Mesh Saliency via Weakly Supervised Classification-for-Saliency CNN

Ran Song, Yonghuai Liu, *Senior Member, IEEE,* Paul L. Rosin

Abstract—Recently, effort has been made to apply deep learning to the detection of mesh saliency. However, one major barrier is to collect a large amount of vertex-level annotation as saliency ground truth for training the neural networks. Quite a few pilot studies showed that this task is difficult. In this work, we solve this problem by developing a novel network trained in a weakly supervised manner. The training is end-to-end and does not require any saliency ground truth but only the class membership of meshes. Our Classification-for-Saliency CNN (CfS-CNN) employs a multi-view setup and contains a newly designed two-channel structure which integrates view-based features of both classification and saliency. It essentially transfers knowledge from 3D object classification to mesh saliency. Our approach significantly outperforms the existing state-of-the-art methods according to extensive experimental results. Also, the CfS-CNN can be directly used for scene saliency. We showcase two novel applications based on scene saliency to demonstrate its utility.

Index Terms—Mesh saliency, deep learning, transfer learning, weak supervision.

1 Introduction

Mesh saliency has been an active topic in computer graphics for a long time. Although conceptually it has several variants such as mesh saliency [1], surface distinction [2] and region distinctness [3], they all represent the understanding of 3D surfaces from the perspective that some regions of a 3D surface are more important than the others in agreement with human perception. In particular, this paper follows the definition in the seminal paper of Lee *et al.* [1] where mesh saliency is defined as a per-vertex map to predict "what most of us would classify as interesting regions in meshes".

Most previous work on mesh saliency relied on hand-crafted features which do not generalise well since their expressive capabilities are limited by the fixed operations that stay the same for meshes of different classes. In recent years, we observed the trend towards learning-based methods, particularly the recent work based on deep learning [4] where a collection of ground truth saliency data must be provided for training some models of machine learning. However, as shown in quite a few user studies (e.g. [3], [4], [5], [6]), gathering such data is a difficult task. We noticed that all of the existing datasets of mesh saliency are very small (e.g. 400 meshes in [5], 79 meshes in [3], 150 meshes in [4] and 32 meshes in [6]). On the other hand, a neural network has to be sufficiently deep to generalise well since learning features at various levels of abstraction requires a sufficiently large number of layers. Accordingly, training has to be supported by massive data to avoid overfitting.

This work is motivated by the dilemma currently widely existing in the 3D object understanding community that the trend of embracing deep learning is hindered by the difficulty of collecting a large amount of accurate and consistent vertex-level annotations for training the learning models, especially the popular deep neural networks [4], [7]. Although some effort such as [8] has been made recently, creating vertex-level annotations for regression problems such as mesh saliency is still expensive. We propose a weakly supervised solution where the training is only based on the classification ground truth. We believe that such a solution is attractive for two reasons. First, a vertex-level annotation in a graphics task is often more expensive than the pixel-level annotation in a corresponding vision task. For instance, the ground truth generation of 3D interest point detection on meshes is more time consuming than that of 2D interest point detection on images since human subjects need to rotate a mesh to mark the vertices of interest [9]. Thus the demand for a weakly supervised method that does not rely on such annotation might be higher in the graphics community. Second, the simplest, most efficient and most consistent annotation that can be collected for most 3D object understanding tasks is the class membership of an object. Due in part to this reason, some large-scale datasets for 3D object classification are already publicly available (e.g. ModelNet [10] and ShapeNet [11]).

We present a new convolutional neural network (CNN) for mesh saliency, namely Classification-for-Saliency CNN (CfS-CNN) which can be trained end-to-end using only the ground truth classification. The first step is to represent a mesh as multiple 2D views. Second, each view is fed into a CNN to generate view-based convolutional features. Next, the CNN branches off a classification channel and a newly designed saliency channel and then multiple view-based features of both classification and saliency are pooled to obtain a descriptor representing the entire 3D mesh. Finally, this descriptor is transformed by the last fully-connected layer to train a classifer. When deploying the

- *R. Song (corresponding author: r.song@brighton.ac.uk) is with the Centre for Secure, Intelligent and Usable Systems, School of Computing, Engineering and Mathematics, University of Brighton, UK*
- *Y. Liu is with the Department of Computer Science, Edge Hill University, UK.*
- *P. L. Rosin is with the School of Computer Science and Informatics, Cardiff University, UK.*

Manuscript revised May 2019

learned CfS-CNN, we first generate view-based 2D saliency maps via classification activation mapping (CAM). Then, we convert view-based 2D saliency into view-based 3D saliency through a novel 2D-to-3D saliency transfer scheme. Note that for the same 3D vertex, its saliency values in multiple views are usually different. Thus finally, we generate a single per-vertex saliency map for the mesh by aggregating multi-view 3D saliency through a linear model. The weights used in this model are output by the saliency channel and essentially derived from the 3D information of the mesh.

It is noteworthy that the 2D views of a 3D model are different from normal 2D images since essentially they do not contain any object colour and material information, and the intensity of each pixel merely reflects local surface geometry. In comparison, 2D image saliency is largely driven by object colour and material while 3D mesh saliency is mainly driven by surface geometry.

Noticeably, here we propose to use a multi-view CNN other than a point-based net such as PointNet [12] or PointNet++ [13] as the baseline net of the CfS-CNN. This is because a multi-view CNN analyses what can be seen in a way similar to humans since it combines surface information from multiple views. To this end, for the particular task of mesh saliency, a measure reflecting human visual perception, a multi-view CNN is more suitable than those based on a point cloud representation. For example, a point not visible in most views is unlikely to be salient (e.g. a point on the inner surface of a vase) since in our method, its saliency is computed by aggregating multi-view saliency. In particular, a multi-view setup is a good choice for our work because we extends mesh saliency to scene saliency where occlusions happen more frequently.

Moreover, the requirement to handle a scene which contains multiple meshes is also an important reason that we develop a multi-view CNN rather than a graph neural network (GNN), which is widely used in geometric deep learning. A GNN learns a deep representation directly over the mesh treated as a non-Euclidean graph by a local operator such as Laplacian [14], [15] and Dirac operators [16]. But such local operators are not good at capturing the global spatial relationship of multiple objects not connected by edges of the mesh. However, such a relationship is recorded in one or multiple 2D views of the scene. To demonstrate the efficacy of scene saliency, we showcase two applications: best view selection of scenes and scene cropping.

Overall, the contribution of our work is threefold:

1) We propose a novel deep neural network for mesh saliency estimation.
2) Our network can be trained end-to-end in a weakly supervised manner with no expensive saliency annotation but mesh category information instead.
3) We demonstrate that our method can be used to generate meaningful saliency maps as well for 3D scenes through two new applications.

2 Related work

Early works on mesh saliency [1], [2], [17] heavily exploited handcrafted local geometric features. For example, Lee *et al.* [1] computed mesh saliency using a center-surround operator on Gaussian-weighted curvatures calculated in a local neighbourhood at multiple scales, and they later demonstrated in [18] that such a mechanism has significantly better correlation with human eye fixations than either a random model or a curvature-based model. However, as shown by psychological evidence [19], [20], [21], saliency also depends on global features. Thus to further improve saliency detection, researchers have proposed methods integrating both local and global features. Wu *et al.* [22] proposed an approach based on the observation that salient features are both locally prominent and globally rare. Song *et al.* [23] analysed the log-Laplacian spectrum of meshes and presented a method which considers both local geometric cues and global information corresponding to the low-frequency end of the spectrum. Wang *et al.* [24] detected mesh saliency using low-rank and sparse analysis in a space of shape features which encode both local geometry and global structure information of mesh. Leifman *et al.* [3] proposed an algorithm for detecting surface regions of interest by looking for regions that are distinct both locally and globally where the global consideration is if the object is 'limb-like' or not. Song *et al.* [25] proposed a local-to-global scheme to integrate both local saliency and the global distinctness of features.

It is natural to consider a data-driven method using data generated by human subjects since as a perceptual measure, mesh saliency reflects the human understanding of 3D data. And in many cases, we hope that artificial intelligence systems interpret data as humans do. However, due to the aforementioned training data problem, existing data-driven methods rely mainly on shallow learning. For example, in [5], a regression model to predict the so-called Schelling distribution is learned on a small dataset of 400 meshes. It is essentially a shallow learning scheme using a selection of handcrafted features. Lau *et al.* [4] proposed the concept of tactile mesh saliency which facilitated a reliable data collection since the concept is well defined and human subjects tend to give highly consistent responses in the process of data collection. Even so, only 150 models are collected for both training and testing. Although such an amount of data are sufficient to train the well designed 6-layer network used in the paper, they might not be enough to support the learning of a sufficiently deep network.

Fundamental differences from closely related works. In comparison with Lau's work [4], our work is fundamentally different in four aspects. First, our neural network is sufficiently deep (21 layers). Second, we train it on a much larger dataset (ModelNet40 [10]). Third, our method is weakly supervised with classification information. Fourth, our method can handle not only a single mesh, but also 3D scenes containing multiple meshes.

Compared to Song's work [26] which requires a separated inference of a Markov Random Field (MRF) component disconnected from the training of the deep neural network, our network is trained end-to-end. This is achieved by the two newly designed layers in the CfS-CNN which not only generate saliency knowledge based on features learned through the convolutional layers, but also inject such knowledge back into the fully-connected layers.

Compared to Shilane's work [2], our method is based on deep learning rather than handcrafted features. Also,

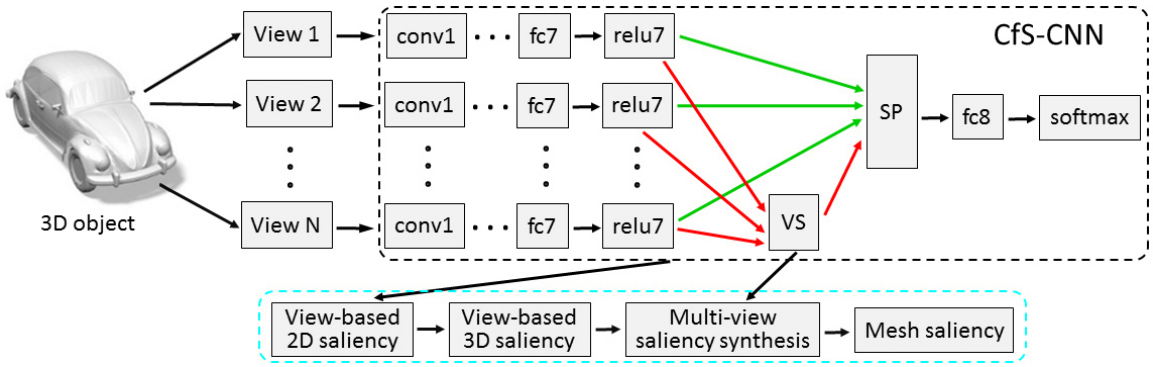


Fig. 1. Overview of the proposed approach based on the CfS-CNN containing two channels. The green and the red arrows denote the classification and the saliency channels respectively. "VS" and "SP" denote the view saliency and the saliency-based pooling layers respectively. The CfS-CNN is trained end-to-end as a classification network composed of the components enclosed by the black dash lines. The components enclosed by the cyan dash lines are implemented only in the deployment.

Shilane's work looks for perceptually important regions on a mesh solely from a classification perspective. But we apply transfer learning from classification to saliency, achieved through the newly designed layers in our network. As demonstrated experimentally in Section 4.5, this strategy significantly outperforms generating saliency solely from classification knowledge.

The proposed CfS-CNN is a multi-view CNN, but it is fundamentally different from other multi-view CNNs such as [26], [27], [28], [29], [30]. This is because the key component of a multi-view CNN, the scheme which aggregates outputs from multiple 2D views in our network is fundamentally different. Our pooling layer is essentially a weighted average pooling while [27], [29], [30] use max-pooling, [28] performs direct concatenation and [26] develops an MRF. This point is further discussed in Section 3.2 and demonstrated experimentally in Section 4.5.

3 METHOD

The pipeline of our method including the CfS-CNN as well as other components is illustrated in Fig. 1. In this section, we first describe each component used in training and/or deployment in a piecewise manner. Then, to help readers better understand how our method works, we explain the details of the implementation as a whole in both training and deployment where each component is contextualised with regard to the pipeline.

3.1 Multi-view mesh representation

Multi-view CNNs have been widely used to adapt CNNs to 3D objects represented as meshes. Compared with other methods which try to generalise deep learning to non-Euclidean domains, multi-view CNNs show state-of-the-art performance in various 3D object understanding tasks [27], [29], [30], [31]. We assume that each mesh is upright oriented along the z-axis and create N 2D views ($N = 24$ in this work) for it by placing N viewpoints (virtual cameras) at the positions defined by the pair $(azimuth, elevation)$. The $azimuth$ variable is the horizontal rotation about the z-axis measured in degrees and subject to $azimuth \in \{0, 30, \ldots, 330\}$. The $elevation$ variable is the vertical elevation of the viewpoint in degrees and subject to $elevation \in \{-30, 30\}$. Then virtual cameras point towards the centre of the mesh and their up vectors are also set as the z-axis. Adding more or different viewpoints is trivial, however, we found that such a viewpoint setup was already enough to achieve high performance. Please refer to Section 4.2 and Table 1 for other viewpoint setups.

3.2 Weakly supervised classification-for-saliency CNN

Motivation and inspiration. As a weakly supervised approach, we need to produce saliency based only on mesh-level annotation (i.e. class membership of meshes). We think this task is feasible due to a simple observation: for 3D objects of the same class, they usually have similar saliency distributions [5]. For example, for the meshes of humans, usually the head, the hands and the feet are detected as salient. For cups, usually the handle is detected as salient. One explanation is that the human perception system tends to capture the most informative features as salient [32] since it can help humans to recognise an object swiftly without the need for scrutinizing all of its details. Thus we argue that the informative features important for distinguishing a 3D object from others belonging to different classes are highly likely to be detected as salient.

Certainly, since mesh saliency and 3D object classification are two different tasks, a fundamental issue is the transferability of the knowledge learned through classification networks, which has only been explored in the context of 2D image understanding [33], [34]. One consensus is that the transferability decreases as the distance between the base task and the target task increases [33]. The CfS-CNN is thus inspired by the hypothesis that the knowledge vital for the base task, 3D object classification, is usually also important for the target task, mesh saliency due to the aforementioned observation. Certainly, it does not mean that an existing state-of-the-art classification network without a specific mechanism for mesh saliency will automatically have a state-of-the-art performance on mesh saliency. We shall further explore this hypothesis in Section 4.6.

We start with the classic VGG-19 model [35] pre-trained on ImageNet as the baseline architecture and then add the newly designed view saliency (VS) and saliency-based pooling (SP) layers onto it. Details of the two layers are described next.

View saliency layer. The VS layer takes as input the outputs of all relu7 (the ReLU layer following the fully connected layer fc7) layers. Since one 3D object is rendered as N views, the input of the VS layer is a matrix of size $4096 \times N$ for a given 3D object. Each of its columns can be regarded as a feature descriptor of one view. The VS layer outputs an N-dimensional vector to the SP layer. Each element of the vector corresponds to the saliency of a particular view, reflecting how salient that view is. The more salient the view, the larger the contribution it will make in the following classification and saliency generation.

Inspired by the works [20], [21] in neuroscience, we propose a simple scheme to compute view saliency. As pointed out in [20] and [21], a basic principle in human perceptual system is to suppress the response to frequently occurring features, while at the same time it remains sensitive to features that deviate from the norm. Thus, the view most different from all other views should be the most salient. Given two views V_i and V_j, their difference can be measured as the Euclidean distance between their feature descriptors F_i and F_j output by the relu7 layer

$$D_{ij} = \|F_i - F_j\|, \text{ s.t. } i, j \in \{1, 2, \ldots, N\} \text{ and } i \neq j. \quad (1)$$

The saliency of V_i is then calculated as the sum of its pairwise differences to all the other views.

$$S_i = \sum_{j \neq i} D_{ij}. \quad (2)$$

Both Eq. (1) and Eq. (2) are differentiable. So for back-propagation, given that the gradient passed to the VS layer (i.e. the gradient of the loss function with respect to the output of the layer) is an N-dimensional vector $\mathcal{S}$, according to the chain rule, the gradient $\mathcal{F}$ of this layer with regard to the input ($4096 \times N$) of the layer can be computed as

$$\mathcal{F}_i = \mathcal{S}_i \frac{\partial S_i}{\partial F_i} \quad (3)$$

where $\frac{\partial S_i}{\partial F_i}$ is a 4096-dimensional vector. Considering Eqs. (1) and (2) and the partial derivative of the Euclidean distance function $\frac{\partial \|x\|}{\partial x_i} = \frac{x_i}{\|x\|}$, it can be computed as

$$\frac{\partial S_i}{\partial F_i} = \sum_{j \neq i} \frac{F_i - F_j}{D_{ij}}. \quad (4)$$

Saliency-based pooling layer. While the VS layer produces saliency information, we need to think about how to incorporate it into a classification network since as a weakly supervised method, the only training data we have is the class membership of a 3D object. Also, as a multi-view CNN, we need to consider how to aggregate the learned knowledge across all the 2D views to create a single descriptor for the 3D object. And very importantly, we need to consider how to cast the influence of saliency into this aggregation process since as mentioned above, we hope that salient views can have larger weights in the classification process. This principle is also based on the simple observation that in many cases, one or two good views of a 3D object are enough for humans to recognise it, while some poor views could be very unhelpful. We propose an SP layer to address the three considerations.

As illustrated in Fig. 1, the SP layer takes as input the outputs of both the relu7 layer and the VS layer. If the output of the relu7 layer is a matrix F of size $4096 \times N$ and that of the VS layer is a N-dimensional vector S, the output of the SP layer, a 4096-dimensional vector P, can be computed as

$$P = FS. \quad (5)$$

Since it is a matrix multiplication, equivalently, we can express it as

$$P = \sum_{i=1}^{N} F_i S_i \quad (6)$$

where F_i is the column vector of F which denotes the feature descriptor of view V_i and S_i is its saliency. Thus it is quite clear that the vector P which can be regarded as the feature descriptor of the 3D object is estimated as the weighted sum of the feature descriptors of all the views where the weights are the estimated saliency of views.

In the back-propagation of the gradient of the classification loss, the gradient passed to the SP layer is a 4096-dimensional vector $\mathcal{P}$. Due to the bilinear form of Eq. (5), by the chain rule of gradients, the gradient $\mathcal{F}$ of the SP layer with regard to its first input F is calculated as

$$\mathcal{F} = \mathcal{P}S^T. \quad (7)$$

Similiarly, the gradient $\mathcal{S}$ with regard to its second input S can be calculated as

$$\mathcal{S} = F^T\mathcal{P}. \quad (8)$$

Interpretation on the saliency channel. As shown in Fig. 1, the VS and the SP layers compose the saliency channel of the CfS-CNN. Essentially, it enables the transfer learning from classification to saliency: the VS layer generates saliency knowledge using the knowledge learned through layers taken from a classification network (VGG-19) and injects such knowledge into the SP layer so that it can be incorporated into the classification network in a meaningful way. Therefore, we are able to train the entire network end-to-end with classification annotations.

It can be seen that the heuristic of the VS layer is manually defined through view differences although it is calculated with the learned features. Note that the most different view does not guarantee that its features are more important than features in other views. Here we actually assume that view importance correlates positively with feature importance in most cases. According to the experimental results, this strategy works well. Also, for implementation, the manipulation of views is much easier than that of local features in a multi-view CNN.

The SP layer is essentially an average pooling weighted by the learned view saliency, which makes it fundamentally different from existing multi-view CNNs. For instance, in [26], an MRF disconnected from the CNN is proposed to gather saliency information from multiple views. In [27], element-wise max pooling is used to synthesize the feature descriptors from multiple views into a single descriptor of the 3D object. It concludes that average pooling is not effective according to their experiments. In [29], a projection layer is employed in a segmentation network to aggregate feature descriptors across multiple views and project the output back onto the 3D object. They also used a max

pooling and reported that an alternative average pooling resulted in lower performance in their experiments. In [30], max pooling is used for part correspondences of 3D objects. They concluded that it offers "significantly higher" performance than average pooling. However, we shall show that our method performs comparably with the state-of-the-art max pooling-based methods in object classification in Section 4.2 and significantly better than max pooling in mesh saliency in Section 4.5 .

3.3 View-based 2D saliency generation

The components elaborated in Sections 3.3-3.5 are only implemented in the deployment mode.

Once the CfS-CNN is trained end-to-end on a classification dataset, given an input mesh, we first use the trained CfS-CNN to predict its class. We then employ the CAM method proposed in [36] to compute a per-pixel saliency map $I(V_i)$ (known as "image-specific class saliency" in [36]) for all the pixels in the view V_i based on their influence on the predicted class. We select this particular CAM method for its efficiency and simplicity since it just requires a simple back-propagation with all the network parameters fixed. The 2D saliency map $I(V_i)$ can be interpreted as a measure of pixel importance with regard to the predicted classification of the mesh. $I(V_i)$ is further normalised to be within the interval of $[0, 1]$. There is no one-to-one correspondence between the pixels in V_i and the vertices of the mesh. We propose the following method to derive a 3D saliency map from a single 2D saliency map.

3.4 View-based 2D-to-3D saliency transfer

Note that the resolution of the CNN views is fixed (224×224 as required by the VGG net) no matter how many vertices the mesh contains. We can thus first generate the view-based 3D saliency maps for a simplified mesh and then compute such maps for the original mesh using point correspondence between the simplified mesh and the original one.

Inspired by the 2D-to-3D operations in Song *et al.* [26] and Kalogerakis *et al.* [29], we project the simplified mesh (containing 2500 faces in our implementation) at each of the N viewpoints and rescale the 2D projections to 224×224. Next, after cropping the rescaled 2D projections to remove the background pixels, we proposed a novel scheme to generate a view-based 3D saliency map from a view-based 2D saliency map. The view-based saliency of a 3D vertex m visible in the view V_i is calculated by considering the view-based saliency of the pixel closest to its 2D projection (i.e. the 2D-to-3D correspondence which associates a 2D pixel with a 3D vertex) and the local density of the vertices:

$$H_m(V_i) = \frac{\exp(1 - Z(m))}{\exp(1 - I_{(x,y)}(V_i))}. \tag{9}$$

Because density can be reflected by the distance between two neighbouring points, we introduce $Z(m)$ computed as the average of the normalised distances (i.e. within the interval of $[0, 1]$) between m and its 1-ring neighbours. (x, y) denotes the coordinates of the pixel in $I(V_i)$ closest to the 2D projection of m. The rationale behind Eq. (9) is that the 2D projection of a 3D vertex is more ambiguous and thus the 2D-to-3D correspondence is less reliable if the local density around the vertex is low. We thus use the term $\exp(1 - Z(m))$ as the numerator in Eq. (9) to reflect the reliability of the 2D-to-3D correspondence. We use $\exp(1 - I_{(x,y)}(V_i))$ as the denominator to define a positive relation between the view-based 2D saliency and the view-based 3D saliency. If a vertex is not visible to a viewpoint, its saliency with regard to that view is set to a constant β. Finally, the view-based saliency of a 3D vertex on the original mesh is computed by finding its closest point on the simplified mesh. Essentially, a view-based 3D saliency map $H_m(V_i)$ indicates the importance of a vertex m with regard to the predicted classification of the object, based on the information recorded in the view V_i. For simplicity, we write $H(V_i)$ as H_i in the rest of the paper.

3.5 Multi-view saliency synthesis

Each of the N 3D saliency maps can be interpreted as an attribute which encodes some information of the 3D object. Among many potential mathematical models of synthesizing such attributes, an intuitive one is a linear model

$$H = \sum_{i=1}^{N} w_i H_i \tag{10}$$

where w_i denotes the contribution of a view-based 3D saliency map H_i. As a weighting parameter, it reflects the importance of a view in the synthesis. Secord *et al.* [37] showed that such a linear model performed well when estimating the importance of views for various 3D objects. In their method, the linear model has to be learned since the weights of the attributes with regard to the importance of views were unknown. In [26], these weights were estimated via an MRF, which leads to an architecture that cannot be trained end-to-end. In our work, however, we simply use the output of the VS layer S as the weights since each of its elements already represents the learned saliency of a view:

$$H = \sum_{i=1}^{N} S_i H_i. \tag{11}$$

3.6 Implementation

Training. We first render the mesh representing a 3D object as 24 2D views as described in Section 3.1 using a standard OpenGL renderer with perspective projection mode. The strengths of the ambient light, the diffuse light and the specular reflection are set to 0.3, 0.6 and 0 respectively. We apply the light uniformly across each triangular face of the mesh (i.e. flat shading). Note that using different illumination models or shading coefficients does not affect our method due to the invariance of the learned convolutional filters to illumination changes, as observed in image-based CNNs. All of the 24 rendered views are then printed at 200 dpi, also in the OpenGL mode, and further resized to the resolution of 224×224. Then we feed these views into the CfS-CNN and train it through stochastic gradient descent. As mentioned in Section 3.2, the baseline VGG-19 network is pre-trained on ImageNet, and the CfS-CNN which contains the newly added VS and SP layers is fine-tuned on the ModelNet40 dataset. The learning rate is set to 5×10^{-4} initially. After 10 epochs, we reduce it to 10^{-4} and after 20 epochs, we further

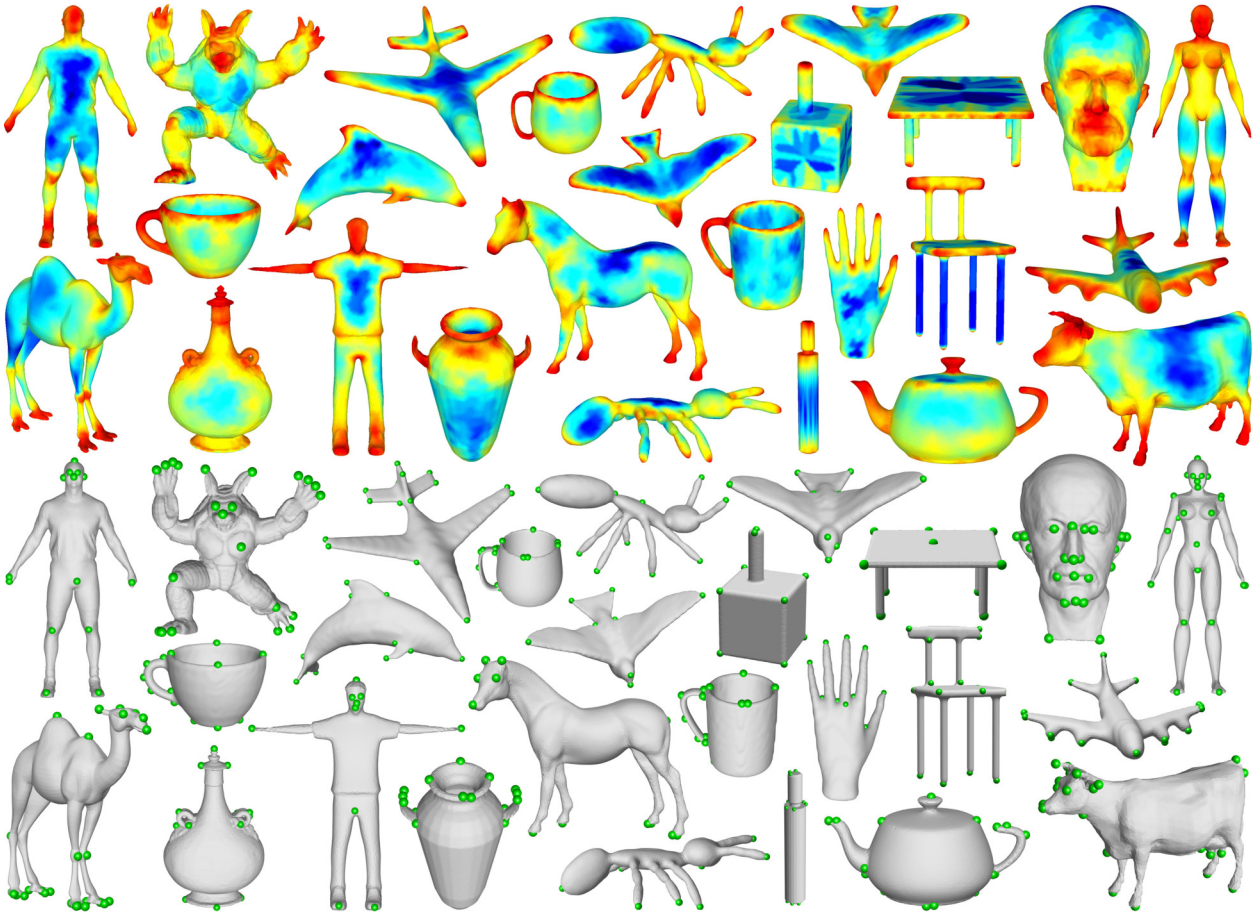


Fig. 2. A gallery of mesh saliency detected by the proposed CfS-CNN method (top half) with the human-picked interest points (Schelling points [5]) of the corresponding meshes (bottom half). Warmer colours show higher saliency.

reduce it to 10^{-5}. We observed that the CfS-CNN converged usually within 30 epochs.

Deployment. First, we again render the mesh as 24 views with the same rendering settings. Once the CfS-CNN is evaluated with the 24 views, we use the schemes described in Sections 3.3 and 3.4 to generate multiple view-based 3D saliency maps. When evaluating the CfS-CNN, we also record the output of the VS layer, which enables the synthesis of multiple view-based 3D saliency maps to finally output a single per-vertex saliency map using the scheme described in Section 3.5.

4 Experimental Results

4.1 Datasets and ground truth generation

We train the proposed 21-layer CfS-CNN on the Princeton ModelNet40 dataset [10] containing 12,311 shapes from 40 common categories. We use the same training and test split as in [10] where 80% of the 3D objects in each category are used for training and 20% are used for evaluating the classification performance of the CfS-CNN.

Since mesh saliency concerned in this paper aims at capturing what most people would classify as interesting regions in meshes, we use human-picked 3D interest points for evaluations. We test our method on the Schelling dataset [5] which provides human-selected 3D interest points (see Fig. 2) for a collection of 400 meshes belonging to 20 object categories. These meshes are all up oriented either by the method proposed in [38] or manually. To generate a per-vertex saliency map from the scattered interest points for quantitative evaluation, we employ a strategy widely used for evaluating image saliency methods: we project a Gaussian distribution on a mesh where each vertex is labeled by either 1 (representing interest point) or 0 (representing non-interest point) and vary the standard deviation to generate different versions of the ground truth saliency maps. When we evaluate our method on these ground truth maps, we essentially estimate whether it can detect saliency at different scales.

Note that human eye fixations could also be used as the ground truth for evaluating mesh saliency. However, human eye fixations are variant to view change, while we intend to produce a saliency map which maps each vertex to a fixed saliency value no matter how the viewpoint changes. Another reason for choosing the Schelling dataset is that it is much larger than existing eye fixation datasets. For example, the dataset proposed in [6] contains 32 meshes and the one proposed in [18] merely contains 5 meshes. As one of the aims is to demonstrate the generalisation capability of our approach, experiments on a larger dataset are more desired.

4.2 Classification results

Although CfS-CNN is primarily designed for weakly-supervised mesh saliency, ultimately it is trained to perform

TABLE 1
Classification results on the ModelNet40 dataset

Method	#Views	View Selection	Classification Accuracy
Su-MVCNN [27]	12	30° elevation	89.9%
Su-MVCNN-avg	12	30° elevation	86.5%
Qi-MVCNN [31]	20	uniform	89.7%
Qi-MVCNN-avg	20	uniform	86.4%
CfS-CNN	12	30° elevation	87.9%
CfS-CNN	24	±30° elevation	88.3%

3D object classification. It is thus interesting to evaluate its performance on the classification task. We compare CfS-CNN against two state-of-the-art 3D object classification methods [27], [31] and their invariants and show the results in Table 1. We pick them for comparison because they are also based on multiple 2D views and, trained on the ModelNet40 dataset using a baseline VGG-19 model pre-trained on ImageNet. Such similarity facilitates a straightforward observation of the effect of saliency over classification.

We find that our method does not significantly hurt the classification performance. Compared with the state-of-art methods [27], [31] specifically designed for classification, our method results in a small performance drop of $1-2\%$. However, it outperforms the variants of both competing methods where we replace the max-pooling scheme used in both of them for synthesizing information from multiple views with average pooling. Qi-MVCNN [31] is largely the same as Su-MVCNN [27] but using a different view selection scheme. Although it uses more views, it is still slightly outperformed by Su-MVCNN. Therefore, not all the views have positive contribution to the classification. So average pooling is generally outperformed by max pooling. The SP layer of our CfS-CNN essentially pools the views as a weighted average, which might reduce the contribution of those bad views but does not rule them out completely. And it seems that views with 30° elevation are generally good choices. Furthermore, that we use 24 views rather than 12 is not driven by the incremental boost of classification performance, but simply because we find that for some objects such as chairs and tables, a large portion of vertices cannot be covered by only 12 views with 30° elevation. In most cases, the 24-view setup guarantees that most vertices of a mesh are visible in at least one view, which facilitates the computation of their saliency.

4.3 Qualitative saliency results

Fig. 2 shows the saliency maps for a variety of 3D objects and the Schelling points [5] of the corresponding objects. One observation is that these saliency maps are highly consistent with the human-generated 3D interest points. Another observation is that objects of the same class tend to have analogous saliency distributions. Fig. 3 also demonstrates this observation where we compare our methods with other state-of-the-art ones. It can be seen that for the models of humans and quadrupeds, our method reliably detects features around the head or the facial region as salient, a behaviour consistent with the ground truth shown in Fig. 2. In comparison, other methods failed to do so. Our method also reliably detects hands and feet of humans or quadrupeds as salient even if the legs are occluded as shown in the first and the second models (Isis and Stanford Bunny) in Fig. 3. This is also desired according to the ground truth of the objects of the same classes shown in Fig. 2.

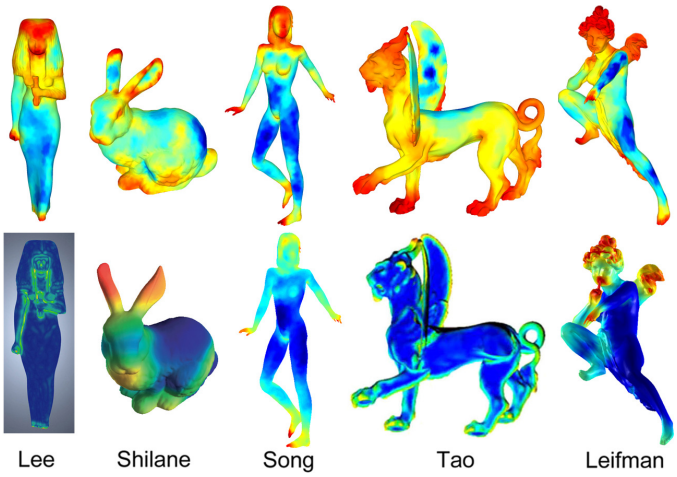


Fig. 3. Saliency detected by our method (top row) and the competing methods (bottom row) including: Lee [1], Shaline [2], Song [23], Tao [40] and Leifman [3].

4.4 Quantitative saliency results

To compare different methods for mesh saliency, we need to measure the similarity between the ground truth derived from humans and the saliency map produced by a competing method. In this work, the ground truth saliency maps are generated through applying Gaussian blurring to the Schelling points as mentioned in Section 4.1. Many metrics have been proposed for evaluating methods for image saliency. Bylinskii *et al.* [39] provided an analysis of 8 different metrics and their properties. According to their recommendations, we selected the Pearson's Linear Correlation Coefficient (CC), computed as

$$CC(H,T) = \frac{cov(H,T)}{\sigma_H \sigma_T} \tag{12}$$

where H and T are the saliency maps produced by a competing method and the ground truth respectively.

We also selected the area under the ROC curve (AUC) suggested in [6]. The ground truth maps are thresholded to be converted into binary maps (in our experiments we threshold to obtain M vertices considered as salient vertices and M is equal to the number of human-selected Schelling points on the 3D object). The saliency map produced by a competing method is then treated as a binary classifier of these salient vertices. The ROC curve represents the relationship between the probability of false positives and the probability of true positives and is obtained by varying the decision threshold on the saliency map. The AUC can then be used as a direct indicator of performance.

As mentioned in Section 3.4, for each viewpoint, the view-based saliency of the vertices invisible to it are set to a constant β. This can be interpreted as a view-based prior assigned to the invisible vertices. However, since it is difficult to create a likelihood model to estimate the representation of such a prior based on the posterior knowledge of the

TABLE 2
Performance on the Schelling dataset in terms of Pearson's Correlation Coefficient (CC). σ denotes the standard deviation of the Gaussian used to generate the pseudo ground truth for mesh saliency. B is the length of the diagonal of the bounding box of the mesh.

Method	$\sigma = 0.1B$	$\sigma = 0.12B$	$\sigma = 0.14B$	$\sigma = 0.16B$	$\sigma = 0.18B$	$\sigma = 0.2B$
Multi-scale Gaussian [1]	0.223	0.213	0.202	0.193	0.186	0.179
Admissible Diffusion Wavelets [41]	0.101	0.091	0.082	0.074	0.068	0.063
Spectral Processing [23]	0.324	0.322	0.313	0.301	0.293	0.284
Salient Regions [3]	0.437	0.421	0.402	0.376	0.360	0.340
Local-to-Global Saliency [25]	0.407	0.390	0.372	0.354	0.340	0.324
CfS-CNN (our method)	**0.455**	**0.457**	**0.454**	**0.447**	**0.439**	**0.427**

TABLE 3
Performance on the Schelling dataset in terms of area under ROC curve (AUC). σ denotes the standard deviation of the Gaussian used to generate the pseudo ground truth for mesh saliency. B is the length of the diagonal of the bounding box of the mesh.

Method	$\sigma = 0.1B$	$\sigma = 0.12B$	$\sigma = 0.14B$	$\sigma = 0.16B$	$\sigma = 0.18B$	$\sigma = 0.2B$
Multi-scale Gaussian [1]	0.700	0.701	0.707	0.705	0.700	0.696
Admissible Diffusion Wavelets [41]	0.777	0.757	0.762	0.755	0.745	0.738
Spectral Processing [23]	0.811	0.814	0.814	0.810	0.804	0.805
Salient Regions [3]	0.855	0.861	0.863	0.870	0.861	0.858
Local-to-Global Saliency [25]	0.856	0.859	0.871	0.867	0.861	0.859
CfS-CNN (our method)	**0.892**	**0.897**	**0.899**	**0.900**	**0.903**	**0.898**

TABLE 4
Evaluation of the effectiveness of the saliency channel using the Schelling dataset in terms of CC and AUC. σ is the standard deviation of the Gaussian used to generate the pseudo ground truth for mesh saliency. B is the length of the diagonal of the bounding box of the mesh.

Metrics	w/ saliency channel	w/o saliency channel
CC ($\sigma = 0.1B$)	0.455	0.416
CC ($\sigma = 0.16B$)	0.447	0.396
AUC ($\sigma = 0.1B$)	0.892	0.848
AUC ($\sigma = 0.16B$)	0.900	0.861

visible vertices, β is just set to a constant and its value is picked empirically. To implement a quantitative evaluation, we make 3 trials ($\beta = 0, 0.5$ and 1 respectively) and select the top-performing one for each 3D object although fine-tuning it will further improve the performance.

Tables 2 and 3 show the overall performance of a selection of state-of-the-art methods on the Schelling dataset in terms of CC and AUC. For CC, 1 represents perfect positive linear relation, 0 represents no relation and -1 represents perfect negative relation. For AUC, 1 represents a perfect classification while 0.5 represents a random one. Both performance metrics demonstrate that our method significantly outperforms all the competing methods. The results indicate that our method can generalise well since some object categories in the testing dataset were not observed during training. This might be because the baseline network of our CfS-CNN is a multi-view CNN pre-trained on ImageNet images from 1k categories although ModelNet40 merely contains 40 categories. This is clearly a benefit derived from the idea of weak supervision since saliency annotations across a large number of object categories are very expensive. It also benefits 3D scene saliency since typically a scene does not belong to any single object category.

4.5 The effect of the saliency channel of the CfS-CNN

To evaluate the effectiveness of the proposed saliency channel shown in Fig. 1, we carry out an ablation study where we compare the CfS-CNN with its ablated version. In the ablated version, the VS and the SP layers which compose the saliency channel are removed and we use the view-pooling layer (essentially a max-pooling layer) proposed in [27] to aggregate the outputs from multiple views. The ablated method thus computes saliency solely from the classification knowledge. For multi-view saliency synthesis, w_is in Eq. (11) are all set to 1 since S_i is not available in the ablated network. All the other components of our method remain the same. Tables 4 shows the result of the ablation study and the quantitative evaluation per class of the ablated method can be found in the supplementary material. We can see that the saliency channel significantly improves the performance of the neural network in terms of CC and AUC. It demonstrates that although the saliency channel leads to a small drop over classification performance, its effect on mesh saliency provides a considerable benefit.

4.6 How well the features learned for 3D object classification are transferred to mesh saliency?

As mentioned in Section 3.2, how well the features learned through a base task is transferred to a target task depends on how "close" the target task is to the base task [33]. While the seminal paper [33] provided an analysis on this issue from a network-specific perspective (i.e. how well features produced by a particular layer of a network transfer from one task to another) in the context of 2D image classification, we investigate this question from a task-specific perspective. We observed from the visual results in Fig. 2 that the objects of the same class tend to have similar saliency distributions. Fig. 4 shows the quantitative performance per class of our method through one AUC plot while more plots per class (including AUC and CC plots with varying σs and a Normalized Scanpath Saliency (NSS) plot) are available in the supplementary material. It can be seen that for the object classes with top performances such as Fish, Person and Quadruped, the salient features (such as head, hands and feet) are also important for classifying the objects. Although

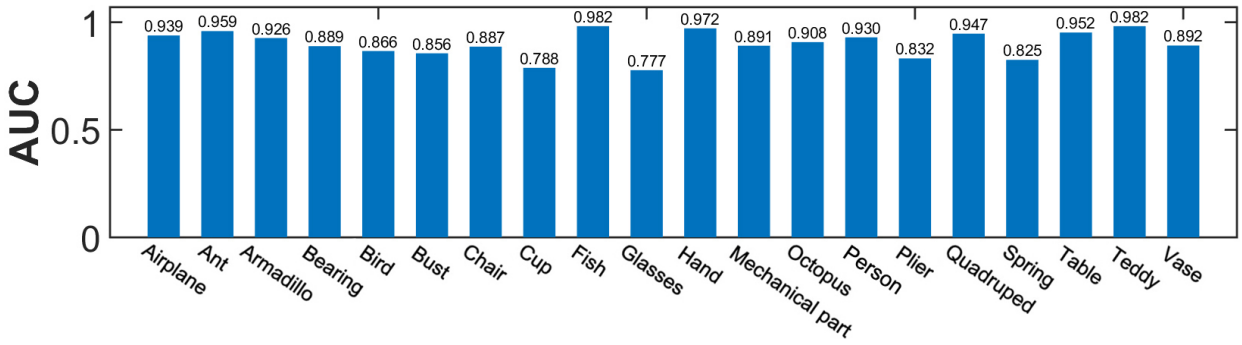


Fig. 4. AUC values per class for our method ($\sigma = 0.16B$).

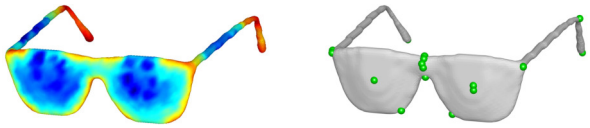


Fig. 5. A failure case of mesh saliency. Left: mesh saliency produced by our method; Right: the Schelling points marked by the human subjects.

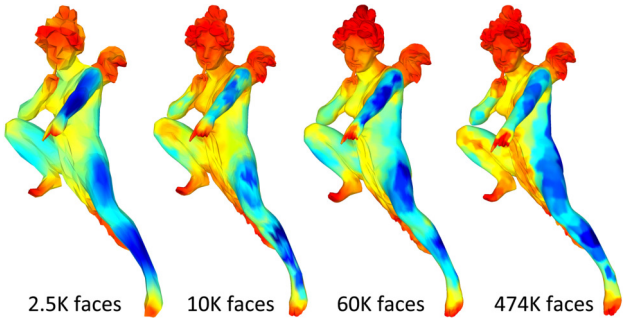


Fig. 6. Saliency maps of the Angel meshes containing significantly different numbers of triangle faces.

the cognitive mechanism of predicting visual saliency is quite complicated, human perception systems do have the capability of quickly recognising most objects without requiring a slow process of scrutiny but through focusing on just important features [42]. Hence, if the perceptually important salient features are consistent with those that are highly discriminative, the features learned for object classification will be transferred to mesh saliency very well. As we observed in Fig. 4, our method performs well in most classes with AUC higher than 0.8, demonstrating a good transfer of such features although this is not the case for the few object classes with lowest performances. For instance, some salient features of glasses such as the centres of the lens picked by human participants in the Schelling dataset as shown in Fig. 5 are not so discriminative since similar features (i.e. a region of planar/slightly curved surface) can be found in the objects of many other classes.

4.7 Consistency over different levels of simplification and efficiency

It is believed that human visual attention is not sensitive to the spatial resolutions since humans can quickly find the salient features without a slow process of scrutinising the details [42]. It is thus desirable that 3D meshes of the same object with different numbers of vertices and triangle faces should have highly consistent saliency maps. We thus carried out an experiment where the input mesh of Angel is subject to different levels of simplification. Fig. 6 shows the results and demonstrates that even if the number of faces of each mesh differs significantly, their estimated saliency maps are still consistent.

TABLE 5
Run times for computing the saliency of some meshes or scenes listed in the paper. Please refer to Table 1 in the supplementary material for the run times of all meshes and scenes listed in the paper.

Mesh or Scene	#Vertices	Run Time (sec.)
Armadillo (Fig. 2)	25.3K	26.04
Cavalry regiment (Fig. 7)	496.7K	137.41
Dining room (Fig. 10)	112.0K	53.97
Dolphin (Fig. 2)	7.6K	21.69
Feline (Fig. 3)	129.0K	51.73
Isis (Fig. 3)	187.6K	64.93
Knights (Fig. 7)	19.5K	45.59
Skateboarding (Fig. 8)	12.6K	39.94
Table (Fig. 2)	13.9K	23.03
Teapot (Fig. 2)	6.9K	21.90

Table 5 shows the run times of 10 models (selected with significantly varying numbers of vertices) listed in this work where we used a computer with an Intel i7-4790 3.6GHz CPU and 32GB RAM without any GPU acceleration. We further reported the run times of all meshes and scenes listed in this paper in Table 1 in the supplementary material. It can be seen that the run time of our method increases only slowly as the number of vertices of the input meshes increases. In general, our method is much faster than some competing approaches such as [22], [40].

5 3D SCENE SALIENCY AND ITS APPLICATIONS

Mesh saliency has been applied to best view selection and mesh simplification of a single object [1], [2], [3], [23]. For novelty purpose, we extend such applications to 3D scenes composed of multiple objects. In this section, we first show and analyze some scene saliency results. Then we show how to use scene saliency to detect the best views of a scene and simplify a scene through scene cropping.

The CfS-CNN is based on a multi-view setup. For a scene, each view encodes the information related to the global positional relationship of multiple objects in it. Thus our method can be directly used for scene saliency.

To analyse the behaviour of scene saliency, we specifically select some scenes composed of similar objects and show their saliency in Fig. 7. Compared with a single horse,

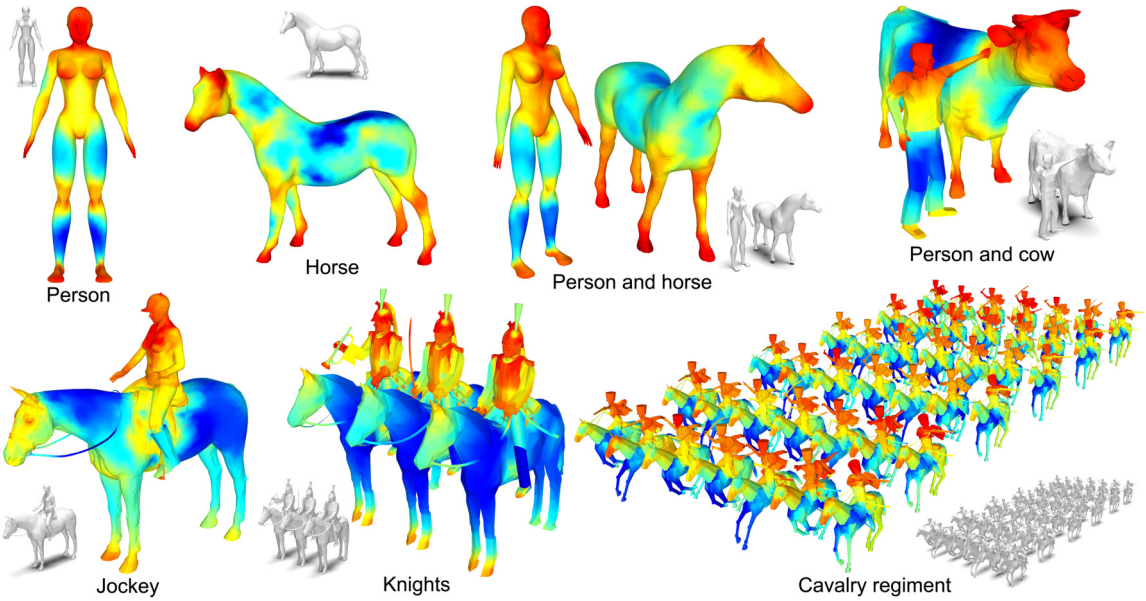


Fig. 7. Examples of mesh and mesh-based 3D scene saliency. Our method can produce saliency maps for not only a single mesh such as a person or a horse, but also 3D scenes containing multiple meshes such as a jockey riding a horse, a row of knights, or even a cavalry regiment.

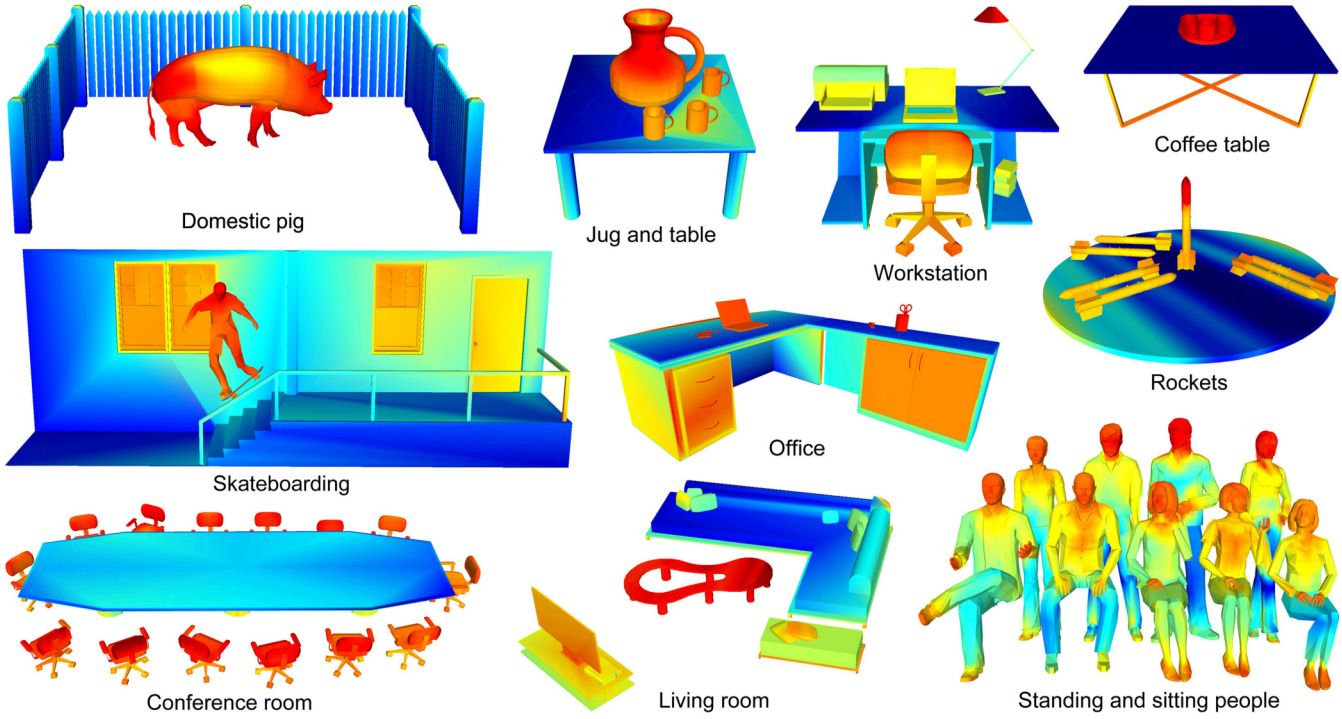


Fig. 8. Saliency of various 3D scenes

in the scenes of 'jockey', 'knights' and 'cavalry regiment', the saliency of the horses, particularly their feet is consistently suppressed due to the coexistence of the humans, which shows that scene saliency is not just a simple concatenation of the saliency of multiple meshes computed individually. However, in the 'person and horse' scene, the saliency of the horse is similar to that of a single horse. We have a similar observation on the 'person and cow' scene where the person and the cow also appear side by side. This means that the saliency of a scene depends on not only the objects it contains, but also the way they coexist.

We further check the 40-dimensional prediction vector output by the CfS-CNN. We found that, for the 'person and horse' scene, this vector has large elements corresponding to the classes of 'person' and 'quadruped'. We also have a similar finding for the 'person and cow' scene. But for the three riding scenes in Fig. 7, both of the two elements are relatively small. Consequently, features such as feet important for classifying a scene as 'person' or 'quadruped' are highlighted in the 'person and horse' and the 'person and cow' scenes but suppressed in the riding scenes. This means that recognising the person and the horse is easier in the

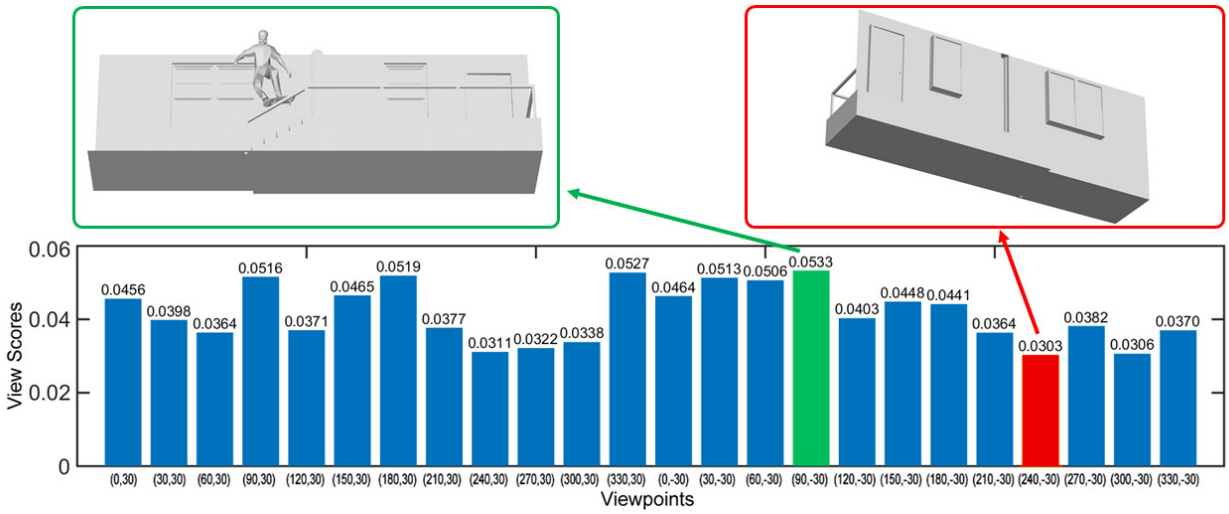


Fig. 9. The best and the worst views of the 'skateboarding' scene shown in Fig. 8. The view within the green outline is the best view of the 3D scene while the view within the red outline is the worst view. In the view scores–viewpoints plot, each viewpoint is denoted by a pair $(azimuth, elevation)$ and the bars corresponding to the largest and the smallest view scores are shown in green and red respectively.

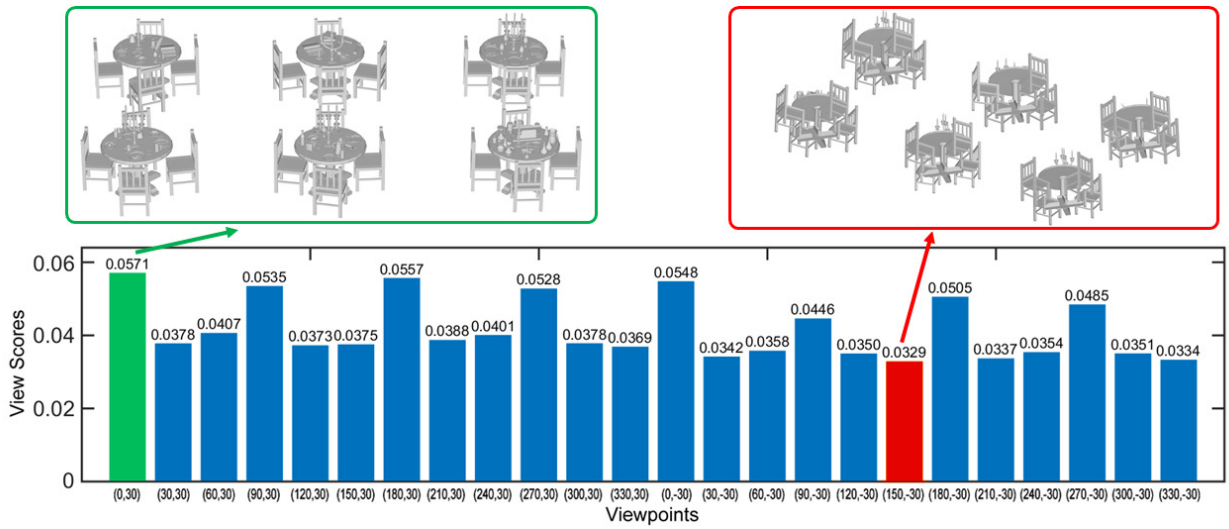


Fig. 10. The best and the worst views of a dining room scene (see Fig. 35 in the supplementary material for the visualised saliency of the scene).

'person and horse' scene than in the riding scenes. We infer that this is because the individual contour of each of the two objects is well preserved in some views of the 'person and horse' scene, but not in the riding scenes due to the different ways that they coexist. Note that contours are important for humans to recognise objects, particularly when colour information is not available. We also observed that in the scenes 'person and horse' and 'person and cow', the feet and legs of the horse and the cow are more salient than those of the persons. This is because they are very important features for classifying horse and cow (as quadruped) but not equally important for person. The head and the hands are more important features for classifying person.

Fig. 8 shows the saliency of some other scenes. More visual results on scene saliency are available in the supplementary material. Again, we observe that whether an object is presented independently or in a scene has a great impact on its saliency. For example, in Fig. 2, we observed that the handle of a mug is usually more salient than the other regions in it but in the scenes 'jug and table' and 'coffee table' in Fig. 8, each mug in general is salient such that its handle is as salient as the other regions in it. In Fig. 2, when the table and chair appear alone there are variations in the saliency of their components, but in the scenes 'conference room' and 'workstation' where they appear jointly, saliency changes greatly: the saliency of the table as a whole is suppressed with many originally salient local features such as some sharp edges and corners becoming not salient while each chair, as a whole becomes salient. Similarly, a person has his or her own salient features (e.g, head, hands, feet) if presented individually (see Fig. 2) or in some scenes (e.g. the 'knights' scene in Fig. 7, the 'standing and sitting people' scene in Fig. 8 and Fig. 33 in the supplementary material). But in some other scenes (e.g. the 'skateboarding' scene in Fig. 8 and Fig. 34 in the supplementary material) where most objects tend to be recognised as background by humans, the person as a whole is detected as salient.

Different from the well-established research on depth-based scene understanding, mesh-based scene understanding is free from some vision biases such as centre bias and

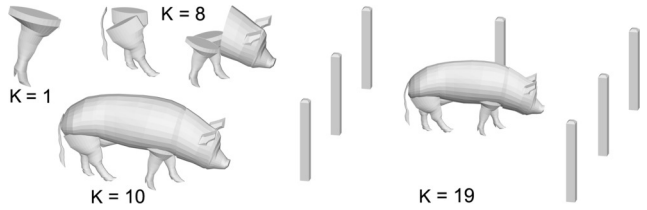


Fig. 11. Saliency-guided cropping of the 'domestic pig' scene

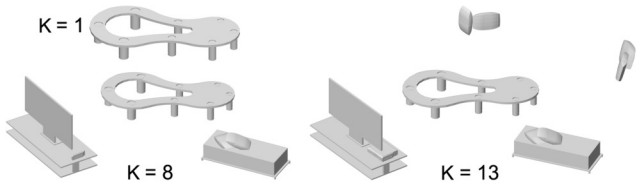


Fig. 12. Saliency-guided cropping of the 'living room' scene

viewing bias [39]. Also, a 3D mesh is usually more informative and less ambiguous than a depth image. Certainly, in some cases, it is expensive to obtain a complete mesh representation of a scene. So mesh-based scene saliency can be a good complement to the depth-based one. To prove this claim, we showcase two applications that mesh-based scene saliency can support but the depth-based one cannot.

5.1 Application 1: Best view selection of 3D scenes

Researchers [37], [43] in computer vision and graphics have explored the question of what are good views of a 3D object. A related question derived from it is how to select the best view of a 3D scene containing multiple objects where each object typically has its own best view when appearing independently. In many cases, this problem is well defined and can be answered with sufficient consistency and confidence.

We propose to use scene saliency for the best view selection of 3D scenes. The best view V_b of a scene is found by $V_b = \arg\max_i L$ where the view score L_i is calculated as

$$L_i = \frac{S_i \sum_m H_m(V_i)}{\sum_j \left(S_j \sum_m H_m(V_j) \right)}. \quad (13)$$

$H_m(V_i)$ denotes the view-based 3D saliency of a vertex m. S is the the output of the VS layer of the CfS-CNN where each of its entry S_i represents the saliency of the view V_i. Therefore, according to Eq. (13), the best view should be salient in comparison with other views and meanwhile contain a large number of salient vertices. Similarly, the worst view V_w is found by $V_w = \arg\min_i L$.

Figs. 9 and 10 show the best and the worst views of two scenes found by our method. According to the view scores shown in Fig. 9, the view with $(azimuth = 90, elevation = 30)$ which is the symmetric view of the detected best view $(90, -30)$ but looking downward is also a good view. Similarly, in Fig. 10, the best view is $(0, 30)$ while the view $(180, 30)$ corresponding to the viewpoint from the other side of the scene is also a good view. These findings are consistent with human perception. Also, the best view of a scene is not necessarily the best view of each individual object in it. For example, in Fig. 9, the best view of the entire scene is not that of the stairs or the handrail. In Fig. 10, the best view of the scene might not be that of some chairs.

5.2 Application 2: Saliency-guided scene cropping

Image cropping is one of the most basic processes of image manipulation. It is the user-controlled removal of the outer peripheral areas from an image. We extend this idea to 3D. To crop a scene, we introduce a saliency-guided approach where the peripheral objects are found by computing the object-level saliency.

In detail, given a 3D scene, we first find all of the disconnected objects it contains by checking the connectivity stored in the face matrix. The saliency of a particular object is then computed as the mean saliency of all vertices it contains. Finally, we rank the objects appearing in the scene based on their object-level saliency. To perform scene cropping, the users just need to indicate an integer K representing the number of objects they want to preserve from the scene. The cropped scene is then composed of the top N salient objects from the original scene.

Figs. 11 and 12 show the cropping results of two 3D scenes. Their saliency maps can be found in Fig. 8. When K is small, only the most salient objects are preserved. Gradually increasing K adds more and more objects which are less and less salient. In Fig. 11, setting $K = 1$ extracts the leg and the foot of the pig since they are the most important features to classify a pig as quadruped. In Fig. 12, less salient objects such as pillows are preserved for a large K.

6 Conclusions and Future Work

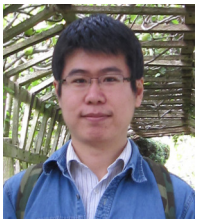
We proposed a novel deep neural network for learning a vertex-level annotation, mesh saliency, from an object-level annotation, class membership of 3D objects. The network, namely CfS-CNN, is trained end-to-end in a weakly supervised manner. Our work reveals that the knowledge of 3D objects learned through a sufficiently deep neural network trained on classification datasets may be transferable to another 3D object understanding task as long as proper heuristics related to the particular task are introduced to guide the feature detection process. We believe that this finding is of broad interest since it provides a promising way to handle potentially challenging 3D object understanding problems hindered by the lack of large-scale fully and consistently annotated training datasets. Therefore, motivated by the performance of this work, one future work is to adapt the proposed approach by considering new heuristics to other 3D object understanding tasks under a certain transfer learning framework.

Also, the current work is not completely invariant to object orientation due to the view-based nature of the CfS-CNN. So another area for future work is a preprocessing scheme for intelligently generating a small number of self-adaptive views of 3D objects and scenes in the hope that the method can be more stable and efficient.

References

[1] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 24, no. 3, pp. 659–666, 2005.
[2] P. Shilane and T. Funkhouser, "Distinctive regions of 3d surfaces," *ACM Trans. Graph.*, vol. 26, no. 2, p. 7, 2007.

[3] G. Leifman, E. Shtrom, and A. Tal, "Surface regions of interest for viewpoint selection," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 38, no. 12, pp. 2544–2556, 2016.
[4] M. Lau, K. Dev, W. Shi, J. Dorsey, and H. Rushmeier, "Tactile mesh saliency," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 35, no. 4, 2016.
[5] X. Chen, A. Saparov, B. Pang, and T. Funkhouser, "Schelling points on 3d surface meshes," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, p. 29, 2012.
[6] G. Lavou, F. Cordier, H. Seo, and C. Larabi, "Visual attention for rendered 3d shapes," *Comput. Graph. Forum (Proc. Eurographics)*, pp. 414–421, 2018.
[7] Z. Shu, X. Shen, S. Xin, Q. Chang, J. Feng, L. Kavan, and L. Liu, "Scribble based 3d shape segmentation via weakly-supervised learning," *IEEE Trans. Vis. Comput. Graph*, 2019.
[8] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph. (Proc. SIGGRAPH Aisa)*, vol. 35, no. 6, p. 210, 2016.
[9] H. Dutagaci, C. Cheung, and A. Godil, "Evaluation of 3d interest point detection techniques via human-generated ground truth," *Vis. Compu.*, vol. 28, pp. 901–917, 2012.
[10] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proc. CVPR*, 2015, pp. 1912–1920.
[11] M. Savva, A. X. Chang, and P. Hanrahan, "Semantically-enriched 3D models for common-sense knowledge," in *Proc. CVPR Workshops*, 2015, pp. 24–31.
[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. CVPR*, 2017, pp. 652–660.
[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NIPS*, 2017, pp. 5099–5108.
[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. ICLR*, 2013.
[15] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016, pp. 3844–3852.
[16] I. Kostrikov, J. Bruna, D. Panozzo, and D. Zorin, "Surface networks," in *Proc. CVPR*, 2018.
[17] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 130–150, 2006.
[18] Y. Kim, A. Varshney, D. Jacobs, and F. Guimbretiere, "Mesh saliency and human eye fixations," *ACM Trans. Appl. Percept.*, vol. 7, no. 2, pp. 12:1–12:13, 2010.
[19] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cogn. Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.
[20] J. M. Wolfe, "Guided search 2.0 a revised model of visual search," *Psychonomic Bulletin & Review*, vol. 1, no. 2, pp. 202–238, 1994.
[21] C. Koch and T. Poggio, "Predicting the visual world: silence is golden," *Nat. Neurosci.*, vol. 2, pp. 9–10, 1999.
[22] J. Wu, X. Shen, W. Zhu, and L. Liu, "Mesh saliency with global rarity," *Graph. Models*, vol. 46, pp. 264–274, 2013.
[23] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, "Mesh saliency via spectral processing," *ACM Trans. on Graph.*, vol. 33, no. 1, 2014.
[24] S. Wang, N. Li, S. Li, Z. Luo, Z. Su, and H. Qin, "Multi-scale mesh saliency based on low-rank and sparse analysis in shape feature space," *Comput. Aided Geom. Des.*, vol. 35, pp. 206–214, 2015.
[25] R. Song, Y. Liu, R. Martin, and K. R. Echavarria, "Local-to-global mesh saliency," *Vis. Comput.*, vol. 34, no. 3, pp. 323–336, 2018.
[26] R. Song, Y. Liu, and P. L. Rosin, "Distinction of 3D objects and scenes via classification network and markov random field," *IEEE Trans. Vis. Comput. Graph.*, 2018.
[27] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proc. ICCV*, 2015, pp. 945–953.
[28] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3d shape retrieval," *Neurocomput.*, vol. 204, pp. 41–50, 2016.
[29] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. CVPR*, vol. 1, no. 2, 2017, p. 8.
[30] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer, "Learning local shape descriptors from part correspondences with multiview convolutional networks," *ACM Trans. Graph.*, vol. 37, no. 1, p. 6, 2018.
[31] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proc. CVPR*, 2016, pp. 5648–5656.
[32] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Proc. SGP*, 2009, pp. 1383–1392.
[33] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 3320–3328.
[34] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015.
[35] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC*, 2014.
[36] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: visualising image classification models and saliency mps," in *ICLR Workshop*, 2014.
[37] A. Secord, J. Lu, A. Finkelstein, M. Singh, and A. Nealen, "Perceptual models of viewpoint preference," *ACM Trans. Graph.*, vol. 30, no. 5, p. 109, 2011.
[38] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer, "Upright orientation of man-made objects," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 27, no. 3, p. 42, 2008.
[39] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.
[40] P. Tao, J. Cao, S. Li, X. Liu, and L. Liu, "Mesh saliency via ranking unsalient patches in a descriptor space," *Computers & Graphics*, vol. 46, pp. 264–274, 2015.
[41] T. Hou and H. Qin, "Admissible diffusion wavelets and their applications in space-frequency processing," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 1, pp. 3–15, 2013.
[42] J. Intriligator and P. Cavanagh, "The spatial resolution of visual attention," *Cognitive psychology*, vol. 43, no. 3, pp. 171–216, 2001.
[43] H. Dutagaci, C. P. Cheung, and A. Godil, "A benchmark for best view selection of 3d objects," in *Proc. ACM workshop on 3DOR*, 2010, pp. 45–50.



Dr Ran Song is a senior lecturer at the University of Brighton, UK since 2014. He received his Ph.D. in 2009 from the University of York, UK and his first degree in 2005 from Shandong University, China. He has published more than 40 papers in peer-reviewed international conferences proceedings and journals. His research interests lie in 3D shape analysis and 3D visual perception.

Prof Yonghuai Liu is a Professor at Edge Hill University, UK since 2018. Before his current post, he was a senior lecturer at Aberystwyth University, UK. He is currently associate editor and an editorial board member for a number of international journals, including Pattern Recognition Letters and Neurocomputing. He has published three books and more than 180 papers in international conference proceedings and journals. His primary research interests lie in 3D computer vision. He is a senior member of IEEE and Fellow of Higher Education Academy of United Kingdom.

Prof Paul L. Rosin is a Professor at the School of Computer Science and Informatics, Cardiff University. Previous posts include lecturer at Brunel University London, UK, research scientist at the Institute for Remote Sensing Applications, Joint Research Centre, Ispra, Italy, and lecturer at Curtin University of Technology, Perth, Australia. His research interests include the representation, segmentation, and early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximation algorithms, etc., medical and biological image analysis, mesh processing, non-photorealistic rendering and the analysis of shape in art and architecture.