice however that the increase in error is not significant for methods such as HKS and our solution, NG, with respect to the standard simplification, SS. The redistribution process will also lead in general to an increased error when a larger number of points of interest are used. As one can notice in the second column of Table 2, the proposed method, NG,

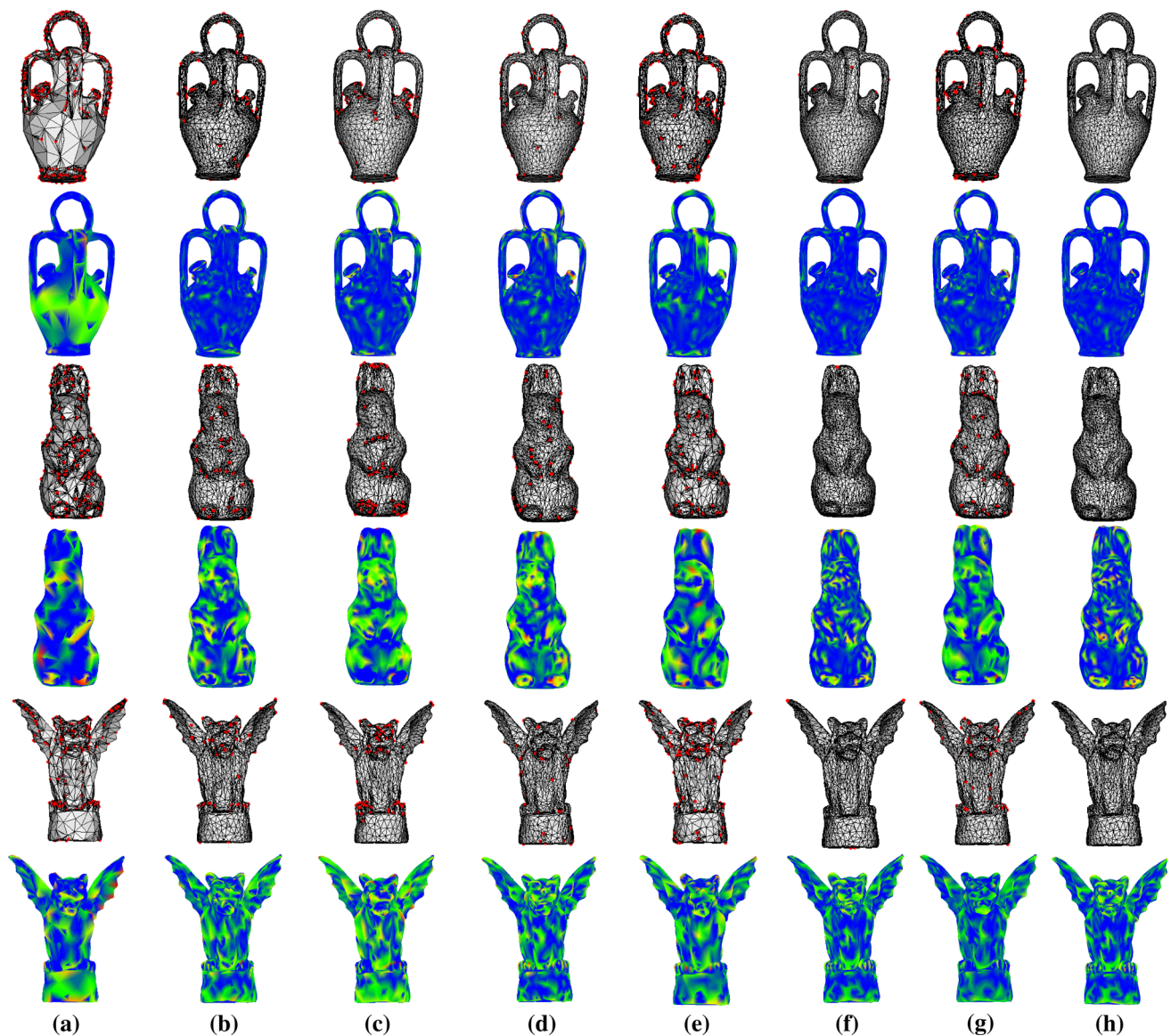


Fig. 7 Comparison in terms of interest points and mesh quality: **a** MS, **b** SP, **c** 3DH, **d** 3DS, **e** SDC, **f** HKS, **g** the proposed solution, NG and **h** standard simplification (SS)

Table 2 Number of interest points, error measures and distortion percentage

Method	No. of interest points	Average maximum error (e^{-4})				Average mean error (e^{-4})				Average variance (e^{-4})				Percentage of distorted objects (%)			
		1/4	1/2	3/4	1	1/4	1/2	3/4	1	1/4	1/2	3/4	1	1/4	1/2	3/4	1
MS	203	112	62	39	25	11	6.8	3.9	2.4	16	9.7	5.8	3.5	80	23.8	7.1	0
SP	75	69	47	38	25	9.9	5.1	3.8	2	13	6.9	4.8	2.9	28.5	4.7	0	0
3DH	87	187	41	27	21	32	4.7	2.7	1.9	44	6.4	4	2.7	14.3	0	0	0
3DS	64	188	65	34	23	22	7.3	3.4	2	33	11	4.7	2.9	28.5	7.2	0	0
SDC	207	375	84	83	56	49	6.9	6.3	4	76	11	11	7.9	45.2	19	7.1	0
HKS	19	57	36	29	22	6.9	3.9	2.7	1.77	9	5.2	3.7	2.6	0	0	0	0
NG	74	80	40	29	23	10	4.5	3.0	1.9	20	6.2	3.9	2.8	7.1	0	0	0
SS	N/A	59	38	29	23	6.4	3.9	2.6	1.7	8.5	5.1	3.6	2.6	0	0	0	0

Fig. 8 Simplified mesh of 3500 faces for the ant model with **a** a large number of interest points (SDC), **b** an appropriate number of interest points (NG), and **c** a small number of interest points (HKS)

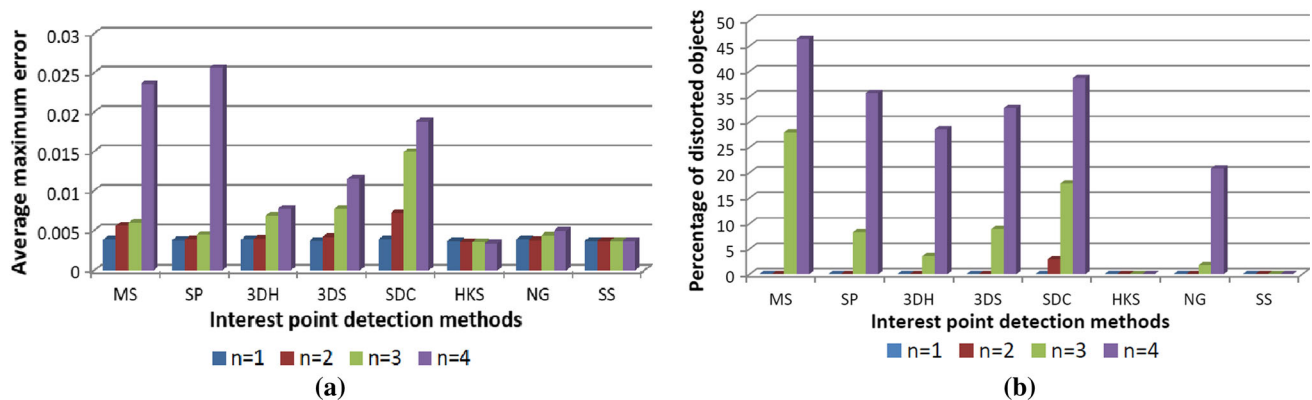
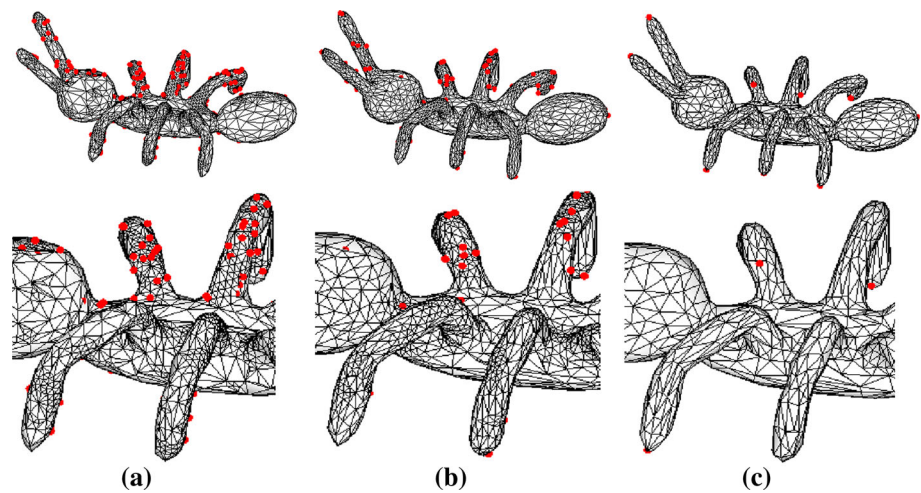


Fig. 9 Impact of the neighborhood size: **a** maximum error and **b** percentage of distorted objects for different interest point detection methods

results in a number of interest points similar the one of the SP method, with more interest points than HKS or 3DS method. A certain correlation is also visible between the number of interest points and the error measures in Table 2 in that a larger number of points seem to lead to higher errors. The fact that more points of interest do not lead to better results is illustrated as well in Fig. 8 that compares a method that obtains more points, namely SDC (Fig. 8a), with the proposed NG approach (Fig. 8b).

Too many points lead to the creation of clusters of dense triangles on the mesh, as those in Fig. 8a. When the number of interest points decreases significantly, as for example in the case of HKS method, the error is not following the drastic decrease in the number of points. From a visual perspective, as shown in Fig. 8c, too few interest points lead to a model with less well-defined characteristics, and closer to uniform simplification.

The error values also tend to increase with an increase in the size of the neighborhood n impacted by the preservation, due to the redistribution process. This is illustrated in Fig. 9a for the 1- to 4-nearest neighborhoods, for a num-

ber of faces equal to three-quarter of the initial number of faces in the mesh. Similar to the approach used in Table 2, the distorted objects are removed from the error calculations and their number is reported separately, in Fig. 9b. The error is reported as an average of maximum errors obtained over all the objects for a given interest point detector.

In general, with coarser meshes, it is preferred to affect smaller neighborhoods ($n = 1$ or 2), while for higher detailed meshes larger neighborhoods ($n = 3$ or 4) can be used. In experimentation with different meshes made of up to 35,000 faces, generally a 3-neighborhood gave good results for meshes larger than 5500 faces, a 2-neighborhood for meshes between 3500 and 5500 and 1-neighborhood for those below 3500. Finally, following the process described in Sect. 3.5 the multiple discrete LOD models can be constructed according to the specific needs of an application. An example is illustrated for the dog model in Fig. 1c, showing that even the discrete copy of lowest resolution maintains the interest regions that characterize the object.

5 Conclusions

This research article proposed the adaptation of QSlim to only simplify the regions that do not contain points in the n -nearest neighborhood of interest points detected using a neural gas solution. The proposed solution is relatively fast and obtains a good performance when compared to other interest point detectors. It is further incorporated in a discrete LOD scheme to be used in virtual reality applications, where a neural network predicts the appropriate number of faces to be used for each of the multiple copies at different resolution. In future work, the proposed solution for interest point detection will be incorporated in a hybrid discrete and view-dependent LOD solution that will allow to adjust the LOD not only according to the distance with respect to the user, but also taking into account the user viewpoint.

Acknowledgments The authors thank H. Monette-Thériault for his help with the implementation of the adapted QSlim algorithm.

Compliance with ethical standards

Funding The authors acknowledge the support to this research from the Natural Sciences and Engineering Research Council of Canada, under the Discovery Grant RGPIN-2014-04953.

Conflict of interest The authors declare that there is no conflict of interest related to this work.

Human and animal rights This article does not contain any studies with human participants performed by any of the authors.

References

- A Benchmark for 3D Interest Points Marked by Human Subjects [Online]. <http://www.itl.nist.gov/iad/vug/sharp/benchmark/3DInterestPoint/>
- Castellani U, Cristiani M, Fantoni S, Murino V (2008) Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Eurographics* 27(2):643–652
- Cignoni P, Rocchini C, Scopigno R (1998) Metro: measuring error on simplified surfaces. *Comput Graph Forum* 17(2):167–174. [AAOnline]. <http://vcg.isti.cnr.it/activities/surfacegrevis/simplification/>
- CloudCompare—3D Point Cloud and Mesh Processing Software, Open Source Project [Online]. <http://www.danielgm.net/cc/>
- Cretu A-M (2009) Experimental data acquisition and modeling of 3D deformable objects. Ph.D. Thesis, University of Ottawa
- Creusot C, Pears N, Austin J (2013) A machine-learning approach to keypoint detection and landmarking on 3D meshes. *Springer Int J Comput Vis* 102(1–3):146–179
- Dugataci H, Cheung CP, Godil A (2012) Evaluation of 3D interest point detection techniques via human-generated ground truth. *Vis Comput* 28(9):901–917
- Feixas M, Sbert M, Gonzalez F (2009) A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans Appl Percept* 6(1):1:2–1:23
- Gal R, Cohen-Or D (2006) Salient geometric features for partial shape matching and similarity. *ACM Trans Graph* 25(1):130–150
- Garland M, Heckbert PS (1997) Surface simplification using quadric error meshes. In: *Proceedings of ACM Siggraph 97*, pp 209–216
- Godil A, Wagan AI (2011) Salient local 3D features for 3D shape retrieval. In: Beraldin, JA, Cheok GS, McCarthy MB, Neuschaeferrube U, Baskurt AM, McDowall IE, Dolinsky M (eds) *Three-dimensional imaging, interaction, and measurement. Proceedings of the SPIE*, vol 7864. San Francisco, USA, pp 78640S-1–78640S-8
- 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
- Kho Y, Garland M (2003) User-guided simplification. In: *Proceedings of ACM symposium on interactive 3D graphics*, pp 123–126
- Lee CH, Varshney A, Jacobs DW (2005) Mesh saliency. *ACM SIGGRAPH* 174:659–666
- Leifman G, Shtrom E, Tal A (2012) Surface regions of interest for viewpoint selection. *IEEE conference on computer vision and pattern recognition*, pp 414–421
- Liu Y-S, Liu M, Kihara D, Ramani K (2007) Salient critical points for meshes. *ACM symposium on solid and physical modelling*, pp 277–282
- Luebke D (2001) A developer's survey of polygonal simplification algorithm. *IEEE Comput Graph Appl* 21(3):24–35
- Luebke D, Reddy M, Cohen JD, Varshney A, Watson B, Huebner R (2003) Level of detail for 3D graphics. In: *The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling*. Elsevier, San Francisco
- Martinetz M, Berkovich SG, Schulten KJ (1993) Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Trans Neural Netw* 4(4):558–568
- Monette-Thériault H, Cretu A-M, Payeur P (2014) 3D object modeling with neural gas based selective densification of surface meshes. *IEEE international conference on systems, man, and cybernetics*, pp 1373–1378, San Diego, US
- Novatnack J, Nishino K (2007) Scale-dependent 3D geometric features. *IEEE international conference on computer vision*, pp 1–8
- Payeur P, Curtis P, Cretu A-M (2013) Computational methods for selective acquisition of depth measurements in machine perception. *IEEE international conference on systems, man, and cybernetics*, pp 876–881, Manchester, UK
- Peure G (2007) Toolbox graph. [Online]. <http://www.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph>
- Pojar E, Schmalstieg D (2003) User-controlled creation of multiresolution meshes. In: *Proceedings of ACM symposium on interactive 3D graphics*, pp 127–130
- Sipiran I, Bustos B (2010) A robust 3D interest points detector based on Harris operator. *Eurographics workshop on 3D object retrieval*, pp 7–14
- Song R, Liu Y, Zhao Y, Martin RR, Rosin PL (2012) Conditional random field-based mesh saliency. *IEEE international conference on image processing*, pp 637–640
- Song R, Liu Y, Martin RR, Rosin PL (2014) Mesh saliency via spectral processing. *ACM Trans Graph* 33(1):6:1–6:17
- Sun J, Ovsjanikov M, Guibas L (2009) A concise and provably informative multi-scale signature based on heat diffusion. *Eurograph Symp Geom Process* 28(5):1383–1392
- Tombari F, Salti S, Di Stefano L (2013) Performance evaluation of 3D keypoint detectors. *Springer Int J Comput Vis* 102(1–3):198–220
- Wu J, Shen X, Zhu W, Liu L (2013) Mesh saliency with global rarity. *Graph Models* 75:255–264
- Yang Y-B, Lu T, Lin J-J (2009) Saliency regions for 3D mesh abstraction. In: Muneesawang P et al (eds) *Advances in multimedia information processing, LNCS 5879*, pp 292–299

- Yu T-H, Woodford OJ, Cipolla R (2013) A performance evaluation of volumetric 3D interest point detectors. *Int J Comput Vis* 102(1–3):180–197
- Zaharescu A, Boyer E, Varanasi K, Horaud R (2009) Surface feature detection and description with applications to mesh matching. *IEEE international conference on computer vision and pattern recognition*, pp 373–380
- Zhao Y, Liu Y, Song R, Zhang M (2012) Extended non-local means filter for surface saliency detection. *IEEE international conference on image processing*, pp 633–636
- Zhao Y, Liu Y, Song R, Zhang M (2013a) A Retinex theory based points sampling method. *Int J Comput Inf Syst Ind Manag Appl* 5:1–10
- Zhao Y, Liu Y, Zeng Z (2013b) Using region-based saliency for 3D interest points detection. In: Wilson R et al (eds) *CAIP 2013, Part II*, LNCS 8048, pp 108–116
- Zou G, Hua J, Dong M, Qin H (2008) Surface matching with salient keypoints in geodesic scale space. *Comput Animat Virtual Worlds* 19:399–410