Local-to-global mesh saliency

Song, Ran; Liu, Yonghuai; Martin, Ralph R.; Rodriguez Echavarria, Karina

Published in:
Visual Computer

DOI:
10.1007/s00371-016-1334-9

Publication date:
2016

Citation for published version (APA):
Song, R., Liu, Y., Martin, R. R., & Rodriguez Echavarria, K. (2016). Local-to-global mesh saliency. *Visual Computer*, 323-336. https://doi.org/10.1007/s00371-016-1334-9

Noname manuscript No.
(will be inserted by the editor)

Local-to-Global Mesh Saliency

Ran Song · Yonghuai Liu · Ralph R. Martin · Karina Rodriguez Echavarria

Received: date / Accepted: date

Abstract As a measure of regional importance in agreement with human perception of 3D shape, mesh saliency should be based on local geometric information within a mesh but more than that. Recent research has shown that global consideration has a significant role in mesh saliency. This paper proposes a local-to-global framework for computing mesh saliency where we offer novel solutions to solve three inherent problems: (1) an algorithm based on statistic Laplacian which does not only compute local saliency, but also facilitates the later computation of global saliency; (2) a local-to-global method based on pooling and global distinctness to compute global saliency; (3) a framework to integrate local and global saliency. Experiments demonstrate that our approach can effectively detect salient features consistent with human perceptual interest. We also provide comparisons to existing state-of-the-art methods for mesh saliency and show improved results produced by our method.

Keywords Mesh saliency · Laplacian · Global distinctness

1 Introduction

In computer graphics, human perception can play an important role in evaluating the results of many tasks such as feature detection, mesh segmentation, shape recognition, shape retrieval, shape-based morphing, mesh simplification and so on. In these tasks, usually the ground truth is influenced by human perception and thus subjective evaluation is used when assessing the success of such methods. To the best of our knowledge, the concept of mesh saliency was first introduced by Lee et al. [19]. It is a computational measure of regional importance on a mesh in accordance with human perception. The hope is to close the gap between the humanly-judged ground truth and computational results.

Human perception of 3D shape is mainly visual. Consequently, computational saliency attempts to predict human visual attention which typically is affected by visual stimuli caused by local geometric features. However, human perception is also influenced by other global and higher-level perceptual cues. For this reason, although some success has been achieved, satisfactory computation of mesh saliency remains a hard problem, and even state-of-the-art methods have shortcomings.

1.1 Related work

Early work on saliency detection for 3D meshed surfaces was largely inspired by related research on 2D images. Guy and Medioni [10] took a scheme for computing a saliency map based on local edges in a 2D image, and applied it to 3D data; the goal was to smoothly interpolate sparse and noisy 3D data to obtain dense surface information. In [36], the saliency of a 3D dynamic scene was computed based on a coarsely rendered 2D projection, employing the 2D method in [13] for adjustment. This saliency-based strategy led to accelerated and improved illumination computation in pre-rendered animations. Mantiuk *et al* [22] used a 2D saliency algorithm to guide real-time MPEG compression of an animated 3D scene. In general, estimating saliency in 2D projections of meshes does not sufficiently take into account

Ran Song
School of Computing, Engineering and Mathematics, University of Brighton, UK
E-mail: r.song@brighton.ac.uk

Yonghuai Liu
Department of Computer Science, Aberystwyth University, UK

Ralph R. Martin
School of Computer Science and Informatics, Cardiff University, UK

Karina Rodriguez Echavarria
School of Computing, Engineering and Mathematics, University of Brighton, UK

depth information within the original data, which, as mentioned in [12], is a key stimulus for human perception of a static scene.

Extensive work has considered computing saliency directly from 3D structure, mainly driven by locally salient geometric features. [25] proposed a multiscale method to extract local line-like features using *surface variation* as a saliency measure, based on eigenvalues of local covariance matrix. [19] computed mesh saliency using a centre-surround operator on Gaussian-weighted curvatures and provided methods to incorporate saliency into mesh simplification and view selection. In [9], salient geometric features based on curvatures were introduced to improve part-in-whole matching. [29] developed an approach to compute the distinctive regions of a 3D surface, and applied it to shape matching, icon generation and mesh simplification. [2] proposed a method for detecting and matching salient points from multiview meshes, where saliency is determined by generating a multiscale representation of the mesh. [7] focused on viewpoint selection and developed a method to compute view-based mesh saliency using mutual information between polygons. [11] proposed the admissible diffusion wavelets on meshes and calculated mesh saliency based on it.

In later work, global cues has been incorporated into various computational models of mesh saliency. [21] proposed an algorithm for detecting surface regions which are distinct both locally and globally where the global consideration is whether the object is 'limb-like' or not. Then they explored how to select viewpoints based on these regions of interest. [3] proposed a regression model to predict mesh saliency based on learning from data collected in a large-scale user study. It also raised the issue that global properties such as segment centeredness and proximity to a symmetry axis are required to explain more subtle salient features. [35] proposed an approach for detecting mesh saliency based on the observation that salient features are both locally prominent and globally rare. [30] analysed the log-Laplacian spectrum of meshes and presented a method where both local geometric cues and global information corresponding to the low-frequency end of the spectrum are considered. Saliency is then determined by transferring information from the spectral domain back to the spatial domain.

1.2 Our work

Although global considerations improve saliency detection, significant differences between computational saliency and human-provided ground truth still remain. As noted in [30], this is because it is difficult to incorporate global perceptual cues into saliency detection. Most mesh saliency methods implicitly assume that human visual perception is driven bottom-up by local, expectation-free reactions to salient stimuli. However, research [5] in neuroscience has demonstrated that vision can also be guided by top-down, expectation-dependent, anticipatory mechanisms. These mechanisms are mostly associated with global cues.

In this work, we move from local geometric features to global cues. One important motivation of our work is the observation that locally salient features may be of low global saliency when analogous features appear many times around the meshed surface, which means that globally they are less distinct. Such observation has been studied in neuroscience and psychophysics, and arguably regarded as the most important global cue influencing human visual attention [34, 16]. For example, [34] concluded that human visual system is sensitive to less frequent features and suppresses frequently occurring features. We accordingly propose a novel framework which detects both local and global saliency and also integrates them. We first compute local mesh saliency based on the newly proposed *stochastic Laplacian*. Then we create a set of feature points by pooling the local saliency map. Next, we develop a method to compute the *global distinctness* of the feature points as their global saliency and finally incorporate it into a saliency adjustment scheme to output a per-point saliency map.

Our approach is presented in detail in Sections 2, 3, 4 and then evaluated in 5. We finally conclude in Section 6.

The main contributions of this paper are threefold:

1. We present a novel algorithm for computing local mesh saliency. It captures salient features at multiple scales very efficiently since compared to previous methods, it does not require a spectral eigendecomposition of the Laplacian [30], or a spatial nearest neighbour search [19] to analyse multiscale neighbourhoods.
2. Based on the local saliency, we propose a novel method to efficiently compute a set of feature points specifically designed for performing a global saliency adjustment.
3. We develop a novel scheme for global saliency adjustment where the global saliency representing the global distinctness of features is integrated with the local mesh saliency to output the final saliency map.

2 Local mesh saliency via stochastic Laplacian

As a local-to-global scheme for mesh saliency, we start from the local cues. In this section, we first describe our algorithm and then analyse it and explain how it works. To compute local saliency, we first calculate the Laplacian of mesh and convert it to the newly proposed *stochastic Laplacian*. Then in each iteration, the stochastic Laplacian which encodes multiscale information is updated by matrix multiplication and numerical operations. Once the termination condition is satisfied, the local saliency map is produced through the aggregation of the difference maps of stochastic Laplacians

which essentially integrates the information about saliency computed at multiple local scales.

The Laplacian is the most widely used local operator for mesh processing. The Laplacian matrix of a mesh is based on the discretisation of a continuous Laplacian (e.g. the Laplace-Beltrami operator) defined for a smooth manifold, using some weighted sum of adjacent vertices [6]. If a mesh M contains m vertices $p_1, \ldots, p_m$, in its simplest form, the Laplacian matrix is:

$$L = D - A \tag{1}$$

where A is the adjacency matrix, given by

$$A(i,j) = \begin{cases} 1 & \text{if } p_i \text{ and } p_j \text{ are neighbours} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

and D is a diagonal matrix in which D_{ii} is the degree of p_i.

We use two further steps after computing the Laplacian. Firsly, we compute $\widehat{L}$ composed of the absolute values of the elements of the Laplacian: $\widehat{L}_{ij} = |L_{ij}|$. Secondly, we normalise $\widehat{L}$ so that the sum of each row is 1: $\mathcal{L} = \left[\widehat{L}\right]_r$ where $[\cdot]_r$ denotes row-based normalisation. These two simple steps are very important since they turn the Laplacian into the *stochastic Laplacian*. Although the Laplacian has a precise definition, we arguably use the term *stochastic Laplacian* to denote a matrix (e.g., $\mathcal{L}$) which is factually a *stochastic matrix* and inspired by the original Laplacian. The stochastic Laplacian provides an effective way to encode multiscale surface geometry.

Our algorithm is given in Algorithm 1. It is based on iterative updates of the stochastic Laplacian so that a series of stochastic Laplacians are generated to encode multiscale features. It can be implemented more easily and efficiently than previous methods requiring distance-based neighbourhood identification [19], eigendecomposition of the Laplacian [30] or the design and dilation of wavelets [11]. In Algorithm 1, each iteration generates a new stochastic Laplacian $F^{(h)}$ and the corresponding difference map $D^{(h-1)}$ of two stochastic Laplacians at adjacent scales.

Stochastic Laplacian. We wish to determine a multiscale neighbourhood of each vertex. The row based normalisation in each iteration guarantees that $F^{(h)}$ is always a stochastic matrix, so describes the transitions of a Markov chain. Given a stochastic matrix $\mathcal{L}$, $\mathcal{L}_{ij}$ denotes the probability of a one-step transition. Therefore, the h-th stochastic Laplacian produced via a series of $h-1$ updates gives the h-step transition probability between different vertices.

Since the stochastic Laplacian matrix is sparse, most transitions are prohibited (have transition probability zero); only transitions within a neighbourhood are permitted as the Laplacian depends on the adjacency matrix. Hence, the h-step transition actually defines a connected $\mathcal{H}$-ring ($\mathcal{H} = \sum_{k=1}^{h-1}(h-k)$) neighbourhood while all other connectivities and paths are prohibited. The Markov chain property allows us to determine vertex neighbourhoods just from the probabilities, so the complexity of neighbourhood identification is linear.

Algorithm 1: Local mesh saliency

Data: A mesh M containing m vertices
Result: A local saliency map I
begin
  Compute $\widehat{L}$ and then the stochastic Laplacian $\mathcal{L}$;
  Initialise $F^{(1)} = \mathcal{L}$ and $H = 2$; `// Superscripts with and without parentheses denote sequential numbers and exponents respectively.`
  for $h \leftarrow 2$ **to** *MaxIter* **do**
    $F^{(h)} = \arctan(h^{1.5} F^{(h-1)} \widehat{L}^{h-1})$;
    $\mathcal{D}^{(h-1)} = \left|F^{(h)} - F^{(h-1)}\right|$;
    $\mathcal{F} = \left[F^{(h)}\right]_r$;
    $F^{(h)} = \mathcal{F}$;
    Calculate the difference map as $D^{(h-1)} = \sum_r \mathcal{D}^{(h-1)}$ where $\sum_r$ denotes row-based summation;
    if $N_Z/m > 0.1m$ where N_Z is the number of nonzero elements in $F^{(h)}$ **then**
      $H = h$; break;
  $\mathcal{I} = \left[\sum_{h=2}^{H} D^{(h-1)}\right]$ where $[\cdot]$ denotes normalisation;
  $I = \text{arccot}(\mathcal{I})$

Multiscale property. Although the saliency computed based on stochastic Laplacian is local, it is of multiple local scales. The local saliency of a vertex is not only decided by its curvature which represents the local geometry at the smallest scale but also the local geometry at larger scales, which depends on larger neighbourhoods of different sizes. To effectively capture salient features at different local scales, the sizes of these neighbourhoods should significantly differ. In Algorithm 1, the core process for multiscale neighbourhood identification is the matrix multiplication $F^{(h-1)}\widehat{L}^{h-1}$, which provides neighbourhood expansion. As h increases, the stochastic matrix $F^{(h)}$ becomes less and less sparse. Apart from the maximum number of iterations $MaxIter$, the termination condition is that the average neighbourhood contains more than 10% of the mesh vertices since we reasonably assume that the maximum size of a salient feature is no larger. For example, for the girl mesh shown in Fig. 1, Algorithm 1 terminates at the 7th scale and thus $H = 7$. From scales $h = 2$ to 7, it detects potential features within neighbourhoods containing various numbers of vertices: 20, 70, 212, 551, 1235, 2443 on average. Fig. 1 illustrates the multiscale representation of the girl mesh using the proposed method. It can be seen that when $h = 2$, small-scale local features such as the tip of nose can be detected. When $h = 4$, medium-scale features such as the entire nose can be detected. When $h = 7$, the algorithm captures some large-scale features such as the entire face and the ribbon which is now, as a whole, a protrusion of the en-

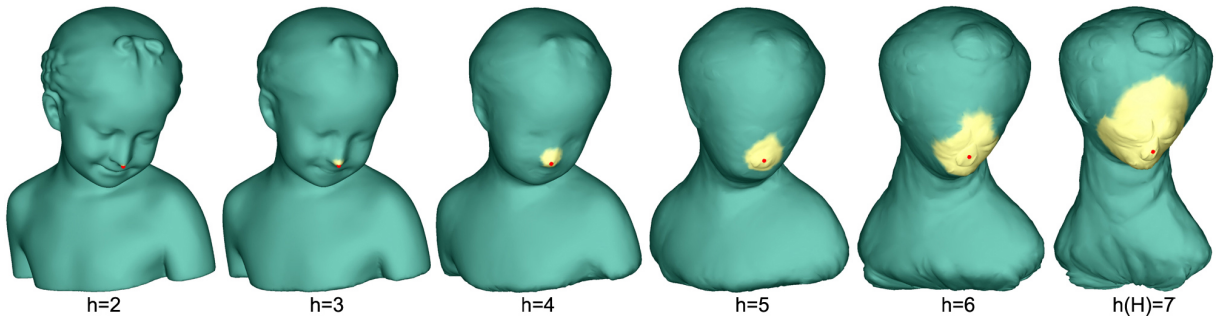

Fig. 1: Multiscale representation of a girl mesh via the stochastic Laplacian. Yellow denotes the neighbourhood of the centred red point at each scale. Note that the *feature neighbourhood* of the red point defined at the 4th scale captures the entire nose as a salient feature. It can be seen that when the stochastic Laplacian is repeatedly applied over the mesh, it produces a 'shrinkage' effect similar to repeated Laplacian smoothing on the mseh.

tire mesh. Empirically, we select the neighbourhood defined by the median scale as the most useful neighbourhood of a vertex and call it the *feature neighbourhood* of that vertex. This concept is used later for local-to-global pooling and global saliency adjustment.

Saliency map. The penultimate line of Algorithm 1 employs the popular idea of aggregating the difference maps $D^{(h-1)}$ at multiple scales [13,19,11,30]. Note that $D^{(h-1)}$ will also be used as a multiscale feature descriptor for computing global distinctness in Section 4. Due to the row-based normalisation in each iteration, as h increases, the entries of the stochastic Laplacian $F^{(h)}$ become smaller, as it becomes less sparse. When h is very large, the entries of the corresponding difference map will be very small, so during aggregation, the difference maps encoding large-scale features will make disproportionately small contributions. Thus in the algorithm, when we update the stochastic Laplacian $F^{(h)}$, based on empirical findings, we use a factor $h^{1.5}$ to enhance the weights of the large scale maps. To enhance salient regions and suppress non-salient regions, Itti et al. [13] proposed a non-maximum suppression approach while Song et al. [30] simply employed a logarithm operation. After testing these approaches, based on our empirical findings, we eventually use an $\arctan$ operation to enhance the differences between salient and non-salient regions in the saliency map, and the final step $I = \text{arccot}(\mathcal{I})$ also further enhances the salient regions.

Relation to heat diffusion. Local features on a mesh are usually interpreted as local extrema of a scalar field of local mesh saliency, with the help of multiscale representation. The generating equation of a multiscale representation is the linear heat diffusion equation [17]. So the multiscale representation of a manifold can be obtained as the solution to a heat diffusion process [4]. The fundamental solution of the heat diffusion equation is known as heat kernel and it has quite a few nice properties [31]. In our work, the multiscale representation as shown in Fig. 1 is generated through iteratively applying the stochastic Laplacian. Given the set of mesh vextices P, the multiscale representation in Fig. 1 can be expressed as $F^{(h)}P$. Heat diffusion and stochastic Laplacian are both effective ways to generate a multiscale representation for computing local mesh saliency. However, our method performs better than heat diffusion in the perspective of saliency computation according to the results of quantitative comparisons shown in Figs. 9 and 10.

Relation to wavelets. One important step in Algorithm 1 is the computation of the difference of two stochastic Laplacians $\mathcal{D}^{(h-1)} = \left|F^{(h)} - F^{(h-1)}\right|$. A graph Laplacian applied on a 3D meshed surface can be viewed as a low-pass filter [33] which removes the high-frequency details of the mesh. In our work, the step of calculating the absolute value of such a discrete Laplacian makes it more Gaussian-like (as negative entries no longer exist). Note that the multidimensional generalisation of a Mexican hat wavelet is the Laplacian of Gaussian function which is usually approximated by the difference of Gaussians function. Therefore, the differences of stochastic Laplacians at adjacent scales behave like wavelets which have powerful localisation property in both frequency and space. In many graphics applications such as saliency detection, mesh segmentation and interest point detection, a nice localisation property of the operator is often attractive due to the need of seeking local features and/or filtering local geometry, while retaining gross shape/structure globally. From this point of view, Algorithm 1 essentially creates a band-pass filter with desired localisation property for highlighting locally salient features. Although our method for computing local saliency can be understood from the perspective of wavelets, as we shall show in Section 5.2, when combining local and global saliency, our method outperforms a recent method [11] based only on wavelets.

3 Local-to-global saliency pooling

As defined in [19], saliency is a measure of regional importance of a mesh. Such a definition suggests that an efficient

computation can be achieved by only considering a small set of feature points which represent local regions. More importantly, performing a saliency adjustment which relies on sparse feature points can effectively guarantee regional consistency of saliency (see Fig. 8 for the experimental validation of this claim). Hence, it is desired to create a sparse set of feature points for performing saliency adjustment. However, due to the following reasons, we tested uniform sampling, thresholding and local maxima extraction based on local saliency or other feature descriptors such as HKS [31] and found none of them generated desired results.

First, the main motivation of applying a global saliency adjustment is that globally salient features may be of low local saliency and vice versa. Thus the set should include some feature points with low local saliency. Second, saliency or other feature descriptors such as HKS are often not so discriminative for points very close to each other. Since global distinctness is associated with the similarity of points defined by their descriptors (see Section 4), a globally salient feature point could be incorrectly regarded as not globally salient as some of its neighbouring points with similar descriptors are also included in the set. Therefore, the set should be composed of sparse feature points spatially far away from each other. Third, the feature points are spatially sparse and each of them is essentially a representative of a large local region. It is thus desired that each point is locally the best representative (e.g., the centre) of the corresponding region. Fourth, since saliency is regional rather than point-wise, it is desired to adjust the saliency of vertices within the neighbourhood of a feature as a whole. However, the sizes of features are highly various. It is not wise to find the local maxima of neighbourhoods of features in some descriptor space since finding the optimal neighbourhood defining every feature is very challenging. State-of-the-art mesh segmentation methods might work, but they are not desired here considering efficiency and complexity.

Based on the above considerations, we propose a local-to-global pooling scheme to solve this problem. Pooling is used in some biologically inspired neural network, such as Convolutional Neural Network (CNN), as an efficient and effective way to extract global or higher-level information from local features. To move from local salient features to global saliency, we use two types of pooling, max-pooling and average-pooling where the former formulates a nonlinear move and the latter corresponds to a linear extraction.

Max-pooling. In CNN, one important pooling operation is max-pooling. It acts as a form of nonlinear downsampling to reduce the amount of computation needed by upper layers. As illustrated in Fig. 2, we employ max-pooling to extract local saliency maxima in this work. In image processing, max-pooling is typically done using a square 2×2 or 3×3 window. As a 3D surface mesh lacks a regular lattice structure, it is necessary to define a valid neighbourhood for the max-pooling operation. Note that the stochastic Laplacian already provides neighbourhood identification at different scales. The neighbourhood of a vertex i at the hth scale can be found by extracting the indices of the nonzero elements in the ith row of $F^{(h)}$. We use the *feature neighbourhood* defined above for each vertex for max-pooling. Then max-pooling finds the maximum in a feature neighbourhood.

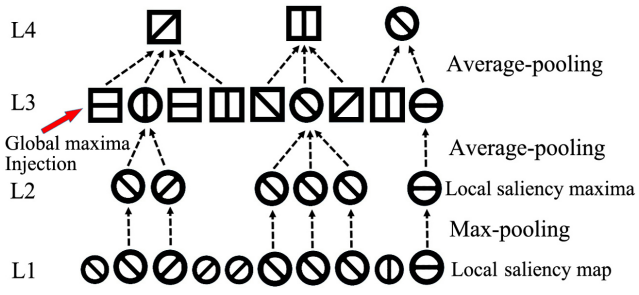

Fig. 2: Local-to-global pooling: circles denote local saliency and squares denote global maxima; the lines in various directions inside them stand for the variety among them.

Average-pooling. Another widely-used pooling operation in CNN is average-pooling which computes the average value of a particular feature over a region of a layer. In computer vision, it is an effective way to encode a large number of image features into a compact image representation. In our model, the average-pooling is a kind of low frequency filtering and thus has the capability of representing the main saliency in the lower layer. We simply employ k-means clustering to perform average-pooling. The details including the input and output to each pooling operation, and the setting of the number of clusters k, is explained shortly.

In total, we do 3 rounds of pooling as shown in Fig. 2. Max-pooling based on the local saliency map is done first, which extracts a set of *locally salient* vertices as the second layer. Then using these locally salient vertices as the input, the average-pooling further reduces the number of points extracted through max-pooling by 90% (k for k-means clustering is set to 10% of the number of locally salient vertices). We replace each cluster centroid produced by k-means clustering by the closest vertex from the set of locally salient vertices. These closest vertices are added to the third layer, and give a sparse representation of *local* mesh features. In the third layer, we incorporate the global maxima which are the top 50% vertices in terms of local saliency. The third round of pooling used to give the fourth layer is again an average-pooling using the vertices in the third layer as input. Since humans have difficulty paying attention to too many events simultaneously [15, 23], we set the number of clusters in the k-means algorithm this time to n_f (set to 100 in our experiments), no matter how large the input mesh is. Again we find the closest vertices to the cluster centroids from the in-

corporated set to form the fourth layer. These n_f vertices are considered to be the *feature points* of the mesh. The following global saliency adjustment is mainly based upon them.

4 Global saliency adjustment

We propose the concept of *global distinctness* to adjust (suppress or enhance) the local saliency map in a global manner. To avoid ambiguity, it is worth noting that our concept of global distinctness is completely different from the *distinctive features* proposed by Shilane and Funkhouser [29]. The global distinctness of a feature point reflects how distinctive it is compared to other feature points while the distinctive features in [29] are the features which can potentially distinguish a shape from objects in a different class. Certainly, there are other global cues which potentially play important roles for mesh saliency. Although some research [34,16] in neuroscience shows that the global distinctness could be the most dominant cue for attracting human visual attention, we still provide a discussion in the final section of the paper and also show some failed cases of our method caused by only considering the global distinctness.

Global distinctness for mesh saliency is inspired by the observation that humans are usually not interested in repeated features no matter whether they are locally prominent but tend to pay attention to globally distinctive and unique features [34,16]. For example, if there is only one red pepper in a stack of green peppers, it will catch all of our attention. But if there are several red peppers in a stack of green peppers, our visual attention will be dispersed although these red peppers are still locally salient. In the two cases, the distinctness can be computed in the color space. For 3D meshes, geometry is the only cue to compute saliency. For instance, Fig. 3 shows the result of our saliency detection on a bumpy sphere. It can be seen that the bumps are usually detected as salient features. In our work, we propose the global distinctness computed in a space formed by the multiscale feature descriptor D^H where each of its columns is a normalised difference map $D^{(h-1)}, h = 2, 3, ...H$ (see Section 2 and Algorithm 1).

4.1 Global distinctness computation

Given a feature point i represented by D_i^H in an $\mathbb{R}^{H-1}$ feature space (i.e., D_i^H is the row corresponding to the ith feature point in D^H), we measure its pairwise distinctness to another feature point j represented by D_j^H as the $H-1$ dimensional Euclidean distance between them

$$d_{ij} = \left\| D_i^H - D_j^H \right\|, \quad i \text{ and } j = 1, 2, \cdots, n_f. \tag{3}$$

To measure the global distinctness of a particular feature point, we construct a matrix X where each entry X_{ij} is the pairwise distinctness d_{ij}. Note that X can be regarded as the weighted adjacency matrix of a graph where every pair of feature points is connected and the length of the edge connecting i and j is determined by the pairwise distinctness d_{ij}. So X encodes the information about how feature points are distinct from each other. We then define the global distinctness as the centrality, a measure of the global influence of a node in a graph. By the Perron-Frobenius theorem [27], X has a unique largest eigenvalue and its corresponding eigenvector V has strictly positive components. Accoding to [24], the ith component of V gives the centrality score of the point i in the graph.

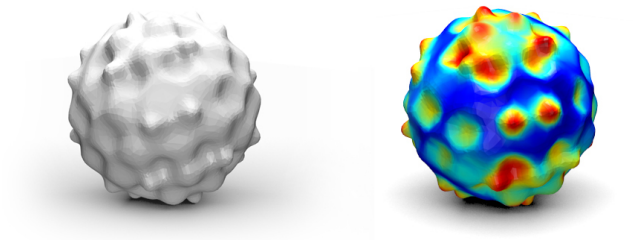

Fig. 3: Our result of saliency detection on a bumpy sphere. The mesh of bumpy sphere is courtesy of the AIM@SHAPE repository.

The above method can also be understood from a perspective of global distinctness maximization. It is a popular idea in mesh or image saliency to perform a global saliency adjustment (enhancement and/or suppression) to make mesh or image regions visually distinct ([13,19]). This can be formulated as a distinctness maximization problem as below

$$\begin{aligned} \arg\max G(S) &= \sum_i S_i \sum_j S_j X_{ij}, \\ s.t. \quad & S \in \mathbb{R}^+ \text{ and } \|S\| = 1 \end{aligned} \tag{4}$$

where the global distinctness S can be understood as weights assigned to local regions. A large S_i means that the point i is globally distinct and thus its pairwise distinctness X_{ij} has a large weighted impact in the overall distinctness $G(S)$. Since human cannot pay attention to too many events simultaneously [15,23], S is subject to $\|S\| = 1$. Eq. (4) can be written as

$$\arg\max G(S) = S^T X S, \ s.t. \ S \in \mathbb{R}^+ \text{ and } \|S\| = 1. \tag{5}$$

Since X is a symmetric real matrix, it is Hermitian. Thus Eq. (5) suggests that $G(S)$ is its Rayleigh quotient. The upper bound of the Rayleigh quotient is the largest eigenvalue of X and can be reached when S is equal to its corresponding eigenvector V. Therefore, as the solution to Eq. (4), the global distinctness $S = V$ maximises the overall distinctness $G(S)$ and essentially suggests whether a local distinction is globally important or not. We then shift V to have a zero mean so that a positive value means enhancement and

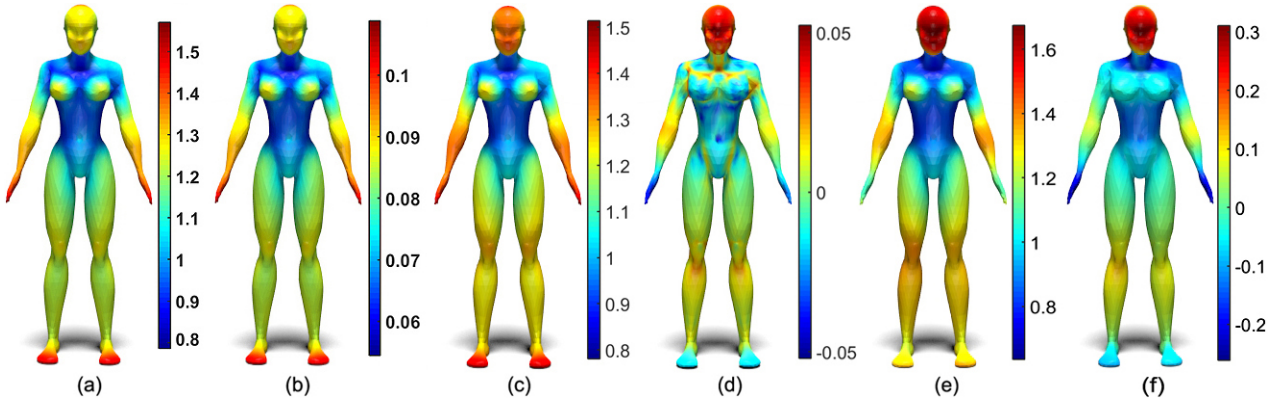

Fig. 4: Global saliency ajustment where warm colours denote high saliency. (a) The local saliency map I; (b) The output saliency map with the popular nonlinear suppression method [13,19] applied; (c) The final saliency map $\mathcal{S}$ with the global saliency adjustment applied where $\alpha = 0.005$; (d) The difference map ($Diff = (\mathcal{S}-I)/I$) between the saliency map shown in (c) and the local saliency map; (e) The final saliency map with the global saliency adjustment applied where $\alpha = 0.03$; (f) The difference map between the saliency map shown in (e) and the local saliency map.

a negative one means suppression. This gives a clue to perform a global saliency adjustment for all mesh vertices.

4.2 Global saliency adjustment based on global distinctness

S only assigns the global distinctness scores to the n_f feature points. To propagate it to each mesh vertex, an interpolation scheme is needed. Although some other interpolation schemes might also work, considering that we have already computed the *feature neighbourhood* of each feature point, we simply calculate the global distinctness of any point p_i as a Gaussian of that of the feature point it associates with.

$$S(p_i) = \frac{S(p_k)\exp(-\|p_i - p_k\|^2/(c \cdot B))}{(\sum_{p_i \in \mathcal{N}} \|p_i - p_k\| / |\mathcal{N}|)^2} \quad (6)$$

where p_i is a mesh vertex within the *feature neighbourhood* $\mathcal{N}$ of a feature point p_k and $|\mathcal{N}|$ denotes the number of vertices in $\mathcal{N}$. $S(p_k)$ is the global distinctness of p_k computed using the method described above. $c \cdot B$ is a normalisation constant where B is the length of the diagonal of the bounding box of the mesh and we set $c = 0.01$ in our experiments. To fuse different saliency cues, linear summation [13] and point-wise multiplication (also known as the matrix Hadamard product) [18] are two popular approaches. We perform the global saliency adjustment for each mesh vertex p as a linear sum

$$\mathcal{S}(p) = I(p) + \alpha S(p) \quad (7)$$

where $I(p)$ is the local salieny map and α is a weighting parameter which balances $S(p)$ and $I(p)$ numerically. It is possible that a vertex does not belong to the neighbourhood of any feature point. In this case its global distinctness $S(p)$ is 0, which means no global adjustment is applied on it. For a vertex within the neighbourhoods of several feature points, its global distinctness is the distance weighted average of the Gaussians computed by Eq. (6).

Figure 4 shows the effect of our global saliency adjustment. In the local saliency map shown in (a), the head of the human model is not that salient compared with the feet and hands. This is a common problem for many other mesh saliency methods [19,30,32] where the tip of a long and thin protrusion has disproportionally high saliency. However, the research in [3] shows that the head or facial region of human should also be salient. Compared to the popular nonlinear suppression [13,19] which visually does not make a significant change as shown in (b), our global saliency adjustment significantly enhances the head and suppresses the feet and hands as shown in (c)–(f).

5 Experimental results

In this section, we show experimental results produced by our method and perform both qualitative and quantitative evaluations.

Fig. 4 shows the impact of the local-to-global weighting parameter α. It can be seen that in (e) and (f), a large α can overturn local saliency such as the hands and feet of the human model while a small α will make the global saliency adjustment pointless. Usually, the larger the sheer size of the mesh (thus the smaller the global distinctness according to Eq. (6)), the larger the α. In our experiments, we set $\alpha = 0.005$ for the 400 SHREC 2007 watertight meshes as their sizes do not vary too much. We do not remove α from our algorithm by first normalising input meshes to have a unit size since in some cases it has an advantage of flexibility. It can be set with the prior knowledge and produce various

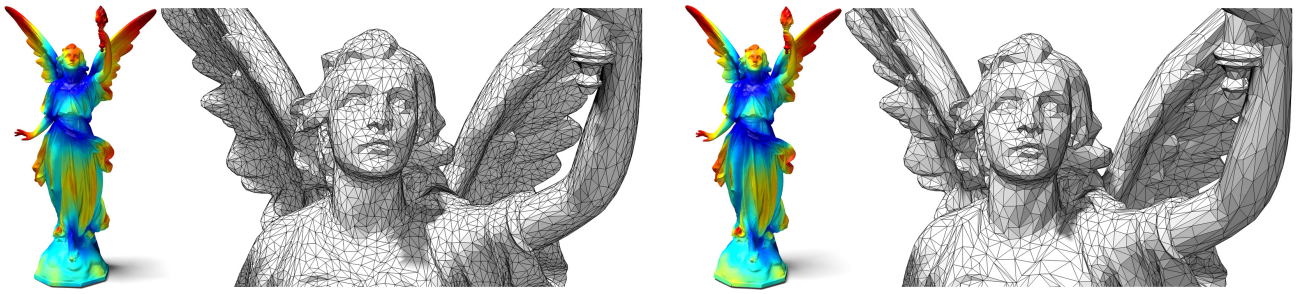

Fig. 5: Our method is insensitive to mesh simplification. Left: the saliency of the Lucy model with 60K triangles; Right: the saliency of the Lucy model with 30K triangles. The model is courtesy of the Stanford 3D Scanning Repository.

saliency as required by applications. For instance, in mesh simplification a smaller α might be desired for preserving local details while in the application of selecting best views, a larger α is typically suitable since the view should show globally distinct features.

Fig. 5 demonstrates that our method is not sensitive to mesh simplification, which is consistent with the definition of mesh saliency (i.e., saliency is regional rather than point-wise as proposed in [19]). This also suggests that our method is insensitive to small local noise, varying sampling and triangulation changes although our local saliency detection is based on Laplacian.

We employ the graph Laplacian other than the widely used cotangent for computing the stocastic Laplacian because we find that it benefits the detection of salient features. Fig. 6 shows the saliency maps produced by using cotangent Laplacian. The largest angle in an ill-shaped triangle has a negative cotangent but with a large absolute value. Since we need to compute the absolute values of the elements of the Laplacian to transfer it to a stochastic matrix, it can be observed that the cotangent Laplacian leads to high saliency values on ill-shaped triangles in Fig. 6. Our method is more meaningful in terms of according with salient features since usually most meshes retain more vertices around features (e.g., shape extremities). We also observed that the cotangent Laplacian is not so efficient as the graph Laplacian.

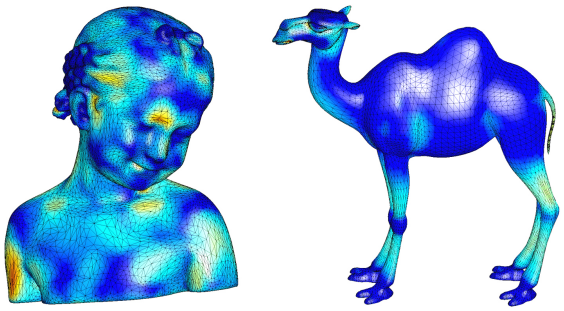

Fig. 6: The cotangent Laplacian is sensitive to ill-shaped triangles when computing mesh saliency where warm colour denotes high saliency.

5.1 Qualitative evaluation

We tested our method on the watertight models of SHREC 2007 containing the same 400 meshes as used in [3]. Fig. 7 shows a gallery of our saliency detection results produced by using the same parameter setting together with corresponding pseudo ground truths provided by [3]. As can be seen, our results are largely consistent with the ground truths.

We next compare our method with the state of the arts. Fig. 8 compares our results with those from [3, 32, 19, 21, 30]. The OutClass regression approach [3] fails to mark certain important features such as the eyes of the girl mesh. It also undesirably highlights the bottom edges of the girl bust and the neck of the Igea model. Similar appearance can be observed on the saliency maps produced by [32]. In general, it seems that their method prefers some local sharp edges such as the bottom edges of the girl bust and the neck of the Igea, the wings of the bird, the belly of the dolphin and the contour of the teddy. In comparison, our method makes a better balance between local sharp features and globally salient features. For example, the head of the human model is not locally sharp and thus is not captured by [32]. But it is highlighted by our method since it is globally important. Since our results are less influenced by local changes in curvature when they happen frequently, our method achieves a better regional consistency and detects large and continuous salient and non-salient regions. For instance, the facial regions of the girl bust and the Igea are considered salient by our algorithm, whereas saliency is indicated in a rather disjointed manner by [19]. Our method detects saliency in a more coherent manner. Compared with the method proposed in [21], our method better localise the eyes of the girl and the Igea. Note that when looking at scenes that include faces, humans have a fundamental bias to consider eyes [14, 26]. We also compares our results with those of [30]. For the chair model, the competing method detects asymmetric saliency for the front and back legs while such asymmetries do not exist in the ground truth data (a similar chair can be found in Fig. 7). The teddy bear shows more inter-

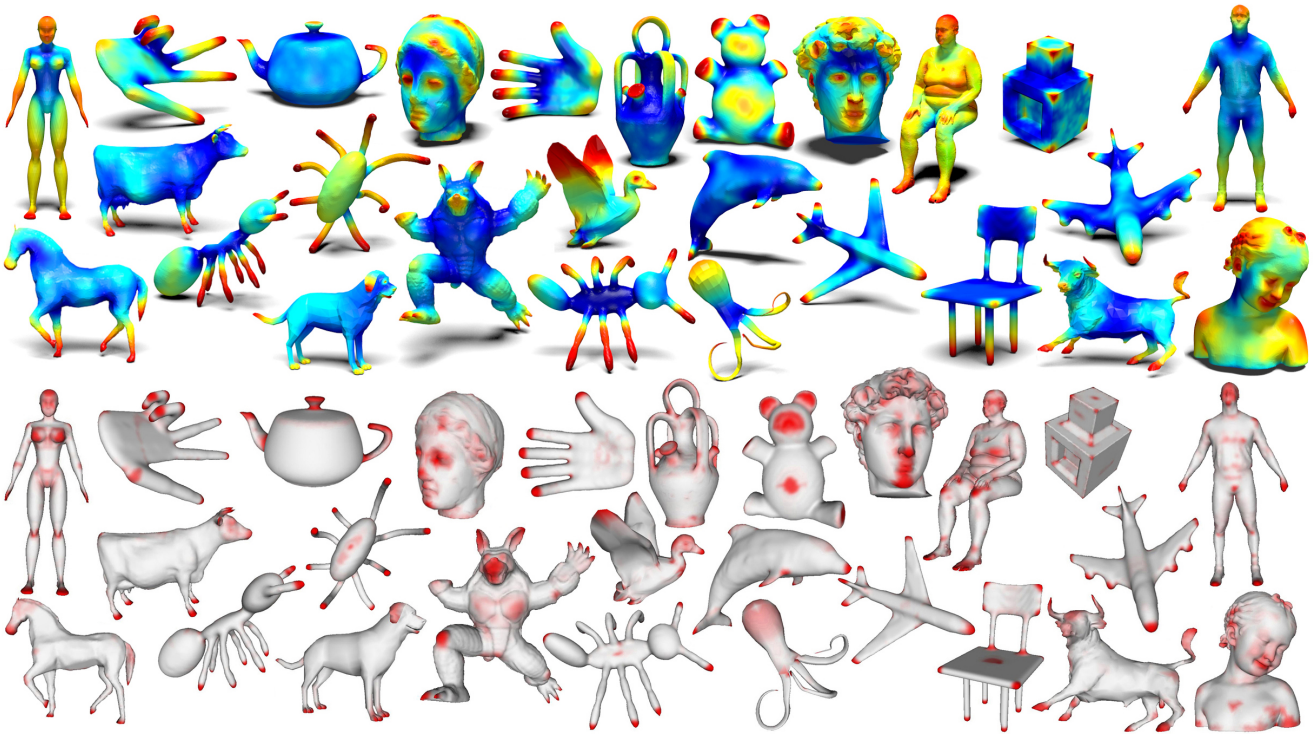

Fig. 7: A gallery of mesh saliency where warm color denotes high saliency. Above: results computed by our method. Below: corresponding pseudo ground truths provided in [3]. These models are courtesy of the Watertight Models of SHREC 2007.

esting results. Our method detects the central regions of the face and the body which are consistent with the ground truth (a similar teddy bear can be found in the middle of Fig. 7) while the competing method completely misses them. And the HKS-based method [31] only captures shape local features or some long protrusions while missing some globally important regions such as the facial regions of the girl and the heads of the horse and the human.

5.2 Quantitative evaluation

We perform quantitative evaluation on the 400 SHREC 2007 watertight models where the ground truths are the Schelling distributions provided by [3].

Based on the results of our work and previous methods, we observed that saliency detection methods seemingly favour some particular classes of objects. Also, due to the varying number of vertices that each mesh contains, a per-mesh error metric could be biased. Thus instead of a per-mesh saliency error which is less reliable and could be highly inconsistent, our evaluation is performed by calculating the *per-class Saliency Error* defined as below:

$$SE_C = \frac{1}{N_C} \sum_{n=1}^{N_C} |\mathcal{S}(n) - \mathcal{S}_{gt}(n)|, \quad C = 1, 2, ..., 20 \qquad (8)$$

where C represents one of the 20 object classes and N_C denotes total number of vertices of all 20 meshes in the class C. $\mathcal{S}(n)$ denotes the normalised saliency value of vertex n computed by a competing method and $\mathcal{S}_{gt}$ is the normalised Schelling distribution value of vertex n regarded as its ground truth saliency.

Although *Saliency Error* is quite intuitive, to make our evaluation less biased, we employ another popular measure which is statistically independent of *Saliency Error*. We compare the competing methods via *Similarity*, a popular metric for evaluating 2D-image saliency detectors [1]. This measure is also called Histogram Intersection [20]. Given two normalised saliency maps $\mathcal{S}_1(n)$ and $\mathcal{S}_2(n)$ containing the same number of vertices (or pixels for 2D-images) N, their Similarity, or Histogram Intersection is computed as

$$SL(\mathcal{S}_1(n), \mathcal{S}_2(n)) = \sum_{n=1}^{N} \min(\mathcal{S}_1(n), \mathcal{S}_2(n)). \qquad (9)$$

The larger the value of the Similarity, the more similar the two maps are deemed to be. So we compute the Similarity between the saliency map of a mesh computed by a competing method and the ground truth saliency map of that mesh. Then to deliver the *per-class Similarity* (with regard to the ground truth), we calculate the mean Similarity for the 20 meshes within the same class.

Ablation study. In order to examine the importance of the global saliency adjustment and demonstrate that it is the

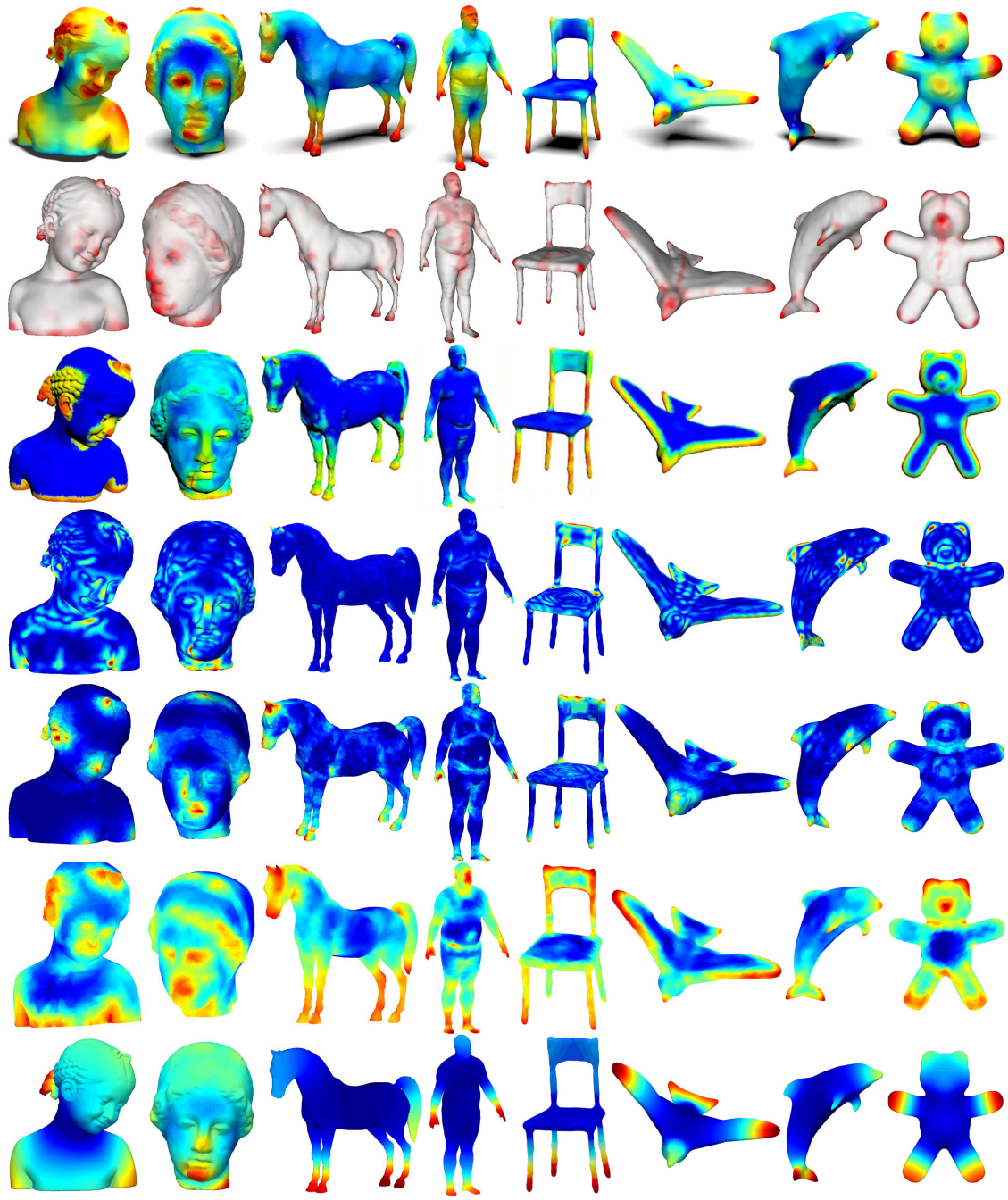

Fig. 8: Saliency detected by our method (first row) and the competing methods including (from the second row to the seventh row) [3], [32], [19], [21], [30] and [31]. Warm colors denote high saliency. These models are courtesy of the Watertight Models of SHREC 2007.

vital factor causing significant improvement, we conduct an ablation study where we ablate the global saliency term in Eq. (7) and only retain the local mesh saliency $I(p)$. We then regard the local mesh saliency as an independent method and evaluate it against other competing methods as well in Figs. 9 and 10.

Figs. 9 and 10 show the comparisons of several state-of-the-art saliency detection methods based on Saliency Error and Similarity with regard to the ground truth data. One intuitive finding is that the two graphs roughly complement one another: the peaks in Fig. 9 roughly correspond to the troughs in Fig. 10 and vice versa. This means that the two metrics generate consistent evaluations. Second, we can see

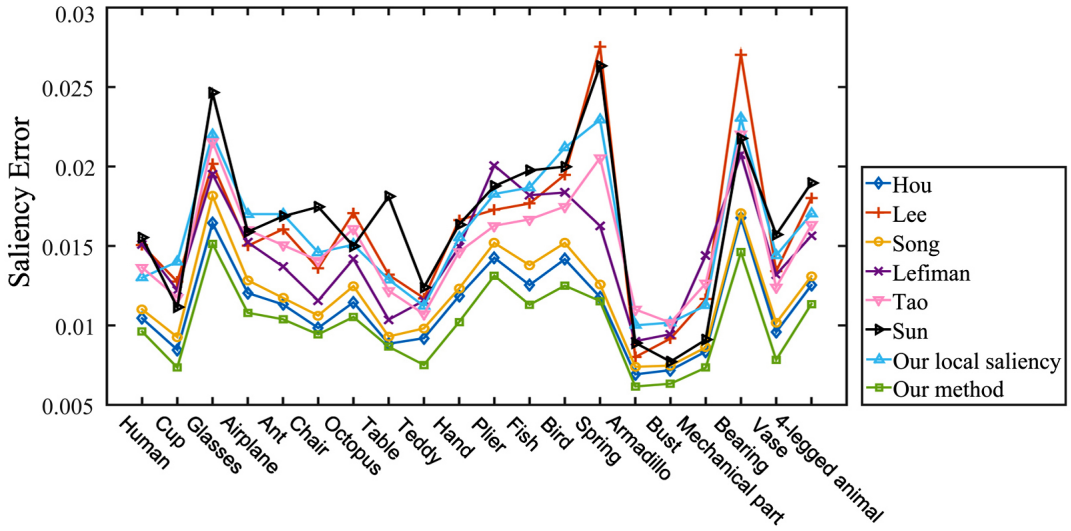

Fig. 9: Evaluation of different methods via Saliency Error: Hou [11], Lee [19], Song [30], Lefiman [21], Tao [32], Sun [31], our local mesh saliency (without global saliency adjustment) and our method (with global saliency adjustment).

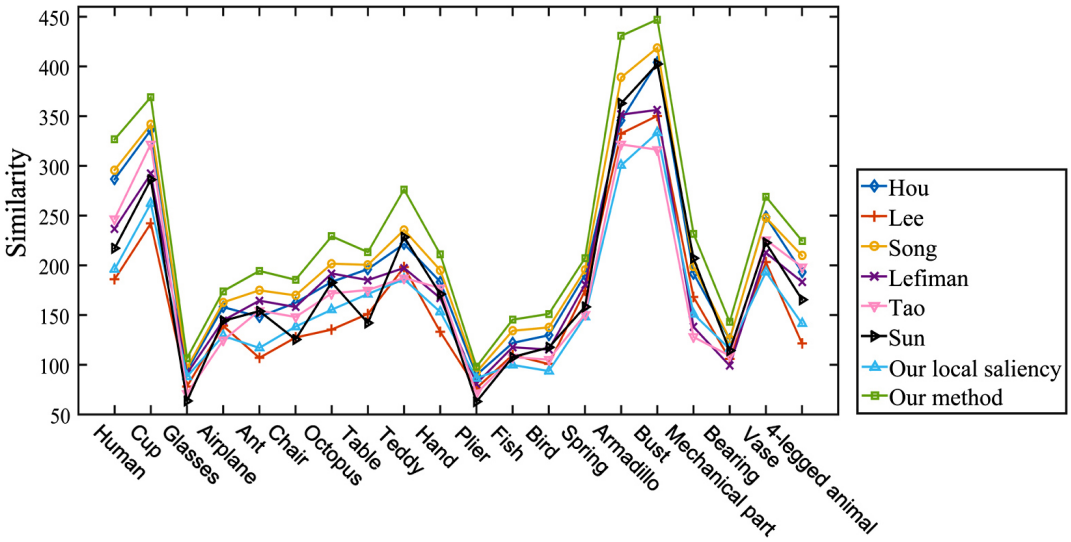

Fig. 10: Evaluation of different methods via Similarity: Hou [11], Lee [19], Song [30], Lefiman [21], Tao [32], Sun [31], our local mesh saliency (without global saliency adjustment) and our method (with global saliency adjustment).

that different methods perform consistently over the 20 classes of meshes. For example, they all perform quite well on the Armadillo and bust objects while glasses, plier and bearing are the most challenging classes to all of them. It demonstrates our observation that most saliency detection methods favour some particular classes of objects and implies the reason that Armadillo and bust objects are widely used in the experiments of these methods. This is because both Armadillo and bust objects have clear and well-defined salient features such as feet, hands, eyes, nose, mouth and ears. But in contrast, for the objects of glasses, plier and bearing, the concept of saliency is ill-defined or even invalid since even human subjects can hardly tell which parts are salient on these objects. Third, we found that Lee's method [19] and our local mesh saliency without any global saliency adjustment have the poorest performance. Different from other methods where more or less global or high-level cues are considered, Lee's method relies only on local cues such as curvature and local neighbourhood determined directly by Euclidean distance. The ablation study shows that the global saliency adjustment proposed in this paper really boosts the performance of saliency detection. This suggests that for mesh saliency detection, the consideration of some kind of global information generally helps. Finally, it can be seen that our method has the best performance in each evaluation

in most classes with roughly 20% improvement to the state of the arts.

All experiments were implemented on a computer with a quad core 3.6GHz i7 CPU and 32GB RAM. The slowest component of our method is the local saliency computation. For instance, for the girl model in Fig. 1 containing 15.5K vertices, computing the local saliency costs 14.5 seconds, the local-to-global pooling costs 3.4s and the global saliency adjustment costs only 0.5s. Since our method is insensitive to simplification as demonstrated in Fig. 5, for meshes containing millions of vertices, we can always compute saliency using the simplified mesh and use the mapping scheme in [30] to output a saliency map for the original input mesh.

6 Conclusions

We have presented a new method for detecting mesh saliency. It is composed of three steps: local saliency computation, local-to-global pooling and global saliency adjustment. When we design the algorithm for computing local mesh saliency, we also consider how it can facilitate the following computation based on global cues and how they can be integrated effectively in a local-to-global scheme. It is worth noting that besides global distinctness, there are also other computational global cues that could benefit saliency detection. For instance, [21] investigated whether a mesh is 'limb-like' or not; [8] concentrated on finding the base of an object. There is also a large amount of work on symmetry within 3D objects which can provide cues for computing global saliency. As a limitation of this work, we only formulate a single cue although it is potentially the most dominant one [34, 16]. As a result, for some classes of objects, our method fails to generate appealing outcomes. For instance, according to the quantitative evaluation, glasses and plier are the most challenging classes for most mesh saliency methods including ours. And Fig. 11 shows failed examples of our saliency detection on the two classes of objects. In these cases, the most important global cue is the 'segment centeredness' [3]. If the mesh saliency method does not include any scheme to capture this particular cue, it is likely to fail on these objects.

However, a challenge for considering various global cues is that most of these cues are computed through completely different schemes. Therefore, it can be foreseen that an interesting direction of future research is to devise a scheme which can benefit the computation of several cues. It will also cause another problem: how to balance and integrate different local and global cues within the same scheme. In this work, we empirically set the local-to-global weighting parameter since the integration is just a simple summation. It can be expected that when more cues are involved, the integration model will be more complicated and cannot be sorted empirically. [28] explored the impacts of different attributes and the combination model for best view selection in a data-driven manner. We believe that similar research can also be conducted for mesh saliency in the near future.

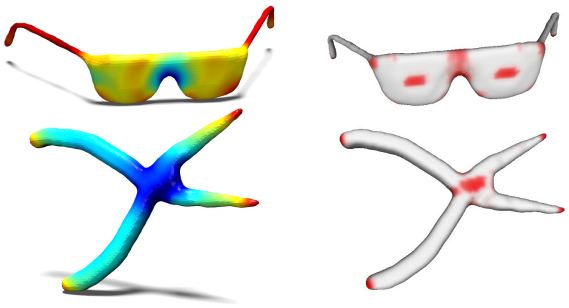

Fig. 11: Failed cases of our mesh saliency detection. Left: our results; Right: the ground truth

Acknowledgements This work is partly funded by EPSRC via the 'Automatic Semantic Analysis of 3D Content in Digital Repositories' project (EP/L006685/1). This support is gratefully acknowledged.

References

1. Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., Torralba, A.: Mit saliency benchmark. http://saliency.mit.edu/
2. Castellani, U., Cristani, M., Fantoni, S., Murino, V.: Sparse points matching by combining 3d mesh saliency with statistical descriptors. In: Proc. Eurographics, pp. 643–652 (2008)
3. Chen, X., Saparov, A., Pang, B., Funkhouser, T.: Schelling points on 3D surface meshes. In: Proc. SIGGRAPH (2012)
4. Coifman, R.R., Maggioni, M.: Diffusion wavelets. Applied and Computational Harmonic Analysis **21**(1), 53–94 (2006)
5. Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. Nature reviews neuroscience **3**(3), 201–215 (2002)
6. Dey, T.K., Ranjan, P., Wang, Y.: Convergence, stability, and discrete approximation of laplace spectra. In: Proc. ACM-SIAM Symposium on Discrete Algorithms, pp. 650–663 (2010)
7. Feixas, M., Sbert, M., González, F.: A unified information-theoretic framework for viewpoint selection and mesh saliency. ACM Transactions on Applied Perception (TAP) **6**(1), 1 (2009)
8. Fu, H., Cohen-Or, D., Dror, G., Sheffer, A.: Upright orientation of man-made objects. In: ACM transactions on graphics (TOG), vol. 27, p. 42. ACM (2008)
9. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. ACM Trans. Graph. **25**(1) (2006)
10. Guy, G., Medioni, G.: Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3d data. IEEE Trans. Pattern Anal. Mach. Intell. **19**(11), 1265–1277 (1997)
11. Hou, T., Qin, H.: Admissible diffusion wavelets and their applications in space-frequency processing. IEEE Transactions on Visualization and Computer Graphics **19**(1), 3–15 (2013)
12. Howard, I.: Seeing in depth. University of Toronto (2002)
13. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell **20**(11), 1254–1259 (1998)
14. Jin, H., Soatto, S., Yezzi, A.: Multi-view stereo reconstruction of dense shape and complex appearance. IJCV **63**(3), 175–189 (2005)
15. Kim, Y., Varshney, A., Jacobs, D., Guimbretiere, F.: Mesh saliency and human eye fixations. ACM Transactions on Applied Perception **7**(2), 12:1–12:13 (2010)

16. Koch, C., Poggio, T.: Predicting the visual world: silence is golden. Nature Neuroscience **2**, 9–10 (1999)
17. Koenderink, J.: The structure of images. Biological cybernetics **50**(5), 363–370 (1984)
18. Lang, C., Nguyen, T.V., Katti, H., Yadati, K., Kankanhalli, M., Yan, S.: Depth matters: Influence of depth cues on visual saliency. In: Proc. ECCV, pp. 101–115. Springer (2012)
19. Lee, C., Varshney, A., Jacobs, D.: Mesh saliency. In: Proc. SIGGRAPH (2005)
20. Lee, S., Xin, J., Westland, S.: Evaluation of image similarity by histogram intersection. Color Research & Application **30**(4), 265–274 (2005)
21. Leifman, G., Shtrom, E., Tal, A.: Surface regions of interest for viewpoint selection. In: Proc. CVPR (oral) (2012)
22. Mantiuk, R., Myszkowski, K., Pattanaik, S.: Attention guided mpeg compression for computer animations. In: Spring Conference on Computer graphics, pp. 239–244. ACM (2003)
23. Matlin, M.W.: Cognition (Textbook) (8 ed.). Wiley (2013)
24. Newman, M.E.: The mathematics of networks. The new palgrave encyclopedia of economics (2008)
25. Pauly, M., Keiser, R., Gross, M.: Multi-scale feature extraction on point-sampled surfaces. Comput. Graph. Forum **22**(3) (2003)
26. Pelphrey, K., Sasson, N., Reznick, J., Paul, G., Goldman, B., Piven, J.: Visual scanning of faces in autism. Journal of autism and developmental disorders **32**(4), 249–261 (2002)
27. Perron, O.: Zur theorie der matrices. Mathematische Annalen **64**(2), 248–263 (1907)
28. Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A.: Perceptual models of viewpoint preference. ACM Transactions on Graphics (TOG) **30**(5), 109 (2011)
29. Shilane, P., Funkhouser, T.: Distinctive regions of 3d surfaces. ACM Transactions on Graphics **26**(2), 7 (2007)
30. Song, R., Liu, Y., Martin, R.R., Rosin, P.L.: Mesh saliency via spectral processing. ACM Transactions on Graphics **33**(1), 6 (2014)
31. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Proc. SGP, pp. 1383–1392 (2009)
32. Tao, P., Cao, J., Li, S., Liu, X., Liu, L.: Mesh saliency via ranking unsalient patches in a descriptor space. Computers & Graphics **46**, 264–274 (2015)
33. Taubin, G.: A signal processing approach to fair surface design. In: Proceedings of SIGGRAPH, pp. 351–358. ACM (1995)
34. Wolfe, J.M.: Guided search 2.0 a revised model of visual search. Psychonomic bulletin & review **1**(2), 202–238 (1994)
35. Wu, J., Shen, X., Zhu, W., Liu, L.: Mesh saliency with global rarity. Graphical Models (2013)
36. Yee, H., Pattanaik, S., Greenberg, D.: Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. ACM Transactions on Graphics **20**(1), 39–65 (2001)