Saliency Detection for 3D Surface Geometry Using Semi-regular Meshes

Se-Won Jeong and Jae-Young Sim, *Member, IEEE*

***Abstract*—In this paper, a unified detection algorithm of view-independent and view-dependent saliency for three-dimensional mesh models is proposed. While the conventional techniques use the irregular meshes, we adopt the semi-regular meshes to overcome the drawback of irregular connectivity for saliency computation. We employ the angular deviation of normal vectors between neighboring faces as geometric curvature features, which are evaluated at hierarchically structured triangle faces. We construct a fully connected graph at each level of semi-regular mesh, where the face patches serve as graph nodes. At the base mesh level, we estimate the saliency as the stationary distribution of random walk. At the higher level meshes, we take the maximum value between the stationary distribution of random walk at the current level and an upsampled saliency map from the previous coarser scale. Moreover, we also propose a view-dependent saliency detection method that employs the visibility feature in addition to the geometric features to estimate the saliency with respect to a selected viewpoint. Experimental results demonstrate that the proposed saliency detection algorithm captures global conspicuous regions reliably and detects locally detailed geometric features faithfully, compared with the conventional techniques.**

***Index Terms*—Mesh saliency, random walk, saliency detection, semi-regular meshes, view-dependent saliency.**

I. Introduction

THESE days a huge amount of multimedia data is created everyday, e.g., speech, text, image, video, and 3D graphics. Therefore, it is drawing much attention to analyze and extract important information from given visual data automatically, which can be applied to facilitate various applications such as event detection [1], video classification [2] and video summarization [3]. Saliency detection is one of the most representative visual analysis techniques which detects visually distinctive and thus noticeable regions from digitized visual media automatically, patterned on the selective perception characteristics of the human visual system (HVS). Most of the saliency detection techniques extract visually distinctive regions by computing feature contrast compared to neighboring regions.

There has been extensive research for saliency detection. Itti *et al.* [4] calculated the difference of features from a target region to its surrounding regions based on the center-surround concept. Hou *et al.* [5] performed the Fourier transform on an input image and applied a filtering in frequency domain to detect spectral residual as saliency. Goferman *et al.* [6] computed local and global saliency using the color difference between image patches. Costa *et al.* [7] constructed a graph on an image and estimated saliency as the steady-state distribution of random walk on the graph. Borji and Itti [8] detected saliency using global rarity of image patches. Imamoglu *et al.* [9] computed saliency using the likelihood of features defined on the wavelet transform domain. Kim *et al.* [10] obtained multiscale saliency distributions by simulating graph-based random walk with restart on an image. Yu *et al.* [11] estimated saliency by spreading the attention across the Gestalt graph using power iteration clustering. Also, saliency detection has been applied to many applications of visual media processing, e.g., image retrieval [12], segmentation [13], and video resizing [14].

3D visual contents have attracted people's attention with the advent of various emerging applications such as virtual and augmented reality [15], 3D printers, and archive of digital heritage [16]. One of the most widely used representation tools of 3D visual contents is polygon meshes. The polygon mesh model is composed of vertices and faces, where the vertices are sampled on the surface of a 3D model and neighboring vertices are connected together to generate the faces. In general, we use larger numbers of faces and vertices to represent detailed 3D geometry for more realistic and immersive visual applications. Therefore, 3D meshes require tremendous amount of storage and require huge computational resources for analysis. To reduce storage space and computational complexity in various practical 3D applications, it is crucial to extract visually salient regions selectively from massive 3D models, which are then processed with high priorities.

During the past decade, several techniques of saliency detection for 3D irregular meshes have been proposed. Early studies were focused on detecting locally distinctive regions by computing the feature contrast compared to neighboring regions. Lee *et al.* [17] computed a mean curvature at each vertex and measured the difference of Gaussian filtered curvatures at various scales. Feixas *et al.* [18] estimated mesh saliency using mutual information based visibility features at vertices. Shilane and Funkhouser [19] found distinctive vertices by comparing

Manuscript received June 25, 2016; revised January 28, 2017 and April 23, 2017; accepted May 26, 2017. Date of publication June 1, 2017; date of current version November 15, 2017. This work was supported in part by Global Ph.D Fellowship Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2015H1A2A1029729), and in part by the NRF under Grant 2016R1D1A1A09919618. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jingdong Wang. *(Corresponding author: Jae-Young Sim.)*

The authors are with the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, South Korea (e-mail: swjeong@unist.ac.kr; jysim@unist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2017.2710802

the harmonic shape descriptor with neighboring vertices. More recently, several approaches have been investigated to capture the overall saliency features of 3D meshes as well using global geometric properties. Leifman *et al.* [20] used the geodesic distance and spin image to estimate the distinctness of each vertex, and spread the distinctness values near the vertices with local extrema. Wu *et al.* [21] detected local saliency by computing the difference of Zernike coefficients, and also estimated global rarity. Song *et al.* [22] obtained mesh spectrum by eigen-analysis of Laplacian matrix, and computed the difference between the original spectrum and a smoothed spectrum. Jeong and Sim [23] computed multiscale saliency values at each vertex by performing random walk on a locally connected graph constructed for an irregular mesh. Liu *et al.* [24] computed saliency based on absorbing Markov chain where the patches associated with the minimum feature variance are selected as absorbing nodes.

Most of the conventional saliency detection techniques first compute local features at multiple scales of an input irregular mesh, and then combine them together to obtain a final saliency distribution. For example, Lee *et al.* [17] computed a weighted sum of the differences of Gaussian filtered curvature maps with various kernel sizes, and Leifman *et al.* [20] calculated the distinctness of vertices at three simplified levels of an irregular mesh. However, due to the irregular connectivity of 3D meshes, it often requires additional computational complexity of searching neighboring vertices to relate the extracted geometric features across multiple scales or to combine the saliency maps computed at multiple scales. Lee *et al.* found different sets of neighboring vertices for each vertex at different sizes of Gaussian kernels, respectively. Jeong and Sim [23] estimated a curvature feature at each vertex in a higher scale mesh as the averaged feature of the K-nearest neighboring vertices in the simplified mesh. Moreover, the obtained neighboring vertices often yield spatially unbalanced distributions across the vertices, since the vertices are irregularly distributed resulting in locally different sampling densities on the surface of a 3D model.

In addition, the conventional techniques provide only a view-independent saliency distribution over the whole surface of an input 3D model, even though 3D models can be dynamically visualized from varying viewpoints especially for interactive applications. This means that the static saliency distribution may not exactly coincide with the actual saliency perceived by HVS according to the changing viewpoint. Additionally, it could be better to reduce the computational complexity of saliency estimation by considering only the visible surface area instead of the whole surface of a 3D model.

In this paper, we propose a novel saliency detection algorithm for 3D mesh models. To overcome the drawback of irregular mesh connectivity, we adopt semi-regular meshes by resampling the surface of input irregular meshes. In semi-regular meshes, the triangle faces at multiple scales are hierarchically associated through systematic subdivision structures. Moreover, the vertices are resampled uniformly on the whole surface area, and therefore the triangle faces at a same scale have almost similar sizes and shapes to one another. By taking advantage of these characteristics of semi-regular meshes, the proposed algorithm can access spatially balanced and topologically consistent neighboring faces or vertices at all scales without explicit computation, which facilitates efficient multiscale saliency detection.

We first assign multiscale geometric features to the triangle faces in a semi-regular mesh by computing the angular deviation of neighboring face normals through regularly structured face hierarchy. We construct a fully-connected directional graph at each level of semi-regular mesh, which takes the faces as nodes. We design the weights of graph edges and perform random walk simulation to obtain the stationary distribution as a saliency map. We combine the stationary distribution of random walk at a current level with the saliency distribution at the coarse level by taking their maximum saliency values. Moreover, based on the unified saliency detection framework, we devise a view-dependent dynamic saliency detection scheme which adaptively changes the saliency distribution according to varying viewpoints. Experimental results show that the proposed algorithm detects global salient regions reliably and also captures local geometric details faithfully, compared with the conventional techniques.

Note that, in this work, we adopt the random walk framework for saliency detection of 3D meshes as did in our previous work [23]. However, the proposed algorithm has the major novelties compared to [23]. First of all, [23] developed a vertex-wise saliency estimation algorithm for *irregular* meshes, and thus accesses the K-nearest neighboring vertices to upsample saliency distributions. However, the proposed algorithm is the first attempt to use *semi-regular* meshes for saliency detection of 3D surface geometry, and computes multiscale face-wise saliency distributions by exploiting the systematic hierarchical structure of subdivision faces. Moreover, [23] used a *locally connected undirected* graph for random walk simulation which is hard to capture global saliency, however, the proposed algorithm constructs a *fully-connected directed* graph and is able to detect local saliency as well as global saliency faithfully.

The remainder of the paper is organized as follows. Section II introduces background knowledge of semi-regular meshes and random walk. Sections III and IV propose the multiscale saliency detection algorithm and the view-dependent saliency detection algorithm, respectively. Section V provides the experimental results. Finally, Section VI concludes the paper.

II. Background

A. Semi-regular Meshes

Triangular meshes have been widely used to represent 3D real-world objects, where the vertices are sampled at arbitrary positions on the surface of 3D objects and nearby vertices are connected together to generate triangle faces. Therefore, as shown in Fig. 1(a), the resulting triangles have different sizes and shapes from one another, and thus yield irregular connectivity which degrades the performance of various geometry processing for 3D meshes. In order to overcome this drawback, semi-regular meshes have been introduced [25], where the vertices are resampled at uniformly distributed positions on the 3D surface. In semi-regular meshes, most vertices are connected to the

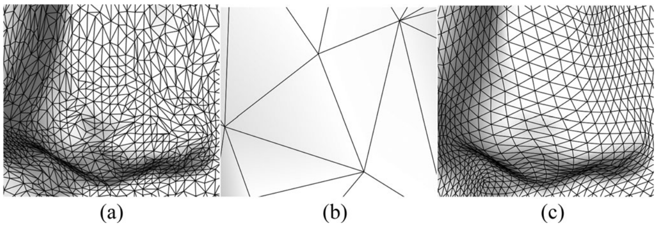


Fig. 1. Remeshing on Venus model: (a) an input irregular mesh, (b) the simplified base mesh, and (c) the refined semi-regular mesh.

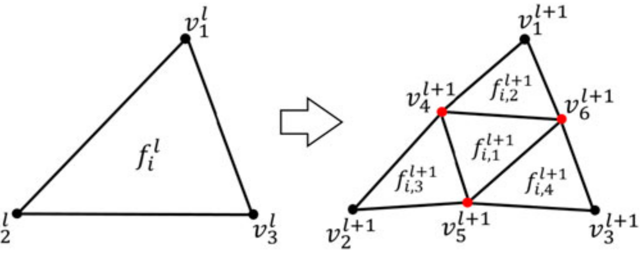


Fig. 2. One-to-four triangle subdivision.

same number of triangles with similar sizes and shapes. Fig. 1(c) shows an example of semi-regular mesh converted from an input irregular mesh in Fig. 1(a). We see that most vertices are regularly distributed on the 3D surface and connected to 6 triangles.

The process of converting an irregular mesh into a semi-regular mesh is called remeshing. A typical remeshing technique is composed of parameterization and refinement. For example, an input irregular mesh in Fig. 1(a) is first simplified to a base mesh as shown in Fig. 1(b) by reducing the number of triangles. During mesh simplification, original vertex positions are additionally embedded onto the triangles in the base mesh by parameterization. Each base triangle is then subdivided into the smaller triangles which are also subdivided iteratively to construct a hierarchical structure of triangle faces. Fig. 2 shows the one-to-four refinement of triangle, where the triangle f_i^l at the l-th level composed of the three vertices v_1^l, v_2^l, and v_3^l, is refined into the four smaller triangles $f_{i,1}^{l+1}$, $f_{i,2}^{l+1}$, $f_{i,3}^{l+1}$, and $f_{i,4}^{l+1}$ at the $(l+1)$th level by inserting new vertices v_4^{l+1}, v_5^{l+1}, and v_6^{l+1}. The positions of newly added vertices are resampled on the surface of 3D object by using the parameterized information of original vertex positions.

Note that the semi-regular meshes have benefits for various geometry processing tasks. First, all the vertices except that of the base mesh have the same number of connected triangles. Thus there is no need to describe or store the connectivity information explicitly. In addition, the resampled vertices yield a relatively uniform and regular distribution on the surface of 3D object, which facilitates the application of traditional signal processing techniques to 3D geometry processing. In particular, geometry data compression techniques achieve state-of-the-art performances with semi-regular meshes [25], [26], where the geometry data in a whole 3D semi-regular mesh are described completely by that of the base mesh and the refinement vectors of higher level vertices predicted from the lower level vertices [27].

B. Random Walk

Random walk (RW) is a mathematical modeling of random movement on a graph, which has been widely used to detect saliency of images [7], [10], [28]. In general, when watching an image, we continually change the gaze direction of eyes from one patch to another according to the appearance features in an image. Hence the saliency at a certain area of an image can be estimated as the visiting frequency of the gaze direction of eyes to the corresponding image patch. This can be naturally simulated by RW on a graph constructed over an image where local image patches are regarded as graph nodes. Let $\mathcal{G}(\mathcal{N}, \mathcal{E})$ be a directional weighted graph, where $\mathcal{N}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively. Each edge $e_{ij} \in \mathcal{E}$ connects two nodes from the j-th node to the i-th node, and is assigned a weight w_{ij}. RW is simulated with a Markov chain characterized by $|\mathcal{N}| \times |\mathcal{N}|$ matrix $\mathbf{P}$, whose (i,j)th element $\mathbf{P}(i,j)$ represents the transition probability that the random walker moves from the j-th node to the i-th node

$$\mathbf{P}(i,j) = \begin{cases} w_{ij}/W_j, & \text{if } e_{ij} \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $W_j = \sum_k w_{kj}$.

When the graph is irreducible and aperiodic, the Markov chain yields a unique stationary distribution $\boldsymbol{\pi}$ satisfying

$$\boldsymbol{\pi} = \mathbf{P}\boldsymbol{\pi} \tag{2}$$

where

$$\boldsymbol{\pi} = [\boldsymbol{\pi}(1), \boldsymbol{\pi}(2), \cdots, \boldsymbol{\pi}(|\mathcal{N}|)]^T. \tag{3}$$

$\boldsymbol{\pi}(i)$ denotes the probability of walker's visit to the i-th node. In general, $\boldsymbol{\pi}$ is obtained by iterative multiplication of the transition matrix $\mathbf{P}$ based on (2). However, when the Markov chain is derived from an undirected graph, i.e., $w_{ij} = w_{ji}$, $\boldsymbol{\pi}$ is in detailed balance with $\mathbf{P}$ and thus directly computed as [29]

$$\boldsymbol{\pi}(i) = \frac{W_i}{\sum_k W_k}. \tag{4}$$

III. Multiscale Geometric Saliency Detection

We propose a novel saliency detection algorithm for 3D semi-regular meshes, which provides static saliency distributions at multiple scales. At each face of a semi-regular mesh, we first compute a local curvature feature and a histogram of angular deviation of face normals. We obtain multiscale saliency distributions by performing RW on a face-based graph at each level, respectively, which are then combined together based on the subdivision hierarchy to capture the global and local saliency simultaneously.

A. Face-Based Geometric Features

The conventional saliency detection techniques for 3D meshes usually compute vertex-wise saliency values [17], [19]–[21], [23], which are hard to be related to one another through multiple scales due to the irregular mesh connectivity. We may address this problem with the semi-regular meshes by

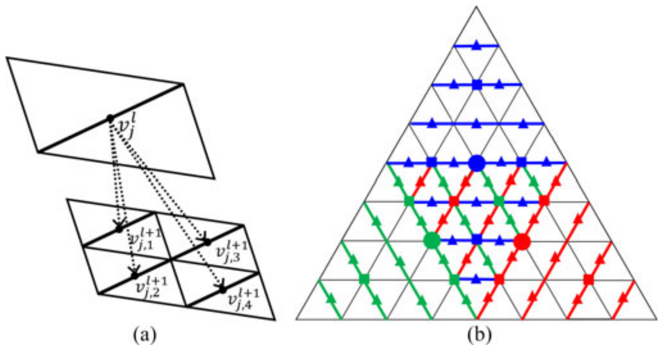


Fig. 3. Hierarchical vertex partitioning in semi-regular meshes. (a) An edge-based vertex tree where a parent vertex v_j^l at the lth level is associated with the four children vertices $v_{j,1}^{l+1}$, $v_{j,2}^{l+1}$, $v_{j,3}^{l+1}$, and $v_{j,4}^{l+1}$ at the $(l+1)$th level. (b) The resulting clusters of vertices, derived from the three circled vertices, respectively, are illustrated in different colors.

systematically clustering the vertices into hierarchical patches and estimating the patch-based saliency. In a semi-regular mesh, the vertices yield an edge-based tree structure as shown in Fig. 3(a), where a vertex v_j^l at the l-th level has the four children vertices $v_{j,1}^{l+1}$, $v_{j,2}^{l+1}$, $v_{j,3}^{l+1}$, and $v_{j,4}^{l+1}$ at the $(l+1)$th level [25]. Fig. 3(b) illustrates the three patches of vertices in different colors, which include all the offspring vertices derived from the three seed vertices indicated by circles, respectively. We see that the resulting patches partition the vertices exclusively without any overlapping, however, their shapes are not locally compact and their regions are intersected with one another within a same surface region.

On the contrary, triangle faces at a certain level in a semi-regular mesh exhibit similar sizes and shapes to one another, and occupy exclusive and locally compact regions on the surface of 3D object. Moreover, as shown in Fig. 2, it is simple and clear to construct a hierarchical quad-tree structure of faces, since the four children triangles $f_{i,1}^{l+1}$, $f_{i,2}^{l+1}$, $f_{i,3}^{l+1}$, and $f_{i,4}^{l+1}$ are subdivided from a parent triangle f_i^l at the coarser level. Therefore, in this work, we take triangle faces as geometric primitives for feature extraction and saliency computation.

In general, the normal vectors between neighboring faces yield large deviation on curved and uneven surface regions, while that on flat and smooth regions yield small deviation [30]. Thus we basically use the angular deviation of face normals as geometric features to estimate the local surface characteristics in semi-regular meshes. Let $\boldsymbol{\rho}(f)$ be the normal vector of a face f. The angle θ_{ik}^l between the face normals of f_i^l and f_k^l can be computed as

$$\theta_{ik}^l = \arccos(\boldsymbol{\rho}(f_i^l) \bullet \boldsymbol{\rho}(f_k^l)) \tag{5}$$

where $\bullet$ denotes the inner product operation.

Let $\Phi(f_i^l)$ be the set of the 1-ring neighboring faces to f_i^l at the l-th level, which share the common edges or vertices with f_i^l. We define a local curvature feature ψ_i^l at f_i^l as the average angle of θ_{ik}^l's associated with all the faces in $\Phi(f_i^l)$, given by

$$\psi_i^l = \frac{\sum_{f_k^l \in \Phi(f_i^l)} \theta_{ik}^l}{|\Phi(f_i^l)|}. \tag{6}$$

We additionally normalize ψ_i^l at each level, respectively, as

$$\bar{\psi}_i^l = \min\left(1, \frac{\psi_i^l}{\psi_{\text{ref}}^l}\right) \tag{7}$$

where ψ_{ref}^l denotes the minimum value between the maximum value of ψ_i^l and the value four times larger than the average of ψ_i^l's for all the faces at the l-th level. We also employ the probability distribution of the angular deviation for the faces in $\Phi(f_i^l)$ as a geometric feature of f_i^l. We construct a normalized histogram $\mathbf{h}_i^l$ of θ_{ik}^l's with 10 bins, where the first nine bins cover the range of angle from 0° having the same bin size of $\psi_{\text{ref}}^l/10$, and the last bin covers the remaining range of angle up to 180°.

B. Graph Modeling

We design a face-based graph structure to detect salient regions in a semi-regular mesh. There have been several graph-based approaches which take the vertices as graph nodes and compute edge weights using the feature differences between neighboring vertices [22]–[24]. However, they consider local connectivity information only to compute saliency values, which is hard to capture global saliency in a whole 3D model. In contrary, we construct a fully-connected graph in this work to detect both of the local and global saliency faithfully. In practice, at each level l in a semi-regular mesh, we construct a fully-connected directional graph $\mathcal{G}^l(\mathcal{N}^l, \mathcal{E}^l)$, where $\mathcal{N}^l$ is the set of nodes and $\mathcal{E}^l$ is the set of edges, respectively. A node $n_i^l \in \mathcal{N}^l$ represents each face f_i^l which is assigned the features of normalized curvature $\bar{\psi}_i^l$ and histogram $\mathbf{h}_i^l$. A directional edge $e_{ij}^l \in \mathcal{E}^l$ connects two nodes from the j-th node n_j^l to the i-th node n_i^l.

We estimate the saliency distribution at the l-th level in a semi-regular mesh by performing the RW simulation on the associated graph $\mathcal{G}^l$. As mentioned in Section II, the probability of random walk via each edge is determined by the associated edge weight: the walker is highly probable to move from n_j^l to n_i^l when the edge e_{ij}^l has a large weight w_{ij}^l. We also reflect the concept of center-surround contrast to design an edge weighting scheme such that w_{ij}^l should be large when the geometric features at f_i^l are highly distinct from that of f_j^l. Specifically, we compute the edge weight as

$$w_{ij}^l = \frac{1}{1 + \exp\left(-\frac{\bar{\psi}_i^l - \bar{\psi}_j^l}{\kappa_1}\right) \cdot \exp\left(-\frac{\beta(\mathbf{h}_i^l, \mathbf{h}_j^l)}{\kappa_2}\right)} \tag{8}$$

where $\beta(\mathbf{h}_i^l, \mathbf{h}_j^l)$ is the Bhattacharyya distance [31] which measures the dissimilarity between $\mathbf{h}_i^l$ and $\mathbf{h}_j^l$. We set $\kappa_1 = 0.3$ and $\kappa_2 = 1.1$ empirically. The exponential terms are positive, and thus w_{ij}^l belongs to the range of $[0, 1]$. Note that the edge weights in (8) force the random walker to visit more curved regions more frequently, since w_{ij}^l becomes large as n_i^l has a high curvature value and n_j^l has a low curvature value.

Many graph-based saliency detection techniques for images employ the spatial distance to design edge weights based on the center prior assumption that the foreground salient objects are usually positioned near the image center. For example, w_{ij}^l is set

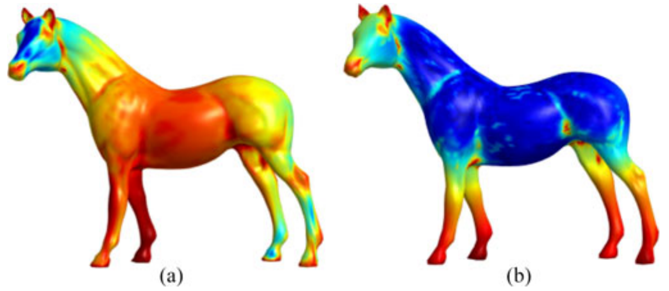


Fig. 4. Effect of the center prior assumption. The resulting saliency distributions at the third level of the Horse model: (a) with the center prior assumption and (b) without the center prior assumption.

to be small when n_i^l is geometrically far from n_j^l, which emphasizes saliency values near the image center by using the Gaussian distribution [10], [32]. However, such assumption does not hold in geometric saliency detection for 3D surfaces. Fig. 4 compares the resulting saliency distributions in 'Horse' model by using the proposed algorithm with and without the distance based edge weighting scheme. As shown in Fig. 4(a), the distance based weighting mainly highlights the smooth body, which is the central region of the whole 3D model, to be more salient than the head and legs which are perceived as more salient by HVS. Therefore, we do not employ the traditional distance based edge weighting scheme for 3D saliency detection. Fig. 4(b) shows the resulting saliency distribution without the center prior assumption where we see that geometrically salient regions are faithfully detected.

C. Multiscale Saliency Detection

We obtain the transition matrix $\mathbf{P}^l$ for the Markov chain defined on the graph $\mathcal{G}^l$ using the edge weights in (8)

$$\mathbf{P}^l(i,j) = \begin{cases} w_{ij}^l / W_j^l, & \text{if } e_{ij}^l \in \mathcal{E}^l \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $W_j^l = \sum_k w_{kj}^l$. Note that $\mathcal{G}^l$ is a fully-connected graph, and thus the matrix $\mathbf{P}^l$ satisfies irreducible and positive recurrent condition [33], which guarantees a unique stationary distribution $\boldsymbol{\pi}^l$ of RW, satisfying

$$\boldsymbol{\pi}^l = \mathbf{P}^l \boldsymbol{\pi}^l. \tag{10}$$

However, $\mathcal{G}^l$ is a directional graph, i.e., $w_{ij}^l \neq w_{ji}^l$, and hence we find a unique stationary distribution by using an iterative method.

We can estimate the single-scale saliency distribution at the l-th level faces in a semi-regular mesh by normalizing the resulting stationary distribution $\boldsymbol{\pi}^l$ in (10), such that

$$\bar{\boldsymbol{\pi}}^l(i) = \frac{\boldsymbol{\pi}^l(i) - \pi_{\min}^l}{\pi_{\max}^l - \pi_{\min}^l} \tag{11}$$

where $\pi_{\max}^l$ and $\pi_{\min}^l$ denote the maximum and the minimum values of $\boldsymbol{\pi}^l(i)$'s at the l-th level, respectively. The left images in Fig. 5 show the normalized stationary distributions at the third level of 'Feline' and 'Bunny' models, obtained by the ordinary RW simulation. The RW highlights only locally distinct regions, for example, the uneven and curvy patterns and the sharp boundary of the wing in 'Feline' model and the boundary of the ears in 'Bunny' model. But it fails to detect globally meaningful regions such as the whole areas of the wing in 'Feline' model and the ears in 'Bunny' model.

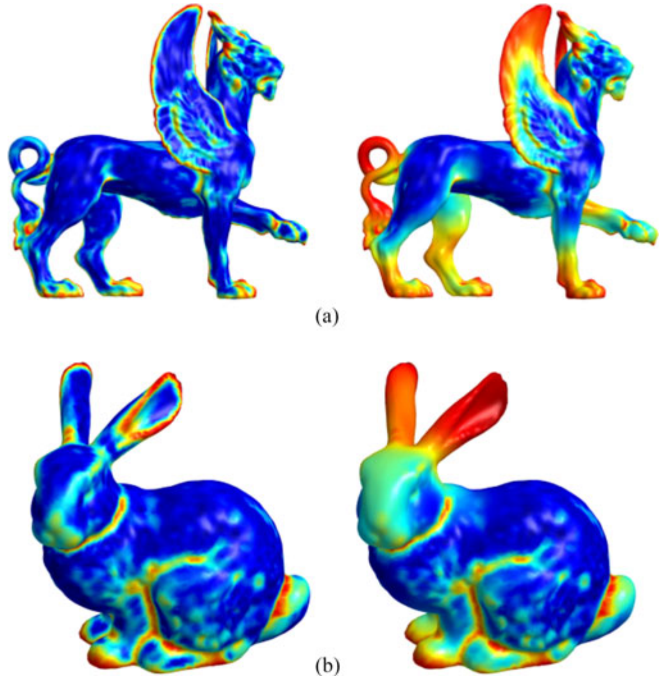


Fig. 5. Comparison of single-scale saliency distributions (left) and multiscale distributions (right) at the third level of semi-regular meshes.

Therefore, we combine the multiscale saliency distributions together to capture the global saliency and the local geometric details simultaneously. We first define the saliency distribution $\mathbf{S}_{\text{face}}^0$ at the coarsest level of mesh as the normalized stationary distribution $\bar{\boldsymbol{\pi}}^0$. Then we recursively find the saliency distribution $\mathbf{S}_{\text{face}}^l$ at a certain level of mesh by taking the maximum saliency between the normalized stationary distribution $\bar{\boldsymbol{\pi}}^l$ at the current level and the distribution $\bar{\mathbf{b}}^l(i)$ which is upsampled from the previous coarser level saliency map $\mathbf{S}_{\text{face}}^{l-1}$, such that

$$\mathbf{S}_{\text{face}}^l(i) = \max(\bar{\boldsymbol{\pi}}^l(i), \bar{\mathbf{b}}^l(i)). \tag{12}$$

The right images in Fig. 5 demonstrate the effect of the multiscale saliency combination, where we see that the multiscale saliency distributions detect the local details highlighted by the single-scale RW as well as the global conspicuous shapes of 3D models, for example, the overall regions of the wing, tail and legs in 'Feline' model and the whole ears in 'Bunny' model.

$\bar{\mathbf{b}}^l$ in (12) is a normalized version of the upsampled distribution $\mathbf{b}^l$, such that

$$\bar{\mathbf{b}}^l(i) = \frac{\mathbf{b}^l(i) - b_{\min}^l}{b_{\max}^l - b_{\min}^l} \tag{13}$$

where $b_{\max}^l$ and $b_{\min}^l$ denote the maximum and the minimum elements in $\mathbf{b}^l$, respectively. We design an upsampling scheme to obtain $\mathbf{b}^l$ from $\mathbf{S}_{\text{face}}^{l-1}$, where $\mathbf{b}^l(i)$ at f_i^l is computed as a weighted

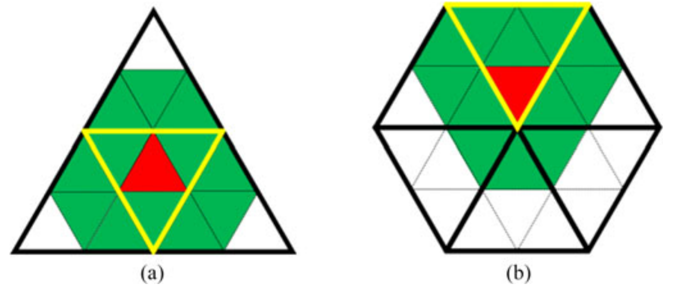


Fig. 6. Topological illustration of the two cases of neighboring faces for upsampling saliency distribution. Target triangle faces are shown in red, and their neighboring faces are in green, respectively. The parent triangles of the target faces are denoted by yellow boundaries.

average of saliency values in $\mathbf{S}_{\text{face}}^{l-1}$ assigned to the faces at the $(l-1)$th level neighboring to f_i^l. Fig. 6 shows the topological illustration of $\Phi(f_i^l)$, where we depict f_i^l in red and the faces of $\Phi(f_i^l)$ in green, respectively. Let $u(f_i^l)$ be the index of the parent face of f_i^l at the $(l-1)$th level. We also represent the parent face $f_{u(f_i^l)}^{l-1}$ using the yellow boundaries. According to the topological relation between f_i^l and $f_{u(f_i^l)}^{l-1}$, different numbers of $(l-1)$th level faces are associated to define $\Phi(f_i^l)$. For example, as shown in Fig. 6(a), when f_i^l is the first child triangle (or center triangle) of its parent $f_{u(f_i^l)}^{l-1}$, the neighboring faces in $\Phi(f_i^l)$ are selected from the children faces of the four $(l-1)$th level neighboring faces. Otherwise, as shown in Fig. 6(b), $\Phi(f_i^l)$ can include the children faces from more $(l-1)$th level faces. In such cases, the numbers of children faces belonging to $\Phi(f_i^l)$ are different across the associated parent faces.

Specifically, we compute $\mathbf{b}^l(i)$ by reflecting the contribution of saliency values at the coarser level faces adaptively, given by

$$\mathbf{b}^l(i) = \frac{\sum_{f_k^l \in \Phi(f_i^l)} \mathbf{S}_{\text{face}}^{l-1}(u(f_k^l)) \cdot \exp\left(-\frac{d(f_i^l, f_k^l)^2}{\kappa_3 \delta_l^2}\right)}{\sum_{f_k^l \in \Phi(f_i^l)} \exp\left(-\frac{d(f_i^l, f_k^l)^2}{\kappa_3 \delta_l^2}\right)} \tag{14}$$

where $d(f_i^l, f_k^l)$ is the Euclidean distance between the two faces f_i^l and f_k^l, i.e., $d(f_i^l, f_k^l) = \|\mathbf{p}(f_i^l) - \mathbf{p}(f_k^l)\|$. The position $\mathbf{p}(f)$ of a face f is defined as follows. At the finest level, $\mathbf{p}(f_i^L)$ is defined as the average position of the three connected vertices of f_i^L. Then the position of a higher level face is defined by the average position of its four children faces in a bottom-up manner, i.e., $\mathbf{p}(f_i^l) = \frac{1}{4}\sum_{k=1}^{4} \mathbf{p}(f_{i,k}^{l+1})$. δ_l is the median of the side lengths of the l-th level triangles. κ_3 is empirically set to 100. Note that, when upsampling higher level saliency values from coarser level ones in (14), we can use spatially balanced and topologically consistent neighboring faces without additional computation by exploiting the systematic face hierarchy in semi-regular meshes. In contrary, the irregular mesh based techniques require additional computational complexity to find neighboring vertices at various sizes of Gaussian kernels [17] or to find K-nearest neighbors [23] when associating multiscale saliency values. Furthermore, the resulting neighboring vertices often cause spatially unbalanced distributions due to locally different sampling densities.

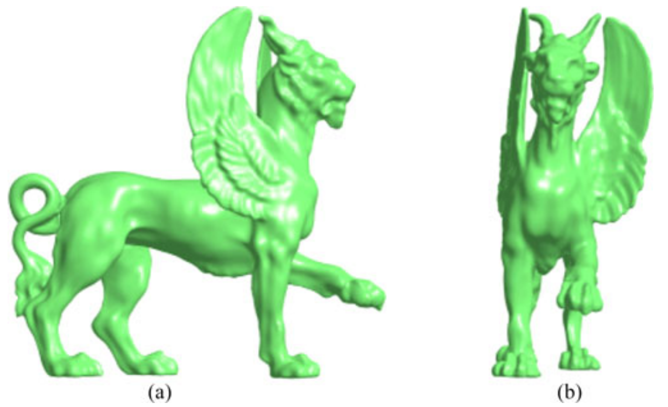


Fig. 7. Visualized images of Feline model from different viewpoints: (a) a side view and (b) a front view.

In addition, we can also estimate the vertex-based saliency distribution $\mathbf{S}_{\text{vertex}}^l$ accordingly, by interpolating $\mathbf{S}_{\text{face}}^l$ at each vertex. Let $\Psi(v_j^l)$ be the 1-ring neighborhood faces to a vertex v_j^l at the l-th level. Then the saliency value $\mathbf{S}_{\text{vertex}}^l(j)$ assigned to v_j^l is interpolated by using the face-wise saliency values of $\Psi(v_j^l)$, given by

$$\mathbf{S}_{\text{vertex}}^l(j) = \frac{\sum_{f_k^l \in \Psi(v_j^l)} \mathbf{S}_{\text{face}}^l(k) \cdot \exp\left(-\frac{d(v_j^l, f_k^l)^2}{\kappa_3 \delta_l^2}\right)}{\sum_{f_k^l \in \Psi(v_j^l)} \exp\left(-\frac{d(v_j^l, f_k^l)^2}{\kappa_3 \delta_l^2}\right)} \tag{15}$$

where $d(v_j^l, f_k^l) = \|\mathbf{p}(v_j^l) - \mathbf{p}(f_k^l)\|$.

IV. View-Dependent Saliency Detection

In many interactive 3D applications, 3D models are visualized dynamically according to the change of user's viewpoint, and thus we only see a partial surface area of whole 3D model at a certain time instance. In such cases, the geometric saliency distribution on 3D surface can be changeable according to the location of viewpoint. For example, when rendered as shown in Fig. 7(a), the right wing in 'Feline' model is considered as one of the most salient parts, but it becomes almost invisible and thus less salient when rendered in Fig. 7(b). In this section, we propose a view-dependent dynamic saliency detection algorithm for 3D semi-regular meshes. Note that the saliency of rendered images of 3D model perceived at some specific viewpoints can be related to various factors such as color, texture, lighting condition, and shading method, which fall into the category of image saliency. In this work, we consider the 3D surface geometry only to evaluate view-dependent saliency.

We first define visibility $\eta_{i,\nu}^l$ for each face f_i^l with respect to a given viewpoint ν. Specifically, at the finest level L, we compute the angle $\phi(f_i^L, \nu)$ between the viewing direction from f_i^L to the viewpoint ν and the face normal $\boldsymbol{\rho}(f_i^L)$, given by

$$\phi(f_i^L, \nu) = \arccos\left(\frac{(\nu - \mathbf{p}(f_i^L)) \bullet \boldsymbol{\rho}(f_i^L)}{\|\nu - \mathbf{p}(f_i^L)\| \cdot \|\boldsymbol{\rho}(f_i^L)\|}\right). \tag{16}$$

When $\phi(f_i^L, \nu)$ is larger than 90°, the face f_i^L is regarded as invisible from the viewpoint. Then the visibility $\eta_{i,\nu}^L$ of finest

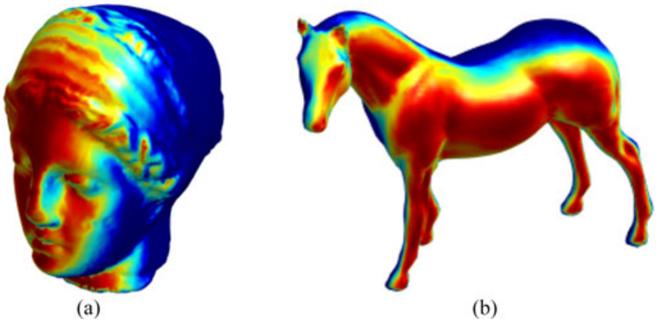


Fig. 8. Visibility distributions for 3D models. The highly visible areas are depicted in red, while the invisible areas are in blue.

level face f_i^L is defined as

$$\eta_{i,\nu}^L = \begin{cases} \cos(\phi(f_i^L, \nu)), & \text{if } 0 \leq \phi(f_i^L, \nu) \leq 90^\circ \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The visibility $\eta_{i,\nu}^l$ at an l-th level face f_i^l is estimated by the average visibility defined at its children faces in a bottom-up manner from the finest level to the coarsest level

$$\eta_{i,\nu}^l = \frac{1}{4} \sum_{k=1}^{4} \eta_{i,k,\nu}^{l+1} \tag{18}$$

where $\eta_{i,k,\nu}^{l+1}$ is the visibility at the child face $f_{i,k}^{l+1}$. Fig. 8 visualizes the distributions of visibility computed at the third level meshes on the surfaces of 3D models. The viewpoints are located in front of the face of 'Venus' and the left side of the body of 'Horse', respectively. We see that the front visible areas are highlighted faithfully, while the rear parts of the models, which are invisible from the viewpoints, are effectively suppressed.

We employ the visibility as an additional feature to estimate the view-dependent saliency of 3D surface geometry. For RW simulation with the visibility feature, we construct a fully-connected directional graph $\mathcal{G}_\nu^l$, which takes only the faces having nonzero visibility as the nodes. Therefore, the number of nodes in $\mathcal{G}_\nu^l$ is always smaller than that in $\mathcal{G}^l$. The associated edge weight is designed by adopting the visibility feature in addition to the curvature and histogram features, given by

$$w_{ij,\nu}^l = \frac{1}{1+\exp\left(-\frac{\eta_{i,\nu}^l - \eta_{j,\nu}^l}{\kappa_4}\right) \cdot \exp\left(-\frac{\bar{\psi}_i^l - \bar{\psi}_j^l}{\kappa_1}\right) \cdot \exp\left(-\frac{\beta(\mathbf{h}_i^l, \mathbf{h}_j^l)}{\kappa_2}\right)} \tag{19}$$

where we set $\kappa_4 = 5$ empirically. Then we perform the RW simulation on $\mathcal{G}_\nu^l$ and obtain the stationary distribution $\boldsymbol{\pi}_\nu^l$

$$\boldsymbol{\pi}_\nu^l = \mathbf{P}_\nu^l \boldsymbol{\pi}_\nu^l \tag{20}$$

where the transition matrix $\mathbf{P}_\nu^l$ is defined similarly to (9) using the modified edge weights in (19). The resulting $\boldsymbol{\pi}_\nu^l$ is also normalized to $\bar{\boldsymbol{\pi}}_\nu^l$ by the maximum and the minimum values in $\boldsymbol{\pi}_\nu^l$. Then $\bar{\boldsymbol{\pi}}_\nu^l$ is regarded as a single-scale saliency distribution at the l-th level of semi-regular mesh with respect to the viewpoint ν.

We also combine the multiscale view-dependent saliency distributions together. The view-dependent saliency distribution $\mathbf{S}_{\text{face},\nu}^0$ in the base mesh is given by the normalized stationary distribution $\bar{\boldsymbol{\pi}}_\nu^0$. We upsample a coarse level saliency distribution $\mathbf{S}_{\text{face},\nu}^{l-1}$ to obtain a higher level distribution $\mathbf{b}_\nu^l$, such that

$$\mathbf{b}_\nu^l(i) = \frac{\sum_{f_k^l \in \Phi_\nu(f_i^l)} \mathbf{S}_{\text{face},\nu}^{l-1}(u(f_k^l)) \cdot \exp\left(-\frac{d(f_i^l, f_k^l)^2}{\kappa_3 \delta_l^2}\right)}{\sum_{f_k^l \in \Phi_\nu(f_i^l)} \exp\left(-\frac{d(f_i^l, f_k^l)^2}{\kappa_3 \delta_l^2}\right)} \tag{21}$$

where $\Phi_\nu(f_i^l)$ is the set of only the visible faces in $\Phi(f_i^l)$ with respect to the viewpoint ν. Then we normalize $\mathbf{b}_\nu^l$ to $\bar{\mathbf{b}}_\nu^l$ similarly to (13). Finally, the view-dependent saliency distribution $\mathbf{S}_{\text{face},\nu}^l$ is obtained by taking the maximum between $\boldsymbol{\pi}_\nu^l$ and $\bar{\mathbf{b}}_\nu^l$, such that

$$\mathbf{S}_{\text{face},\nu}^l(i) = \max(\bar{\boldsymbol{\pi}}_\nu^l(i), \bar{\mathbf{b}}_\nu^l(i)). \tag{22}$$

Note that whereas $\mathbf{S}_{\text{face}}^l(i)$ estimates the static geometric saliency value at each face f_i^l regardless of viewpoints, $\mathbf{S}_{\text{face},\nu}^l(i)$ represents a changeable saliency value at f_i^l when the 3D model is visualized from the viewpoint ν. Also, the vertex-based view-dependent saliency distribution $\mathbf{S}_{\text{vertex},\nu}^l$ is obtained by using the (15) where $\mathbf{S}_{\text{face}}^l$ is replaced with $\mathbf{S}_{\text{face},\nu}^l$.

V. Experimental Results

We evaluate the performance of the proposed saliency detection algorithm for 3D semi-regular meshes using twelve test models as shown in Fig. 12. 'Bunny,' 'Feline,' 'Horse,' 'Rabbit,' and 'Venus' models are provided from the Caltech multiresolution modeling group[1], and 'Armadillo,' 'Bust,' 'Dinosaur,' 'Isis,' 'Max Planck,' 'Santa,' and 'Screwdriver' models are obtained by using the normal remeshing scheme [26], respectively. All the test models are decomposed to the five levels of semi-regular meshes including the base mesh level. Note that the remeshing changes the shape of irregular meshes, however, an irregular mesh itself is a sampled version of the original surface of 3D object. In this work, we regard a semi-regular mesh as another version of sampled surface to evaluate the performance of saliency detection for 3D surface geometry. Also, we measure the remeshing errors between irregular meshes and semi-regular meshes using METRO [34]. The properties of the test models are given in Table I.

A. *Multiscale Static Saliency Detection*

We first demonstrate the performance of the multiscale static saliency detection. Fig. 9 displays the face-wise saliency distributions $\mathbf{S}_{\text{face}}^l$ obtained by using the proposed algorithm from the base level to the third level of semi-regular meshes for 'Venus' and 'Dinosaur' models. We see that a single triangle face at the coarsest level covers a relatively large surface area, for example, the overall eye region in 'Venus' or most of the head region in 'Dinosaur'. Thus it is assigned a representative saliency value reflecting global surface characteristics. On the other hand, as the level of semi-regular meshes becomes finer, each triangle face exhibits, in addition to the global features, more detailed

[1] 2000. [Online]. Available: http://www.multires.caltech.edu/software/pgc/

TABLE I
PROPERTIES OF TEST SEMI-REGULAR MESH MODELS

	$\lvert V\rvert$	$\lvert T\rvert$	$\lvert BT\rvert$	L	Remeshing error (10^{-4})
Bunny	34,835	69,473	230	4	2.50
Feline	49,919	99,732	504	4	1.43
Horse	48,485	96,966	220	4	1.57
Rabbit	67,039	134,074	210	4	1.31
Venus	50,002	100,000	388	4	1.24
Armadillo	21,623	43,242	450	4	4.53
Bust	25,467	50,930	500	4	1.10
Dinosaur	28,098	56,192	280	4	7.91
Isis	46,912	93,820	216	4	1.76
Max Planck	27,726	55,448	500	4	0.66
Santa	56,836	113,668	250	4	1.42
Screwdriver	27,152	54,300	340	4	0.67

$\lvert V\rvert$ and $\lvert T\rvert$ denote the numbers of vertices and faces in original irregular meshes used for remeshing. $\lvert BT\rvert$ denotes the number of faces in base meshes. L means the finest level of semi-regular meshes. Remeshing error is computed by METRO [34].

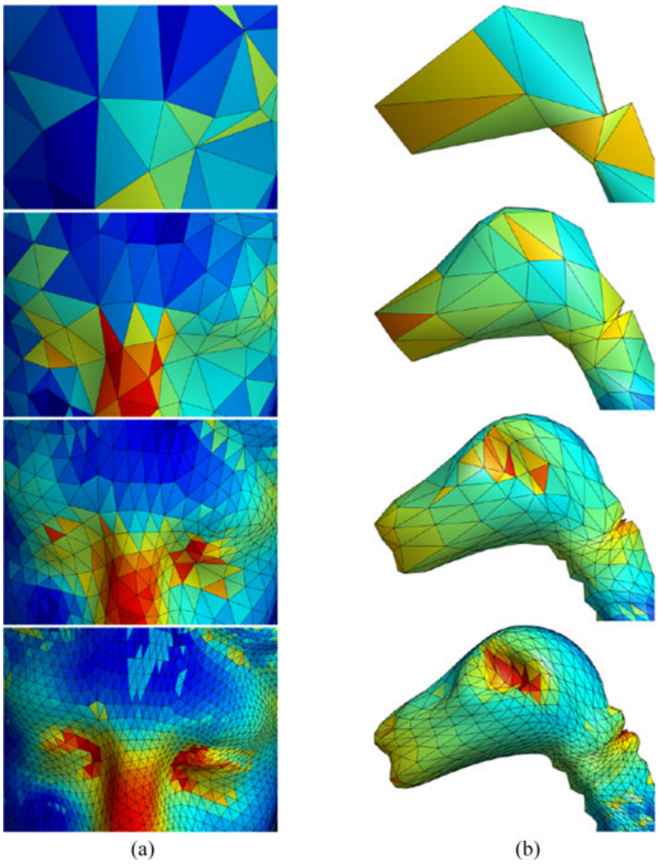


Fig. 9. Face-wise multiscale saliency distributions obtained by using the proposed algorithm. From top to bottom, the levels of semi-regular meshes are $l = 0, 1, 2,$ and 3, respectively. (a) Venus. (b) Dinosaur.

local features such as the curved shapes of the eyes and nose in ‘Venus’ and ‘Dinosaur.’

Fig. 10 depicts the resulting vertex-wise saliency distributions $\mathbf{S}^l_{\text{vertex}}$ of the twelve test models estimated through the four levels of mesh hierarchy from the base level to the third level. We also observe that the coarse scale saliency maps capture the global conspicuous features across relatively large surface areas: for example, the ears in ‘Bunny,’ the wings in ‘Feline,’ the head and legs in ‘Horse,’ and the tail in ‘Dinosaur’. Moreover, at the third level of mesh, the proposed algorithm detects the above global regions as well as the local geometric details such as the boundaries of the wings in ‘Feline,’ the nose and ears in ‘Horse,’ and the eyes in ‘Dinosaur’. Note that the detected saliency distributions are different according to the relative size of employed face patches where the geometric features are evaluated. For example, in ‘Horse’ model, the head and legs are regarded as salient compared to the whole body at coarse scales, but the eyes, nose, and ears become more noticeable than the whole head at fine scales. Consequently, the proposed algorithm can capture both of the global salient regions and the locally detailed geometric features together by combining multiscale saliency distributions.

We also compare the multiscale saliency detection results obtained from semi-regular meshes and irregular meshes, respectively. We employ irregular meshes with similar visual quality to that of the semi-regular meshes at the third level, which are then simplified to generate a hierarchy of irregular meshes. We modified the proposed algorithm designed for semi-regular meshes to be applicable to irregular meshes. Specifically, we compute the angular deviation of vertex normals using the one-ring neighbors, and we replace (14) with the up-sampling scheme used in [23]. Fig. 11 shows the resulting multiscale saliency maps on ‘Venus’ and ‘Horse’ models. We see that the hierarchical saliency distributions obtained on irregular meshes tend to highlight non-salient surface regions through all scales, such as the smooth top area in ‘Venus’ head and the neck and body of ‘Horse’ model, which are not highlighted on semi-regular meshes.

B. *Qualitative Comparison With Conventional Algorithms*

We compare the performance of the proposed algorithm with that of the four existing saliency detection algorithms for 3D meshes: Lee *et al.* [17], Leifman *et al.* [20], Jeong and Sim [23], and Liu *et al.* [24]. Note that while the proposed algorithm detects saliency for semi-regular meshes, all the existing algorithms detect saliency for irregular meshes. Therefore, we regard the finest level of semi-regular meshes as input irregular meshes to test the conventional algorithms, which are illustrated in the first row in Fig. 12. However, due to computational complexity in Leifman *et al.*’s algorithm, we find the saliency on simplified meshes having similar visual quality to the third level of semi-regular meshes, and then interpolate the obtained saliency distributions to the original input meshes. Also, we normalize the spin image to make the sum of image elements to be one, and compute the distinctness of each vertex with respect to the k-nearest neighbors when implementing Leifman *et al.*’s algorithm. Jeong and Sim’s algorithm is also a multiscale scheme, and thus we generate five scales of simplified irregular meshes using Qslim [35]. In addition, for better illustration of final saliency distributions, the top 3% and bottom 3% of saliency values are truncated, respectively, for Lee *et al.*’s and Jeong and Sim’s algorithms. In our implementation of Liu *et al.*’s algorithm, we set $\sigma = 0.5$, and normalize the feature distance between two vertices by the maximum distance.

In the proposed algorithm, the more memory space and computational complexity are required, as the larger number of faces

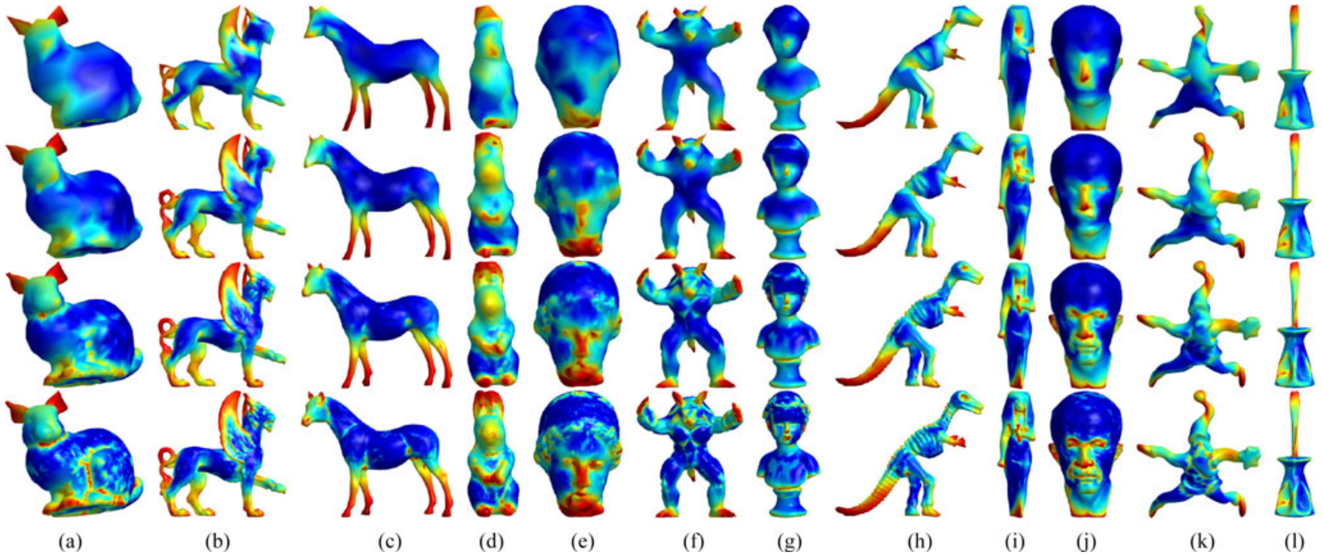


Fig. 10. Multiscale saliency detection results of the proposed algorithm. From top to bottom, we visualize the vertex-wise saliency maps at the levels of semi-regular meshes of $l = 0, 1, 2,$ and 3, respectively. (a) Bunny, (b) Feline, (c) Horse, (d) Rabbit, (e) Venus, (f) Armadillo, (g) Bust, (h) Dinosaur, (i) Isis, (j) Max Planck, (k) Santa, and (l) Screwdriver.

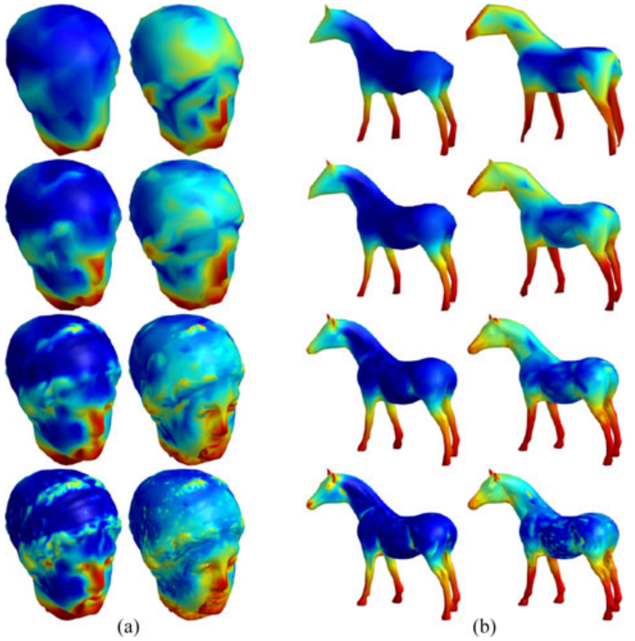


Fig. 11. Comparison of multiscale saliency distributions between semi-regular meshes and irregular meshes. For each model, the resulting saliency maps are obtained on semi-regular meshes (left) and irregular meshes (right). From top to bottom, semi-regular meshes at $l = 0, 1, 2,$ and 3, respectively. (a) Venus. (b) Horse.

we process. To be specific, it requires $O(\mathcal{N}^2)$ complexity to perform random walk simulation on the fully-connected graph with $\mathcal{N}$ nodes. Therefore, in this work, we perform RW simulation from the base mesh up to the third level of semi-regular meshes, and estimate the saliency distribution at the finest level ($l = 4$) by directly upsampling the saliency distribution obtained at the third level.

Fig. 12 compares the resulting saliency distributions of the proposed algorithm and the conventional algorithms at the finest level of semi-regular meshes. We see that Lee *et al.*'s algorithm detects local features well such as the detailed hair in 'Venus' and the bumpy pattern of the tail in 'Dinosaur'. But it often highlights visually unimportant details too much, for example, around the body regions of 'Horse,' 'Bunny,' and 'Armadillo'. Also, it fails to capture the overall shapes of global salient regions in most cases. Leifman *et al.*'s algorithm captures the global conspicuous regions such as the eyes, noses, ears, and feet in 'Bunny' and 'Rabbit,' but relatively less emphasizes some prominent shapes such as the overall wings in 'Feline' and the tail in 'Dinosaur'. Jeong and Sim's algorithm highlights the local features mainly since it employs locally-connected graphs, and fails to capture the global features in most cases: for example, the ears in 'Bunny' and 'Armadillo,' the eyes and ears in 'Horse,' the hair in 'Venus,' and the face in 'Isis'. Liu *et al.*'s algorithm highlights globally salient regions such as the legs in 'Horse' and 'Feline' and the facial regions in 'Armadillo' and 'Feline'. However, it often captures the boundaries of large salient regions, such as the ears in 'Bunny' and the wings in 'Feline'. We also provide the saliency detection results of sing-scale RW simulation which mainly detects locally salient regions. On the contrary, the proposed multiscale saliency detection algorithm captures the visually conspicuous global regions as well as the locally detailed features reliably in most cases by computing the multiscale saliency distributions through systematically structured face hierarchy of semi-regular meshes. In particular, it clearly highlights the wing in 'Feline' and the tail in 'Dinosaur,' which are not captured or considered as less salient by the other algorithms. However, the proposed algorithm including Leifman *et al.*'s algorithm and Liu *et al.*'s algorithm detect the boundaries of the necks in 'Venus' and 'Max Planck,' which are not regarded as salient in human-annotated saliency maps. This is because most of the saliency detection algorithms including the proposed one detect geometrically distinct features only without considering the perception characteristics of HVS.

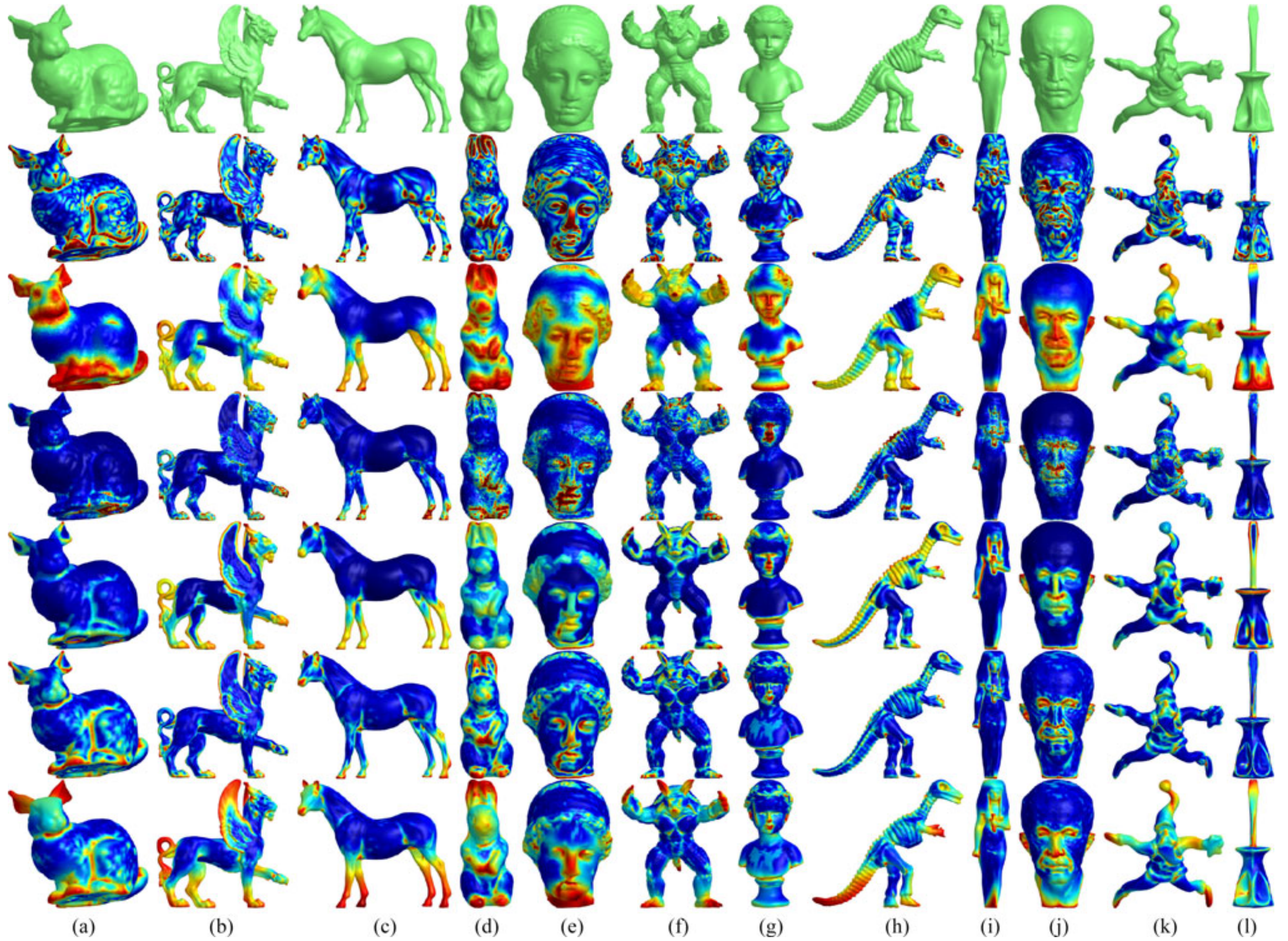


Fig. 12. Comparison of the resulting vertex-wise saliency maps obtained by the conventional algorithms and the proposed algorithm. From top to bottom, we show the finest level of semi-regular meshes and the resulting saliency maps obtained by Lee *et al.* [17], Leifman *et al.* [20], Jeong and Sim [23], Liu *et al.* [24], single-scale RW, and the proposed multiscale saliency detection algorithm, respectively. (a) Bunny, (b) Feline, (c) Horse, (d) Rabbit, (e) Venus, (f) Armadillo, (g) Bust, (h) Dinosaur, (i) Isis, (j) Max Planck, (k) Santa, and (l) Screwdriver.

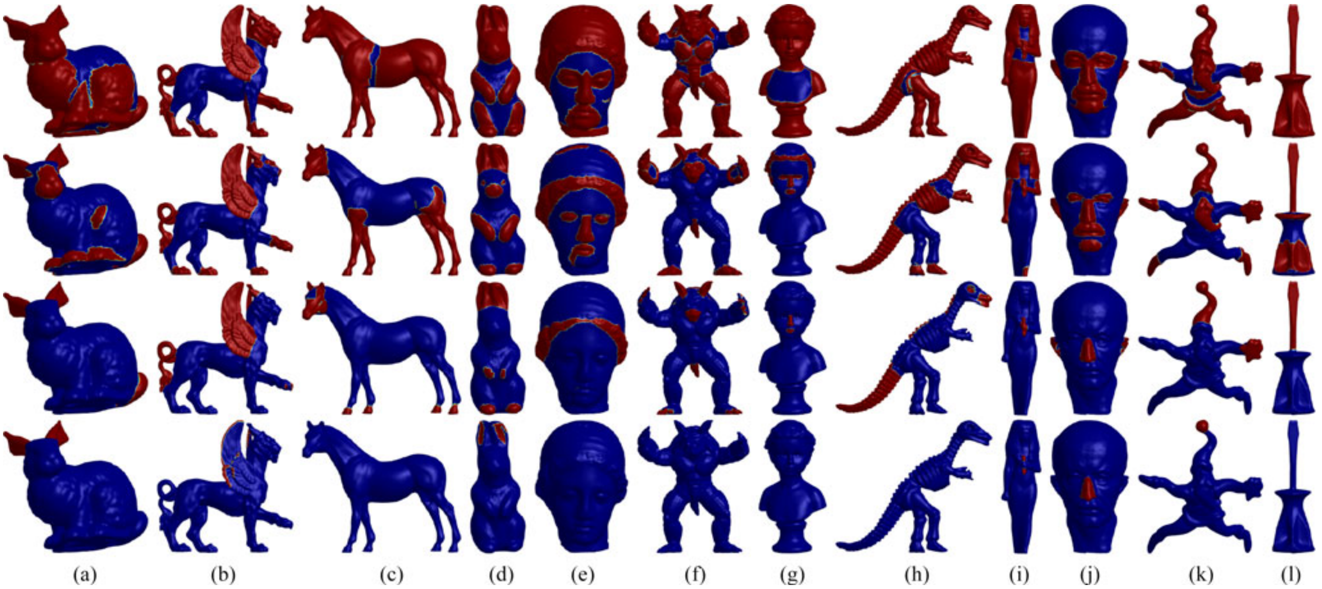


Fig. 13. Four different sets of ground truth binary saliency maps where the salient and non-salient regions are depicted in red and blue, respectively. From top to bottom, $\mathbf{\Omega}_1$, $\mathbf{\Omega}_5$, $\mathbf{\Omega}_{12}$, and $\mathbf{\Omega}_{20}$. (a) Bunny, (b) Feline, (c) Horse, (d) Rabbit, (e) Venus, (f) Armadillo, (g) Bust, (h) Dinosaur, (i) Isis, (j) Max Planck, (k) Santa, and (l) Screwdriver.

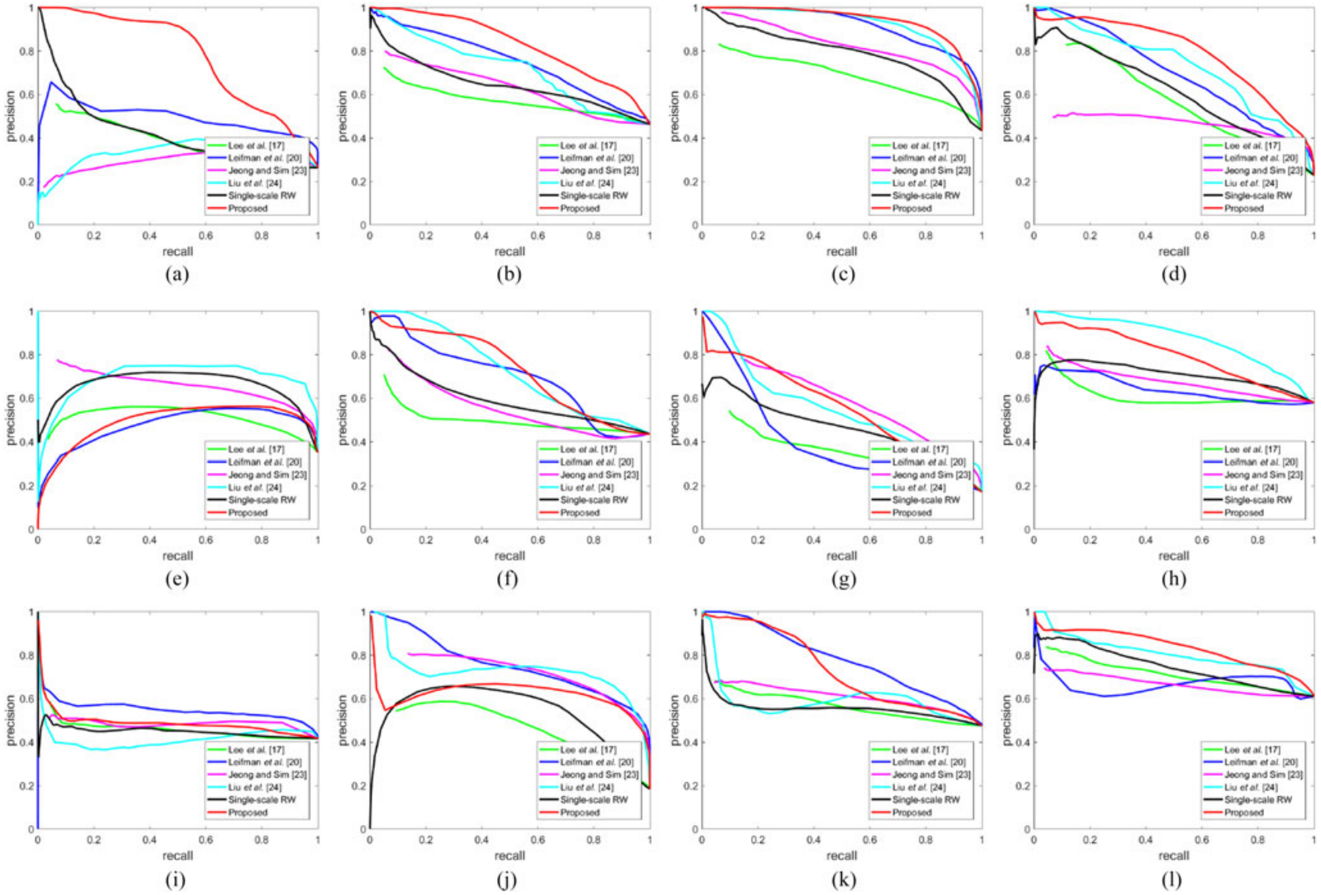


Fig. 14. Quantitative comparison of saliency detection performances in terms of the precision and recall curve. (a) Bunny, (b) Feline, (c) Horse, (d) Rabbit, (e) Venus, (f) Armadillo, (g) Bust, (h) Dinosaur, (i) Isis, (j) Max Planck, (k) Santa, and (l) Screwdriver.

C. Quantitative Comparison With Conventional Algorithms

We attempt to evaluate the accuracy of the resulting saliency distributions of 3D models quantitatively. To this end, we manually generate ground truth binary saliency maps for the twelve test models. In practice, we first obtain 24 human-annotated binary saliency maps for each 3D model where each vertex is labeled as salient or not, respectively. Then we generate a set of ground truth saliency maps $\mathbf{\Omega}_n$ where a vertex is regarded as salient when it is marked as salient by at least n people out of the 24 participants. Fig. 13 shows four different sets of $\mathbf{\Omega}_1$, $\mathbf{\Omega}_5$, $\mathbf{\Omega}_{12}$, and $\mathbf{\Omega}_{20}$, where the salient and non-salient vertices are depicted in red and blue colors, respectively. We see that, in general, as n increases, relatively small portions of surface areas are highlighted as salient in $\mathbf{\Omega}_n$. For example, $\mathbf{\Omega}_1$ often represents most of the vertices as salient, and all the vertices in 'Screwdriver' become salient in $\mathbf{\Omega}_1$ as an extreme case. In contrary, $\mathbf{\Omega}_{12}$ often fails to capture locally salient regions, for example, the eyes, nose, and mouth in 'Venus' and the eyes and mouth in 'Max Planck,' respectively. Furthermore, $\mathbf{\Omega}_{20}$ rarely highlights even the globally salient regions and has no salient vertex at all on 'Horse,' 'Armadillo,' 'Dinosaur,' and 'Screwdriver' models. In this work, we select $\mathbf{\Omega}_5$ as ground-truth saliency, since it captures globally salient areas as well as locally detailed regions together in most 3D models.

Then we plot the precision and recall (PR) curves of the proposed algorithm and the existing algorithms together in Fig. 14. A PR curve is drawn by measuring the precision and recall values of the resulting binarized saliency maps at various thresholds with respect to a ground truth saliency map. We see that the proposed algorithm achieves the best performance on 'Bunny,' 'Feline,' 'Horse,' 'Rabbit,' and 'Screwdriver' models, since it captures globally salient regions, for example, the ears in 'Bunny,' the legs in 'Horse,' and the wings in 'Feline,' more reliably compared with the other methods. Also, the proposed algorithm yields the comparable performance to the best on 'Armadillo' and 'Bust' models. The proposed algorithm holds the second rank on 'Santa' model, since it considers the whole arms and legs as salient while the ground truth depicts only the hands and feet regions as salient. The proposed algorithm performs relatively worse on 'Venus' and 'Max Planck' since the overall neck regions exhibit globally distinct feature contrast and hence are assigned relatively high saliency values, which is inconsistent with human-annotated ground truth saliency. In addition, the proposed multiscale saliency detection algorithm improves the performance of single-scale RW in all test models except 'Venus,' where a large portion of the neck region is missed to be detected as salient by the single-scale RW simulation.

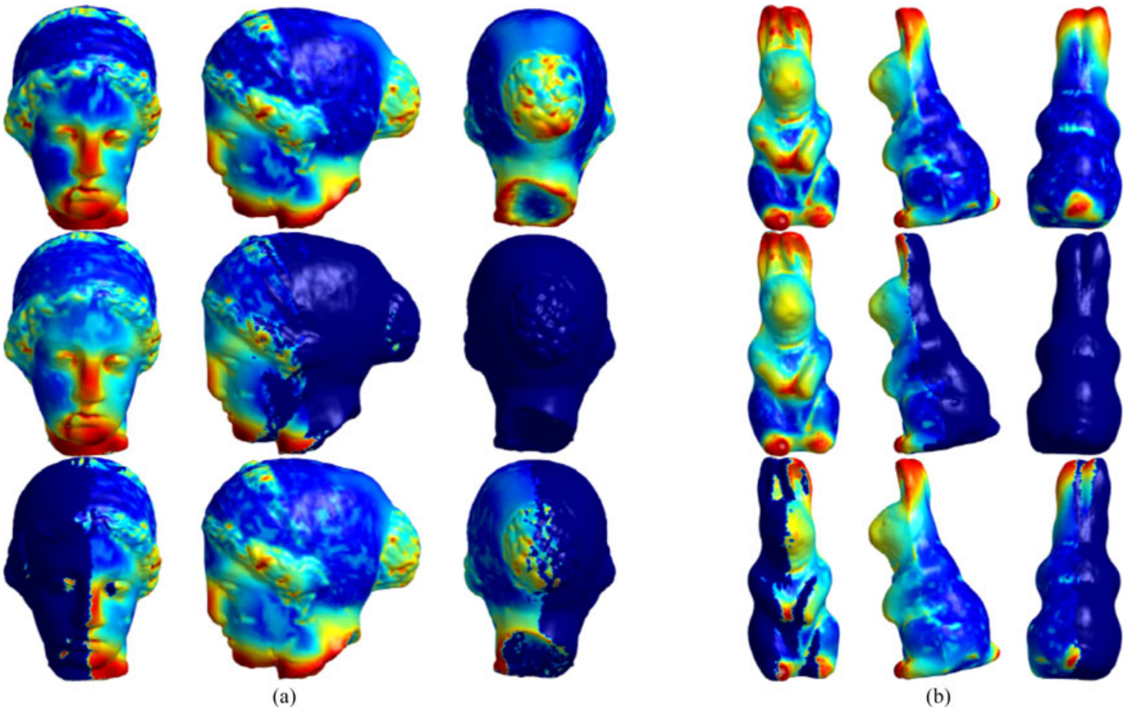


Fig. 15. Comparison between view-independent and view-dependent saliency distributions at the finest level of semi-regular meshes for Venus and Rabbit models. From left to right for each model, we show the saliency images rendered from the three viewpoints, ν_1, ν_2, and ν_3, respectively. The first row shows the static view-independent saliency maps. The second and third rows show the dynamically changing view-dependent saliency maps when the associated viewpoints are located at ν_1 and ν_2, respectively. (a) Venus. (b) Rabbit.

Note that all the methods achieve the maximum precision score of 1 with respect to $\mathbf{\Omega}_1$ on 'Screwdriver' model, since all vertices are marked as salient in $\mathbf{\Omega}_1$. On the other hand, the quantitative evaluation with respect to $\mathbf{\Omega}_n$ with large n is almost pointless on some test models due to very low precision values. In particular, the recall scores with respect to $\mathbf{\Omega}_{20}$ cannot be defined on 'Horse,' 'Armadillo,' 'Dinosaur,' and 'Screwdriver' models where all the vertices are marked as non-salient in $\mathbf{\Omega}_{20}$.

D. View-Dependent Dynamic Saliency Detection

We test the view-dependent saliency detection performance of the proposed algorithm. Fig. 15 compares the static view-independent saliency distributions and dynamically changing view-dependent saliency distributions at the finest level of semi-regular meshes for 'Venus' and 'Rabbit' models. The first, second, and third columns for each model show the saliency maps rendered from three different viewing positions ν_1, ν_2, and ν_3, respectively, rotating around the 3D models. The first row represents the static view-independent saliency distributions. The second and third rows illustrate the resulting view-dependent saliency distributions when the user's viewpoints are assumed to be located at ν_1 and ν_2, respectively. We see that, with the view-dependent mode, most of the salient regions located in the frontal areas of 3D models facing to the selected viewpoints are highlighted reliably, for example, the face of 'Venus' and the front area of 'Rabbit' at the viewpoint ν_1, and the left sides of 'Venus' head and 'Rabbit' at the viewpoint ν_2, respectively. Moreover, as the surface normal is more perpendicular to the viewing direction, the associated view-dependent saliency value becomes smaller compared to the associated static view-independent saliency value. Also, we can achieve computational efficiency by avoiding to consider most of the invisible back side of 3D models for view-dependent saliency computation. Note that even though the resulting view-dependent saliency distributions are changed adaptively from the second row to the third row in Fig. 15 when the viewpoint changes from ν_1 to ν_2, the view-dependent saliency distributions are almost similar to the view-independent one on the visible areas of 3D models with respect to the viewpoint.

We apply the view-dependent saliency detection algorithm to adaptive transmission of 3D semi-regular meshes. We consider an application of interactive communication of 3D models between a server and a client through a transmission channel with constrained bandwidth, where a client wants to download most important information first to achieve the best quality of 3D model at his/her viewpoint for a given available bitrate. The semi-regular meshes are suitable for this application, since the geometry data are composed of the coarse base mesh and the progressively refined geometry data at higher levels [25]. The server first transmits a base mesh to the client. The client sends the viewpoint to the server, and the server allocates a major portion of bitrates to refine geometrically salient and visible surface regions of a semi-regular mesh accordingly. To this end, we partition a semi-regular mesh surface into disjoint local segments, and encode each segment separately to generate progressively compressed bitstream by using [26]. During transmission, we use the saliency value computed at each vertex as a weighting factor to evaluate the contribution of each vertex to surface geometry. Fig. 16 shows the reconstructed 3D meshes on the

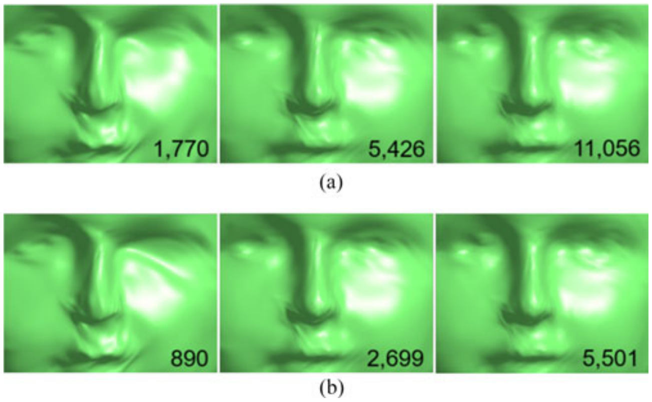


Fig. 16. Saliency guided transmission of progressively compressed bitstreams for a semi-regular mesh on the Bust model. The numbers denote the bytes of transmitted bitstreams. (a) View-independent. (b) View-dependent.

'Bust' model at various bitrates of compression for the refined geometry data, which are obtained by using the static view-independent saliency distribution and the dynamically changing view-dependent saliency distribution, respectively. We see that the view-dependent saliency guided transmission reduces about half of the total bitrates compared with the static saliency guided transmission, while achieving similar visual qualities of reconstructed 3D models from a users viewpoint.

VI. Conclusion

We proposed a novel saliency detection algorithm for 3D semi-regular meshes. We first defined multiscale face-wise features which reflect the angular deviation of the face normal vectors. At each level of mesh, we constructed a fully-connected graph taking each face as a graph node. The weights of directional edges are designed to achieve the center-surround feature contrast such that a random walker moves frequently from low curvature regions to high curvature regions. We obtained multiscale saliency distributions by performing RW on the face graphs at different levels of mesh, which are then combined together. Moreover, we also developed a view-dependent saliency detection algorithm by computing a visibility feature according to the varying viewpoint. Experimental results demonstrated that the proposed algorithm provides reliable multiscale saliency distributions by capturing the global conspicuous regions as well as the locally detailed features faithfully, while the conventional techniques often fail to capture both features together.

Future research topics include a thorough investigation of the perception characteristics of HVS on both of the photometry and the 3D surface geometry, a reasonable methodology to generate ground truth saliency distributions, and a design of a more robust saliency detection algorithm accordingly.

References

[1] A. Tamrakar *et al.*, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3681–3688.

[2] J. Fan, H. Luo, Y. Gao, and R. Jain, "Incorporating concept ontology for hierarchical video classification, annotation, and visualization," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 939–957, Aug. 2007.

[3] B. W. Chen, J. C. Wang, and J. F. Wang, "A novel video summarization based on mining the story-structure and semantic relations among concept entities," *IEEE Trans. Multimedia*, vol. 11, no. 2, pp. 295–312, Feb. 2009.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[5] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 2280–2287.

[6] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 2376–2383.

[7] L. da Fontoura Costa, "Visual saliency and attention as random walks on complex networks," *CoRR*, Jul. 2007. [Online]. Available: http://arxiv.org/abs/physics/0603025v2

[8] A. Borji and L. Itti, "Exploiting local and global patch rarities for saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 478–485.

[9] N. Imamoglu, W. Lin, and Y. Fang, "A saliency detection model using low-level features based on wavelet transform," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 96–105, Jan. 2013.

[10] J.-S. Kim, J.-Y. Sim, and C.-S. Kim, "Multiscale saliency detection using random walk with restart," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 198–210, Feb. 2014.

[11] J. G. Yu, G. S. Xia, C. Gao, and A. Samal, "A computational model for object-based visual saliency: Spreading attention along gestalt cues," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 273–286, Feb. 2016.

[12] Y. Gao, M. Shi, D. Tao, and C. Xu, "Database saliency for fast image retrieval," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 359–369, Mar. 2015.

[13] F. Meng, H. Li, G. Liu, and K. N. Ngan, "Object co-segmentation based on shortest path algorithm and saliency model," *IEEE Trans. Multimedia*, vol. 14, no. 5, pp. 1429–1441, Oct. 2012.

[14] D. Y. Chen and Y. S. Luo, "Preserving motion-tolerant contextual visual saliency for video resizing," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1616–1627, Nov. 2013.

[15] K. Kiyokawa, "Trends and vision of head mounted display in augmented reality," in *Proc. Int. Symp. Ubiquitous Virtual Reality*, Aug. 2012, pp. 14–17.

[16] L. Gomes, O. R. P. Bellon, and L. Silva, "3D reconstruction methods for digital preservation of cultural heritage: A survey," *Pattern Recog. Lett.*, vol. 50, pp. 3–14, Dec. 2014.

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

[18] M. Feixas, M. Sbert, and F. González, "A unified information-theoretic framework for viewpoint selection and mesh saliency," *ACM Trans. Appl. Perception*, vol. 6, no. 1, pp. 1-1–1-23, Feb. 2009.

[19] P. Shilane and T. Funkhouser, "Distinctive regions of 3d surfaces," *ACM Trans. Graph.*, vol. 26, no. 2, pp. 6-1–6-15, Jun. 2007.

[20] G. Leifman, E. Shtrom, and A. Tal, "Surface regions of interest for viewpoint selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 414–421.

[21] J. Wu, X. Shen, W. Zhu, and L. Liu, "Mesh saliency with global rarity," *Graphical Models*, vol. 75, pp. 255–264, Sep. 2013.

[22] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, "Mesh saliency via spectral processing," *ACM Trans. Graph.*, vol. 33, pp. 6-1–6-17, Feb. 2014.

[23] S.-W. Jeong and J.-Y. Sim, "Multiscale saliency detection for 3d meshes using random walk," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2014, pp. 455–458.

[24] X. Liu, P. Tao, J. Cao, H. Chen, and C. Zou, "Mesh saliency detection via double absorbing Markov chain in feature space," *Vis. Comput.*, vol. 32, no. 9, pp. 1121–1132, Sep. 2016.

[25] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Technol.*, Jul. 2000, pp. 271–278.

[26] J.-Y. Sim, C.-S. Kim, C.-C. Kuo, and S.-U. Lee, "Rate-distortion optimized compression and view-dependent transmission of 3-d normal meshes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 854–868, Jul. 2005.

[27] M. Lounsbery, T. D. DeRose, and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Trans. Graph.*, vol. 16, no. 1, pp. 34–73, Jan. 1997.

[28] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs for salient object detection in images," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3232–3242, Dec. 2010.

[29] J. R. Norris, *Markov Chains*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[30] A. Hubeli and M. Gross, "Multiresolution feature extraction for unstructured meshes," in *Proc. IEEE Vis.*, Oct. 2001, pp. 287–294.

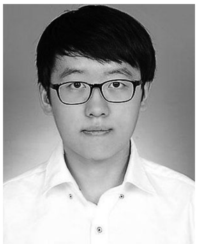
[31] C. H. You, K. A. Lee, and H. Li, "An SVM kernel with GMM-supervector based on the bhattacharyya distance for speaker recognition," *IEEE Signal Process. Lett.*, vol. 16, no. 1, pp. 49–52, Jan. 2009.

[32] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 596–603.

[33] R. Serfozo, *Basics of Applied Stochastic Processes*. New York, NY, USA: Springer-Verlag, 2013.

[34] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Comput. Graph. Forum*, vol. 17, no. 2, pp. 167–174, Jun. 1998.

[35] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. 24th Annu. Conf. Comput. Graph. Interactive Techn.*, Aug. 1997, pp. 209–216.



Se-Won Jeong received the B.S. degree in electrical and computer engineering from Ulsan National Institute of Science and Technology, Ulsan, South Korea, in 2014, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include saliency detection for 3D models.

Jae-Young Sim (S'02–M'05) received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 1999, 2001, and 2005, respectively.

From 2005 to 2009, he was a Research Staff Member with the Samsung Advanced Institute of Technology, Samsung Electronics Company, Ltd., Suwon, South Korea. In 2009, he joined the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea, where he is now an Associate Professor. His research interests include image, video, and 3D visual processing, computer vision, and multimedia data compression.