Learning to Predict 3D Mesh Saliency

Dalia A. ALfarasani, Thomas Sweetman, Yu-Kun Lai, Paul L. Rosin
School of Computer Science & Informatics, Cardiff University

***Abstract*—Mesh saliency, which measures the perceptual importance of different regions on a mesh, benefits a wide range of applications. However, existing mesh saliency models are largely built with hard-coded formulae, which cannot capture true human perception. Some existing techniques utilise indirect measures to capture user perception (e.g., mouse clicks), which can be unreliable. In this work, we collect eye-tracking data for 3D objects seen from different views, and develop an optimisation-based approach to fusing heat-maps captured from individual views to form consistent saliency maps on meshes. To predict mesh saliency on a new shape, we further develop a learning-based approach that regresses local surface characteristics based on a set of input features. Experimental results show that our learning-based method achieves better performance than state-of-the-art methods for unseen shapes. We will make our dataset publicly available.**

I. Introduction

Mesh saliency can play an important role in computer graphics in determining the outcomes of many tasks, such as feature detection [1], shape recognition [2], mesh segmentation [3], mesh watermarking [4], 3D printing [5], etc. Mesh saliency measures perceptual importance of local regions on a mesh, which is clearly subjective. It can be considered from a generic perspective where some 3D surface regions are considered more important than others [6], [7], but also task-dependent, e.g., in relation to touching [8]. Mesh saliency is also related to other metrics that measure 'uniqueness' or 'distinctiveness', e.g. surface distinction [9] and region distinctness [10]. However, distinctiveness measures focus on regions which set a shape apart from other shapes, which is different from mesh saliency. The ground truth for mesh saliency is typically determined by human perception, and subjective judgement is therefore involved in assessing the performance of such approaches.

Lee et al. [6] were the first to introduce the concept of mesh saliency, which is a computational measure of regional importance on a mesh. Their approach is based on differences of Gaussian, which is a geometric measure that aims to approximate human perceptual importance. Kim et al. [11] applied mesh saliency techniques to human eye movements using a 2D method in a user research. They used the standardised chance-adjusted saliency to measure the relationship between mesh saliency and fixation positions for 3D rendered images, demonstrating that existing computational models of mesh saliency can predict human eye movements significantly better than a purely random or curvature model. Although the importance of regions on 3D shapes can be considered in a general way, it can also be task-specific. For example, the work [8] considers the problem of tactile mesh saliency, where saliency is defined in the context of grasping, pressing and touching. In this paper, we focus on general visual saliency (i.e., without a specific task).

Many computational models for visual saliency of images have been both proposed and implemented. In 1985, Itti et al. [12] proposed an early model which stated that image locations with saliency will have some distinction from their surrounding environment. Some researchers in this field such as [13], [14], [15] have described in their works various other models of saliency. Stove and Straßer et al. [16] used saliency information acquired from the eye movements of an individual to simplify images, generating a non-photorealistic, painterly rendering. However, these works focus on using eye tracking for image saliency, rather than saliency of 3D shapes, which we investigate in this paper.

In this paper, we investigate using eye tracking data of rendered views of 3D shapes as a way to obtain ground truth saliency on meshes. As each view is only able to cover part of the mesh, and different views may contain shape parts with significantly different levels of saliency, an optimisation approach is developed to fuse saliency derived from individual views to take into account their relative saliency levels, while ensuring consistent saliency values in the shared regions. Based on this, we further build machine learning models to predict mesh saliency, based on local geometric features and existing 3D saliency prediction models. Our experiments show that a learning based approach achieves better performance than existing saliency methods on unseen shapes.

II. Related work

Recently, several algorithms for computing the saliency of 3D models have been developed. Many algorithms are based on the 'centre-surround' method of Lee et al. [6] that uses the absolute difference between the Gaussian-weighted average of the mean curvature at scales σ and 2σ, with the Gaussian filtering limited to neighbourhoods of size 2σ. Several scales with different σ values are jointly used to capture saliency at different scales. Yang et al. [17] proposed a method for quantitatively calculating visual attention based on eye-tracking data for 3D scene maps by obtaining the participants' gaze behaviour differences to establish a quantitative relationship between eye movement indexes and visual saliency. Liu et al.'s [18] use of virtual agents to simulate how humans interact with objects helps to understand shapes and to identify their salient parts in relation to their functions. Moreover, Chen et al. [19] investigated human perception, and considered 3D mesh Schelling points, which are feature points chosen by people in a coordination task. They found that Schelling point sets

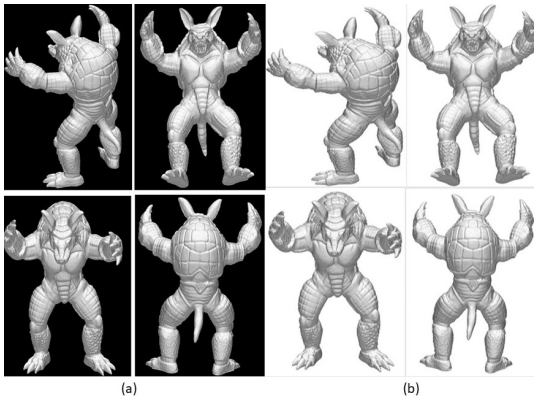

Fig. 1. Example views of one mesh (Armadillo) rendered in (a) black and (b) white backgrounds. 4 out of 20 views are shown here.

are usually highly symmetric, and local curvature properties are the most useful method for identifying Schelling points. They propose using sophisticated deep learning approaches to discover mesh Schelling points automatically, without the need for participant observations. The authors use mesh convolution and pooling to extract meaningful characteristics from mesh objects and then predict the 3D heat map of Schelling points end-to-end [20].

Mesh saliency has many interesting applications. Leifman et al. [10] presented an algorithm for identifying regions of interest on 3D surfaces. Their method studies 3D regions of distinctiveness from both local and global perspectives, and demonstrates that saliency derived from their method is effective for viewpoint selection. Howlett et al. [21] demonstrated the value of saliency for guiding 3D simplification, where saliency was captured using an eye-tracker for recording the two-dimensional image area in which an individual has looked at a three-dimensional model.

Although there are many ways to detect 3D mesh saliency, there have been few techniques developed to evaluate their effectiveness. Many papers utilise heatmaps for showing salient model parts or utilise saliency-led mesh simplification for demonstrating the methodology while preserving attention-grabbing parts. Even though such approaches succeed in showing how they operate at a high level, they make it hard to compare the effectiveness of various methods as they only provide subjective evaluation.

In this paper, we investigate a methodology to produce ground truth mesh saliency through fusion of eye tracking data for different views of rendered 3D shapes, and based on this, further develop machine learning methods for predicting saliency on 3D meshes.

III. Proposed Method

Our method involves a methodology to capture ground truth mesh saliency by fusing eye tracking data for multiple rendered views of 3D shapes. We then split the captured mesh saliency dataset into training and test sets, and build machine learning models using the training set data to predict saliency on the unseen test set.

A. Obtaining ground truth mesh saliency through eye tracking

1) Shape rendering and eye tracking data capture: Eye tracking provides an intuitive way for collecting user interest given visual stimuli. However, it is not possible for the participant to see an entire 3D shape at once. To address this, we place each shape at the origin, scale it to fit in the unit sphere, and render 20 evenly distributed views (using face centres of an icosahedron as the camera location with the camera direction pointing to the origin) to provide sufficient coverage of the shape. For each rendered image, we also keep the mapping between the rendered image and the 3D mesh so that image-based saliency values can be mapped back to the 3D mesh. For implementation convenience, this is represented as a vertex map, corresponding to each rendered view, where at the projected position of each vertex, we use a unique RGB colour to record its index. This can be efficiently achieved using the standard OpenGL rendering pipeline. When producing these view images, we need to ensure they express clear clues for 3D shapes, but avoid introducing artifacts that may distract user attention. We considered two alternative ways of choosing the background, namely using black background and white background (see Figure 1 for some examples).

Our preliminary user evaluation shows that black background is less distracting than white background so that participants will concentrate on the actual shapes rather than their attention wandering around in the background, and so this is chosen for rendering. This is evidenced by the fact that users spent significantly more time focusing on the background when white background is used, compared with black background. We also set the light source to be in the same location as the camera and pointing towards the centre of the object, and ensure the captured view is well lit (but not over-exposed).

As the subjective result does not have a unique correct answer, it can be hard to measure the effectiveness of user input, and so before we start the experiment we show the participants a trailer of the experiment to make sure they fully understand the task before starting. Some other studies allow the user to rotate and zoom in/out of the model; however, to effectively capture eye tracking data, a series of eye gaze locations and corresponding durations needs to be recorded for each view, and so we pre-render and fix each view as a static image. Also, we let the participants ask questions during the trailer so the user can feel more confident during the experiment, leading to more accurate results.

The participants are then shown images of these rendered views, and their eye tracking data is captured. Note that adjacent views have large overlaps, which not only happens naturally, but is also useful, as this allows saliency captured from different views to be reliably fused. However, this leads to a potential problem of memorising: when a user is presented with a similar view shortly before, he/she may not actively try to explore interesting features of the shape. To avoid this, we carefully group the 20 views into 5 groups, each with 4 views, such that these 4 views are as widely separated as possible, and

each user is only asked to look at one group of views for each shape. Before starting the experiment, we had a quick mock experiment to test the time needed to view the 3D shape. We found that showing each rendered image for 5 seconds gives more accurate results than 10 seconds, and a gap of 2 seconds between images is sufficient to rest the eyes and lose any fixation from the previous image and to provide a pause in order to allow the subjects' eyes to relax and focus.

The previously generated vertex map can then be used to remap eye-tracking data on a two-dimensional image back to the three-dimensional model. When the eye-tracking data has marked a fixation on a point in the image we can find the nearest non-black pixel on the vertex map and use the RGB colour data in that pixel to find the exact vertex the subject had fixated on allowing us to assign fixations on a two-dimensional rendered image back to the original model. However, a vertex is likely to be seen from multiple rendered views, so these need to be fused to obtain a consistent saliency value for each vertex.

2) Ground truth mesh saliency generation and fusion: In the following, we discuss how we work out the ground truth saliency map on a mesh M. Let R_i $(i = 1, 2, \ldots, 20)$ be the 20 rendered views of M. For each view R_i, the eye tracking data of all the users is collected, and represented as a sequence of eye fixation points $(x_j^{(i)}, y_j^{(i)}, t_j^{(i)})$, where j is the sample index, $(x_j^{(i)}, y_j^{(i)})$ are the coordinates of the fixation point in the image domain, and $t_j^{(i)}$ is the duration of the fixation. As fixation points tend to be sparse, following the common practice in image saliency research that applies Gaussian blurring to the fixation map to estimate the saliency map, we map discrete fixation maps to meshes to obtain per-vertex saliency values as follows. We first map 2D fixation point $(x_j^{(i)}, y_j^{(i)})$ to the corresponding fixation vertex $v_j^{(i)}$ on the 3D mesh M. It iterates over each fixation in the experiment. Each fixation takes the vertex map corresponding to the 2D image fixation, takes the fixation x and y position in pixels, finds the nearest coloured pixel in the vertex map, and decodes the RGB value into a vertex index. Let $d_{\max}$ be the distance between the two farthest apart vertices on the mesh. Each vertex v in the neighbourhood $\mathcal{N}_j^{(i)}$ on the mesh M receives a saliency contribution from the fixation vertex $v_j^{(i)}$ according to the following formula:

$$s(v, v_j^{(i)}) = \exp\{-d(v, v_j^{(i)})/\bar{d}\} \cdot t_j^{(i)} \qquad (\forall v \in \mathcal{N}_j^{(i)}). \tag{1}$$

In practice, $\bar{d}$ is set to 0.05 times $d_{\max}$, and $\mathcal{N}_j^{(i)}$ is defined as those vertices v with distance to the fixation vertex $d(v, v_j^{(i)}) \leq \bar{d}$. This ensures each fixation point influences a reasonably sized neighbourhood, with the influence dropping where the distance from the fixation point increases. The distance measure $d(\cdot, \cdot)$ is ideally geodesic distances, although in practice Euclidean distance gives a decent approximation and is used in our experiments due to the relatively small neighbourhood size, and shapes not having highly folded structures.

Then, the contributions of all fixation points from the same view are summed up to work out the saliency value for each vertex w.r.t. the given view $s_v^{(i)}$:

$$s_v^{(i)} = \sum_j s(v, v_j^{(i)}). \tag{2}$$

However, the saliency values for different views are not directly comparable. For example, if one view contains highly salient regions, e.g., faces, some potentially important but less significant regions, e.g. hands, may receive low saliency, whereas if the hands are seen without faces at the same time, they may be seen as highly salient in that particular view. Therefore, the relative importance of each vertex needs to be normalised when fusing inputs from different views. Let the rendered view $\mathcal{V}^{(i)}$ be the vertices that are visible from view R_i, we further introduce a weight w_i for the i-th view, and use the commonly seen regions as anchors for normalisation, formulated as the following optimisation problem:

$$\min \sum_{i_1, i_2 \in \{1,2,\ldots,20\}, i_1 \neq i2} \sum_{v \in \mathcal{V}^{(i_1)} \bigcap \mathcal{V}^{(i_2)}} \left(w_{i_1} s_v^{(i_1)} - w_{i_2} s_v^{(i_2)} \right)^2, \tag{3}$$

where i_1 and i_2 iterate over all adjacent views (with at least one shared vertex). This ensures shared vertices across multiple views have saliency values as consistent as possible. To avoid getting trivial solutions with $w_1 = w_2 = \cdots = w_{20} = 0$, we additionally introduce a constraint:

$$\sum_i w_i = 1. \tag{4}$$

The above least-squares optimisation problem can be easily solved by solving a (small) linear system with the weights of individual views as unknowns. The final saliency value for s_v is obtained by averaging over values, linearly scaled to $[0, 1]$:

$$s_v = \frac{\sum_{i=1, v \in \mathcal{V}^{(i)}}^{20} w_i \cdot s_v^{(i)} - s_{\min}}{s_{\max} - s_{\min}}, \tag{5}$$

where $s_{\min}$ and $s_{\max}$ are the minimum and maximum values of s_v (before linear scaling).

B. Machine learning based approach to predicting mesh saliency

Existing methods for mesh saliency are largely based on handcrafted rules. In this paper, we investigate using learning based approaches to predict mesh saliency. To make this task feasible, given relatively limited training data, we take features at each vertex as input, and predict saliency values such that they are as close as possible to the ground truth saliency as described in the previous subsection. The features we use include a combination of geometric features (i.e. Heat Kernel Signature (HKS) [22], conformal factor [23], MeshSIFT [24], SHOT [25], Gaussian curvature and off-centre bias [26]) and existing 3D saliency models, namely Lee et al. [6], [9] and Song et al. [7]. Here, off-centre bias simply measures the Euclidean distance of a vertex position from the centre of the object such that the further away from the centre, the more

salient it is considered. This is intuitive as protrusions tend to have higher saliency values.

Let $\mathbf{f}_v = (f_{v,1}, f_{v,2}, \ldots, f_{v,N})$ be the feature vector containing both geometry related and existing saliency estimation results for vertex v, where N is the total number of feature values for a vertex. We split our mesh dataset with ground truth saliency into training and test sets. Machine learning models are built using all the vertices of the meshes in the training set, and then we retain test set mesh vertices for testing purpose.

For this purpose, since we have relatively limited training data, we focus on traditional machine learning models, rather than deep neural networks. We tried three different models: linear least squares regression, a feed-forward neural network and support vector regression (SVR). The least squares regression aims to work out the optimal per-feature weight ω_k and bias b such that the model best predicts the saliency values, i.e.

$$\min \sum_v \left(\sum_{k=1}^{N} \omega_k f_{v,k} + b - s_v \right)^2 . \tag{6}$$

The feedforward neural network is a shallow neural network with three layers where the input layer contains N nodes corresponding to the input features, the hidden layer contains 10 nodes, and the output layer contains 1 node corresponding to the predicted mesh saliency value [27]. For SVR, the standard model is used, which takes per-vertex features as input and predicts the saliency value for the vertex [28].

IV. Experimental Setup and Results

A. Experimental Procedure

The experiments were administered within the School of Computer Science and Informatics, Cardiff University. The experiment was carried out in a computer vision lab space, with an occasional reflective surface and constant close light. The viewing distance was kept at around 60 cm. The participants' eye movements were recorded using a non-invasive SensoMotoric Instrument (SMI) Red-m eye-tracker running at a rate of 250 Hz. Gaze data was retrieved from raw eye-tracking data obtained during the experiment using SMI's BeGaze™Analysis Software. For every 3D mesh this data contains the number of fixation points, and for each fixation point, its coordinates and duration. Fixation was captured by SMI's computer software using the distribution and duration-based formula established, with minimum period of 100 ms. The mean duration of fixations μ_i for a subject i is:

$$\mu_i = \frac{1}{n} \sum_{j=1}^{n} x_j \tag{7}$$

where n is the total number of fixations recorded over the 400 stimuli utilised in our study and x_j is the duration of the fixation j.

Participants: Forty female and twenty male members from the School of Computer Science at Cardiff University, with ages in the range of 20 to 39, volunteered to participate.

Design and procedure: Participants were informed that their task in the experiment was to look at the region on the model that they think is of most interest, and the participant was not allowed to move their head during the experiment, so that if there is an issue we can pause the experiment and then recommence. The experiment took only ten minutes per participant.

B. Eye Tracking and Mesh Saliency Ground Truth Results

In particular for ensuring that participants can concentrate while doing the study, we used different models from the Stanford repository and the AIM@SHAPE-VISIONAIR shape. Our goal is to include a variety of high-resolution shapes (in terms of triangle counts) from benchmark datasets to enable efficient prototyping and practical evaluation of real-world and large-scale shape models. We select 20 shapes in our study. This leads to 400 rendered images (20 views per shape) at 1920×1080 pixel resolution. As described before, each participant is shown 4 views of each shape, leading to a total of 80 images. The eye tracking data of all users is then fused to produce ground truth mesh saliency on 20 shapes. We split the dataset into training and test sets, each containing 10 shapes. Note that although 20 shapes are not many, each shape contains thousands of vertices, and thus it is sufficient for training machine learning models when applied at the vertex level.

We now show some examples demonstrating the effectiveness of our fusion strategy for saliency from individual views. As shown in Figure 2, the initial eye tracking data is captured on individual views, which are then fused using our method to produce a consistent saliency map on each mesh (with a normalised saliency value assigned to each vertex of the mesh). As shown, the fusion works well, with salient areas that get a lot of attention from multiple different views enhanced like both faces in the examples in Figure 2. Regions which receive little attention from the participants correspond to those boring/less distinctive areas of the model, and they have low saliency values.

More examples of ground truth saliency maps from eye tracking are shown in Figure 3. Similar trends are observed, although for objects not including faces (e.g. the Falling shape), salient regions tend to be more variable, and some regions on the body are also relatively salient (although less so than faces); see e.g. the Bulldog and Gargoyle shapes.

Participants paid most attention to models with visible facial features. As shown in Figure 3, the saliency map of the Falling mesh is rather different from those of the other models where the head on them has a very high salience. This might be because of the saliency values around the face greatly outweigh any other values as each participant will look at the face at some point. This means that there is a low salience value for areas that are not in the facial region when the saliency map is normalised 0 to 1. One way to circumvent this would be to use a non-linear scaling to stop strong salient areas from blocking out the rest of the shape. The other observation is that, obviously, people first look at the most salient area of an image. The most salient area in the case of these models might be the head, but as participants have only 5 seconds

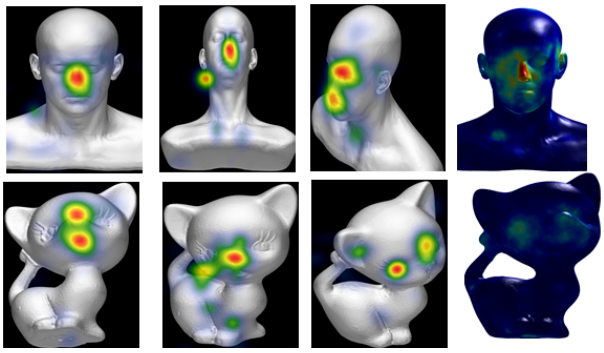

Fig. 2. Examples of 2D fixation maps, and the results of fusing them to form consistent saliency maps on 3D models (rightmost column); red and yellow are salient areas, while green and blue are non-salient areas.

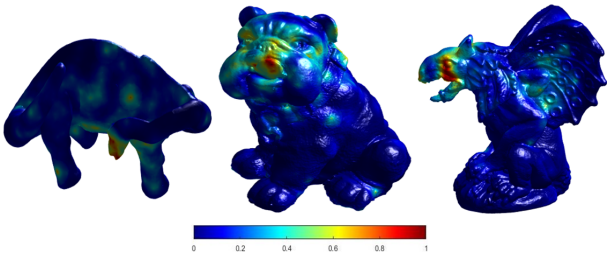

Fig. 3. Examples of ground truth salient maps derived from eye tracking data; red and yellow are salient areas, while green and blue are non-salient areas. From left to right Falling, Bulldog and Gargoyle shapes.

to view each image, they may not have time to look at other slightly less salient areas on the mesh. A possible solution for this would be to reduce the exposure time of each object and assign a higher saliency weight to fixations at the start of a viewing so that the first objects viewed were given more salience than a point viewed at the end of the viewing time. This is left to explore as future work.

V. Learning new methods of measuring 3D mesh saliency

A. Evaluation with Ground Truth Saliency Maps

Our collected ground truth saliency maps can be used to evaluate the effectiveness of existing mesh saliency methods in a quantitative way. To evaluate the existing methods of measuring saliency, methods are required for comparing two saliency maps on the same mesh, i.e. the ground truth generated by the eye tracking experiment and the saliency map output by a saliency prediction method. The prediction method performance is considered better if it has a closer distribution to the ground truth.

A basic measure for the similarity between these maps is Mean Squared Error (MSE), which is 0 if they are a perfect match, and a high value if they are dissimilar. This measure is simple, but it only works well when the absolute salient values of two saliency maps are close. In practice, however, it is the relative importance which is more important. For instance, if one region is more important than another, it is hard to know how much the salience value of the first region should be larger than that of the second.

TABLE I

Average SSIM value and MSE for each existing method and our learning based method for evaluating the quality of predicted saliency maps against the ground truth derived from eye tracking. Only test set is used to ensure fair comparison. For SSIM, larger is better, and for MSE, smaller is better.

	Models	SSIM	MSE
Existing models	Off-centre bias	0.620	0.020
	Lee et al	0.629	0.016
	Song et al.	0.751	0.010
	Conformal factor	0.600	0.026
	Gaussian curvature	0.616	0.022
Learnt models	Least squares regression	0.906	0.004
	Feedforward neural network	0.895	0.006
	Support vector regression (SVR)	0.861	0.009

To address this, we also utilise SSIM which is a method for predicting the perceived quality when measuring the similarity between two images. The SSIM values ranges between (-1 to 1), where 1 means perfect match the reconstruct image with original one [29] and is known to be better correlated to perceptual similarity and less sensitive to absolute value differences. We extend the standard SSIM defined in the image domain to 3D mesh heatmaps. Changing SSIM to operate on 3D heat maps requires adapting the neighbourhoods of the standard SSIM. SSIM takes a window around each pixel when working on 2D images, but this does not work directly for meshes due to their irregular connectivity. We therefore replace such windows with neighbourhoods on meshes within a certain distance to the vertex of concern. For this purpose, a smaller neighbourhood than that used in the eye tracking mapping is more meaningful, and we set it to $0.02 \times d_{\max}$ where $d_{\max}$ is the farthest distances between pairs of vertices on the mesh.

Pixels are always spaced apart uniformly and are perfectly uniform in size, neither of which is true with vertices. So we developed a replacement window for 3D SSIM where the neighbourhood is defined as vertices within a set distance from the vertex of concern, if it can be reached via vertices also within the distance through graph traversal. This ensures that vertices which are close in 3D space but from disjoint parts are not included. This is essentially very similar to the remapping neighbourhood, except that a smaller neighbourhood size (2% of $d_{\max}$ rather than 5%).

B. Evaluation Results of Existing and Our Learning based Methods

We now apply our evaluation methodology to existing mesh saliency methods and our learning based methods. To ensure fair comparison, in particular between existing methods and learning based methods, we only report the average performance on the test set. For existing methods, we test representative methods Lee et al. [6], Song et al. [7] and baseline methods Gaussian curvature, conformal factor, and off-centre bias. Quantitative evaluation on our eye tracking

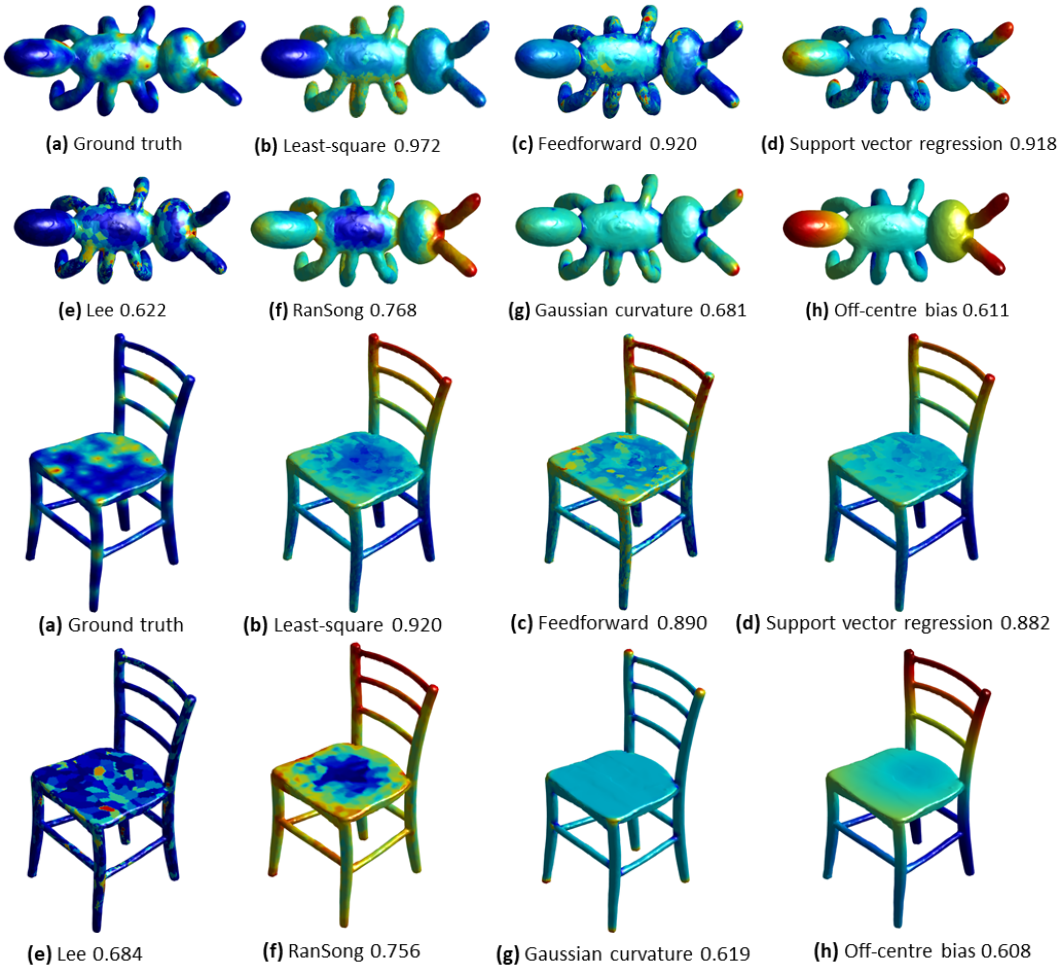

Fig. 4. Examples of saliency results: (a) ground truth, (b,c,d) our learning based methods, (e,f) existing methods and (g,h) geometry feature based baseline methods.

based test set is reported in Table I. As can be seen, Song et al.'s method achieves better performance than other existing and baseline methods, according to both SSIM and MSE. Other methods tend to give similar performance, with Lee's results better than the three baseline methods in both metrics.

In comparison, the three variants of learning based methods all perform better than existing methods, according to both SSIM and MSE metrics. In general we found that least-squares regression outperforms more complicated methods including feedforward neural networks and SVR. This is probably because of the relatively limited data, and the simpler linear model avoids overfitting and generalises better to unseen data. Our learning based method achieves 0.906 SSIM and 0.004 MSE, which are significantly better than state-of-the-art methods (0.751 SSIM and 0.010 MSE for Song et al. [7]).

Visual comparisons of different results on two shapes are shown in Figure 4, along with SSIM values. As can be seen, ground truth is generally plausible, and learning based methods, in particular the one based on least-squares regression, predict saliency maps which are more similar to the ground truth.

VI. Conclusion

Estimating saliency on meshes is a fundamental tool that benefits many downstream applications. Existing methods largely focus on developing dedicated formulae to achieve this, but it is difficult to fully capture perceptual importance using these methods. In this paper, we investigate a methodology to produce ground truth saliency maps on meshes using eye tracking data. In particular, we fuse saliency maps from individual views to produce a single consistent saliency map for a given mesh. The dataset will be made publicly available. Based on this, we further develop learning based methods that take existing saliency prediction results and geometric features at each vertex as input to predict the local saliency value. Qualitative and quantitative results show that our learning based methods, in particular the model based on least squares regression, outperform state-of-the-art methods. In future work we would like to build a larger dataset and evaluate the effectiveness of more machine learning methods, including methods based on deep neural networks. Different eye tracking data capture protocols will also be investigated, to identify a more suitable one in the presence of highly salient regions.

REFERENCES

[1] Ivan Sipiran and Benjamin Bustos, "Key-components: detection of salient regions on 3D meshes," *The Visual Computer*, vol. 29, no. 12, pp. 1319–1332, 2013.

[2] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 945–953.

[3] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–12, 2009.

[4] Nassima Medimegh, Samir Belaid, Mohamed Atri, and Naoufel Werghi, "3D mesh watermarking using salient points," *Multimedia Tools and Applications*, vol. 77, no. 24, pp. 32287–32309, 2018.

[5] Weiming Wang, Haiyuan Chao, Jing Tong, Zhouwang Yang, Xin Tong, Hang Li, Xiuping Liu, and Ligang Liu, "Saliency-preserving slicing optimization for effective 3D printing," in *Computer Graphics Forum*. Wiley Online Library, 2015, vol. 34, pp. 148–160.

[6] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 24, no. 3, pp. 659 – 666, 2005.

[7] R. Song, Y. Liu, R.R. Martin, and P.L. Rosin, "Mesh saliency via spectral processing," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 1, pp. 6, 2014.

[8] Manfred Lau, Kapil Dev, Weiqi Shi, Julie Dorsey, and Holly Rushmeier, "Tactile mesh saliency," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 52, 2016.

[9] P. Shilane and T. Funkhouser, "Distinctive regions of 3D surfaces," *ACM Trans. Graph.*, vol. 26, no. 2, pp. 7, 2007.

[10] G. Leifman, E. Shtrom, and A. Tal, "Surface regions of interest for viewpoint selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2544–2556, 2016.

[11] Youngmin Kim, Amitabh Varshney, David W Jacobs, and François Guimbretiere, "Mesh saliency and human eye fixations," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 2, pp. 1–13, 2010.

[12] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 11, 1998.

[13] P.S. Heckbert and M. Garland, "Optimal triangulation and quadric-based surface simplification," *Computational Geometry*, vol. 14, no. 1-3, pp. 49–65, 1999.

[14] R. Milanese, H. Wechsler, S. Gill, J. Bostl, and T. Pun, "Integration of bottom-up and top-down cues for visual attentionusing non-linear relaxation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 781–785.

[15] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1–2, pp. 507–545, 1995.

[16] S.L. Stoev and W. Straßer, "A case study on automatic camera placement and motion for visualizing historical data," 2002, pp. 545–548.

[17] Bincheng Yang and Hongwei Li, "A visual attention model based on eye tracking in 3d scene maps," *ISPRS International Journal of Geo-Information*, vol. 10, no. 10, pp. 664, 2021.

[18] Zhenbao Liu, Xiao Wang, and Shuhui Bu, "Human-centered saliency detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1150–1162, 2015.

[19] Xiaobai Chen, Abulhair Saparov, Bill Pang, and Thomas Funkhouser, "Schelling points on 3D surface meshes," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–12, 2012.

[20] Geng Chen, Hang Dai, Tao Zhou, Jianbing Shen, and Ling Shao, "Automatic Schelling points detection from meshes," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[21] S. Howlett, J. Hamill, and C. O'Sullivan, "An experimental approach to predicting saliency for simplified polygonal models," in *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*. 2004, vol. 57-64, ACM.

[22] Wei-Zu Yang, Liang-Chang Yu, Po-Chou Chen, and Tai-Liang Chen, "The design of multimedia web-based phone and billing system with freeware over the VoIP network," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*. IEEE, 2006, vol. 1, pp. 4–pp.

[23] D. Pickup, X. Sun, P.L. Rosin, and R.R. Martin, "Euclidean-distance-based canonical forms for non-rigid 3D shape retrieval," *Pattern Recognition*, vol. 48, no. 8, pp. 2500–2512, 2015.

[24] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[25] Federico Tombari, Samuele Salti, and Luigi Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision*. Springer, 2010, pp. 356–369.

[26] Ali Borji and James Tanner, "Reconciling saliency and object center-bias hypotheses in explaining free-viewing fixations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1214–1226, 2015.

[27] G Ososkov and P Goncharov, "Shallow and deep learning for image classification," *Optical Memory and Neural Networks*, vol. 26, no. 4, pp. 221–248, 2017.

[28] Zeynep Cipiloglu Yildiz, Abdullah Bulbul, and Tolga Capin, "Modeling human perception of 3d scenes," in *Intelligent Scene Modeling and Human-Computer Interaction*, pp. 67–88. Springer, 2021.

[29] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.