

3D Object Modeling with Neural Gas Based Selective Densification of Surface Meshes

Hugues Monette-Thériault, Ana-Maria Cretu
Department of Computer Science and Engineering
Université du Québec en Outaouais
Gatineau, Canada
[monh06, ana-maria.cretu]@uqo.ca

Pierre Payeur
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
ppayeur@eecs.uottawa.ca

Abstract—The paper proposes an automated method for the modeling of objects using multiple discrete levels of detail for virtual reality applications. The method combines classical discrete level of detail approaches with a novel solution for the creation of selectively-densified object meshes. A neural gas network is used to capture regions of interest over a sparse point cloud representing a 3D object. Meshes at different resolutions that preserve these regions are then constructed by adapting a classical simplification algorithm to allow the simplification process to affect only the regions of lower interest. Different interest point detectors are incorporated in a similar manner and compared with the proposed 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.

Keywords—interest points; 3D object modeling; level of detail; selectively-densified surface mesh

I. INTRODUCTION

Current 3D data acquisition devices are capable to collect a large amount of data in a limited time. Similarly, computer graphics software packages offer enhanced capabilities to create extremely detailed and realistic 3D models. However, the increased size and complexity of datasets and models frequently represent impediments for further processing, manipulation and visualization at a reasonable computational cost. The problem is even more acute in interactive applications where the users expect to be able to execute operations on a model and see results in real time. From here stems the interest into finding appropriate procedures that would minimize the complexity and size of models by selecting only those points that are relevant for the characteristics of the studied object.

To respond to the real time quest, one approach that was proposed and is often employed in interactive computer graphics applications is to manage the level of detail (LOD) of an object. There are three large categories of methods for managing the level of detail of an object [1]: discrete, continuous and view-dependent. The traditional and most used approach is discrete LOD, where multiples copies of different resolutions of an object are created during an offline preprocessing step. The details are uniformly reduced. Distant objects use a coarser resolution, therefore a smaller number of polygons leading to a higher rendering speed. In spite of its simplicity, the method can take long to prepare the multiple copies. Continuous LOD methods create a data structure

encoding a continuous spectrum of details and the desired level is extracted from the structure at run-time. It is advantageous because it provides progressive rendering and interruptible loading properties. Finally 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. However this fidelity comes at the price of higher computation time and higher memory consumption. While these methods succeed to provide appealing results in many cases, they generally perform poorly at very low levels of detail because they simplify the object geometry uniformly without taking into account the fact that some localized areas or features could be visually more important than others. As a solution to this latter problem, some researchers proposed the use of human inputs on the desired quality of an object model where the user can select desired areas for local improvements [2-5]. Alternatively a few automatic solutions were proposed to guide the local mesh density of a model [6].

In this paper we propose the use of points of interest on the surface of an object as a mean to manage the level of detail. A previously proposed method by the authors for the detection of regions of interest in the context of adaptive scanning [7] is integrated here in a novel solution for the automated creation of selectively-densified object meshes. 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, is then adapted to only simplify the regions that contain neither points of interest nor points in the n -nearest neighborhood of points of interest. The proposed solution is compared with the case in which other interest point detectors are integrated in a similar fashion and with the classical simplification process using QSlim. Finally the proposed solution is incorporated in a discrete LOD scheme for virtual reality applications, where 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.

II. LITERATURE REVIEW

There has been a lot of interest in the research community for the detection of points of interest for applications such as matching of objects [8, 9], mesh and shape retrieval [10, 11],

adaptive scanning of 3D objects [7] or guiding mesh simplification [5, 6]. Starting from the observation that changes in curvature lead generally to regions that are different from their surroundings, and therefore of interest, Lee *et al.* [6] compute mesh saliency using the curvature map, a mapping from each vertex to its mean curvature. The approach is further improved by Liu *et al.* [12] who combine the mesh saliency with Morse theory. In [8], a given mesh is re-meshed at different levels of decimation and a Difference-of-Gaussian (DoG) operator is applied to identify the points of interest. An adaptive inhibition process chooses only the vertices characterized by values larger than 85% of their mean neighborhood values. From these, only vertices that are local maximum and that have a value higher than 30% of the global maximum are retained. In [10], vertex saliency is calculated based on the distance of a vertex with respect to its neighborhood and included in face saliency computation. Salient regions are then ranked based on their face saliency and the ones with higher values are retained. The results are extremely limited and no formal evaluation of the models is performed. Another improvement of [6], proposed by Song *et al.* [5], uses the geodesic measure instead of the Euclidean distance to compute the neighborhood of a point and further incorporates multi-scale information in a conditional random field framework to impose consistency constraints between neighboring points. Sipiran and Bustos [13] extend the Harris corner detector for a 3D object. Also exploiting the corner detection idea is the method proposed by Novatnack and Nishino [14]. The surface of a mesh model is parameterized on a 2D plane. A distortion map encodes the relative change in the edge length and a normal map is constructed by interpolating the normals at each vertex. 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 geometric corners. The corners detected at several scales are then projected back on the 3D surface. In Zaharescu *et al.* [9], the corners detected as extrema in across scales DoGs are processed with a Hessian operator in order to eliminate additional non-stable responses. The solution proposed by Godil and Wagan [11] 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. From these, only those located on the surface are retained. Sun *et al.* [15] 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. Dugataci *et al.* [16] compare some of the previously discussed methods for interest point detection with human-generated ground truth.

In terms of mesh simplification methods, a good survey of polygonal simplification methods along with their strengths and weaknesses is presented in [1, 17]. As the interest in this paper is to guide the mesh simplification process using the points of interest, a brief overview is presented on different solutions that allow for user guidance in the simplification process. It is worth mentioning that in most of the papers from the literature, the simplification algorithm of choice is QSlim, [17, 18]. The reason is that this algorithm is considered to

ensure one of the best balances between speed, fidelity and robustness among all the simplification algorithms [17]. A user-guided version of the QSlim algorithm is proposed in [2]. The user controls selectively the importance of different regions, 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. Lee *et al.* [6] guide the order of simplification contractions of the QSlim algorithm using a weight map that is constructed by thresholding the mesh saliency map. The weighting is therefore stronger in salient regions. The vertex pairs to be repeatedly contracted are ordered by increasing quadric error (collapsing cost) as in [18]. Based on the same approach, the authors of [5] choose to weigh the simplification process by amplifying the saliency values of the points situated in salient regions. The approach of [4] enables the user to control the simplification of a mesh by painting the desired regions in a Maya plug-in. Their algorithm weighs the collapsing cost, but also takes into account appearance attributes. Ho *et al.* [3] allow as well the user to 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 input for the local improvement of a mesh.

III. PROPOSED METHOD FOR DETECTING REGIONS OF INTEREST AND THEIR INTEGRATION IN A SELECTIVELY DENSIFIED 3D OBJECT MODEL

In this paper we propose a novel automated method to create selectively-densified 3D object models. It employs a neural gas network to determine automatically the regions of interest. These regions are then incorporated in an adapted version of the QSlim simplification algorithm to create selectively densified meshes to encode the shape of 3D objects. Using a discrete LOD method inspiration, multiples copies of an object are created during an offline preprocessing 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.

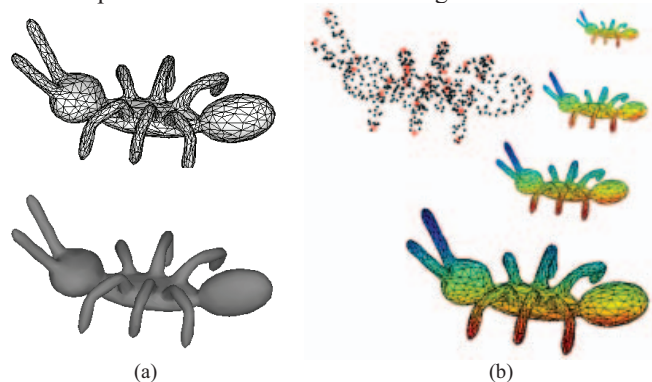


Fig. 1. (a) Selectively-densified mesh with and without material properties, (b) discrete levels of details with regions of interest preservation for an ant.

Distant objects use a coarser resolution, therefore a smaller number of polygons, but ensure that the characteristics of an object are kept. An example of the approach is illustrated in

Fig. 1 for an ant model. Fig. 1a shows the selectively-densified model at its highest resolution as a mesh and with a dull grey material. The sparse point cloud with the points of interest marked by red dots is shown on the left side of Fig. 1b, while the right side of the same figure shows four copies at different resolution for the ant model.

A. Region of Interest Detection

In our previous work [7] we have shown that self-organizing architectures can be successfully incorporated in an automated selective scanning scheme to identify regions of interest for further refinement. In particular, neural gas was studied for this purpose. In a few words, starting from an initial sparse 3D point cloud, a neural gas network is trained to adapt 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 regions of interest are detected by finding higher density areas of nodes in the adaptation result. This is accomplished by first applying a Delaunay triangulation to the neural gas output, knowing that areas of high density of nodes are represented by smaller triangles in the resulting mesh. It is worth mentioning that the meshes used in this previous work are too sparse to offer modeling capabilities. The mean value of the length of vertices between every pair of nodes for every triangle is set as a threshold and all the edges of triangles that are larger than this threshold are removed from the tessellation. The removal of the edges longer than the threshold ensures the identification of close points and, therefore, dense areas of features. The interested reader is invited to consult [7] for the specific details of the implementation, such as the theoretical framework and the selected parameters. In the context of the present work, the same approach is used to identify regions (and implicitly points) of interest over the surface of a 3D object. Fig. 2 shows an example for the detection of points of interest over a point cloud representing a horse. Fig. 2a illustrates with green points the neural gas map for a size of 1500 nodes in the map, while Fig. 2b shows the points of interest, in red, using the procedure described above.

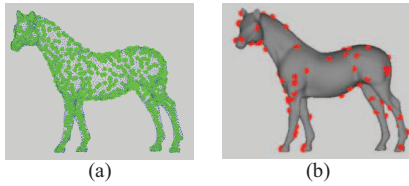


Fig. 2. (a) Neural gas map after adaptation and (b) points of interest detected using the proposed approach overlapped over the horse model.

B. Incorporation of Regions of Interest in the Selectively Densified 3D Model

As stated in the introduction, QSlim [18] is the algorithm of choice to incorporate the detected areas of interest. 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 3 vertices that each create a face in the mesh, in counter-clockwise order, the QSlim algorithm, in particular its software package, is adapted and incorporated into an

automated solution to create a selectively simplified mesh M_s . QSlim simplifies a mesh by repeated edge contractions (collapses) considering the quadric error metric of each vertex of the mesh. This metric is calculated as a 4×4 matrix that represents the sum of squared distances from the vertex to the planes of neighboring triangles. If the value of the metric is large, it means the corresponding vertex could be a distinctive feature on the mesh and therefore will be removed later from the mesh. Otherwise, it will be removed earlier. This metric is used to calculate the cost of a contraction and the optimal position for the unified vertex. After extracting all the edges from a mesh, each is associated to a cost and all the edges are stored in an ordered list of costs. The idea is to remove at each step the edge with the least 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_s , to be retained after the simplification. The aim of most solutions available in the literature is to weigh stronger the regions of interest after the creation of a simplified mesh [3, 5, 6] and adjust their cost, in order to delay their simplification. Instead, in this paper we do not allow the simplification process 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. The choice of the neighborhood size, n , is further discussed in the experimental results section. Because the targeted number of faces to preserve 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 pseudo-algorithm for our approach is the following:

Step 1:

```
// interest points detection for an object O represented by a mesh M
Obtain sparse 3D point cloud representing an object in a mesh M by
uniform decimation:
O =  $\{P_1, \dots, P_i, \dots, P_{NS}\}$ ,  $P_i = [X_i, Y_i, Z_i] \in \mathbf{M}$ ,  $i=1 \dots NS$ ;
Initialize neural gas network;
Apply neural gas adaptation on O to obtain the 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 are not points of interest or neighbors
of points of interest:
```

$$\mathbf{E}_D = \{e(u,v) \mid e(u,v) \in \mathbf{M}, e(u,v) \notin \mathbf{PN}\}$$

```
Initialize Ms=M | 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

P, the set of interest points, can be as well obtained using any interest point detector, for example any of the detectors that

are used for comparison in section IV. Because the proposed region-of-interest detector in section III.A runs on Matlab, the Matlab wrapper of QSlim is adapted from [19] and the same platform is used to implement the proposed solution.

C. Comparison of Simplified Meshes

The Metro software [20] is used for comparison between the different selectively simplified meshes. Metro is a tool for measuring the output quality of simplified meshes using a point-to-surface distance metric. It takes as input a pair of surfaces (meshes in our case) and outputs the maximum distance, as well as the mean and root mean square distance 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.

D. Multi-resolution Object Modeling

In order to obtain multiple copies of an object at different resolutions, the simplification algorithm with region of interest preservation is applied for an increasing number of faces, varying from 1500 up to half the number of faces in the initial mesh of an object. Because it is difficult for a user to judge the number of faces required for a certain object, in the current approach a neural network is trained over the database of objects in order to learn the mapping between the number of faces to be used for each of the multiple copies at different resolution and the initial mesh. It takes as input a full size mesh (at its maximum resolution) and returns, in this case, a series of 4 increasing number of faces. It is a two-layer feed-forward architecture with an empirically determined size of 40 neurons in the hidden layer. In order to train this network, the measures of error are computed as detailed in section III.C within a certain range of resolutions, and the one that obtains the lowest error within the range is chosen for training. The range of resolutions are calculated by dividing by 4 (or by the desired number of copies at different resolutions to be used) the interval between 1500 faces and half the number of faces in the entire mesh. At run-time, the network is provided with the mesh of an unknown object and it outputs the number of faces to retain for each of the copies, from coarse to detailed. The simplification algorithm with regions of interest preservation is applied in turn for each of the copies by constraining the simplification to the number of faces specified.

IV. EXPERIMENTAL RESULTS

In order to evaluate the proposed framework, several objects are tested from the dataset associated with the benchmark for 3D interest point detection algorithms [16, 21]. This allows a direct comparison with the neural gas based detection of regions of interest. The interest point detectors identified in the benchmark, namely mesh saliency (MS) [6], salient points (SP) [8], 3D-SIFT (3DS) [11], 3D Harris (3DH) [13], scale-dependent corners (SDC) [14] and Heat Kernel Signature (HKS) [15] as well as the proposed neural gas-based interest points detector (NG) are incorporated in an adapted version of the QSlim simplification algorithm as described in section III.B. Fig. 3 shows the results obtained with the neural gas approach for the girl model.

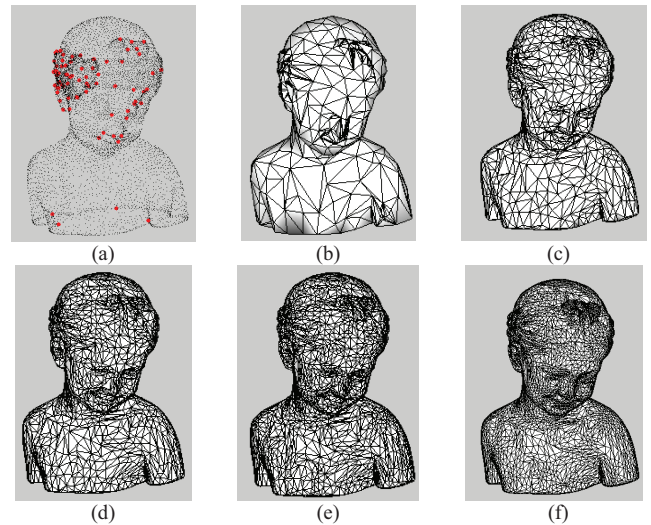


Fig. 3. Proposed method with regions of interest preservation for girl model: (a) point cloud with interest points, and selectively-densified models with (b) 1500, (c) 3500, (d) 5500, (e) 7500 and (f) 15000 faces in the simplified mesh.

Fig. 3a illustrates the point cloud and the points of interest, in red, while Fig. 3b to 3f show the results for an increasing number of faces starting from coarse (5% of the size of the initial mesh) to detailed (half the size of the initial mesh). One can notice that the simplified mesh is denser in regions with fine details. This is also visible in the series of results shown in Fig. 4, obtained using the proposed approach based on neural gas for different objects extracted from the database.

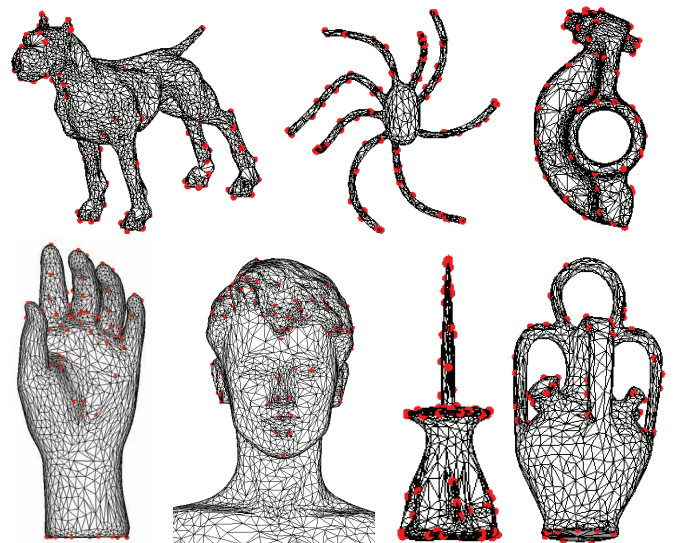


Fig. 4. Examples of other selectively-densified object models obtained with the proposed approach for a number of faces in the simplified mesh equal to half of the number of faces in the initial mesh and for a neighborhood $n=3$.

Fig. 5 compares the proposed approach with regions of interest preservation (Fig. 5d and enlarged in 5e) with the case where the QSlim is not constrained in the regions of interest (Fig. 5b and 5c) for the same number of faces, that is 15000 faces in the simplified mesh.

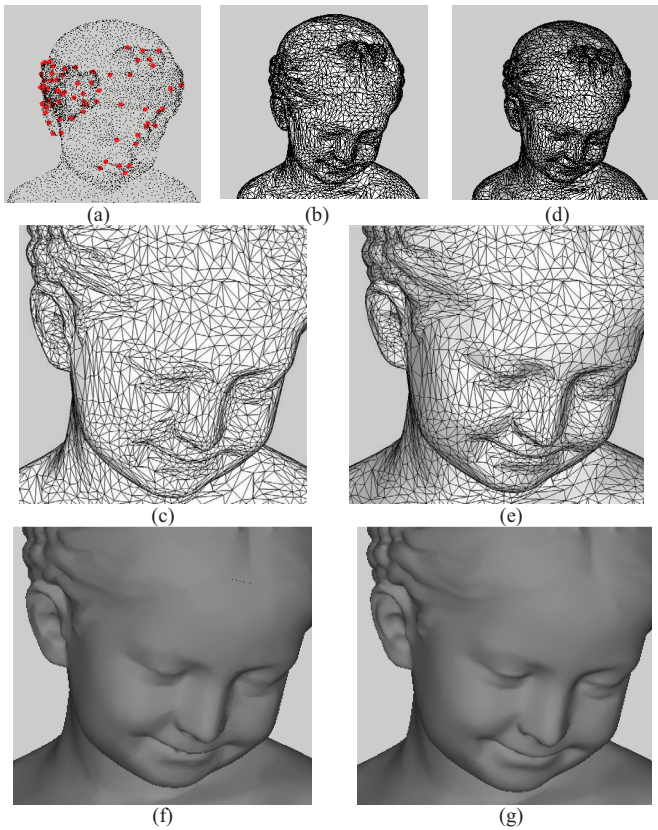


Fig. 5. Comparison of QSlim with the proposed approach: (a) identified interest points, (b), (c), (f) QSlim results without interest regions preservation for 15000 faces in the simplified mesh, and (d), (e), (g) proposed approach with interest regions preservation for 15000 faces.

The proposed method achieves a better representation of the interest regions that characterise the object, in this case the facial features. The eyes, lips, and the ears are better defined in Fig. 5d and 5e than in Fig. 5b and 5c. Because in general, objects in virtual environments are presented as colored and textured meshes, in Fig. 5f and 5g a comparison is illustrated between QSlim simplified mesh and the selectively-densified mesh using a dull grey material and Gouraud lighting. The same observations stand in this case and also when a shiny material is used, as shown in Fig. 6.

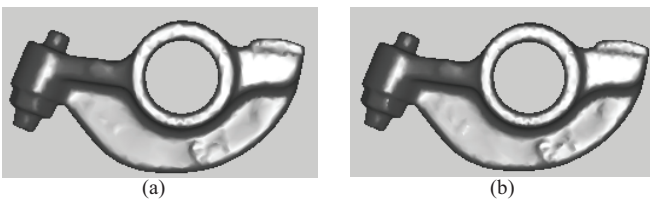


Fig. 6. Comparison of (a) QSlim with (b) proposed approach for the same number of faces (5500) when a shiny material is used.

The neural gas point based identification takes roughly 10s for each 10000 vertices of an initial model on Matlab platform, running on a Windows machine with an Intel Pentium CPU at 4.5GHz and 2GB of RAM, while the adapted QSlim adds roughly 0.14s per each 10000 faces of the initial model. To enable a quantitative comparison of performance, the simplified meshes obtained using the different interest

point detectors are also evaluated using the approach described in Section III.C. Fig. 7 illustrates the error difference between the different methods, calculated as an average over all the objects in the database. The error measures are computed in each case with respect to the initial (full size) mesh. It is worth mentioning that a slightly higher error is in general expected with the preservation of regions of interest, because the triangles of the mesh are redistributed in order to better cover the detailed regions and therefore resulting in a less detailed coverage (larger triangles) of regions that are less important. Because 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 simple simplification. Fig. 7a shows the error measures computed for each interest point detector (MS, SP, 3DH, 3DS, SDC, HKS, NG) as an average over different simplification levels: namely 1500, 3500, 5500, 7500 and 15000 faces and also using simple QSlim simplification with no region of interest preservation (SS).

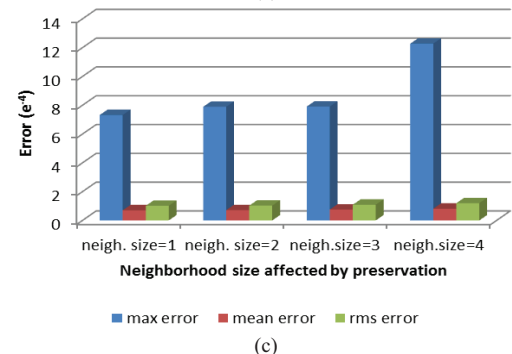
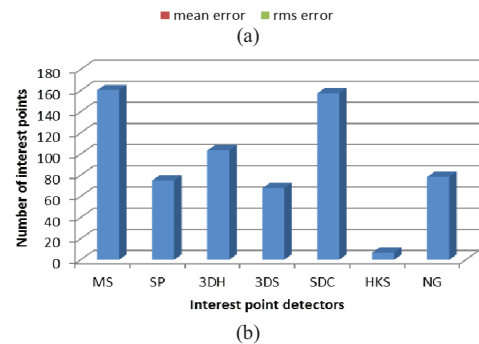
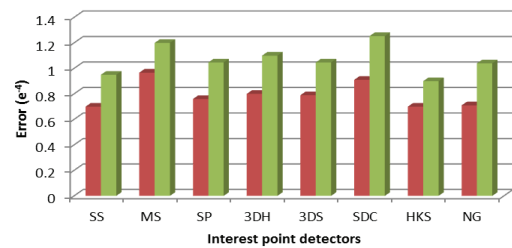


Fig. 7. (a) Error between simplified meshes with interest regions preservation as an average over different simplification levels, (b) number of interest points detected by different methods, and (c) effects of the neighbourhood size.

It can be noticed however that the increase in error is barely evident for methods such as HKS and the proposed solution (NG) with respect to the simple QSlim simplification (SS).

At the same time, 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 when comparing Fig. 7a and 7b. The latter shows the number of interest points detected by the different methods. A certain correlation is visible between this number and the error measures in Fig. 7a in that a larger number leads to higher errors. However this is not necessary true when the number of points decreases significantly, as in the case of HKS. For this method, the error is not following the drastic decrease in the number of points. The proposed method, NG, results in a number of interest points in the range of the SP method, with more interest points than HKS or 3DS. 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. On the other hand fewer points of interest, as those of HKS (Fig. 8c), lead to a model closer to uniform simplification, with less well defined characteristics.

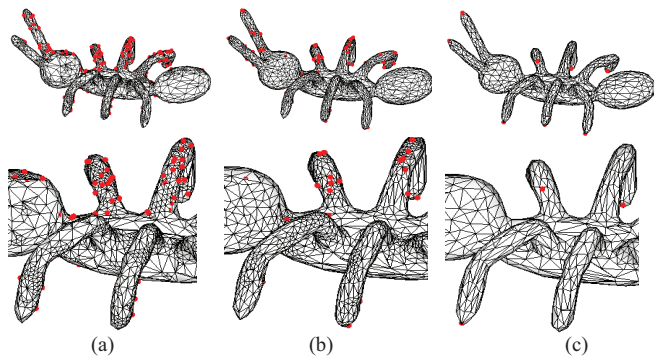


Fig. 8. Simplified mesh of 3500 faces for (a) a large number of interest points (SDC), (b) an intermediate number of interest points (NG), and (c) a small number of interest points (HKS).

The error also tends to increase with an increase in the size of the neighborhood n impacted by the preservation. This is illustrated for the 1- to 4-nearest neighborhoods, for a constant number of 15000 faces in Fig. 7c. This size is chosen because it ensures that for all objects in the database the shape is preserved at $n=4$. For example, for certain methods that return many points of interest, the number of faces at $n=4$ is not sufficient to preserve the general shape of the object. The error is reported as an average over all detectors. 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 35000 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.

CONCLUSION

The paper 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 approach 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.

REFERENCES

- [1] D. Luebke, M. Reddy, J.D.Cohen, A. Varsheny, B. Watson, and R. Huebner, "Level of Detail for 3D Graphics", *Morgan Kaufmann*, 2003.
- [2] Y. Kho and M. Garland, "User-guided Simplification", *Proc. ACM Symp. Interactive 3D Graphics*, pp. 123-126, 2003.
- [3] T.-C. Ho, Y.-C. Lin, J.-H. Chuang, C.-H. Peng, and Y.-J. Cheng, "User-Assisted Mesh Simplification", *Proc. ACM Int. Conf. Virtual Reality Continuum and its Applications*, pp. 59-66, 2006.
- [4] E. Pojar, and D. Schmalstieg, "User-Controlled Creation of Multiresolution Meshes", *Proc. ACM Symp. Interactive 3D Graphics*, pp. 127-130, 2003.
- [5] R. Song, Y. Liu, Y. Zhao, R.R. Martin and P.L. Rosin, "Conditional Random Field-Based Mesh Saliency", *IEEE Int. Conf. Image Processing*, pp. 637-640, 2012.
- [6] C. H. Lee, A. Varshney, and D.W. Jacobs, "Mesh Saliency," *ACM SIGGRAPH*, vol. 174, pp. 659-666, 2005.
- [7] P. Payeur, P. Curtis, and A.-M. Cretu, "Computational Methods for Selective Acquisition of Depth Measurements in Machine Perception", *IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 876-881, Manchester, UK, 2013.
- [8] U. Castellani, M. Cristiani, S. Fantoni and V. Murino, "Sparse Points Matching by Combining 3D Mesh Saliency with Statistical Descriptors", *Eurographics*, vol. 27, no. 2. pp. 643-652, 2008.
- [9] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface Feature Detection and Description with Applications to Mesh Matching", *IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 373-380, 2009.
- [10] Y.-B. Yang, T. Lu, and J.-J. Lin, "Saliency Regions for 3D Mesh Abstraction", P. Muneesawang *et al.* (Eds.), *Advances in Multimedia Information Processing*, LNCS 5879, pp. 292-299, 2009.
- [11] A. Godil, and A.I. Wagan, "Salient Local 3D Features for 3D Shape Retrieval", *SPIE 3D Image Processing and Applications*, 2011.
- [12] Y.-S. Liu, M. Liu, D. Kihara and K. Ramani, "Salient Critical Points for Meshes", *ACM Symp. Solid and Physical Modelling*, pp. 277-282, 2007.
- [13] I. Sipiran, and B. Bustos, "A Robust 3D Interest Points Detector Based on Harris Operator", *Eurographics Workshop on 3D Object Retrieval*, pp. 7-14, 2010.
- [14] J. Novatnack, and K. Nishino, "Scale-Dependent 3D Geometric Features", *IEEE Int. Conf. Computer Vision*, pp. 1-8, 2007.
- [15] J. Sun, M. Ovsjanikov, and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion", *Eurographics Symp. Geometry Processing*, vol. 28, no. 5, pp. 1383-1392, 2009.
- [16] H. Dugataci, C.P. Cheung, and A. Godil, "Evaluation of 3D Interest Point Detection Techniques via Human-Generated Ground Truth", *The Visual Computer*, vol. 28, no. 9, pp. 901-917, 2012.
- [17] D. Luebke, "A Developer's Survey of Polygonal Simplification Algorithm", *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 24-35, 2001.
- [18] M. Garland, and P.S. Heckbert, "Surface Simplification Using Quadric Error Meshes", *Proc. ACM Siggraph 97*, pp. 209-216, 1997.
- [19] G. Peyre, *Toolbox Graph*, 2007. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph>.
- [20] P. Cignoni, C. Rocchini and R. Scopigno, "Metro: Measuring Error on Simplified Surfaces", *Computer Graphics Forum*, vol. 17, no. 2, 1998, pp. 167-174, [Online]. Available: <http://vcg.isti.cnr.it/activities/surfacegrevis/simplification>.
- [21] A Benchmark for 3D Interest Points Marked by Human Subjects, [Online]. Available: <http://www.itl.nist.gov/iad/vug/sharp/benchmark/3DInterestPoint>.