Neurocomputing 400 (2020) 11–23

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Sparse metric-based mesh saliency

Shanfeng Hu[a], Xiaohui Liang[b], Hubert P.H. Shum[a,*], Frederick W.B. Li[c], Nauman Aslam[a]

[a] Northumbria University, Newcastle upon Tyne NE1 8ST, UK
[b] Beihang University, Beijing 100191, China
[c] Durham University, South Road, Durham DH1 3LE, UK

ARTICLE INFO

Article history:
Received 29 January 2019
Revised 16 November 2019
Accepted 25 February 2020
Available online 10 March 2020

Communicated by Prof. Wei Feng

Keywords:
Mesh saliency
Visual attention
Metric
Sparsity

ABSTRACT

In this paper, we propose an accurate and robust approach to salient region detection for 3D polygonal surface meshes. The salient regions of a mesh are those that geometrically stand out from their contexts and therefore are semantically important for geometry processing and shape analysis. However, a suitable definition of region contexts for saliency detection remains elusive in the field, and the previous methods fail to produce saliency maps that agree well with human annotations. We address these issues by computing saliency in a global manner and enforcing sparsity for more accurate saliency detection. Specifically, we represent the geometry of a mesh using a metric that globally encodes the shape distances between every pair of local regions. We then propose a sparsity-enforcing rarity optimization problem, solving which allows us to obtain a compact set of salient regions globally distinct from each other. We build a perceptually motivated 3D eye fixation dataset and use a large-scale Schelling saliency dataset for extensive benchmarking of saliency detection methods. The results show that our computed saliency maps are closer to the ground-truth. To showcase the usefulness of our saliency maps for geometry processing, we apply them to feature point localization and achieve higher accuracy compared to established feature detectors.

1. Introduction

The human visual system has a remarkable ability to quickly and effortlessly identify a small number of interesting objects in a visual field. This ability appears mainly stimulus-driven and is commonly referred to as *visual attention*, which helps suppress the vast amount of visual inputs that are not essential to subsequent cognitive processing tasks [1].

In computer vision, numerous computational methods have been proposed to mimic the visual attention mechanism for efficient image understanding [2–5]. Inspired by the idea of saliency-guided image processing, Lee et al. [6] introduced the concept of *mesh saliency* to computer graphics, highlighting its advantages over traditional geometric quantities (e.g. curvatures) for assessing the perceptual importance of mesh regions. By prioritizing the processing of mesh regions according to their saliency values, the perceptual quality of processed meshes can be largely retained and the processing time can be effectively reduced. Such examples include shape simplification [6–9], shape matching [7,10], realistic rendering [11–14], shape segmentation [9,15], shape reconstruction [9], and crowd modeling [16].

Despite the vast use of mesh saliency in geometry processing and computer graphics, the definition of what constitutes saliency remains elusive in the field. The fundamental challenge is that the attention mechanism of human vision is far from being fully understood [1]. Regarding this, many efforts have been devoted to hand-crafting some computational methods that take a 3D polygonal mesh as input and produce a saliency map as output. For example, the local contrast methods of [6,10,17] compute the saliency of a local region as the difference of its differential properties from its surroundings. The global rarity methods of [8,9,18,19] compute the saliency of regions in wider surroundings and are able to highlight more distinct shape features. However, the former mainly respond to local geometric variations and suffer from surface noises and bumps, while the latter are sensitive to topological flaws due to the use of mesh connectivity for saliency computation.

There are also methods using high-level semantic annotations for saliency computation. One example is the method of [7] that detects salient regions effective at distinguishing shapes of different object categories. The other example is the tree-regression-based method of [20], which learns to predict saliency from low- and high-level geometric properties such as curvatures and symmetries. However, both methods require category-specific

* Corresponding author.
E-mail address: hubert.shum@northumbria.ac.uk (H.P.H. Shum).

https://doi.org/10.1016/j.neucom.2020.02.106
0925-2312/

human annotations for saliency computation and therefore cannot generalize to more object categories without annotations.

In view of the above challenges, we propose a new saliency detection method that does not require human annotations, generates accurate saliency maps much closer to ground-truth, and is robust to mesh noises, simplification and holes. Our method is mathematically derived from two fundamental principles of saliency: *rarity* and *sparsity*. The rarity principle regards those regions distinct from others to be salient, and the sparsity principle ensures that only a small number of truly distinct regions can pop out from the saliency computation process. Without enforcing sparsity while optimizing rarity, the computed saliency maps would become overly uniform and very few regions can stand out and be correctly recognized as salient.

Specifically, we propose a sparse metric-based rarity optimization problem for saliency computation. The optimization variable of the problem is the optimal saliency map to be solved for, and the optimization objective is the continuous rarity of a metric encoding the shape distance between every pair of local regions. We incorporate the sparsity principle of saliency by constraining the L0-norm of any feasible saliency map solutions. As a result, our saliency detection method amounts to solving a sparse eigenvalue problem [21], with the optimal saliency map being the sparse eigenvector of the metric of a mesh. By averaging multi-scale metrics into a scale-free metric, our method is able to discover a compact set of multi-scale salient regions from raw geometric features. By avoiding the use of mesh connectivity in metric representation, our method maintains robust to mesh flaws such as simplifications, noises, and holes.

To evaluate the performance of saliency detection methods, we build a perceptually motivated 3D eye fixation dataset from 50 graphics meshes and 8 human subjects through a 3D eye-tracking experiment. We also implement the highly cited saliency detection methods of [6,18,19], whose original source codes are not publicly available for large-scale quantitative benchmarking. We perform extensive method evaluations on our eye fixation dataset and the Schelling saliency dataset of [20]. The results show that our computed saliency maps are closer to the ground-truth than that generated by the competing methods of [6,8,9,18,19]. To showcase the usefulness of our computed saliency maps, we apply them to feature point localization [22] and compare to the established feature detectors of [6,23–25]. The results show that our saliency-guided feature detector outperforms others in terms of feature point localization accuracy.

We propose three contributions in this study:

- We propose a sparse metric-based rarity optimization method for saliency detection. Our method is shown to be able to produce accurate saliency maps without relying on human annotations while being robust to mesh simplifications, noises, and holes.
- We build a perceptually motivated 3D eye fixation dataset for saliency detection benchmarking. The dataset extends the eye fixation experiment of [26] from 2D to 3D, and complements the Schelling saliency dataset of [20] that is not constructed from real captured human eye movements. Our publicly available dataset can be downloaded from this link: http://info.hubertshum.com.
- We implement the highly cited saliency detection methods of [6,8,18,19] and perform benchmarking on our eye fixation dataset and the Schelling saliency dataset of [20]. The results show that our computed saliency maps are closer to the ground-truth annotations compared to that of [6,9,18,19]. Our open-source codes can be downloaded from this link: http://info.hubertshum.com.

2. Related work

Visual Attention Modeling. The theoretical foundation of visual attention can be traced back to [27], where Treisman and Gelade proposed "Feature-Integration Theory" which suggests what and how visual features are combined to direct human visual attention. Koch and Ullman [28] developed a feed-forward computational method to incorporate these features, indicating that salient image locations are visually distinct from their surroundings. Itti et al. [2] implemented a center-surround operator on low-level image features for saliency detection. After that, a large body of visual saliency methods have been proposed in computer vision [1,3–5]. Recently, [29] proposed to extract the most distinguishable information from image color, texture and location features for saliency detection. Further, [30] incorporated both low- and high-level image features into a variety of learning algorithms for more accurate saliency detection. Additionally, the detected saliency maps can be enforced to be more accurate and consistent using Markov Random Field (MRF) on multiple feature maps [31]. Our proposed method is also for visual attention modeling but works on 3D surface meshes instead of 2D images. Another difference is that we only use one set of features for saliency detection [32], without leveraging multiple sets of features as in [29–31]. Incorporating more features is a future direction.

Image Co-saliency Detection. Recently, image co-saliency detection that aims to extract the common salient objects from a group of similar images has been widely studied. Different from traditional image saliency detection that handles each individual image separately [1–5], co-saliency detection needs to address the foreground consistency and background variations of a group of images[33]. proposed to estimate the co-saliency priors of these images for co-saliency detection using the Gaussian Mixture Model (GMM) on the corresponding binary salient masks[34]. further observed that the common salient objects usually have similar color distributions and therefore can be effectively captured in a shared color space without the explicit combination of single saliency maps. More recently, [35] achieved more accurate co-saliency detection results by learning from both labeled exemplar images and regions. We focus on single saliency detection for 3D meshes in this work instead of co-saliency detection.

Saliency Detection for 3D Scenes. To accelerate realistic rendering, Yee et al. [36] used the method of [2] to detect salient regions of coarsely rendered scenes and focused rendering resources on these important regions. Similar to [36], Longhurst et al. [37] controlled per-pixel ray sampling density based on detected salient regions. Afterwards, [14] extended the idea to participating media rendering and achieved realistic results with low computational costs. Mantiuk et al. [38] made an attempt to compress animated scenes with the guidance of image saliency. By only using salient regions of rendered images, these methods have no access to any depth information of 3D scenes, which plays an important role in human visual attention [39]. In contrast, our method analyses 3D geometry directly and therefore can detect structure-related saliency information.

Saliency Detection for Surface Meshes. We classify existing mesh saliency detection methods into the following five categories:

Local Contrast. Inspired by [2], Lee et al. [6] introduced the concept of mesh saliency using a center-surround operator on Gaussian-weighted mean curvatures. Gal and Cohen-Or [10] defined the saliency of a region based on its relative size, curvatures, and curvature changes. They detected and segmented salient regions for partial shape matching. Feixas et al. [17] proposed an information-theoretical framework for viewpoint selection and mesh saliency computation. Zhao et al. [40] computed saliency from local normal information for subsequent refinement. These models compute saliency as local contrast of surface prop-

erties, generally by comparing local regions to their neighbours. Jeong et al. [41] used normal information to compute both view-independent and view-dependent saliency. However, local contrast methods tend to wrongly identify bumpy and noisy regions as salient. Our method addresses this issue by extending region comparison from a local to a global context.

Spectral Irregularity. Based on the work of [4], Song et al. [9] assumed that saliency is hidden in the irregularity of the log-Laplacian spectrum of a mesh. They extracted such irregularity and transformed it back to the spatial domain to compute saliency. Mesh simplification [42] was used to tackle the costly eigendecomposition of a large Laplacian matrix. Due to the spatial unawareness of spectral basis, this method has difficulties in capturing some individual local salient regions. Our method, instead, works in the spatial domain and can capture salient regions from small to large scales.

Shape Discrimination. Analogous to feature selection in classification problems, Shilane and Funkhouser [7] detected salient surface regions for distinguishing shapes of different object categories. The detection results depend not only on a semantically categorized shape database but also on the object categories of input meshes. In many applications, however, these semantic annotations remain scarce and their collection is labour-intensive. Our method does not require any semantic data for saliency detection, thereby allowing the use for geometry processing applications that do not have semantic annotations.

Learning-based Detection. Through a large-scale online user study, Chen et al. [20] obtained massive amounts of salient surface points for a library of meshes. They used the obtained data to train regression models for saliency prediction. A variety of low-, middle-, and high-level cues, such as curvatures, geodesics, segmentation, and symmetries, were incorporated. The trained models performed well on the training set but showed limited generalization abilities for novel meshes out of training datasets. Our method does not require training before use and shows good generalization to diverse object categories.

Global Rarity. Leifman et al. [18] detected more globally rare surface regions for viewpoint selection. Wu et al. [8] combined local contrast with global rarity to compute mesh saliency. Pingping et al. [19] detected salient regions by identifying non-salient backgrounds via manifold ranking. These methods aim to suppress repeated patterns across a mesh surface by extending region comparison from a local to a wider context. However, they require mesh segmentation and geodesic distance computation which are not robust to topological flaws such as holes and non-manifold structures of meshes. Our method belongs to this category and is based on a global metric representation, which is robust to underlying potentially poor mesh tessellations.

Mesh Saliency Evaluation. Howlett et al. [43] computed saliency maps from human eye fixations to guide mesh simplification. They demonstrated that preserving salient details could improve the fidelity of simplified meshes. Using eye-tracking experiments, Kim et al. [26] validated that saliency [6] was better compared to mean curvature for eye fixation prediction. We extend the eye-tracking experiment of [26] from 2D to 3D and build a 3D eye fixation dataset suitable for public saliency detection benchmarking.

Mesh Saliency Applications. Numerous graphics applications have benefited from mesh saliency. Kim and Varshney used saliency to edit surface [12] and volume [11] regions for highlighted visualization. Liu et al. [15] and Miao et al. [13] employed saliency to detect feature points and extremum lines for mesh segmentation and depiction. Recently, Gu et al. [44] combined saliency with Poisson sampling for adaptive depth image compression. Other applications include mesh simplification [6–9], viewpoint selection [6,7,18,45], shape matching [7,10], mesh sampling [8], surface reconstruction[9], and crowd modeling [16]. We apply our method to the task of feature point detection which is a fundamental building block in many geometry processing applications.

3. Our method

3.1. Overview

We illustrate the computation steps of our proposed saliency detection method in Fig. 1. Given a 3D polygonal surface mesh as input, we first sample a set of random points on the surface (a). For each sample point, we construct a shape descriptor that characterizes its local shape information (b). We then compute a matrix of squared Euclidean distances among all sample points using their shape descriptors (c). From this metric representation, we derive the optimal saliency map by solving a sparse metric-based rarity optimization problem (d). Finally, we map the computed saliency from the sampled points back to the underlying mesh vertices via Gaussian filtering (e).

3.2. Mesh sampling

To achieve the translation and uniform scaling invariance of saliency detection, we normalize an input mesh by locating its centroid at the origin and uniformly scaling its radius (i.e. the half diagonal length of its tight bounding box) to 1. As a surface mesh is sometimes either under- or over-tessellated, we randomly sample a set of points on the surface of the normalized mesh [7], so that the quality of computed saliency maps is maintained while the computational cost remains invariant to the original size of the mesh. We denote the sample point set as $\mathcal{P} = \{p_1, p_2, \cdots, p_n\}$ and empirically find that 5000 points are sufficient to cover the whole surface. We use this value in all of our experiments.

As observed in [6,7], salient regions can range from small surface details to large surface parts. To accommodate the multi-scale nature of saliency, we define a succession of increasingly larger regions for each sample point on the surface, $\mathcal{S} = \{0.02, 0.04, 0.06, 0.08, 0.1\}$. We denote a region p_i^k as a spherical volume of the radius $s_k \in \mathcal{S}$, with the volume centered at the sample point $p_i \in \mathcal{P}$. We use this region representation because it is independent of the underlying potentially poor mesh tessellation (i.e. irregular meshes, non-manifold edges, and disconnected components) and easily supports multi-scale saliency computation by iterating through each scale in $\mathcal{S}$. We find that $\mathcal{S}$ works well for capturing both small- and large-scale salient features in practice.

3.3. Shape descriptor construction

For each region p_i^k defined in the above, we compute a feature vector f_i^k to characterize its local shape information. We choose the harmonic shape descriptor [32] because it is rotation-invariant and has minimal information loss. It was also used by [7] to facilitate the multi-scale computation of saliency.

To compute the descriptor, we convert a mesh into a Gaussian distance field of resolution $256 \times 256 \times 256$ and partition each region (i.e. a spherical volume) into 8 equally-spaced concentric shells [7]. We sample the Gaussian distance field on these shells and compute the amplitudes of the first 8 spherical harmonic frequency bands for each shell [32]. Therefore, the shape descriptor length of each region is $8 \times 8 = 64$. We find that this feature granularity is sufficient to discriminate regions for effective saliency computation.

3.4. Sparse metric-based rarity optimization

Traditionally, the methods of [6–8,10,17–19,40,41] compute saliency from some hand-crafted rules, lacking a principled goal of

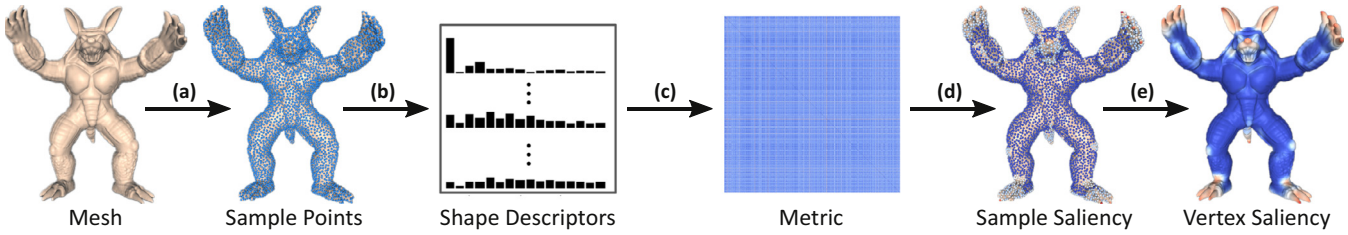


Fig. 1. Overview of our sparse metric-based mesh saliency detection method. The steps from (a)-(e) are mesh sampling, shape descriptor construction, metric computation, saliency optimization, and vertex saliency interpolation respectively. We use red and blue colours to indicate high and low saliency values respectively.

optimization. In contrast, we derive saliency from optimizing the rarity of a global metric representation while enforcing the sparsity of saliency. The rarity principle regards those regions that have the maximum distinction from others as salient, while the sparsity principle ensures that only a compact set of truly distinct regions can stand out. To this end, we propose the following optimization problem:

$$\arg\max R(\varphi) = \varphi^T M\varphi, \text{ s.t. } \varphi \geq 0\ ,\ \|\varphi\| = 1 \text{ and } \|\varphi\|_0 \leq \mu n \quad (1)$$

where $\varphi \in \mathbb{R}^n$ is the saliency map of the sample points $\mathcal{P}$ to be solved for (φ_i is the saliency value of point p_i), and $M \in \mathbb{R}^{n\times n}$ is the metric representation that encodes the pairwise shape contrasts among all sample points. Additionally, the first constraint ensures the solution saliency map to be element-wise nonnegative, and the second constraint $\|\varphi\| = \sqrt{\sum_i \varphi_i^2} = 1$ requires the map to have a unit Euclidean norm.

The Rarity Principle of Saliency. We refer to the objective of the optimization problem (1) as the rarity principle of saliency. After rewriting it as $R(\varphi) = \sum_{i,j} \varphi_i \varphi_j M_{ij}$ and assuming that the saliency map φ is binary (i.e. 1 for salient points and 0 otherwise), we can see that the objective exactly sums the shape contrasts among all sample points together. By globally optimizing this combinatorial problem with the sparsity constraint, we would obtain a set of salient regions (associated with the salient points) that are the most distinct from others. However, this problem is known to be NP-hard to solve [46] and the resulting map is not continuous for many applications. Therefore, we relax a saliency map to be continuous-valued and arrive at our continuous metric-based rarity optimization problem (1), so that it can be much more efficiently solved and the optimal saliency map is inherently continuous.

The Sparsity Principle of Saliency. We refer to the third constraint of the optimization problem (1) as the sparsity principle of saliency. The constraint $\|\varphi\|_0 = \sum_i I(\varphi_i \neq 0) \leq \mu n$ enforces the fraction of detected salient regions to be less than $0 < \mu \leq 1$, where $I(\cdot)$ is the indicator function. When $\mu = 1$, it has no use because all sample points are feasible to be identified as salient. When $0 < \mu < 1$, it guarantees that only a fraction of truly unique salient regions can be retained. We find that setting μ to 0.2 works well in practice.

To finely quantify and compare the sparsity patterns of saliency maps, we consider the Lorenz curves and Gini indices of them for analysis [47]. Let $\varphi_{(1)} \leq \varphi_{(2)} \leq \cdots \leq \varphi_{(n)}$ be the non-decreasing order statistics of a saliency map φ. The Lorenz curve is a piecewise linear function interpolating $n+1$ points $(F_i, L(F_i))$, where for $0 \leq i \leq n$, $F_i = \frac{i}{n}$ denotes the proportion of the i least salient regions and $L(F_i) = \frac{\sum_{j=1}^{i} \varphi_{(j)}}{\sum_{j=1}^{n} \varphi_{(j)}}$ encodes the proportion of saliency values assigned to these regions. As F_i varies evenly from 0 to 1, $L(F_i)$ grows increasingly from 0 to 1, tracing out a concave curve from the origin to (1,1). The Gini index associated with a Lorenz curve is one minus two times the area under the curve. As shown in Fig. 2, the Lorenz curve of a uniform saliency map is the straight line from the origin to (1,1), with the lowest Gini index of 0 indicating the absolutely even distribution of saliency values to all regions. The other extreme is the singular saliency map, which distributes all the saliency values only to a single region and produces the highest Gini index of 1. We also show the saliency maps of the Dragon with and without sparsity in Fig. 2. It can be seen that the map without sparsity is visually and quantitatively very close to the uniform one, suggesting very weak discrimination between salient (i.e. the long body) and non-salient (i.e. the head and claws) regions of the Dragon. By enforcing the sparsity constraint in (1), we dramatically push the map away from the uniform one and highlight the salient regions of the Dragon much more clearly.

Multi-Scale Saliency Computation. To capture small- and large-scale salient regions, we use a metric for each scale to represent the global pairwise shape contrasts among all sample points for saliency computation. Specifically, we compute $M_{ij}^k = \|f_i^k - f_j^k\|^2$ as the squared Euclidean distance between the descriptors of a pair of regions, and $M^k \in \mathbb{R}^{n\times n}$ as the metric consisting of these descriptor distances among all points at scale k. Due to the use of such a global metric representation, we are able to avoid the ambiguity of manually choosing a suitable context for saliency detection, as traditionally done in the methods of [6–8,10,17–19,40,41]. More importantly, the representation is decoupled with the underlying mesh tessellation, which may contain topological flaws that prevent robust saliency computation.

Without the sparsity constraint in (1), the objective is the Rayleigh quotient of a metric M^k and the saliency map φ^k globally optimizing it is the principal eigenvector of M^k [48]. Due to the nonnegativity of M^k, its principal eigenvector is guaranteed to be nonnegative and unique. It can be efficiently computed from M^k using the power method [21]. With the sparsity constraint, we can also efficiently solve the problem (1) using the truncated power method [21]. We describe the solution process as follows:

- **Initialization**. We shift all the eigenvalues of M^k to $(0, \infty)$ to make it positive definite, $\widetilde{M}^k \leftarrow M^k + \nu I_{n\times n}$, where ν is the principal eigenvalue of M^k computed from the power method and $I_{n\times n}$ is the identity matrix.
- **Iteration**. We start from the principal eigenvector of M^k computed from the power method,and then alternate between setting the $(1-\mu)n$ smallest values of the current map to zeros and multiplying it by $\widetilde{M}^k$ followed with normalization, until converging to the optimal sparse saliency map.

Multi-Scale Saliency Integration To capture both small- and large-scale salient features, we integrate multi-scale shape information by summing the metrics of all scales together: $M = \sum_k M^k$. We then compute the integrated saliency map φ from M by solving the problem (1) using the method described in the above. This way, we obtain a scale-free saliency map that fully adheres to the rarity and the sparsity principles os saliency. We also successfully avoid the cost of computing, summing, and then discarding multi-scale saliency maps as traditionally done in [6–10,17–19].

As shown in Fig. 3, the smallest-scale saliency map responds strongly to the local surface bumps and textures of the Bunny. As the scale is increased, larger salient regions such as the mouth,

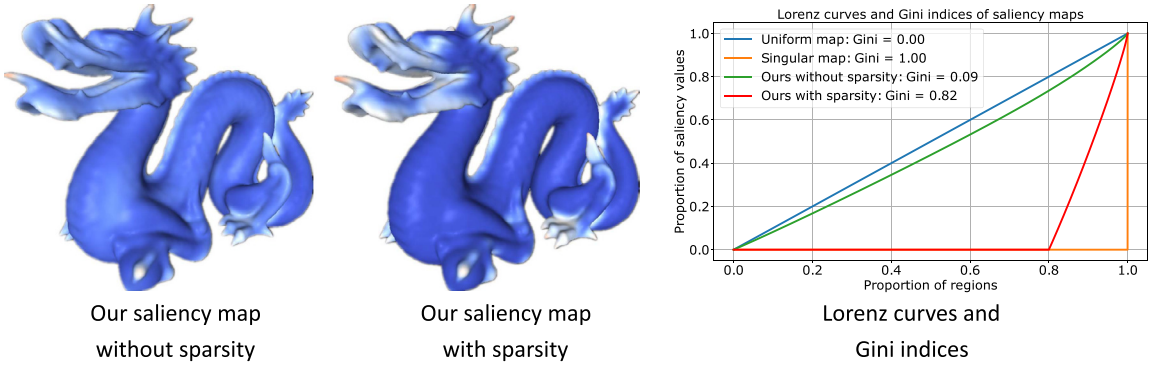


Fig. 2. Saliency detection with and without sparsity. *Left*: our computed saliency map without sparsity. *Middle*: our computed saliency map with sparsity. *Right*: Lorenz curves and Gini indices of saliency maps. Our map without sparsity is both visually and quantitatively similar to the uniform one, while our map with sparsity is quantitatively sparser and visually more concentrated on distinct regions.

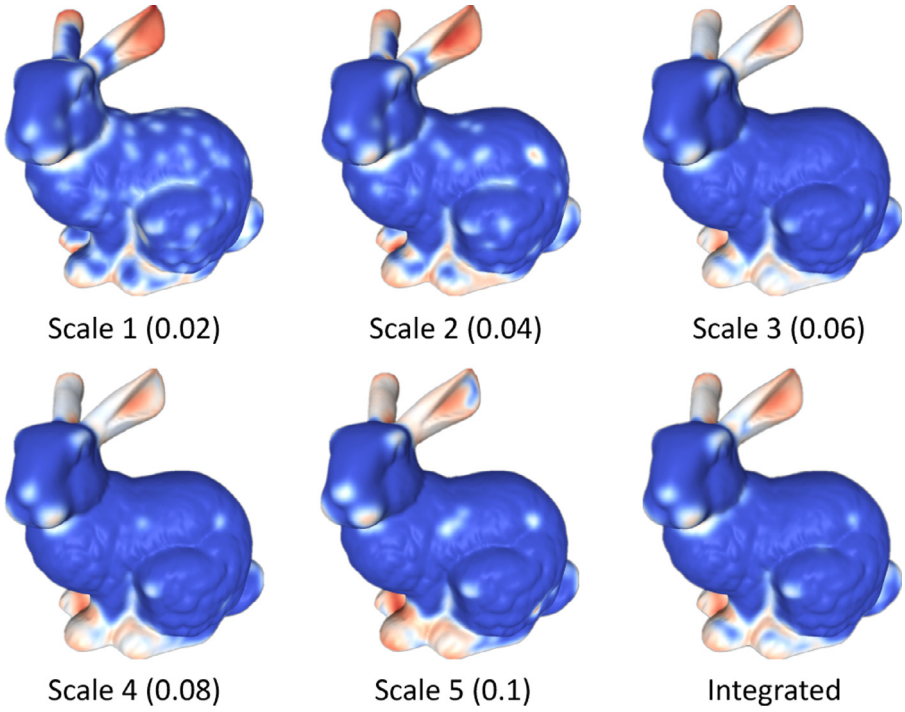


Fig. 3. Multi-scale saliency integration. The saliency map of each scale is computed from the corresponding metric and the integrated map is computed from the sum of the metrics of all scales.

eyes, ears, and feet are accurately captured. The final scale-free map effectively retains these visually salient regions while suppressing other undesirable local surface variations.

3.5. Vertex saliency interpolation

After computing the saliency values of sample points, we map them back to the underlying mesh vertices using Gaussian filtering. Let $\xi(v)$ denotes the saliency of a vertex v. We compute $\xi(v)$ as the Gaussian-weighted average of the saliency values of the sample points close to v:

$$\xi(v) = \frac{\sum_{p\in\mathcal{N}(v,3\sigma)} exp[-||v-p||^2/(2\sigma^2)]\varphi(p)}{\sum_{p\in\mathcal{N}(v,3\sigma)} exp[-||v-p||^2/(2\sigma^2)]} \quad (2)$$

where $\mathcal{N}(v,3\sigma) = \{p \in \mathcal{P} \mid ||v-p|| < 3\sigma\}$ and σ is the scale parameter of the Gaussian filter. We use a KD-tree to organize and query sample points for more efficient Gaussian filtering. We find that $\sigma = 0.02$ works well in practice.

4. Our 3D eye fixation dataset

As visual saliency is inherently a pre-attentive mechanism of the human visual system [1], it is important to evaluate the performance of saliency detection methods using real captured human eye movements on 3D surface meshes. However, the previously constructed saliency datasets are either too small [43], only for 2D rendered images of 3D meshes [26], or not captured from real human eye movements on 3D meshes [20]. Therefore, we propose our 3D eye fixation dataset for public saliency detection benchmarking (http://info.hubertshum.com), which is built as follows (see an example mesh and the collected eye fixations on the left of 6):

Mesh Dataset. We collected 50 meshes that are popularly used in computer graphics research from the Stanford 3D Scanning Repository [49] and the SHREC2007 Challenge [50]. For each mesh, we fixed the non-manifold edges and remeshed the surface into a good quality, so that all of our evaluated saliency detection methods can work well on it [6,9,18,19]. In the future, we will include

meshes of poorer qualities (e.g. with holes and non-manifold structures) for more realistic benchmarking in real-world applications.

Participating Subjects. We hired 8 undergraduate and master students from Beihang University as human subjects for our study. They were aged 23 – 28 and have normal or corrected visions. They were kept unknown about the purpose of our study to reduce the bias of collected data.

Eye-Tracking Experiments. To capture 3D eye movement data, we generated a 48s video for each mesh that shows its whole surface from 12 key viewpoints. We kept each viewpoint static for 3 s and then smoothly switched to the next viewpoint in 1 s, so that the visual attention of a subject can be directed through the whole surface of a mesh. For each subject, we instructed him/her to sit in a distance of 95 – 110 cm from a 1680 × 1050 LED display. Before the onset of each video stimulus (corresponding to each mesh in our dataset), we calibrated our used SMI RED250 eye-tracker by letting the subject gaze at 9 successive black dots on the screen. We considered the calibration successful if the gaze error was less than 0.8°, otherwise we repeated the calibration process. After calibration, we let the subject freely view the displayed video of a mesh and used the eye-tracker to capture his/her gaze positions on the screen at 250HZ sampling rate, with a gaze capturing accuracy of 0.4°.

Data Pre-processing and Aggregation. For each mesh in our dataset, we discarded the first two and the last two eye fixations of each subject because they can be noisy due to the onset and offset of the video stimulus. We only retained the 15 2D eye fixations of the longest durations (average $\geq$ 300ms) for each subject, because these eye fixations were much less noisy and represented where a subject was gazing towards on the screen. To obtain 3D eye fixations on the surface, we synchronized the timestamps of the captured 2D eye fixations with camera viewpoints and then projected them back to the nearest mesh vertices on the surface. Finally, we aggregated the 15 3D eye fixations from each of the 8 subjects to form 120 ground-truth eye fixations on each mesh surface in our dataset.

5. Results

5.1. Saliency detection results

We show the saliency maps of 18 meshes computed by our method in Fig. 4, along with the ground-truth maps provided by Chen et al. [20]. Each mesh is randomly chosen from the 20 of the corresponding object category [20]. These results indicate several strengths of our method for saliency detection:

- **Shape Distinction**. Our method successfully detects the globally distinct regions of surface meshes, such as the protruded parts (the horns, ears, and legs of the Cow), shape extremities (the hands, feet, and head of the Human), sharp edges (the perpendicular borders of the Mech), and corners (the claws of the Armadillo).
- **Curvature Insensitivity**. Our method is shown to be robust to the local curvature changes of surface regions. As shown for the legs of the Armadillo, they are bumpy and textured but are effectively suppressed by our method.
- **Compactness of Saliency**. The saliency maps computed by our method are visually quite compact, which only highlight a small number of salient regions with a clear boundary between non-salient ones.

We can also see that our saliency maps are visually close to the ground-truth. This suggests that they capture the true unknown human visual attention towards surface meshes to some extent.

5.2. Visual comparisons with other methods

We compare our saliency maps with those generated by 6 representative methods in Fig. 5. We choose these methods for comparison because they are the most cited in the field and have distinct methodologies. We highlight the merits of our method over each of them as follows:

- **Local Contrast**. The method of [6] computes saliency as the local contrast of mean curvatures and is thus unable to suppress bumpy surface regions such as the body of the Bunny. In comparison, our method only detects the globally rare mouth, eyes, ears, and feet regions.
- **Shape Discrimination**. The method of [7] detects category-specific distinctive regions and only marks the whole heads of the Horse and the Dog as salient. Our method more finely captures the individual salient regions, including their mouths, eyes, ears, and legs.
- **Global Rarity**. While the method of [18] used shape extreme points, patch distinction and patch association for saliency detection, the method of [19] induced saliency from their dissimilarities to non-salient backgrounds. As shown for the facial features of the Horse and the Bunny, our method highlights them more accurately compared to that of [18]. For the body of the Dragon and the head of the Igea, our method is more robust to the noisy surface variations.
- **Learning-based Detection**. The method of [20] trains a tree-regression function for saliency detection. However, the trained function shows limited generalization abilities for novel meshes. For example, the handles of the Vase and the legs of the Camel are not well detected. Our method captures these regions without using any semantic data.
- **Spectral Irregularity**. The method of [9] leverages the residuals of the Laplacian spectrum for saliency detection. Due to the spatial unawareness of spectral basis, the method has difficulty in localizing individual salient regions. While the forelegs of the Dog and the centers of the Glasses are of interest, they are erroneously neglected. Our method recovers them correctly.

5.3. Quantitative comparison with other methods

We also evaluate saliency detection methods on our own 3D eye fixation dataset and the Schelling saliency dataset of [20]. We choose the two datasets because the former is directly captured from our human eye tracking experiments and the latter reflects human subjective agreements on what constitute semantically prominent regions on surface meshes. Both characterize human visual attention on surface meshes to some extent.

Implementing Other Methods for Comparison. Since the introduction of mesh saliency to computer graphics [6], a number of saliency detection methods have been proposed in the past [6–8,10,17–20,40,41], which show some progress of the visual quality of the generated saliency maps. However, their source codes are not publicly available, preventing a large-scale quantitative benchmarking of their true performance. Kim et al. [26] only evaluated the performance of the method of [6] and the evaluation is only based on 2D eye fixation data. Therefore, we have made efforts to implement the highly cited methods of [6,8,18,19] and use the source code of [9] for quantitative evaluation (http://info.hubertshum.com). For the methods of [6,9], we use the original parameters and find that the reproduced results align well with that reported in the original papers. For the methods of [18,19], we have tried to tune their shape descriptor construction and saliency computation parameters to match the reported images in the original papers. To our knowledge, this is the first time that saliency detection methods are quantitatively evaluated in the field. In the

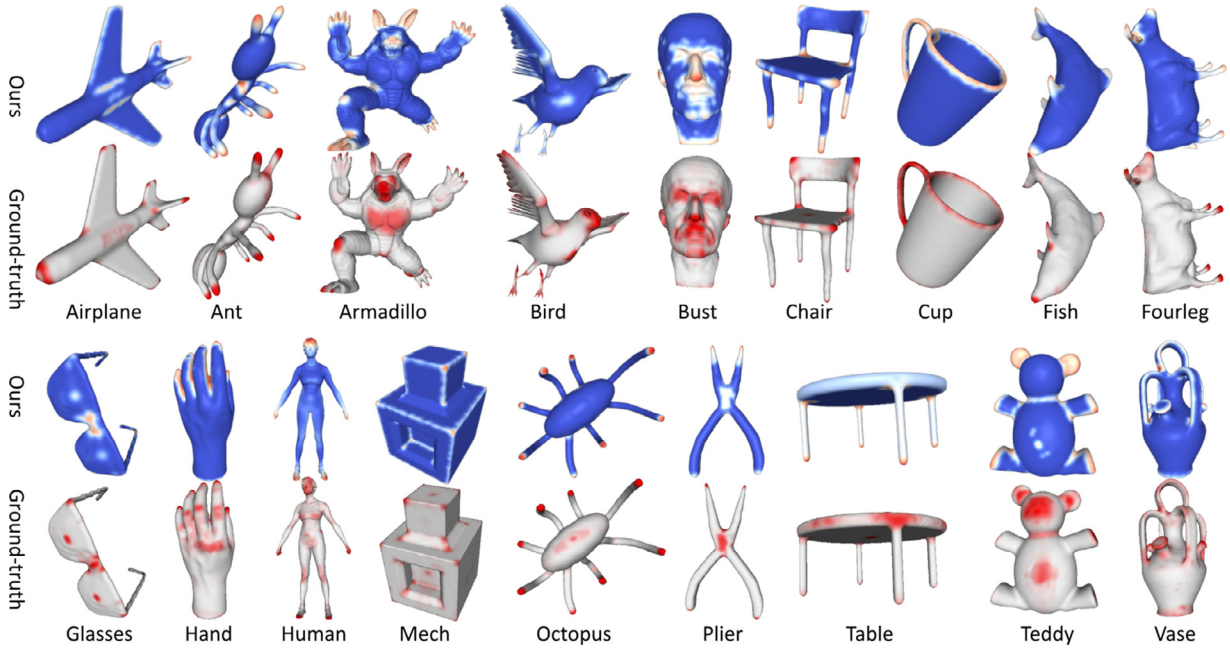


Fig. 4. Comparison of our saliency maps with the ground-truth of [20]. Each of the 18 shown meshes belongs to a different object category from the dataset of [20]. Following [9], we take the rendered pseudo ground-truth images from [20] because we do not have access to their source code.

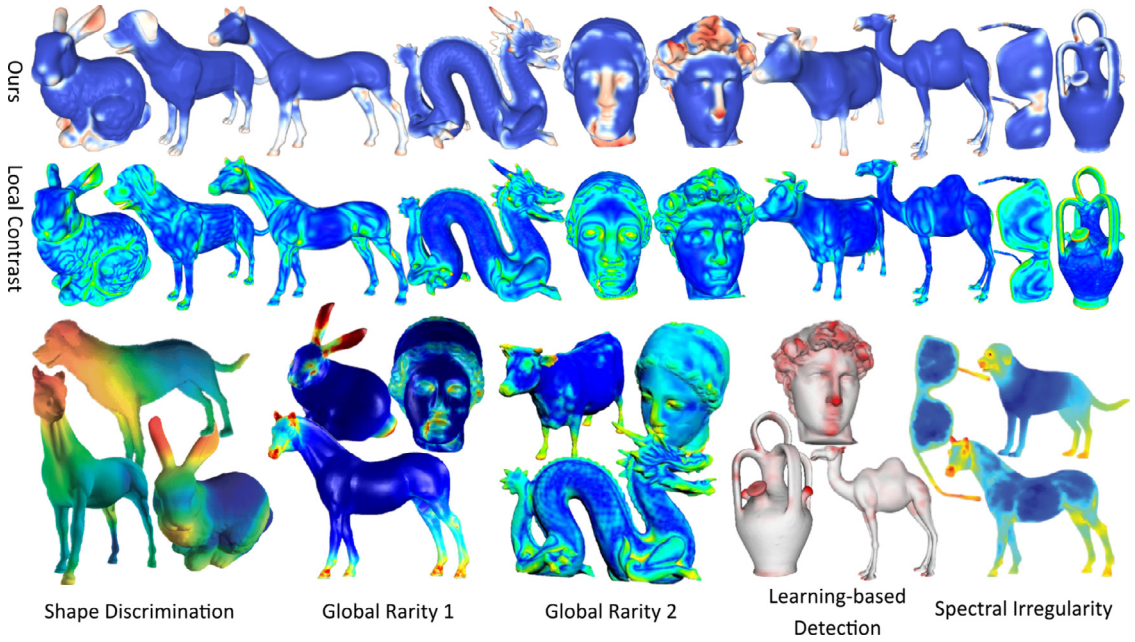


Fig. 5. Comparison of our saliency maps with that generated by other methods. The competing methods include Local Contrast [6], Shape Discrimination [7], Global Rarity 1 [18], Global Rarity 2 [19], Learning-based Detection [20], and Spectral Irregularity [9]. Following [9,18,19], we take the rendered saliency maps from the original papers of [7,9,18–20] because we do not have access to their source codes. For the method of [6], we generate the saliency maps using our own implementation and visualize them to match the original color themes.

future, we plan to implement other methods for more thorough evaluation.

Evaluation on Our Eye Fixation Dataset. As described in Section 4, we build a 3D eye fixation dataset from 50 meshes and 8 human subjects, who were instructed to freely view these meshes on a computer screen while wearing a high-precision eye-tracking device. After data pre-processing and aggregation, we retain 120 most frequently attended points on each mesh in our study. We use the popular Area Under Curve (AUC) metric to quantify how well a saliency map captures these eye fixations on a surface [1].

We present an example mesh with the captured eye fixations and the evaluation results on our eye fixation dataset in Fig. 6. As pointed out by [1], the AUC metric is sensitive to the blurring of saliency maps, so we filter each saliency map on a surface using the Gaussian kernels with scales from 0 to 0.2 and compute the corresponding AUC score for each scale. The Human method amounts to filtering the eye fixations of randomly selected half subjects into a saliency map and evaluating on the other half subjects for each mesh. A stable AUC score is computed by averaging the results of this random process.

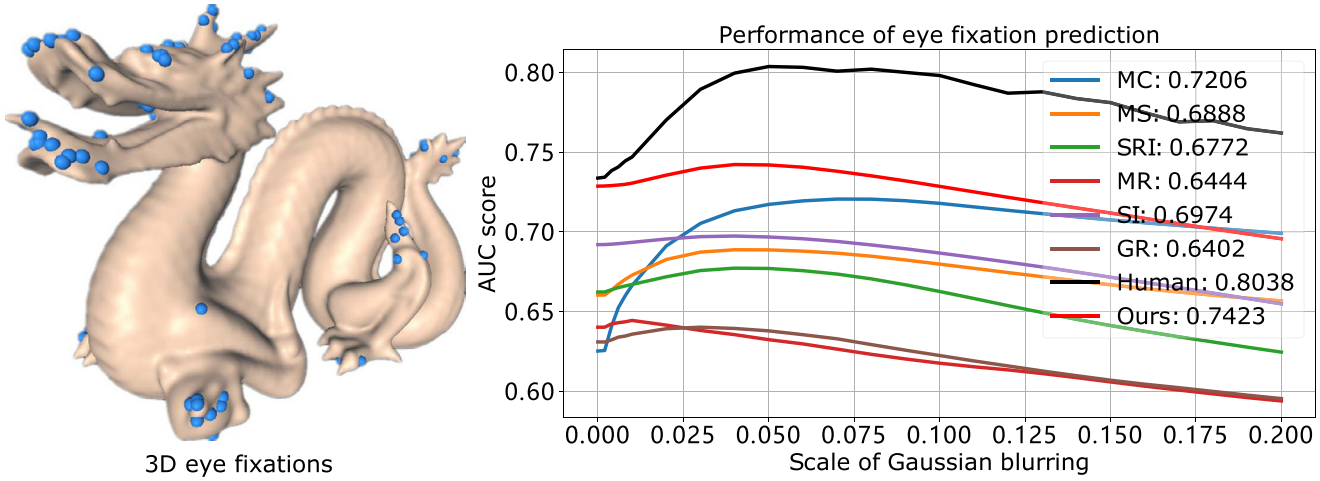


Fig. 6. Performance of eye fixation prediction. *Left*: The Dragon and our captured ground-truth 3D eye fixations on the surface. *Right*: The Human method uses the captured eye fixations of half subjects to predict that of the other halves, thereby measuring the self-consistency of our eye fixation dataset. The competing methods are MC [51], MS [6], SRI [18], MR [19], SI [9], and GR [8]. The peak AUC scores of these methods are displayed in the plot legend.

It can be seen from Fig. 6 that the Human method detects the captured eye fixations very accurately, achieving the peak score around scale 0.05. This shows that our recorded eye fixations agree well among subjects and therefore qualify as ground-truth for method benchmarking. By focusing on the peak scores, it is surprising to see that MC performs better than MS on eye fixation prediction, which appears to contradict the findings of [26]. This may be because Kim et al. [26] did not take blurring into consideration for evaluation. It is also interesting to see that the more global SRI and MR methods achieve lower accuracy compared to that by the local MC and MS methods, indicating their limited eye fixation localization performance. The spectral SI method, compared to them, performs slightly better but is still worse than the optimally blurred MC method. In contrast, our method achieves much higher eye fixation localization accuracy around the optimal blur scale 0.04.

Evaluation on the Saliency Dataset of [20]. We also use the Schelling saliency dataset of [20] for method benchmarking. This dataset has 400 meshes evenly split into 20 object categories, with a collection of human annotated salient points on each mesh and the corresponding filtered saliency map. We use the AUC score and the Linear Correlation Coefficient (LCC) to quantify the accuracy of a saliency map for predicting discrete salient points and continuous saliency values respectively [1].

To finely compare different methods, we report their AUC scores for each object category separately and the scores for all categories together in Table 1. It can be seen that overall our method is the best performing one, and the followings are the MS, MC, SRI, SI, MR, and GR methods. It is interesting to see that the AUC scores achieved by these methods are generally higher than that on our eye fixation dataset. This may be because the salient points in the dataset of [20] are fewer and less spread on mesh surfaces. As visually shown in Fig. 7, The sparsity of salient points can explain the poor performance of the SI and MR methods, which produce overly large patches of salient regions and therefore lack feature localization ability. The GR method also performs poorly because it fails to clearly highlight salient regions of surfaces. In contrast, our method localizes salient points more accurately by optimizing the rarity and sparsity principles together.

We report the LCC scores of these methods for predicting continuous saliency distributions in Table 2. We note that continuous saliency distributions are generally harder to predict than discrete points because a method needs to discriminate salient and non-salient regions more finely. Therefore, we expect the LCC metric to be a more comprehensive performance metric than AUC. It can be seen that our method produces saliency maps that correlate with the ground-truth considerably better compared to other methods. We observe that while MS and SRI may be good at localizing sparse salient points, they have limited abilities to finely separate less salient from totally non-salient ones. As expected, SI and MR perform poorly because they produce overly large patches of salient regions that contain many non-salient backgrounds as well. GR does not perform well either because it fails to clearly separate salient and non-salient regions.

5.4. Robustness comparison with other methods

To show the robustness of our method, we compare our saliency maps of corrupted meshes with those computed by the competing methods of [6,9,18,19] in Fig. 8. It can be seen that our method copes well with surface noises, simplifications, and holes. The computed saliency maps remain very close to that of the original Armadillo. In contrast, the method of [6] is fairly sensitive to curvature changes, responding strongly to the bumps around the legs of the Armadillo. The methods of [18,19] are also not sufficiently resilient to the introduced mesh flaws, producing inconsistent saliency maps for the damaged versions of the Armadillo (e.g. at the claws and facial regions). The method of [9] appears more robust than that of [18,19] but fails to localize small-scale salient features such as the eyes, knees, and claws of the Armadillo.

5.5. Feature point localization

To showcase the usefulness of our computed saliency maps, we apply them to the task of feature point localization on surface meshes and evaluate on the dataset of [22]. We choose this task as our application because it is a fundamental building block of geometry processing and shape analysis.

The dataset of [22] consists of 43 commonly used graphics meshes and the feature points annotated by 16 human subjects for each mesh. We evaluate six feature detectors: MS [6], SP [24], SDC [23], HKS [25], and ours. For MS, SP, SDC and HKS, Dutagaci et al. [22] detected feature points using the published source codes. For our method, we classify a mesh vertex to be a candidate feature point if it has a local maxima saliency value that is also higher than the average of all local maxima saliency values. After sorting these candidates in the descending order of saliency, we sequentially retain each point with the constraint that it has at least 0.15

Table 1
Performance (AUC) of salient point detection on the dataset of [20]. The first row is the list of evaluated methods: MC [51], MS [6], SRI [18], MR [19], SI [9], GR [8], and Ours. The second row shows the scores computed on the 19 categories (excluding Helix as in the original paper) of the dataset of [20] together. The remaining rows show the scores computed on each category separately. The highest score is highlighted in each row.

	MC	MS	SRI	MR	SI	GR	Ours
All Categories	0.7839	0.8028	0.7605	0.6826	0.7097	0.7402	**0.8168**
Airplane	0.8952	0.8356	**0.9004**	0.7404	0.8690	0.8135	0.8790
Ant	0.7728	**0.8806**	0.7975	0.7331	0.7039	0.7216	0.7925
Armadillo	0.8620	**0.9022**	0.7603	0.8165	0.7858	0.8183	0.8878
Bearing	0.7814	0.8313	0.6350	0.6445	0.5312	0.7729	**0.8555**
Bird	0.8442	0.7792	**0.8468**	0.7532	0.7878	0.7580	0.8232
Bust	**0.8134**	0.8120	0.7696	0.6307	0.6714	0.651	0.7690
Chair	0.7570	0.7821	**0.8398**	0.6665	0.7154	0.6956	0.8012
Cup	0.7845	0.7829	0.7888	0.5622	0.7891	0.6684	**0.8031**
Fish	**0.9432**	0.9109	0.9231	0.8451	0.8950	0.884	0.9015
Fourleg	**0.8613**	0.8394	0.7682	0.7996	0.8281	0.8047	0.8331
Glasses	0.5201	0.5981	**0.7057**	0.4969	0.5226	0.5343	0.6947
Hand	0.7895	**0.8242**	0.8159	0.6977	0.7440	0.8142	0.8066
Human	0.7515	**0.8015**	0.6661	0.6745	0.7085	0.6325	0.6890
Mech	**0.8880**	0.8231	0.8060	0.6098	0.5934	0.5745	0.8780
Octopus	0.7295	0.8339	0.7534	0.7063	0.6059	0.7278	**0.8796**
Plier	0.5677	0.7083	0.8718	0.5119	0.6370	0.732	**0.9128**
Table	0.7734	0.7458	0.8123	0.7102	0.7355	0.6769	**0.8317**
Teddy	0.6909	**0.7131**	0.5457	0.6160	0.6775	0.6379	0.6975
Vase	0.7747	0.7822	0.7874	0.7201	0.7819	0.8098	**0.8299**

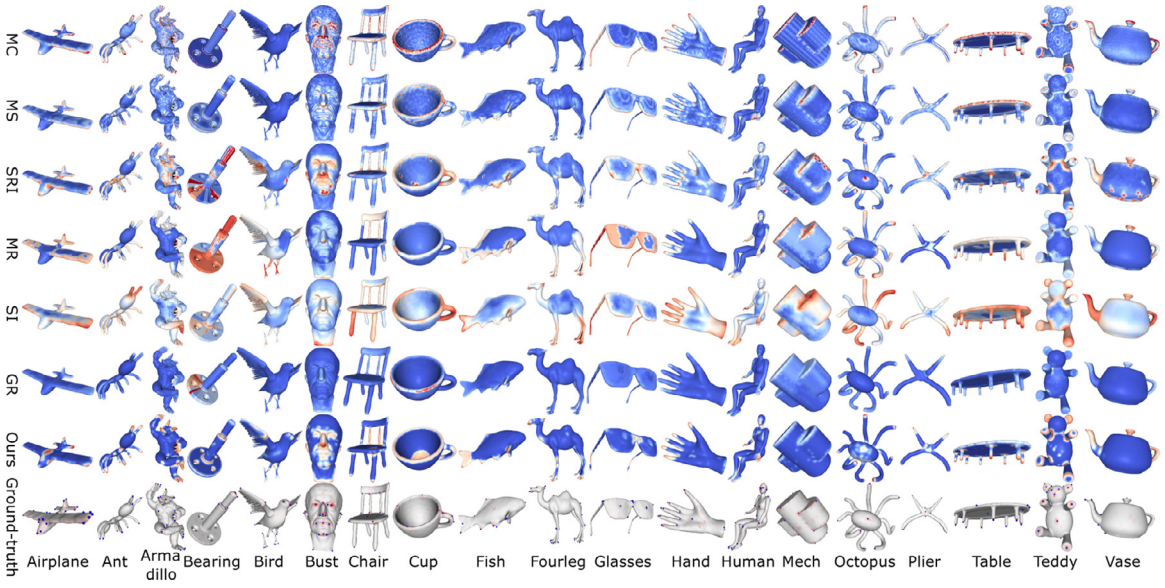


Fig. 7. Comparison of the saliency maps generated on the dataset of [20]. The first six rows correspond to the competing methods of MS [6], SRI [18], MR [19], SI [9], and GR [8]. The seventh row is our proposed method and the eighth shows the ground-truth salient points (in blue) and saliency distributions (in red). The columns correspond to the 19 categories (excluding Helix as in the original paper) of the dataset of [20]. Please zoom in to compare the details of the saliency maps.

geodesic distances to the already selected points. This way, we obtain a set of feature points that are spread out on mesh surfaces. We use the popular Recall, Precision, and F-measure scores for detector performance evaluation [22].

We show the Recall, Precision, and F-measure scores of the evaluated feature detectors in Fig. 9. The Recall score of a detector measures the fraction of true feature points it correctly finds in the ground-truth, and the Precision score measures the fraction of true feature points in the detector output. The F-measure score is the harmonic mean of Recall and Precision, which indicates the overall performance of a detector. As the localization error threshold is increased, all three scores of the evaluated detectors grow because the output points become more probably to be identified as correct matches. The MS, SP, and SDC methods achieve higher Recalls compared to HKS and our method by detecting excessive numbers of points. The cost, however, is that they produce many non-salient points and therefore score considerably lower on Precision. HKS, on the other end, sacrifices Recall for Precision by only generating sufficiently prominent points. Our method is shown to strike the best balance between Recall and Precision, identifying many true feature points while not incurring many false positive ones. This can be seen from the F-measure plot, where our method is shown to outperform other feature detectors from very small localization threshold.

5.6. Run time of our method

Table 3 reports the run time of our method and the competitors on a commodity PC with an Intel Dual Core 3.1GHZ CPU and a 4GB RAM. For each mesh, the run times of the main steps and the total time used are listed. It can be seen that our method scales well from medium-size meshes (e.g. the Bunny and the Dinosaur) to very large meshes (e.g. the Buddha and the Lucy). This high scalability would allow it to be used as an efficient preprocessing tool for many saliency-guided graphics applications. MC is the fastest method because it only only involves very efficient differential cur-

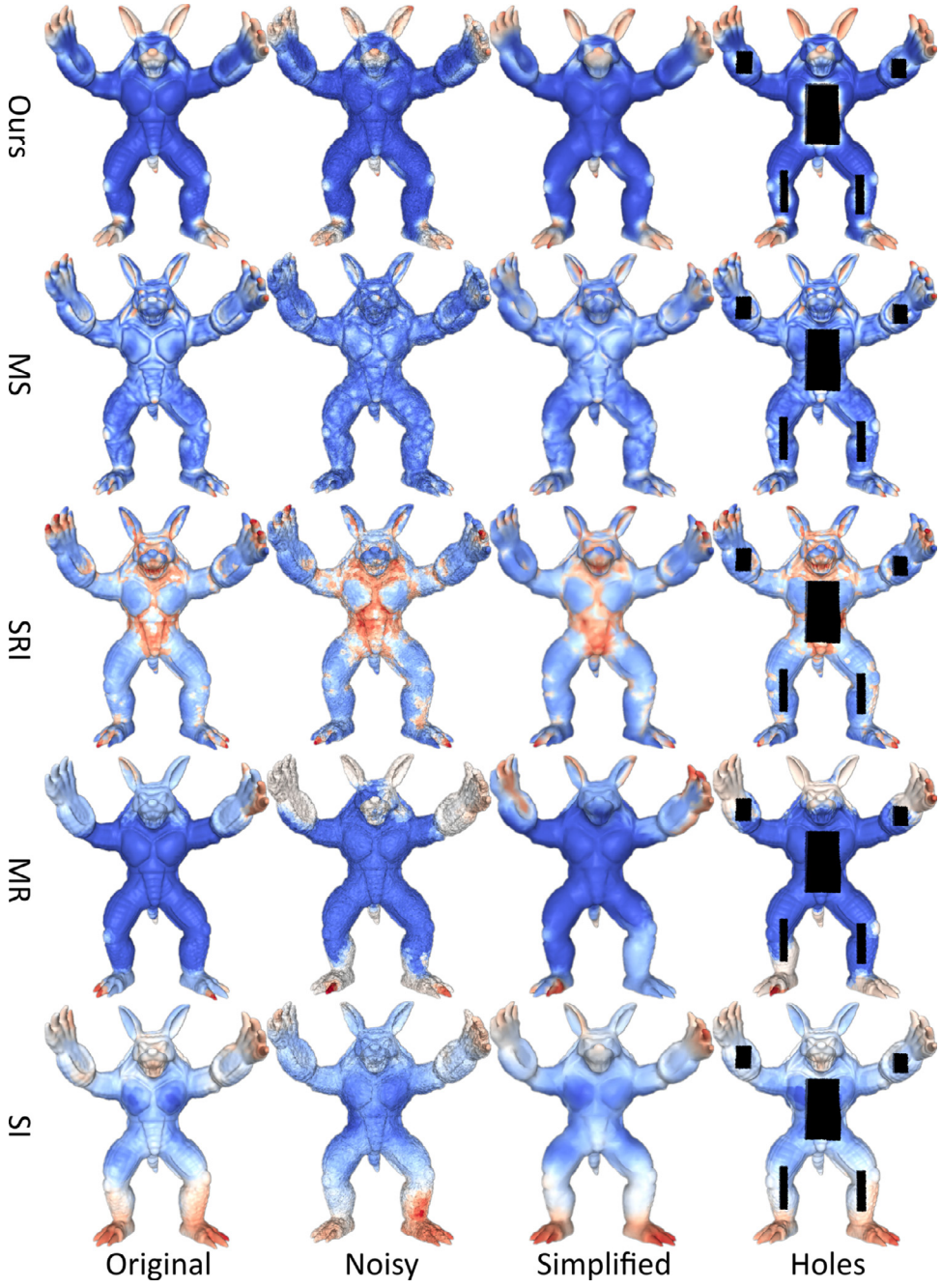


Fig. 8. Comparison of the robustness of our method with that of others. The columns from left to right correspond to the original Armadillo mesh, the noisy version (20% noises in vertex normal directions), the simplified version (with 5k vertices), and the broken version with holes. The competing methods are MS [6], SRI [18], MR [19], and SI [9].

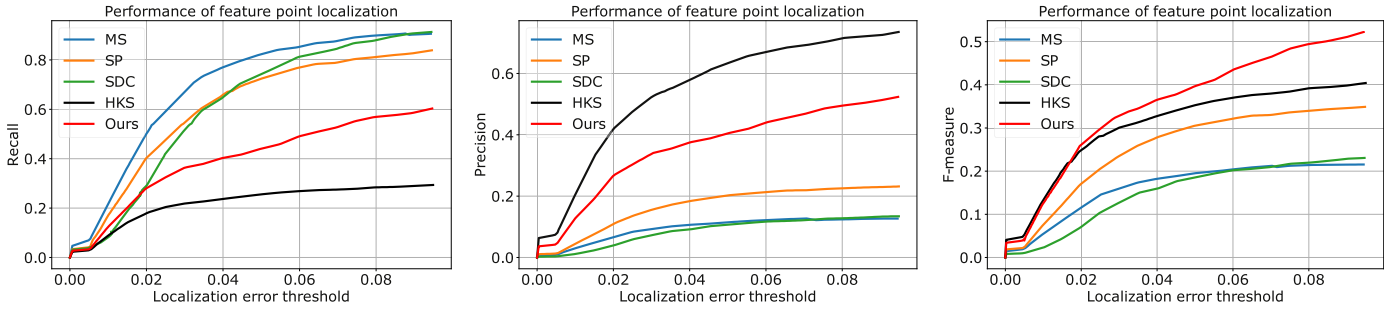


Fig. 9. Performance of feature point localization on the dataset of [22]. Four competing methods that are highly cited in the field are included: MS [6], SP [24], SDC [23], and HKS [25]. We use the popular Recall, Precision, and F-measure for performance evaluation [22].

Table 2
Performance (LCC) of saliency value prediction on the dataset of [20]. The first row is the list of evaluated methods: MC [51], MS [6], SRI [18], MR [19], SI [9], GR [8], and Ours. The second row shows the scores computed on the 19 categories (excluding Helix as in the original paper) of the dataset of [20] together. The remaining rows show the scores computed on each category separately. The highest score is highlighted in each row.

	MC	MS	SRI	MR	SI	GR	Ours
All Categories	0.3442	0.3131	0.2898	0.2158	0.1987	0.3083	**0.4303**
Airplane	0.4908	0.3270	0.4271	0.2221	0.3049	0.3555	**0.6049**
Ant	0.3779	0.4648	0.3349	0.3324	0.2116	0.3869	**0.6138**
Armadillo	0.4248	**0.4801**	0.1690	0.3096	0.2417	0.3570	0.4658
Bearing	0.2949	0.3055	0.1835	0.0760	0.0213	0.2588	**0.3578**
Bird	0.4594	0.3496	0.3738	0.2381	0.2371	0.3666	**0.5319**
Bust	**0.3018**	0.2979	0.2614	0.1005	0.1577	0.1025	0.2295
Chair	0.2484	0.2441	0.3574	0.2130	0.1803	0.3015	**0.4618**
Cup	**0.4011**	0.3624	0.3789	0.2094	0.3455	0.2289	0.3306
Fish	**0.5824**	0.4708	0.4412	0.3745	0.3499	0.4659	0.5303
Fourleg	**0.4211**	0.2945	0.3004	0.2558	0.2794	0.3055	0.4089
Glasses	0.1736	0.1499	0.2662	0.0680	0.1648	0.1880	**0.3825**
Hand	0.3904	0.2714	0.3669	0.2806	0.2013	**0.4492**	0.4208
Human	**0.3914**	0.3404	0.2579	0.2017	0.2020	0.1629	0.2528
Mech	0.3862	0.2903	0.2166	-0.0082	0.0670	0.02484	**0.4329**
Octopus	0.3029	0.3623	0.3319	0.1833	0.1048	0.3457	**0.5367**
Plier	0.2799	0.1426	0.3719	0.3532	0.1330	0.4037	**0.5450**
Table	0.3233	0.2315	0.3279	0.2484	0.2033	0.3612	**0.5225**
Teddy	0.2137	**0.2419**	0.0942	0.1302	0.1466	0.1813	0.2282
Vase	0.3810	0.3303	0.3160	0.2981	0.3300	0.3479	**0.3830**

Table 3
Run time of our method and the competitors in seconds. A: multi-scale metric computation from a mesh. **B**: saliency computation from a metric. **C**: vertex saliency interpolation. Please note that we implement our method, MC [51], and MS [6] in C++. We implement SRI [18], MR [19], and GR [8] in Matlab. The original implementation of SI [9] we use in this work is also in Matlab.

Mesh	#Vert	A	B	C	Ours	MC	MS	SRI	MR	SI	GR
Bunny	35k	40.3	3.0	0.1	43.5	0.5	12.2	56.9	35.8	13.7	43.4
Dinosaur	56k	42.0	2.8	0.4	45.2	1.8	28.2	100.2	67.3	23.6	86.3
Armadillo	172k	40.5	3.1	0.8	44.3	2.3	56.1	245.8	221.3	36.5	276.2
Dragon	437k	41.4	3.0	1.6	46.0	4.7	245.1	657.9	622.1	43.6	676.3
Buddha	543k	40.5	2.9	1.9	45.3	5.2	376.9	873.2	825.4	56.4	982.4
Lucy	604k	43.3	2.9	2.6	48.9	7.3	448.0	1178.2	912.5	67.4	1264.0

vature estimation on surface meshes. MS scales poorly because the multi-scale Gaussian filtering of curvatures becomes very expensive for large meshes. Despite in Matlab, the original implementation of SI remains efficient as it applies mesh simplification before saliency computation. In contrast, the methods of SRI, MR, and GR are quite time-consuming for especially large meshes such as Buddha and Lucy. We emphasize again that the methods of SRI, MR, SI, and GR are implemented in Matlab and could be more efficiently implemented in C++ in the future. We leave the fairer comparison of the computational costs of these methods as future work.

6. Conclusion and future work

We have proposed an accurate and robust sparse metric-based saliency detection method for 3D polygonal surface meshes. Our method was rigorously derived from optimizing the rarity principle of saliency while enforcing the sparsity principle of saliency. This makes it able to optimally discover a compact set of salient regions that have the maximum distinction from others. Our method was formulated as solving for the sparse eigenvector of a global metric, which enjoys the robustness to the flaws of surface noises, simplifications, and holes. The results on our eye fixation dataset, the Schelling saliency dataset of [20], and the feature localization dataset of [22] show that our method produces more accurate saliency estimations compared with existing ones.

We have selected 50 commonly used graphics meshes for our 3D eye fixation dataset construction. In the future, we could scale the construction to the SHREC2007 dataset [50] so that we can have each of the 400 meshes annotated with 3D eye fixations and Schelling saliency values [20]. This would greatly facilitate the large-scale benchmarking of saliency detection methods for further progress. Implementing more methods (e.g. of [17,40,41]) will also help this.

We have instructed each subject to sit in a distance of 95 – 110 cm from a 1680 × 1050 LED display to capture our 3D eye fixation dataset. This ensures that all of the meshes in our dataset have a proper size on the display. There is evidence that visual attention and the distance of observers are correlated [52]. This suggests us to explore the impact of the distance of subjects from the display on captured eye fixations in the future.

We have used a fixed number of sample points for mesh saliency computation in this work. This is not optimal as some large surfaces may be under-sampled while some small surfaces may be over-sampled. We are investigating the method of adaptive Poisson sampling [53] to generate high-quality saliency maps for both small and large meshes as future work.

Our assumption of saliency is a bottom-up computational approximation to the pre-attentive mechanism of the human visual system. As a result, it may fail to capture a few visually salient but not necessarily rare regions of a mesh, such as the chest of the Armadillo and the face of the Teddy, as shown in Fig. 4. To capture these challenging regions, more high-level cues like symmetries, segmentation, and semantic annotations are expected to be helpful. Benefiting from the large volume of image saliency annotations [54], deep learning methods have recently demonstrated excellent performance on image saliency detection [55–57]. However, the size of 3D mesh saliency datasets [20,58] is still very limited and may not well support the generalization of over-parameterized deep learning models. As a result, designing a more efficient representation of meshes with much fewer parameters for saliency

detection will be crucial for deep learning methods to generalize well. We are actively investigating this direction.

Different from visual saliency we addressed in this work, there is tactile saliency that measures the importance of surface regions for human-object interaction (e.g. grasp, press, and touch) [58]. The interaction-oriented nature of tactile saliency makes it a desirable complement to the traditional visual saliency for more effectively assessing the semantic importance of surface regions. We are seeking to jointly investigate visual saliency and tactile saliency as future work for 3D shape analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Shanfeng Hu: Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing - original draft. **Xiaohui Liang:** Resources, Supervision, Funding acquisition. **Hubert P. H. Shum:** Resources, Writing - review & editing, Supervision, Funding acquisition, Project administration. **Frederick W. B. Li:** Writing - review & editing. **Nauman Aslam:** Resources, Supervision, Funding acquisition, Project administration.

Acknowledgement

This work was supported in part by the Erasmus Mundus Action 2 Programme (Ref: 2014-0861/001-001), the Royal Society (Ref: IES\R2\181024), the Defence and Security Accelerator (Ref: DSTLX-1000140725), the National Key Research and Development Program of China (Ref: 2017YFB1002702), and the National Nature Science Foundation of China (Ref: 61572058).

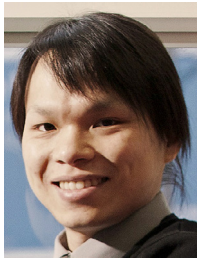
References

[1] A. Borji, L. Itti, State-of-the-art in visual attention modeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 185–207.
[2] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[3] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in neural information processing systems, 2006, pp. 545–552.
[4] X. Hou, L. Zhang, Saliency detection: a spectral residual approach, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[5] Y. Ji, H. Zhang, K.-K. Tseng, T.W. Chow, Q.J. Wu, Graph model-based salient object detection using objectness and multiple saliency cues, Neurocomputing 323 (2019) 188–202.
[6] C.H. Lee, A. Varshney, D.W. Jacobs, Mesh saliency, ACM Trans. Graph. 24 (3) (2005) 659–666.
[7] P. Shilane, T. Funkhouser, Distinctive regions of 3d surfaces, ACM Trans. Graph. 26 (2) (2007).
[8] J. Wu, X. Shen, W. Zhu, L. Liu, Mesh saliency with global rarity, Graph. Models 75 (5) (2013) 255–264.
[9] R. Song, Y. Liu, R.R. Martin, P.L. Rosin, Mesh saliency via spectral processing, ACM Trans. Graph. 33 (1) (2014) 1–17.
[10] R. Gal, D. Cohen-Or, Salient geometric features for partial shape matching and similarity, ACM Trans. Graph. 25 (1) (2006) 130–150.
[11] Y. Kim, A. Varshney, Saliency-guided enhancement for volume visualization, IEEE Trans. Vis. Comput. Graph. 12 (5) (2006) 925–932.
[12] Y. Kim, A. Varshney, Persuading visual attention through geometry, IEEE Trans. Vis. Comput. Graph. 14 (4) (2008) 772–782.
[13] Y. Miao, J. Feng, Perceptual-saliency extremum lines for 3d shape illustration, Vis. Comput. 26 (6–8) (2010) 433–443.
[14] V. Sundstedt, D. Gutierrez, O. Anson, F. Banterle, A. Chalmers, Perceptual rendering of participating media, ACM Trans. Appl. Percept. 4 (3) (2007) 15.
[15] Y.-S. Liu, M. Liu, D. Kihara, K. Ramani, Salient critical points for meshes, in: Proceedings of the ACM Symposium on Solid and Physical Modeling, ACM, 2007, pp. 277–282.
[16] R. McDonnell, M. Larkin, B. Hernández, I. Rudomin, C. O'Sullivan, Eye-catching crowds: saliency based selective variation, ACM Trans. Graph. 28 (3) (2009) 1–10.
[17] M. Feixas, M. Sbert, F. González, A unified information-theoretic framework for viewpoint selection and mesh saliency, ACM Trans. Appl. Percept. 6 (1) (2009) 1–23.
[18] G. Leifman, E. Shtrom, A. Tal, Surface regions of interest for viewpoint selection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 414–421.
[19] T. Pingping, C. Junjie, L. Shuhua, L. Xiuping, L. Ligang, Mesh saliency via ranking unsalient patches in a descriptor space, Comput. Graph. 46 (2015) 264–274.
[20] X. Chen, A. Saparov, B. Pang, T. Funkhouser, Schelling points on 3d surface meshes, ACM Trans. Graph. 31 (4) (2012) 1–12.
[21] X.-T. Yuan, T. Zhang, Truncated power method for sparse eigenvalue problems, J. Mach. Learn. Res. 14 (1) (2013) 899–925.
[22] H. Dutagaci, C. Cheung, A. Godil, Evaluation of 3d interest point detection techniques via human-generated ground truth, Vis Comput 28 (9) (2012) 901–917.
[23] J. Novatnack, K. Nishino, Scale-dependent 3d geometric features, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.
[24] U. Castellani, M. Cristani, S. Fantoni, V. Murino, Sparse points matching by combining 3d mesh saliency with statistical descriptors, Comput. Graph. Forum 27 (2) (2008) 643–652.
[25] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, Comput. Graph. Forum 28 (5) (2009) 1383–1392.
[26] Y. Kim, A. Varshney, D.W. Jacobs, F. Guimbretière, Mesh saliency and human eye fixations, ACM Trans. Appl. Percept. 7 (2) (2010) 1–13.
[27] A.M. Treisman, G. Gelade, A feature-integration theory of attention, Cogn. Psychol. 12 (1) (1980) 97–136.
[28] C. Koch, S. Ullman, Shifts in selective visual attention: towards the underlying neural circuitry, in: Matters of Intelligence, 188, Springer Netherlands, 1987, pp. 115–141.
[29] Q. Wang, Y. Yuan, P. Yan, Visual saliency by selective contrast, IEEE Trans. Circuits Syst. Video Technol. 23 (7) (2012) 1150–1155.
[30] Q. Wang, Y. Yuan, P. Yan, X. Li, Saliency detection by multiple-instance learning, IEEE Trans. Cybern. 43 (2) (2013) 660–672.
[31] G. Zhu, Q. Wang, Y. Yuan, P. Yan, Learning saliency by mrf and differential threshold, IEEE Trans. Cybern. 43 (6) (2013) 2032–2043.
[32] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d shape descriptors, in: Proceedings of the Eurographics Symposium on Geometry Processing, Eurographics Association, 2003, pp. 156–164.
[33] R. Huang, W. Feng, J. Sun, Saliency and co-saliency detection by low-rank multiscale fusion, in: IEEE International Conference on Multimedia and Expo, 2015, pp. 1–6.
[34] R. Huang, W. Feng, J. Sun, Color feature reinforcement for cosaliency detection without single saliency residuals, IEEE Signal Process. Lett. 24 (5) (2017) 569–573.
[35] R. Huang, W. Feng, Z. Wang, Y. Xing, Y. Zou, Exemplar-based image saliency and co-saliency detection, Neurocomputing (2019).
[36] H. Yee, S. Pattanaik, D.P. Greenberg, Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments, ACM Trans. Graph. 20 (1) (2001) 39–65.
[37] P. Longhurst, K. Debattista, A. Chalmers, A gpu based saliency map for high-fidelity selective rendering, in: Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, ACM, 2006, pp. 21–29.
[38] R. Mantiuk, K. Myszkowski, S. Pattanaik, Attention guided mpeg compression for computer animations, in: Proceedings of the 19th Spring Conference on Computer Graphics, ACM, 2003, pp. 239–244.
[39] I.P. Howard, Seeing in depth, University of Toronto Press, 2002.
[40] Y. Zhao, Y. Liu, Y. Wang, B. Wei, J. Yang, Y. Zhao, Y. Wang, Region-based saliency estimation for 3d shape analysis and understanding, Neurocomputing 197 (2016) 1–13.
[41] S.-W. Jeong, J.-Y. Sim, Saliency detection for 3d surface geometry using semi-regular meshes, IEEE Trans. Multimedia 19 (12) (2017) 2692–2705.
[42] M. Garland, P.S. Heckbert, Surface simplification using quadric error metrics, in: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1997, pp. 209–216.
[43] S. Howlett, J. Hamill, C. O'Sullivan, Predicting and evaluating saliency for simplified polygonal models, ACM Trans. Appl. Percept. 2 (3) (2005) 286–308.
[44] M. Gu, S. Hu, X. Wang, X. Liang, X. Shen, A. Qin, Saliency-driven depth compression for 3d image warping, in: Pacific Graphics (short paper), 2014, pp. 91–96.
[45] A. Secord, J. Lu, A. Finkelstein, M. Singh, A. Nealen, Perceptual models of viewpoint preference, ACM Trans. Graph. 30 (5) (2011) 1–12.
[46] U. Feige, D. Peleg, G. Kortsarz, The dense k-subgraph problem, Algorithmica 29 (3) (2001) 410–421.
[47] F.A. Farris, The gini index and measures of inequality, Am. Math. Mon. 117 (10) (2010) 851–864.
[48] C.D. Godsil, G. Royle, C. Godsil, Algebraic graph theory, Springer New York, 2001.
[49] The stanford 3d scanning repository. http://graphics.stanford.edu/data/3Dscanrep/
[50] Shrec'2007 watertight mesh database. http://watertight.ge.imati.cnr.it/
[51] M. Meyer, M. Desbrun, P. Schroder, A.H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, Math. Visualiz. 6 (8–9) (2002) 35–57.

[52] C. Wardak, S. Denève, S.B. Hamed, Focused visual attention distorts distance perception away from the attentional locus, Neuropsychologia 49 (3) (2011) 535–545.
[53] D.-M. Yan, P. Wonka, Gap processing for adaptive maximal poisson-disk sampling, ACM Trans. Graph. 32 (5) (2013) 148.
[54] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, A. Borji, Revisiting video saliency prediction in the deep learning era, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[55] S. He, R.W. Lau, Q. Yang, Exemplar-driven top-down saliency detection via deep association, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5723–5732.
[56] G. Li, Y. Yu, Deep contrast learning for salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 478–487.
[57] H. Xiao, J. Feng, Y. Wei, M. Zhang, S. Yan, Deep salient object detection with dense connections and distraction diagnosis, IEEE Trans. Multimedia 20 (12) (2018) 3239–3251.
[58] M. Lau, K. Dev, W. Shi, J. Dorsey, H. Rushmeier, Tactile mesh saliency, ACM Trans. Graph. 35 (4) (2016) 52.

Shanfeng Hu is a final-year PhD student from the Department of Computer and Information Sciences at Northumbria University, UK. His recently submitted thesis is about incorporating structured metric representations for 3D geometric deep learning. He also works part-time as a Senior Data Scientist at tenokonda Ltd. on deep learning for quantitative data analytics. He received his Bachelor degree from the Department of Computer Science and Technology at Henan University, China.

Xiaohui Liang received his Ph.D. degrees in computer science and engineering from Beihang University, China. He is currently a Professor, working at the School of Computer Science and Engineering at Beihang University. His main research interests include computer graphics and animation, visualization and virtual reality.



Hubert P. H. Shum is an Associate Professor in Computer Science at Northumbria University, UK, and the Director of Research and Innovation of the Computer and Information Sciences Department. Before that, he was a Senior Lecturer at Northumbria University, a Lecturer at the University of Worcester and a Post-doctoral Researcher at RIKEN Japan. He received his PhD degree from the University of Edinburgh, his Master and Bachelor degrees from the City University of Hong Kong. He serves as an Associate Editor in Computer Graphics Forum. He has over 100 publications in the areas of computer graphics, computer vision, human motion analysis and machine learning.

Frederick W. B. Li received a B.A. and an M.Phil. degree from Hong Kong Polytechnic University, and a Ph.D. degree from the City University of Hong Kong. He is currently an Assistant Professor at Durham University. He served as a guest Editor for several special issues of World Wide Web Journal, Journal of Multimedia and JDET. He has served as a Program Co-Chair of ICWL for four years and IDET for two years. His research interests include distributed virtual environments, computer graphics and e-Learning systems.

Nauman Aslam is an Associate Professor in the Department of Computer Science and Digital Technologies, Northumbria University, UK. He is also an Adjunct Assistant Professor at Dalhousie University, Canada. He received his PhD in Engineering Mathematics from Dalhousie University, Halifax, Nova Scotia, Canada in 2008. Prior to joining Northumbria University, he worked as an Assistant Professor at Dalhousie University, Canada. His research interests include wireless sensor network, energy efficiency, security and WSN health applications.