MESH SALIENCY DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

Stavros Nousias[1,2], *Gerasimos Arvanitis*[1], *Aris S. Lalos*[2] *and Konstantinos Moustakas*[1]

[1] Department of Electrical & Computer Engineering, University of Patras, Greece
[2] Industrial Systems Institute, Athena Research Center, Greece
{snousias,arvanitis,moustakas}@upatras.gr,{nousias,lalos}@isi.gr

ABSTRACT

Mesh saliency has been widely considered as the measure of visual importance of certain parts of 3D geometries, distinguishable from their surroundings, with respect to human visual perception. This work is based on the use of convolutional neural networks to extract saliency maps for large and dense 3D scanned models. The network is trained with saliency maps extracted by fusing local and global spectral characteristics. Extensive evaluation studies carried out using various 3D models, include visual perception evaluation in simplification and compression use cases. As a result, they verify the superiority of our approach as compared to other state-of-the-art approaches. Furthermore, these studies indicate that CNN-based saliency extraction method is much faster in large and dense geometries, allowing the application of saliency aware compression and simplification schemes in low-latency and energy-efficient systems.

Index Terms— Saliency mapping extraction, Convolutional Neural Network, saliency features of 3D objects.

1. INTRODUCTION

Visual saliency, investigated by numerous researchers, has been considered as the metric that evaluates the significance of a region or point in space, with respect to human visual perception. Saliency is mainly a stimulus-driven process [1] where the salient part of the scene differs from its neighbouring region [2] due to lack of correlation [3]. Psychophysical studies have shown that low-level salient features are processed in the early visual cortex and extract the most important and basic information of the visual scene [4]. Saliency mapping has been employed in various applications in the area of image [5] and video [6] processing, proving a number of benefits in terms of computational and storage efficiency. Similarly, in the area of geometry processing, 3D mesh saliency, correspond to a measure of regional importance of 3D meshes, was coined by Lee *et al.* [7]. Several other approaches followed utilizing spectral methods [8, 9], curvature-based methods [7, 10] multiscale descriptors [11], entropy-based methods [12] and hybrid methods [13] taking into account both geometry and color.

Convolutional neural networks(CNNs) have been also extensively used for the extraction of saliency maps from images[14]. Recent works have used also CNNs to extract saliency maps for 3D representations generated by a multiview setup [15], though the literature on CNN based saliency extraction for other 3D data representations including meshes and point clouds is limited. Motivated by the aforementioned challenges and open issues this work employs CNNs to automatically extract saliency maps from 3D geometries facilitating the processing of very large and dense meshes with lower complexity while enabling parallelization. To the best of our knowledge, this is the first CNN-based approach to extract 3D mesh saliency maps, applied directly on the geometric data, thus contributing to the field of geometric deep learning where the sampling of the latent space is non-uniform. More specifically, the contributions of the proposed approach can be summarized in the following points: i) We employ a method based on fusion of robust principal component analysis and eigenvalues to extract saliency maps of 3D scanned meshes and formulate the training set for the effective training process of a convolutional neural network. ii) We formulate a geometric 3D patch descriptor to facilitate the learning process, significantly reducing the required training dataset. iii) We provide an experimental evaluation of the proposed methods in terms of execution times and visual inspection. iv) We provide use case examples, namely compression and simplification, evaluated with relevant metrics. The rest of this paper is organized as follows: Section 2 focuses on preliminaries and basic assumptions. Section 3 describes in detail the workflow of the proposed approach. Section 4 is dedicated to the experimental evaluation while conclusions are drawn in Section 5.

2. PRELIMINARIES AND BASIC ASSUMPTIONS

RPCA has been employed [16, 17], for the decomposition of a matrix $\mathbf{E}$ into a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ by

Research supported by the EU research and innovation program Horizon 2020 Marie Sklodowska WARMEST :loW Altitude Remote sensing for the Monitoring of the state of cultural hEritage Sites: building an inTegrated model for maintenance, under Grant Agreement No. 777981

Research supported by the EU research and innovation program Horizon 2020 CPSoSaware under Grant Agreement No. 871738

solving: $\arg\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1$, s.t. $\mathbf{L}+\mathbf{S}=\mathbf{E}$, where $\|\mathbf{L}\|_*$ is the nuclear norm of a matrix $\mathbf{L}$ (i.e, $\sum_i \sigma_i(\mathbf{L})$ is the sum of the singular values of $\mathbf{L}$). We cope with this convex optimization problem using a very fast approach, as described in [18], according to:

$$\arg\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}\|\mathbf{L}+\mathbf{S}-\mathbf{E}\|_F + \lambda\|\mathbf{S}\|_1 \quad \text{s.t. rank}(\mathbf{L}) = K \quad (1)$$

which can be solved using alternative minimization:

$$\mathbf{L}^{(t+1)} = \arg\min_{\mathbf{L}} \|\mathbf{L}+\mathbf{S}^{(t)}-\mathbf{E}\|_F \quad \text{s.t. rank}(\mathbf{L}) = K \quad (2)$$

$$\mathbf{S}^{(t+1)} = \arg\min_{\mathbf{S}} \|\mathbf{L}^{(t+1)}+\mathbf{S}-\mathbf{E}\|_F + \lambda\|\mathbf{S}\|_1 \quad (3)$$

In each (t) iteration, the Eq. (2) is updated with rank $= K$. We define the condition

$$\frac{u_K}{\sum_{i=1}^{K} u_i} > \epsilon \quad (4)$$

where u denotes the singular values and ϵ is a small threshold. If the condition is satisfied then the rank is increased by one (i.e., $K = K+1$) and the Eq. (3) is also updated. To update the Eq. (2), a partial SVD$(\mathbf{E}-\mathbf{S}^{(t)})$ is estimated keeping K components. To update the Eq. (3), a shrinkage operator is used $\mathcal{D}(.)$, where $\mathcal{D}(\mathbf{E}-\mathbf{L}^{(t+1)}, \lambda) = \text{sign}(\mathbf{E}-\mathbf{L}^{(t+1)})\max\{0, |\mathbf{E}-\mathbf{L}^{(t+1)}| - \lambda\}$

Triangular meshes $\mathcal{M}$ are traditionally described by n vertices $\mathbf{v}$ and n_f faces f. Each vertex $\mathbf{v}_i$ is denoted by $\mathbf{v}_i = [x_i,\ y_i,\ z_i]^T$, $\forall\, i = 1, \cdots, n$. Each f_j face is a triangle that can be described by its centroid $\mathbf{c}_j = {(\mathbf{v}_{j1}+\mathbf{v}_{j2}+\mathbf{v}_{j3})}/{3}$ and its outward unit normal:

$$\mathbf{n}_{c_i} = \frac{(\mathbf{v}_{j2}-\mathbf{v}_{j1}) \times (\mathbf{v}_{j3}-\mathbf{v}_{i1})}{\|(\mathbf{v}_{j2}-\mathbf{v}_{j1}) \times (\mathbf{v}_{j3}-\mathbf{v}_{j1})\|} \quad (5)$$

where $\mathbf{v}_{j1}$, $\mathbf{v}_{j2}$ and $\mathbf{v}_{j3}$ are the position of the vertices that define face $f_j = \{\mathbf{v}_{j1}\ \mathbf{v}_{j2}\ \mathbf{v}_{j3}\}$, $\forall\, j = 1, \cdots, n_f$. The first-ring area of a vertex $\mathbf{v}_i$ is defined as the neighborhood $\mathcal{N}_i$ in which the vertex $\mathbf{v}_i$ is connected to other vertices by only one edge (i.e., with topological degree equal to 1).

Space filling curves (SFCs) [19] are maps from a 1D interval to a region in n-dimensional space. Intuitively, they can be considered as curves that traverse all points of the n-dimensional space, thus inducing an order on those points. SFCs represent a bijective function so that $f : [0,1] \rightarrow [0,1]^2$. Common subsets of $\mathcal{R}^n$ can be denoted as $\mathcal{I} = [0,1]$, $\mathcal{Q} = [0,1]^2$, $\mathcal{W} = [0,1]^3$. A widely used SFC curve is the classical Hilbert space-filling curve, visualized in Figure 1. Hilbert curve is defined by the partitioning of interval $\mathcal{I}$ into 4^n subintervals of length 4^{-n} and the square $\mathcal{Q}$ into 4^n sub-squares of side 2^{-n} and can be generated by recursion. A bidirectional mapping is generated between the sub-intervals of $\mathcal{I}$ and the sub-squares of $\mathcal{Q}$. Even though the original Hilbert curve handles only regions of size $2^n \times 2^n$ this is not a problem in our case. Hilbert curve exhibits the highest level of coherency described by the metric defined as the distance along the curve with respect to multi-dimensional Euclidean metric [20]. Voothries [21] concludes that the Hilbert curve is superior to other curves in this respect. We will highlight the importance of this observation in the following sections.

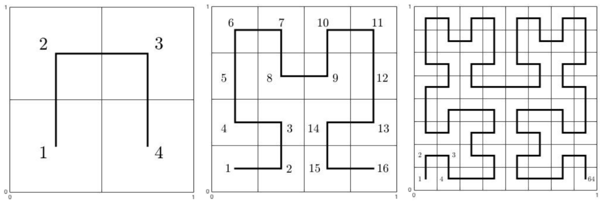


Fig. 1: Hilbert curve.

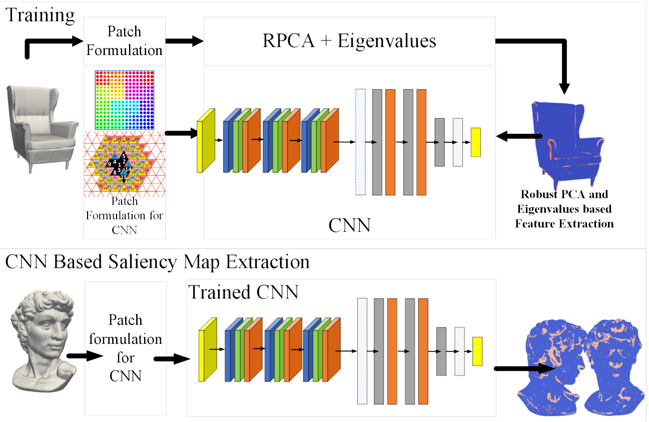

Training
Patch Formulation
RPCA + Eigenvalues
Patch Formulation for CNN
CNN
Robust PCA and Eigenvalues based Feature Extraction
CNN Based Saliency Map Extraction
Trained CNN
Patch formulation for CNN

Fig. 2: CNN based saliency map extraction pipeline.

3. PROPOSED CNN BASED SALIENCY EXTRACTION FOR 3D MESHES

This section presents the pipeline of the proposed approach for the extraction of the 3D saliency mapping. Training data were generated from 3D geometric data and a ground truth saliency mapping, extracted via traditional methods [22]. After the training process, the generated output can be used for the automatic extraction of saliency mapping for any other new 3D model. Figure 2 briefly presents the pipeline of our approach. We start by separating the whole mesh into n_f (i.e., equal to the number of centroids) overlapped and equal-sized patches. Then, we follow two different steps for the estimation of the spectral and geometrical saliency. The final result is a combination of these two values. Once the saliency mapping has been estimated, we use it as ground truth for the training of the CNN.

3.1. Saliency Analysis

For each face f_i of the mesh, we estimate the corresponding patch, represented as a matrix $\mathbf{N}_i \in \mathbb{R}^{(k+1)\times 3}$, consisting of the $k+1$ neighboring centroid normals: $\mathbf{N}_i = [\mathbf{n}_{ci}, \ \mathbf{n}_{ci_1}, \ \mathbf{n}_{ci_2}, \ \cdots, \ \mathbf{n}_{ci_k}]^T \ \forall \ i = 1, \cdots, n_f$ Each matrix $\mathbf{N}_i$ is used for the estimation of the covariance matrices $\mathbf{R}_i = \mathbf{N}_i^T \mathbf{N}_i \in \mathbb{R}^{3\times3}$. Then, $\mathbf{R}_i = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ is decomposed to a matrix $\mathbf{U}$, consisting of the eigenvectors, and a diagonal matrix $\mathbf{\Lambda} = \text{diag}(\lambda_{i1}, \lambda_{i2}, \lambda_{i3})$, consisting of the corresponding eigenvalues $\lambda_{ij}, \ \forall \ j = 1, 2, 3$. Finally, the spectral saliency s_{1i} of a centroid $\mathbf{c}_i$ is denoted as the value given by the inverse l^2-norm of the corresponding eigenvalues:

$$s_{1i} = \frac{1}{\sqrt{\lambda_{i1}^2 + \lambda_{i2}^2 + \lambda_{i3}^2}} \ \forall \ i = 1, \cdots, n_f \tag{6}$$

Additionally, we normalize the values in order to be in the range of [0-1], according to:

$$\bar{s}_{1i} = \frac{s_{1i} - \min(s_{1i})}{\max(s_{1i}) - \min(s_{1i})} \ \forall \ i = 1, \cdots, n_f \tag{7}$$

Observing the Eq. (6), we can see that large values of the term $\sqrt{\lambda_{i1}^2 + \lambda_{i2}^2 + \lambda_{i3}^2}$ correspond to small saliency features indicating that the centroid lies in a flat area, while small values correspond to large saliency values, characterizing the specific centroid as a geometric feature. This can be easily justified by the fact that the centroid normal of a face lying in a flat area is represented by one dominant eigenvector, the corresponding eigenvalue of which has a very large value. On the other hand, the centroid normal of a face lying in a corner is represented by three eigenvectors, that correspond to eigenvalues with small but almost equal amplitude.

The geometrical saliency features are estimated by exploiting the sparsity of the centroid normals. For the construction of three matrices $\mathbf{E}_l \in \mathbb{R}^{n_f \times (k+1)} \ \forall \ l \in \{x, y, z\}$ we use the $\mathbf{N}_i$ matrices according to: $\mathbf{E}_l = [\mathbf{N}_{1l}; \mathbf{N}_{2l}; \cdots; \mathbf{N}_{n_f l};]$ containing the centroid normals values of the corresponding coordinate.

Then, we apply the RPCA approach to each one of these matrices, as described in 2, taking advantage of the geometrical coherence between neighbouring guided normals. By the decomposition, the low-rank $\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z$ and sparse $\mathbf{S}_x, \mathbf{S}_y, \mathbf{S}_z$ matrices are estimated. However, the estimation of the geometric saliency feature s_{2i} of the centroid $\mathbf{c}_i$ requires only the values of the first column of the sparse matrices, according to:

$$s_{2i} = \sqrt{S_{i1_x}^2 + S_{i1_y}^2 + S_{i1_z}^2} \ \forall \ i = 1, \cdots, n_f \tag{8}$$

where S_{i1_x} denotes the scalar value of the i^{th} row, of the 1^{st} column, of the $\mathbf{S}_x$ matrix. Finally, in a similar way with the spectral analysis, we estimate the normalized saliency $\bar{s}_{2i}$ (Eq. (7)). The saliency mapping is a combination of the normalized spectral $\bar{s}_1$ and geometrical $\bar{s}_2$ saliency features, according to: $s_{ci} = \frac{\bar{s}_{1i} + \bar{s}_{2i}}{2} \ \forall \ i = 1, \cdots, n_f$. The proposed method is robust, even when we assume complex surfaces with different geometrical characteristics, since it exploits spectral characteristic (i.e., over-sensitivity in the variation of neighboring centroid normals) and geometrical characteristics (i.e., sparsity property of intense features). For the estimation of the saliency value of each vertex we use the following equation,

$$s_i = \frac{\sum_{\forall \mathbf{c}_j \in \mathcal{N}_i} s_{cj}}{|\mathcal{N}_i|} \ \forall \ i = 1, \cdots, n \tag{9}$$

Finally, s is quantized into four classes to facilitate CNN based saliency prediction. For face f_i with saliency value s_i the corresponding class c_i is computed as follows, $c_i = \lfloor s_i/\Delta \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function and $\Delta = 1/4$. For the rest of the paper we will refer to this fusion of geometrical and spectral saliency components as **fusion-based approach**.

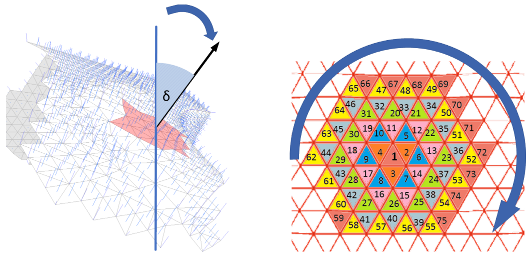


Fig. 3: Example of a patch neighbourhood (n = 69). The normals of the faces are also apparent.

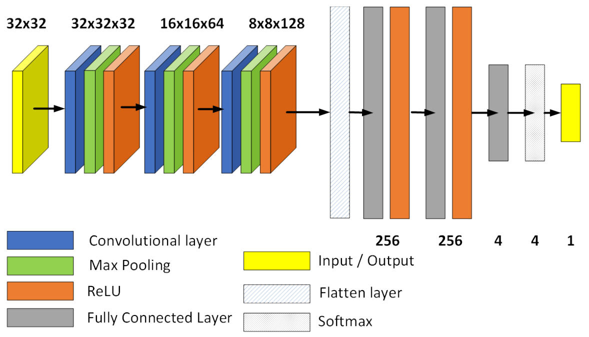


Fig. 4: CNN architecture.

3.2. CNN architecture, training and saliency map extraction

The CNN architecture consists of three convolutional layers, each one followed by a max-pooling layer and a ReLU activation function. The dimension of the convolutional kernels is 3×3 performing a padded convolution operation. A flatten layer succeeds the convolutional layers followed by three fully connected layers and a softmax activation function. An overview of the CNN architecture is visualized in Figure 4.

The patch descriptor facilitates the training process by constraining the latent space. Each patch $\mathcal{P}_i$ consists of the face normals of N topologically closest neighbours of face f_i arranged by the distance of the face centroid c_i to face centroid c_j, $f_j \in \mathcal{P}_i$ and organized in rings as visualized in Figure 3. Working on the normal space makes the descriptor scale and translation invariant. To deal with rotation invariance we assume a local coordinate system and rotate the patch by angle δ_{n_i} around rotation axis $\mathbf{a}_{n_1}$ so that $\mathbf{n}_f = \frac{1}{N}\sum_{i\in\mathcal{N}_i} A_i \cdot \mathbf{n}_{c_i} = \mathbf{c}$, where $\mathbf{c}$ is a user defined vector. Such a transformation would imbue the patch with rotation invariance across local x and y axes, while the face arrangement itself renders the patch descriptor rotation invariant w.r.t local z-axis. The motivation behind rotating each patch towards the same direction is that it allows efficient training with smaller training sets.

As a next step, to allow the convolutional kernels to capture the relation between locally consistent faces, we reshape the patch with a Hilbert curve function. Figures 5 and 6 highlight the different regions a patch captured by a Hilbert curve setting compared to a simple reshape function. Finally, to form the training set we extract patches from 3D scanned meshes and assign a class to each patch employing the saliency map extraction process presented in subsection 3.1

4. EXPERIMENTAL EVALUATION

4.1. Comparison Between the Traditional and the Deep Learning Approach

Our CNN-based saliency map extraction approach [1] adequately extracts salient features as Figure 7 presents. The first column presents the 3D scanned model and the second corresponds to the saliency map taken for groundtruth. The third column corresponds to Hilbert curve based patch face arrangement while the fourth column to simple reshape function patch face arrangement. Even though qualitative inspection shows that the saliency map is captured in both learning-based methods, the confusion matrices reveal, that Hilbert curve arrangement yields better accuracy. In terms of performance and computational complexity, even though, for smaller meshes, traditional and CNN-based approaches exhibit similar performance, the CNN approach is much faster in larger and models with higher density, as presented in Figure 8. It is important to highlight that for the head model fusion-based method required a patch size of $k = 58$ faces, for centurion model $k = 80$ and for stonecorner model $k = 150$. Patch size varies with model density and feature size. However, the patch size, that the CNN can handle, in similar or better execution times, is many times larger i.e. $k = 1024$ for the presented deep architecture. Fusion-based saliency mapping was executed in MATLAB while the CNN in Python Tensorflow. For the latter, the training took place in an NVIDIA GeForce GTX 1080 graphics card with 8GB VRAM and compute capability 6.1. For the testing and evaluation, the saliency map extraction took place in an Intel(R) Core(TM) i7-4790 CPU @ 3.60Hz with 32GB of RAM, while the I/O operations where not taken into account for the execution times evaluation.

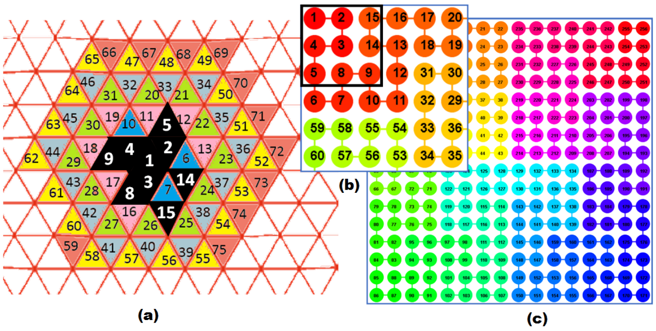


Fig. 5: Faces captured by convolutional kernel and Hilbert-based arrangement of patch faces.

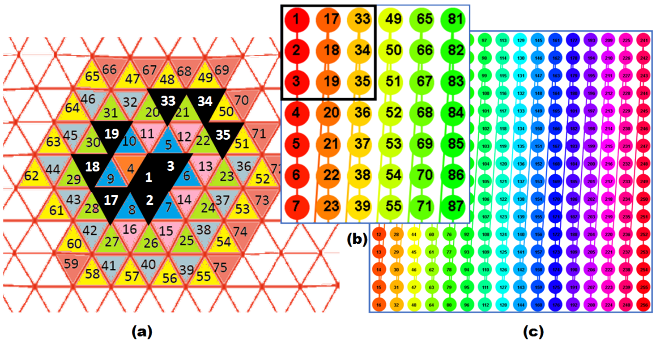
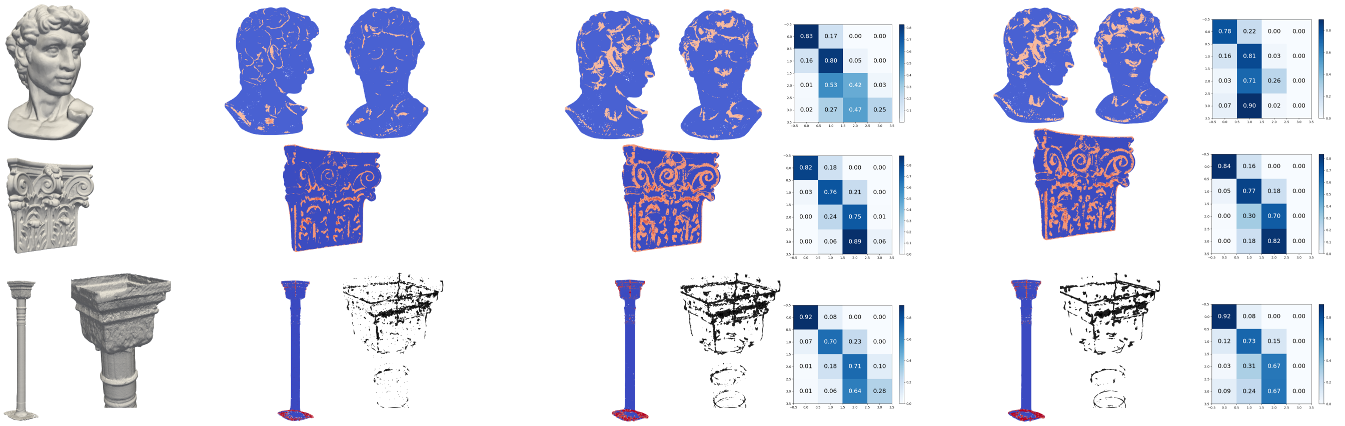


Fig. 6: Faces captured by convolutional kernel and simple reshape function arrangement of patch faces.

4.2. Simplification use case

Simplification of 3D meshes is a low-level application that focuses on representing an object using a lower resolution. The main objective of a successful simplification process is to preserve meaningful perceptual information. The most important geometrical information of a 3D object is the high-frequency spatial features which are mostly represented by corners and edges. However, in some special cases of geometrically complex 3D meshes, this binary classification of points, into features and non-features, is not always well-defined since more geometrical categories could be apparent. In Figure 10, we present the reconstruct results of different simplifications scenarios (i.e., $\sim 75\%$, 85%, and 90%) using two different approaches. More specifically, (i) we use only the features extracted by the fusion-based method, as presented in [23], and (ii) we firstly use our approach for the estimation of the saliency and then, we take different percentages of random points from each on of the four classes giving emphasis to the most salient class. We also provide the Hausdorff distance

[1] https://github.com/snousias/mesh-saliency-detection-using-convolutional-neural-networks



(a) (b) (c) (d)

Fig. 7: Saliency maps of 3D scanned geometries. (a) 3D mesh geometry, (b) groundtruth estimation, (c) CNN based saliency map extraction employing Hilbert curve arrangement with corresponding confusion matrices (d) CNN based saliency map extraction employing simple reshape based arrangement with corresponding confusion matrices.

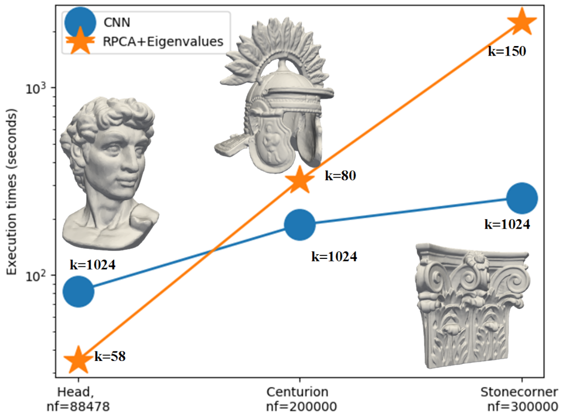


Fig. 8: Execution times evaluation. n_f is the number of faces an k is the patch size.

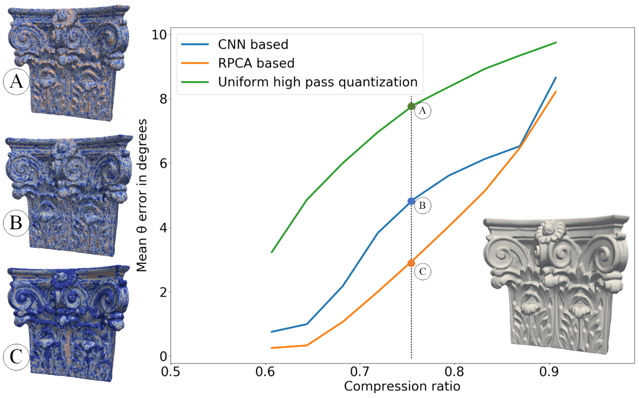


Fig. 9: Mean θ error as a function of compression ratio and θ error visualization. Red colors correspond to higher θ.

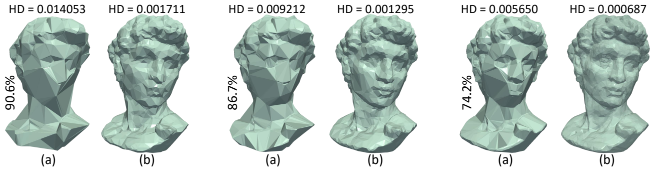


Fig. 10: Reconstructed results for different simplification scenarios, using: (a) the feature extraction method, described in [23], (b) our proposed method.

(HD) error applied to the triangle faces for easier comparison among the methods.

4.3. Compression use case

Compression of 3D meshes is another low-level geometry processing application allowing to reconstruct a mesh given the delta coordinates and a subset of mesh vertices according to the approach presented in [23]. To provide further evidence for the usability of CNN-based saliency map extraction we use the saliency map result to order the delta coordinates as follows $\mathcal{D}' = \{d \in \mathcal{D}|v_c = 4\}\{d \in \mathcal{D}|v_c = 3\}\{d \in \mathcal{D}|v_c = 2\}\{d \in \mathcal{D}|v_c = 1\}$, where $\mathcal{D}$ is initial set of delta coordinates and $v_c = 1$ to $v_c = 4$ is the predicted class index with the largest value to correspond to more significant salient features. The anchor points are uniformly selected, while only a portion of $\mathcal{D}'$ is non-zero. Figure 9 presents a reconstruction accuracy comparison between uniform, fusion-based and CNN-based non-zero delta selection revealing that CNN and fusion-based approaches converge while a uniform selection setting exhibits worse performance. For the evaluation of the efficiency of the compression achieved with CNN based saliency maps we employ mean theta error metric defined as the mean value of θ angle difference between the groundtruth face normal position and the reconstructed.

5. CONCLUSIONS

In this work, we presented a CNN-based 3D mesh saliency extraction approach, formulating a training set from spectral and geometrical 3D mesh processing. More specifically, we use the outcome of these methods in order to train a CNN model which can be used for the fast and accurate saliency map extraction of very dense 3D scanned models. Our aim is to provide a fast 3D saliency mapping method which could be beneficially used by applications that required the use of very dense 3D models and very fast processing. Extensive evaluation studies carried out, assuming a variety of evaluation scenarios, including heatmap visualization for visual perception, simplification and compression use cases, verify the superiority of our approach as compared to other state-of-the-art approaches.

6. REFERENCES

[1] Maria Nordfang and Jeremy M Wolfe, "Guided search for triple conjunctions," *Attention, Perception, & Psychophysics*, vol. 76, no. 6, pp. 1535–1559, 2014.

[2] Thomas Töllner, Michael Zehetleitner, Klaus Gramann, and Hermann J Müller, "Stimulus saliency modulates pre-attentive processing speed in human visual cortex," *PLoS One*, vol. 6, no. 1, pp. e16276, 2011.

[3] Sang-Hun Lee, Randolph Blake, and David J Heeger, "Traveling waves of activity in primary visual cortex during binocular rivalry," *Nature neuroscience*, vol. 8, no. 1, pp. 22, 2005.

[4] Shiva Kamkar, Hamid Abrishami Moghaddam, and Reza Lashgari, "Early visual processing of feature saliency tasks: A review of psychophysical experiments," *Frontiers in systems neuroscience*, vol. 12, 2018.

[5] A. Nguyen, J. Kim, H. Oh, H. Kim, W. Lin, and S. Lee, "Deep visual saliency on stereoscopic images," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1939–1953, April 2019.

[6] T. Alshawi, Z. Long, and G. AlRegib, "Unsupervised uncertainty estimation using spatiotemporal cues in video saliency detection," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2818–2827, June 2018.

[7] Chang Ha Lee, Amitabh Varshney, and David W Jacobs, "Mesh saliency," *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 659–666, 2005.

[8] Ran Song, Yonghuai Liu, Ralph R. Martin, and Paul L. Rosin, "Mesh saliency via spectral processing," *ACM Trans. Graph.*, vol. 33, no. 1, pp. 6:1–6:17, Feb. 2014.

[9] Ran Song, Yonghuai Liu, Ralph R Martin, and Karina Rodriguez Echavarria, "Local-to-global mesh saliency," *The Visual Computer*, vol. 34, no. 3, pp. 323–336, 2018.

[10] N. Wei, K. Gao, R. Ji, and P. Chen, "Surface saliency detection based on curvature co-occurrence histograms," *IEEE Access*, vol. 6, pp. 54536–54541, 2018.

[11] Jinliang Wu, Xiaoyong Shen, Wei Zhu, and Ligang Liu, "Mesh saliency with global rarity," *Graphical Models*, vol. 75, no. 5, pp. 255–264, 2013.

[12] P. Tao, L. Zhang, J. Cao, and X. Liu, "Mesh saliency detection based on entropy," in *2016 6th International Conference on Digital Home (ICDH)*, Dec 2016, pp. 288–295.

[13] G. An, T. Watanabe, and M. Kakimoto, "Mesh simplification using hybrid saliency," in *2016 International Conference on Cyberworlds (CW)*, Sep. 2016, pp. 231–234.

[14] MN Favorskaya and LC Jain, "Saliency detection in deep learning era: trends of development," *Management Information Systems*, , no. 3, pp. 10–36, 2019.

[15] Ran Song, Yonghuai Liu, and Paul Rosin, "Mesh saliency via weakly supervised classification-for-saliency cnn," *IEEE transactions on visualization and computer graphics*, 2019.

[16] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[17] Zhouchen Lin, Minming Chen, and Yi Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *CoRR*, vol. abs/1009.5055, 2009.

[18] Paul Rodriguez and Brendt Wohlberg, "Fast principal component pursuit via alternating minimization," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 69–73.

[19] Hans Sagan, *Space-filling curves*, Springer Science & Business Media, 2012.

[20] Craig Gotsman and Michael Lindenbaum, "On the metric properties of discrete space-filling curves," *IEEE Transactions on Image Processing*, vol. 5, no. 5, pp. 794–797, 1996.

[21] Douglas Voorhies, "Space-filling curves and pace-filling curves and a measure of coherence," *Graphics Gems II*, p. 26, 1991.

[22] G. Arvanitis, A.S. Lalos, and K. Moustakas, "Saliency mapping for processing 3d meshes in industrial modeling applications," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, July 2019.

[23] Aris S Lalos, Gerasimos Arvanitis, Aristotelis Spathis-Papadiotis, and Konstantinos Moustakas, "Feature aware 3d mesh compression using robust principal component analysis," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.