Mesh Saliency via Spectral Processing

Ran Song and Yonghuai Liu
Aberystwyth University, UK
and
Ralph R. Martin and Paul L. Rosin
Cardiff University, UK

We propose a novel method for detecting mesh saliency, a perceptually-based measure of the importance of a local region on a 3D surface mesh. Our method incorporates *global* considerations by making use of spectral attributes of the mesh, unlike most existing methods which are typically based on *local* geometric cues. We first consider the properties of the log-Laplacian spectrum of the mesh. Those frequencies which show differences from expected behaviour capture saliency in the frequency domain. Information about these frequencies is considered in the spatial domain at multiple spatial scales to localise the salient features and give the final salient areas. The effectiveness and robustness of our approach are demonstrated by comparisons to previous approaches on a range of test models. The benefits of the proposed method are further evaluated in applications such as mesh simplification, mesh segmentation and scan integration, where we show how incorporating mesh saliency can provide improved results.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*

General Terms: Algorithms, Applications

Additional Key Words and Phrases: Mesh saliency, Spectral mesh processing, Mesh simplification, Mesh segmentation

1. INTRODUCTION

Mesh saliency is a measure that attempts to capture the importance of a point or local region on a 3D surface mesh in a similar way to human visual perception. The human perceptual system is able to detect visual saliency extraordinarily quickly and reliably, even for novel scenes. The term 'saliency' is often considered in the context of bottom-up computations [Itti et al. 1998]. However, development of bottom-up computational models which simulate this basic intelligent behavior remains a profound challenge in computer vision and graphics. Mesh saliency detection methods usually merge perceptual criteria inspired by low-level human visual cues with mathematical measures based on discrete differential geometry, such as curvatures. Overall, however, saliency must efficiently and effectively reflect perceptually important regions on a 3D mesh, which curvature alone may not capture. While mesh saliency may not outperform mesh curvature as a surface analysis tool in all applications, it provides an alternative approach when processing 3D meshes, based on perceptual mechanisms rather than purely local geometric measures of shape.

1.1 Related work

Research into 3D mesh saliency is largely inspired by corresponding work on 2D images [Itti et al. 1998; Walther and Koch 2006; Hou and Zhang 2007; Goferman et al. 2010]. In particular, the concept of *scale space* has been successfully extended to the mesh domain. In 2D vision, scale spaces are usually constructed by manipulating downsampled images with various operators. According to cognitive psychology, within a certain band of spatial resolutions, saliency should be invariant under scale changes [Intriligator and Cavanagh 2001]. Typically, the regions which capture our visual attention in an image should be consistent with those which do so in a downsampled version of the same image; downsampling, in essence, discards detail. Analogously, for 3D meshes, scale space is usually constructed by producing a bank of meshes of different smoothness. To simulate scale-invariance, existing methods for detecting 3D mesh saliency often perform operations using a difference-of-Gaussian (DoG) scale space. Experiments have demonstrated that such a multi-scale computational model of mesh saliency has significantly better correlation with human eye fixations than either a random model or a curvature-based model [Kim et al. 2010].

Early work on saliency detection for 3D models focused on finding saliency in 2D projections of meshes. Guy and Medioni [Guy and Medioni 1997] took a scheme for computing a saliency map based on edges in a 2D image, and applied it to 3D data; the goal was to smoothly interpolate sparse and noisy 3D data to obtain dense surface information. In [Yee et al. 2001], the saliency of a 3D dynamic scene was computed based on a coarsely rendered 2D projection then use of the method in [Itti et al. 1998]. This saliency-based strategy led to accelerated and improved global illumination computation in pre-rendered animations. Mantiuk *et al* [Mantiuk et al. 2003] used a 2D saliency algorithm to guide real-time MPEG compression of an animated 3D scene. In general, estimating saliency in 2D projections of meshes does not sufficiently utilise depth information within the original data, which, as men-

Authors' addresses: Ran Song, Yonghuai Liu, Department of Computer Science, Aberystwyth Univesity, UK; email: sora1998@hotmail.com; yyl@aber.ac.uk; Ralph R. Martin, Paul L. Rosin, School of Computer Science and Informatics, Cardiff University, UK; email: {ralph.martin, paul.rosin}@cs.cardiff.ac.uk

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

tioned in [Howard 2002], is a key stimulus for human perception of a static scene.

More recently, extensive work has considered computing saliency directly from 3D structure. This work shows that mesh saliency is a flexible, and indeed, somewhat ill-defined concept that can vary according to application, while being of broad interest to the community. [Pauly et al. 2003] proposed a multi-scale method to extract line-like features using *surface variation* as a saliency measure based on eigenvalues of the local covariance matrix. [Lee et al. 2005] computed mesh saliency using a center-surround operator on Gaussian-weighted curvatures in a DoG scale space. They also provided methods to incorporate saliency into mesh simplification and view selection. [Howlett et al. 2005] evaluated saliency for simplified meshes. In [Gal and Cohen-Or 2006], salient geometric features based on curvatures were introduced to improve part-in-whole matching. [Shilane and Funkhouser 2007] developed an approach to compute the distinctive regions of a 3D surface and applied it to shape matching, icon generation and mesh simplification. [Castellani et al. 2008] proposed a method for detecting and matching salient points from multi-view meshes, where saliency is determined by generating a multi-scale representation of a mesh in a DoG scale space. [Feixas et al. 2009] focused on viewpoint selection and developed a method to compute view-based mesh saliency using mutual information between polygons. [Leifman et al. 2012] proposed an algorithm for detecting surface regions of interest and explored how to select viewpoints based on these salient regions. [Chen et al. 2012] proposed a regression model to predict mesh saliency based on learning from data collected in a large-scale online user study.

While local geometric cues do indeed influence where people focus their attention in an image or a mesh, saliency actually depends on a few basic principles of human visual attention, as shown by psychological evidence [Treisman and Gelade 1980; Wolfe 1994; Koch and Poggio 1999]. These include not only local considerations, but also global considerations. Responses to frequently occurring features are suppressed while at the same time sensitivity is retained to features that deviate from the norm [Koch and Poggio 1999]. [Shilane and Funkhouser 2007] and [Chen et al. 2012] deliver semantic saliency with a certain amount of global meaning provided via learning rather than a specific mechanism. [Shilane and Funkhouser 2007] uses a training database, but the detected distinctive regions undesirably changes with training database. A large-scale user study reported in [Chen et al. 2012] showed that saliency should remain highly consistent even with different prior knowledge. Unfortunately, the method proposed in [Chen et al. 2012] requires a per-category training with an extremely large training set—$19/20$ of the entire category size. When training is performed only on meshes from different categories, the results deteriorate. Our work is also closely related to [Song et al. 2013] which detects points of interest by considering saliency from a spectral point of view. However, that paper merely targets a specific application, and the approach differs from the one presented here in several significant ways: in terms of how the spectrum is constructed, how it is analysed, and how the resulting information is transferred back to the spatial domain. In [Song et al. 2013], the spectrum is constructed by calculating the eigenvalues of the Laplace-Beltrami operator, which only considers mesh topology. Due to the high computational cost, [Song et al. 2013] merely analyses a fixed small number of smallest eigenvalues, which means that most of the spectral information is wasted. To transfer the information to the spatial domain, a curvature-weighted method is used, based on diffusion, which however tends to detect shape extremities and is sensitive to noise. These lead to poor performance on saliency localisation as shown in our comparisons in Section 4.1.

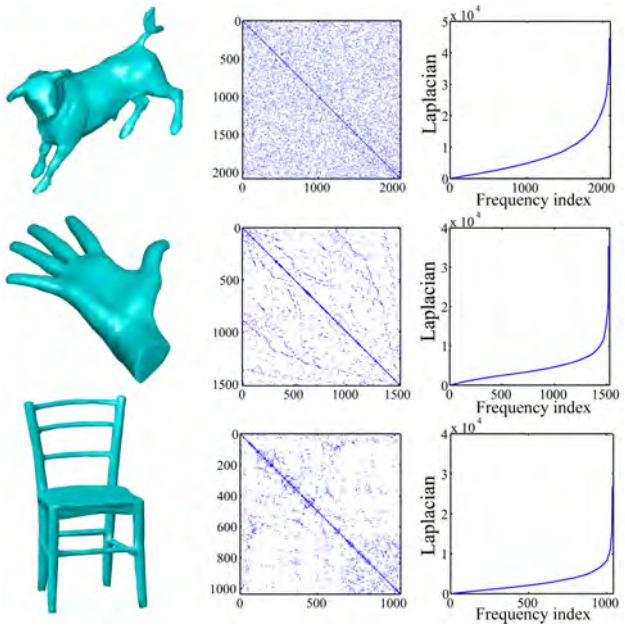

Fig. 1. Left: some meshes; Middle: Laplacian matrices of the meshes; Right: Laplacian spectra of the meshes

1.2 Our work

This paper proposes a general and fully automatic method for detecting mesh saliency, requiring no prior information and no learning. On one hand, the global nature of mesh saliency is captured by considering the *spectral* attributes of the *log-Laplacian* spectrum of a mesh. By carefully analyzing these attributes, we reveal and demonstrate that the irregularity of the log-Laplacian spectrum of a 3D mesh is highly related to the saliency of the mesh. We also offer a computational model to efficiently and effectively capture such irregularity and transfer it from the frequency domain to the spatial domain. Nevertheless, the proposed algorithm considers the mesh at multiple *spatial* scales to detect salient features of various sizes. This multi-scale analysis allows the saliency estimates from different scales vote against each other, leading to accurate localisation of more reliable saliency values. To tackle the major hurdle in the use of spectral mesh analysis, the high computational cost caused by the eigendecomposion of large Laplacian matrices, we use a saliency mapping scheme. Combination of both local and global considerations results in improved detection of saliency compared to methods merely considering local spatial cues.

Our approach is presented in detail in Sections 2 and 3. We then experimentally demonstrate its robustness, effectiveness and efficiency in Sections 4 and 5; we also explore applications of mesh saliency to mesh simplification, mesh segmentation and scan integration. Finally, we draw conclusions in Section 6.

The main contributions of this paper are threefold:

(1) We use spectral mesh processing to develop a generic, bottom-up method for 3D mesh saliency detection which does not require any learning process. It combines local multi-scale saliency with global spectral response, leading to improved

Fig. 2. Plots of the eigenvectors of the Laplacian matrix corresponding to the eight smallest nonzero eigenvalues

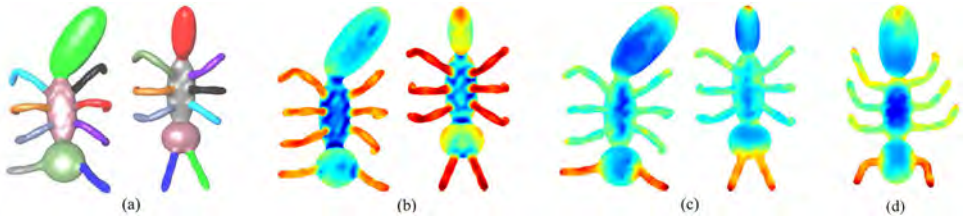

Fig. 3. (a) Mesh segmentation (courtesy of [Zhang et al. 2012]); (b) Mean curvature where warmer colours show higher curvatures; (c) and (d) Saliency from our method: warmer colours show higher saliency.

saliency detection for 3D meshes, as demonstrated by comparative experiments.

(2) We integrate our approach to saliency detection with mesh simplification, a classic application of mesh saliency. We demonstrate that simplification guided by our mesh saliency outperforms not only traditional simplification methods, but others based on saliency.

(3) We explore novel applications of mesh saliency to the problems of mesh segmentation and scan integration. These are active and significant topics in computer graphics, yet little attention has been paid to perception-inspired strategies. Our saliency-guided methods are shown to provide promising and robust results.

2. MESH SALIENCY VIA SPECTRAL PROCESSING

In information theory, a signal can be decomposed into two components separately representing innovation and redundancy. From the perspective of statistics, such redundancy corresponds to statistically-invariant properties of the environment [Ruderman 1994]. [Hou and Zhang 2007] notes that the innovation component of a 2D image can be approximated by a spectral residual based on the Fourier transform.

Matrix theory [Lancaster and Tismenetsky 1985] shows that the eigenvectors of the Laplacian of a 3D mesh play an analogous role to the Fourier transform in terms of representing information at different spatial frequencies. In this section, we show that the log spectrum of the geometric Laplacian of a 3D mesh has some appealing attributes from the perspective of saliency detection, and suggest how to exploit its properties in the computation of mesh saliency.

2.1 Mesh Laplacian

In [Taubin 1995], the frequencies of a triangular mesh were defined as the eigenvalues of the Laplacian matrix based on discretisation of the Laplacian using a weighted sum of adjacent vertices. If a mesh M contains m vertices $p_1, \ldots, p_m$, in its simplest form, the Laplacian matrix can be computed as:

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

where A is the adjacency matrix between vertices, given by

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

and D is a diagonal matrix in which D_{ii} is the degree of vertex p_i. This simplest computational model merely takes into account the topology. To incorporate local geometric information, the adjacency matrix should include weights taking into account the distance between neighbouring vertices:

$$W(i,j) = \frac{1}{\|p_i - p_j\|^2} A(i,j); \tag{3}$$

W is subsequently normalised so that the sum of each row is 1. This leads to the the geometric Laplacian, which we use in the remainder of the paper:

$$L = W - D. \tag{4}$$

It is easy to show that L is symmetric and its smallest eigenvalue is 0. Let $\Lambda = \text{Diag}\{\lambda_f, 1 \le f \le m\}$ denote a diagonal matrix formed from the eigenvalues (also called the frequencies) λ_f of L arranged in increasing order of magnitude (f is the frequency index), and let B denote the orthogonal matrix whose columns b_i are the eigenvectors of L. The Laplacian matrix L can be decomposed as follows:

$$L = B\Lambda B^T. \tag{5}$$

The Laplacian spectrum is defined as $\mathcal{H}(f) = \{\lambda_f, 1 \le f \le m\}$. Fig. 1 shows the Laplacian matrices and Laplacian spectra for several meshes. Although the sparse Laplacian matrices are quite different from each other, the Laplacian spectra share analogous trends. It was argued in [Hou and Zhang 2007] that analogous spectra share much typical, uninformative content, in turn implying that the process of detecting saliency should be consistent with removing such redundant information, or alternatively, detecting the atypical information.

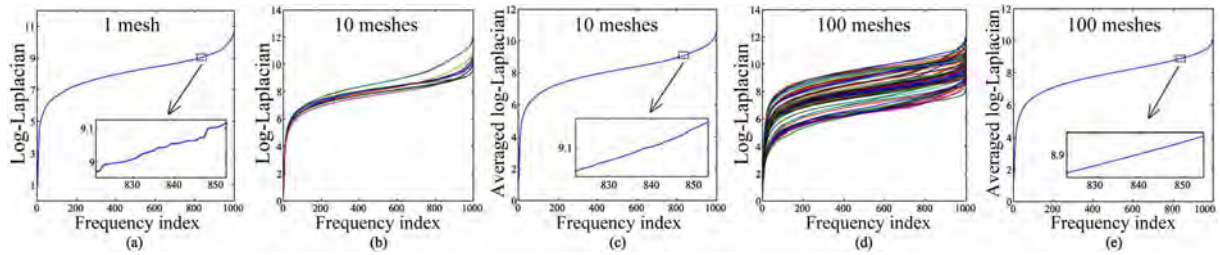

Fig. 4. Log-Laplacian spectra and averaged log-Laplacian spectra for 1, 10 and 100 meshes from the Princeton Mesh Benchmark [Chen et al. 2009]: (a) log-Laplacian spectrum of a single mesh; (b) log-Laplacian spectra of 10 meshes; (c) averaged log-Laplacian spectrum of 10 meshes; (d) log-Laplacian spectra of 100 meshes; (e) averaged log-Laplacian spectrum of 100 meshes.

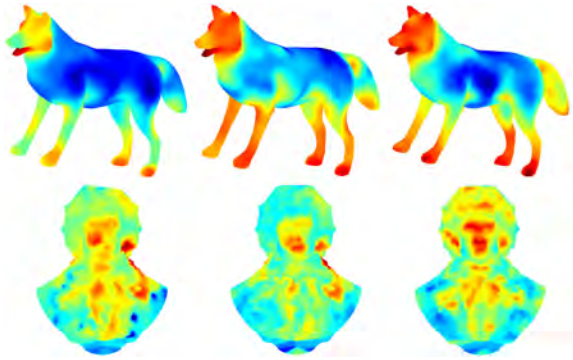

Fig. 5. Saliency results of using different methods to compute the norm. Both meshes come from the same database [Chen et al. 2012]. Left: OutClass learning. The norm is the average spectrum of all 400 meshes (comprising 20 classes of objects) in the database. Middle: InClass learning. The norm is the average spectrum of the 20 meshes in the same class of objects. Right: No learning. The norm is computed using our method. Ground truths can be found in Fig. 12.

Several works have explored properties and applications of Laplacian spectra and eigenstructures of 3D meshes [Pauly et al. 2006; Zhang et al. 2010; Lévy and Zhang 2010; Zhang et al. 2012]. It is known that the eigenvectors of the Laplacian approximate the characteristic functions for each component of the mesh, and that a mesh can be represented as a linear sum of basis functions formed by the eigenvectors. Fig. 2 shows plots of the eigenvectors of the Laplacian corresponding to the first eight nonzero eigenvalues for an ant model. The spectral mesh decomposition does not give direct cues which indicate saliency, although it clearly encodes some global information about the underlying mesh model. [Zhang et al. 2012] utilised the global information derived from spectral attributes for automated mesh segmentation: in Fig. 3(a), the antennae and legs of the ants are clearly recognised to be separate from the body. On the other hand, as shown in Fig. 3(b), mean curvature merely focuses on local cues, yet again manages to segment out the legs and antennae. However, both processes fail to distinguish the antennae and the legs as belonging to two different categories. By contrast, Fig. 3(c) shows that our saliency detection approach distinguishes the antennae from the legs and regards the antennae as more salient. Geometrically, the legs are very similar to the antennae since they are both locally cylindrical and have similarly shaped tips. However, globally, the legs are articulated but the antennae are not. Furthermore, Fig. 3(d), using a different ant model, demonstrates that our approach can distinguish articulated antennae

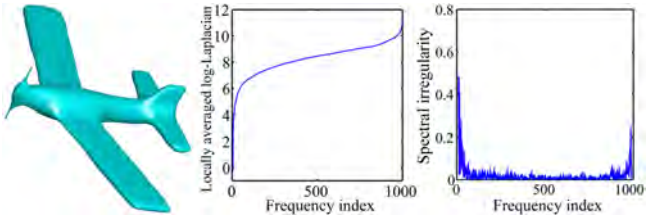

Fig. 6. Locally averaged log-Laplacian spectrum of a mesh and its irregularity curve

from the legs which are articulated in a different manner. Another global cue is the number of legs or antennae; a basic property of the human perception system is to suppress responses to frequently occurring features [Koch and Poggio 1999]: it is desirable that the antennae are regarded as more salient than the legs.

2.2 From spectrum to saliency

Spectra have been widely used to capture global or local statistics of 2D images. Among them, the log spectrum has proved to be a powerful tool for the analysis of natural images. [van der Schaaf and van Hateren 1996] analysed the log-power spectra of an extensive set of natural images and proposed a scheme to reduce second-order redundancy in images. [Oliva and Torralba 2001] suggested a computational model for scene recognition using a logarithmic energy spectrum. [Hou and Zhang 2007] exploited the power of the log-Fourier spectrum in saliency detection for 2D natural images.

In this work, we use the log-Laplacian spectrum:

$$\mathcal{L}(f) = \log(|\mathcal{H}(f)|) \tag{6}$$

where log denotes the natural logarithm. The logarithmic transform acts as a spectral redistributor. Unlike spectral equalisation, which forces spectrum amplitudes to the same level at all frequencies while honoring local variations and without modifying the phase, the logarithmic transform changes the histogram of the Laplacian spectrum by giving the few lowest frequencies a large range of amplitude while placing the rest of frequencies in a narrow range. As shown by the Laplacian spectrum in Fig. 1, the low-frequency end has low amplitude while after the logarithmic transform, in the log-Laplacian spectrum shown in Figure. 4, this end dominates the range of amplitudes. The logarithmic transform amplifies variations (in local terms) and deviations (in global terms) at low-frequencies while suppressing them in the rest of the spectrum.

Now, as pointed out in [Koch and Poggio 1999], the human visual system regards features that deviate from the norm as informative, and is sensitive to them. It is thus to be expected that deviations

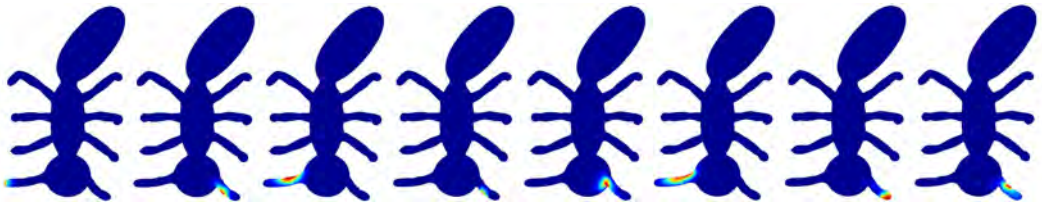

Fig. 7. Eigenvectors of the matrix S corresponding to the first eight eigenvalues

in the spectrum are indicators of visual saliency. It is also known that the eigenvectors of the mesh Laplacian approximatie the characteristic functions for each component of a 3D mesh, which depend on its global and local structure [Von Luxburg 2007]. The eigenvectors corresponding to the smallest eigenvalues (frequencies) of the Laplacian code information representing the most fundamental components [Lévy and Zhang 2010].

Thus, using the log-Laplacian spectrum to amplify the deviations in the low-frequency section of the spectrum helps us to find the most fundamental saliencies. Furthermore, the rapid growth of the Laplacian spectrum at the highest frequencies makes it too sensitive—meshes from laser-scanning devices often include high frequency noise [Huang and Ascher 2008], and a further benefit of the log-Laplacian spectrum is that it effectively suppresses such sensitivity at high-frequencies.

Using the idea that spectral deviations from the norm represent saliency, naturally, our target is to capture such deviations via a computational model. Here, it is worth querying what should be taken as the norm. Is saliency a property that distinguishes an object from all other objects? Or is it a property that tells us which parts of a single object are unusual, and worthy of further attention?

The first idea leads to the idea of using an averaged spectrum computed from many objects as the norm. Fig. 4 shows the log-Laplacian spectra of varying numbers of objects and their averaged spectra; the latter are locally smooth. However, 3D mesh databases are generally small (compared with 2D image databases), so an average spectrum may not be generic and representative. Consequently, saliency based on computing spectral deviations as the difference between the spectrum of the target mesh and the average spectrum of a number of meshes can be unreliable, as shown in Fig. 5. For the animal, InClass learning provides better results compared to OutClass learning. However, neither approach provides reliable saliency detection for the bust model.

The second idea means that we should find the unusual frequencies within an individual spectrum. An effective way to do this is to locally average that spectrum, smoothing it, and then look for frequencies significantly different from the local average. Results of doing so are also shown in Fig. 5, and it can be seen that this approach produces reliable results for both the animal and the bust, where the face is now indicated as salient. The latter approach has the further benefit of not needing a set of reference objects, and the inherent difficulty of choosing the models for such a reference set. No learning is required.

We thus adopt a local averaging filter $J_n(f)$ to compute the norm:

$$\mathcal{A}(f) = J_n(f) * \mathcal{L}(f) \tag{7}$$

where $J_n(f) = \frac{1}{n}[1\ 1\ \ldots\ 1]$ is an $n \times 1$ vector. Spectral deviation can now be computed as the spectral irregularity $\mathcal{R}$:

$$\mathcal{R}(f) = |\mathcal{L}(f) - \mathcal{A}(f)| \tag{8}$$

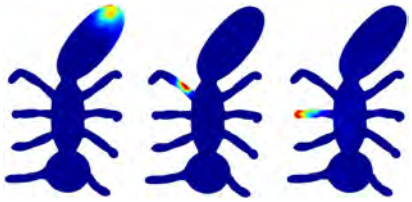

Fig. 8. Plots of the eigenvectors of the matrix S corresponding to the 194^{th}, 300^{th} and 1499^{th} eigenvalues

In our implementation, $n = 9$; changing it alters the final result only slightly. Fig. 6 shows the locally averaged log-Laplacian spectrum and the spectral irregularity for a particular mesh. It can be viewed as a compressed representation of the innovation of a mesh. To bring this representation back to the spatial domain, we perform a composition

$$S = BRB^T W, \tag{9}$$

where $R = \text{Diag}\{\exp(\mathcal{R}(f)) : 1 \leq f \leq m\}$ is a diagonal matrix whose entries are exponentials of the elements of $\mathcal{R}(f)$. We introduce a weighting using the distance-weighted adjacency matrix W to diminish the impact of more distant and unconnected vertices. Note that each row of the Laplacian matrix L provides the *spectral transform coefficients* of a vertex [Lévy and Zhang 2010]. Similarly, each row of S corresponds to a vertex of the mesh. The saliency map $S(i)$ for vertex i is thus derived by summing S along each row.

To demonstrate the proposed scheme, we calculate the eigenvalues and eigenvectors of the matrix S in Eq. (9) using the ant model. Fig. 7 provides plots of the eigenvectors of the matrix S corresponding to the first eight eigenvalues (eigenvalues arranged in ascending order). In sharp contrast to Fig. 2, Fig. 7 clearly shows the saliency captured by our method. It can be seen that the primary saliencies correspond to the most important, globally salient regions, and in particular the antennae are captured by the smallest frequencies (eigenvalues). In comparison, Fig. 8 shows how other less salient regions are captured by higher frequencies.

3. MULTI-SCALE MESH SALIENCY

In [Hou and Zhang 2007], the log-Fourier spectrum is considered for detecting saliency in 2D images (at a single fixed scale) while we investigate the log-Laplacian spectrum for detecting mesh saliency. The Fourier transform is a powerful, well-established tool for spectral processing of 2D images. In particular, it is invertible as the information captured in the spectral domain can be easily transferred back to the spatial domain via the inverse Fourier transform. However, spectral mesh processing based on the mesh Laplacian does not provide a reliable inverse transform, which makes our problem significantly more challenging. In this work, we propose

Fig. 9. Dispersed results when computing single-scale mesh saliency via spectral processing

Eq. (9) as the basis for the transfer. However, the Laplacian spectrum provides a global characterisation of shape. While it has its virtues, it does not allow for local control. Spatial localisation of the saliency originally captured in the spectral domain is thus difficult. Also, since the spectrum is merely a discrete representation of mesh information at a limited number of frequencies (especially as we actually analyse the spectrum of a simplified mesh—see Section 3.3), high frequency details of local surfaces can potentially be improperly discarded. In fact, the results of computing a single-scale mesh saliency via spectral processing are usually rather dispersed as shown in Fig. 9, which was generated by constructing a local filter tuned to the specific salient frequencies detected by the spectral analysis (a 'salient-pass' filter) and applying that filter at each point of the mesh. Salient features (e.g., the eyes of the head, an ear of the horse) are not localised, as the filter operates point-wise. To ensure more reliable results, we conduct saliency detection at multiple scales to better determine salient feature regions.

Multi-scale analysis has been successfully used in both vision and graphics for feature detection; it is capable of localising features with various spatial sizes. One goal of such methods is to provide robustness to noise, since noise is usually influential at only a few spatial scales. Multi-scale techniques have been applied in both 2D and 3D saliency detection [Goferman et al. 2010; Lee et al. 2005; Castellani et al. 2008].

3.1 DoG scale space in 3D

The two saliency detection methods [Lee et al. 2005; Castellani et al. 2008] have both employed difference-of-Gaussian (DoG) scale space which can approximately achieve scale invariance by fixing a constant size difference between adjacent scales. Given a d-dimensional signal $U : \mathbb{R}^d \to \mathbb{R}$, its linear scale-space representation $F : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ is defined as the convolution:

$$F(\cdot, t) = U(\cdot) \otimes g(\cdot, t), \tag{10}$$

where t is the scale parameter and

$$g(x, t) = \frac{1}{(2\pi t)^{1/2}} \exp(-x^T x/(2t)) \tag{11}$$

is a Gaussian kernel with standard deviation σ determined by the scale parameter: $\sigma = \sqrt{t}$. The generating equation of F in Eq. (10) is the diffusion equation [Koenderink 1984], i.e. F can also be obtained as the solution to a diffusion process:

$$\frac{\partial F}{\partial t} - \lambda \Delta F = 0, \tag{12}$$

where λ is the diffusion constant and Δ denotes the Laplacian operator. We may extend the DoG scale space [Lowe 2004] to 3D:

$$\begin{aligned} D(p, t) &= U(p, t) \otimes (g(p, kt) - g(p, t)) \\ &= F(p, kt) - F(p, t) \end{aligned} \tag{13}$$

where k is a constant multiplicative factor and $p \in M(t)$ denotes a vertex on the mesh $M(t)$. The neighbourhood region of p, over which Gaussian smoothing is applied, is built by collecting all vertices within a distance equal to $2.5\sqrt{t}$, where $t = t_1, t_2, \ldots, t_s$ (we use $s = 5$) is the discretisation of the scale parameter; usually the first scale corresponds to the original mesh with $t_1 = 0$. Such a discrete multi-scale representation means that ΔF can be computed by a finite difference approximation to $\partial F/\partial t$, using the difference of nearby scales at kt and t:

$$\lambda \Delta F = \frac{\partial F}{\partial t} \approx \frac{F(p, kt) - F(p, t)}{kt - t} \tag{14}$$

and therefore

$$F(p, kt) - F(p, t) \approx (k - 1)t\lambda \Delta F. \tag{15}$$

Lowe [Lowe 2004] claimed that by carefully fixing t to a suitable constant, the DoG function can be viewed as a close approximation to the scale-normalised Laplacian of Gaussian, $t\Delta F$, as studied by Lindeberg [Lindeberg 1994]. Lindeberg showed that the normalisation of the Laplacian by the factor t, or σ^2 is required for true scale invariance.

3.2 Dynamic DoG scale space

To achieve scale invariance, according to Eq. (15), we have

$$(k - 1)t\lambda \Delta F = t\Delta F, \quad \text{so} \quad k = \frac{1}{\lambda} + 1. \tag{16}$$

Lowe's claim holds for a 2D image because it is a discrete representation produced by uniform and isotropic sampling. Although a 3D mesh is also a discrete representation of a 3D surface, the sampling is usually inconsistent and anisotropic. Such differences lead to instability, particularly in areas with very low sampling densities [Pauly et al. 2006], if Lowe's DoG scale space is directly adopted in 3D.

In the diffusion equation Eq. (12), λ is the diffusivity, which depends on density. While constructing DoG scale space for a 2D image we can simply assume that λ is a constant, but a 3D mesh has different local densities of points. Because density can be reflected by the distance between two neighbouring points, we introduce a function relating density to the average of the normalised distances between a point and its 1-ring neighbours.

$$k(i) = \frac{1}{\lambda} + 1 = \frac{cn}{\sum_{j \in \mathcal{N}(i)} \|p_i - p_j\|} + 1 \tag{17}$$

where n denotes the number of vertex i's 1-ring neighbours and $\mathcal{N}(i)$ denotes its 1-ring neighbourhood. c is a normalisation constant set to the average interpoint distance of the mesh. To compute the Gaussians used to construct the DoG scale space, at different vertices, we find their nearest neighbours using different distance thresholds equal to $2.5\sqrt{k(i)t}$. Eq. (13) is thus replaced by the more sophisticated version:

$$D(p_i, t) = |F(p_i, k(i)t) - F(p_i, t)| \tag{18}$$

Following [Lee et al. 2005], for multi-scale saliency detection, we fix 5 scales of smoothing, corresponding to $t \in \{\epsilon^2, 2\epsilon^2, 3\epsilon^2, 4\epsilon^2, 5\epsilon^2\}$ where ϵ is 0.2% of the length of the diagonal of the bounding box of the model. In a region with low point cloud density, $k(i)$ is small and few points are detected as neighbours, so this area undergoes little smoothing. Where there are only few points, the algorithm has lower confidence that their behaviour indicates some visually important feature. If the same behaviour is indicated by more points, it is more likely to be a reliable feature.

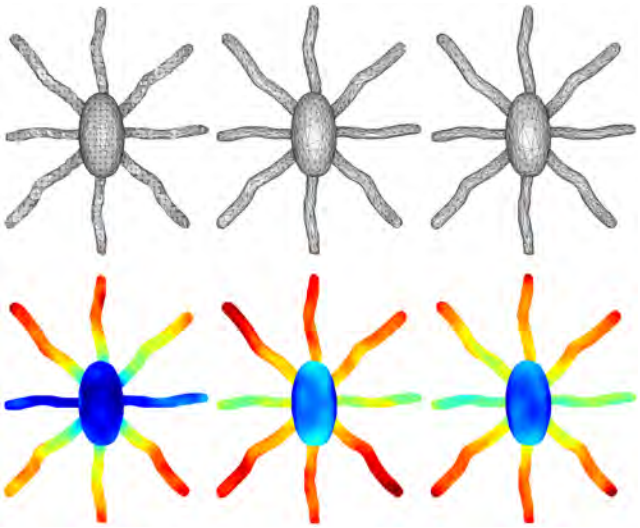

Fig. 10. Saliency maps of an octopus using different simplification methods. Top row: simplified meshes where each mesh contain 1552 vertices. Bottom row: saliency maps. Left column: the clustering decimation of MeshLab's implementation of [Cignoni et al. 2008]. Middle column: the quadric edge collapse decimation of MeshLab's implementation of [Cignoni et al. 2008]. Right column: QSlim [Garland and Heckbert 1997]

3.3 Saliency mapping

Having constructing the scale space as a group of smoothed meshes $M(t)$ via Gaussian filtering using Eqs. 10 and 11, we can compute the scale saliency map $\tilde{\mathcal{S}}(i,t)$ at each scale t by computing spectral mesh saliency at scales $k(i)t$ and t:

$$\tilde{\mathcal{S}}(i,t) = |S(i,k(i)t) - S(i,t)|\,. \tag{19}$$

We then sum the scale saliency maps at all scales to output the final saliency map.

However, computation of eigenvalues and eigenvectors of the Laplacian matrix ($O(m^3)$, where m is the number of vertices) is time consuming. It is believed that the saliency detection process in the human visual system operates in parallel, quickly and simply, over a very limited number of spatial resolutions [Intriligator and Cavanagh 2001]: the human visual system is only sensitive to changes at a limited number of fixed scales. Without a slow process of scrutiny, humans cannot perceive details in a high-resolution image or mesh. This has been used to justify downsampling of images, which greatly speeds up algorithms. For example, Alexe *et al* [Alexe et al. 2010] rescaled images to $s \times s$ for five scales using $s \in \{16, 24, 32, 48, 64\}$ to carry out multi-scale saliency detection. In [Rahtu et al. 2010], four downsampling scales were applied. Thus, for 3D meshes, we consider reducing the number of vertices through simplification.

Fig. 10 shows the results of employing different simplification methods. It can be seen that saliency maps do no change significantly in the presence of local shape variation caused by different simplification methods. This is because our method is based on the spectral attributes of the log-Laplacian spectrum which is a global shape characterisation. Since QSlim [Garland and Heckbert 1997] is also an edge collapse method based on quadric error metric, its resultant saliency map is highly similar to the one produced by the quadric edge collapse decimation of the open-source MeshLab implementation [Cignoni et al. 2008]. The result of clustering decimation which collapses vertices by creating a grid enveloping the mesh and simply discretises them based on the cells is also consistent, particularly given that the pattern of the distribution of saliency is similar to the other two. This demonstrates that our proposed saliency detection method is reasonably insensitive to small variations in local geometry, which is consistent with related findings reported in [Intriligator and Cavanagh 2001].

Thus, in this work, we employ QSlim for mesh simplification. The number of vertices retained in the simplified mesh does not affect the final saliency map significantly as the very finest details are usually not as salient as mid-scale details. For example, we notice the feet of a dog, but not individual toenails. Clearly it is possible to construct counterexamples, such as a sphere with a sharp spike, but such cases infrequently arise amongst real objects. Furthermore, if done in an appropriate manner, simplification will preserve features such as the spike—and if there are many such features, they are not salient.

Having obtained a saliency map for the simplified mesh, we map it to the original mesh using a k-d-tree for speed. The saliency of a vertex p on the original mesh is set to that of the closest point

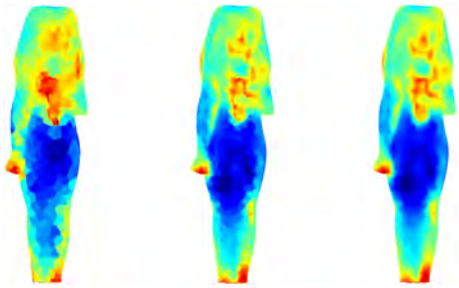

Fig. 11. Saliency maps using simplification to different numbers of points D, and different numbers of iterations i of smoothing. Left: $D = 1000$, $i = 1$. Middle: $D = 2000$, $i = 5$. Right: $D = 2000$, $i = 15$.

Algorithm 1: Spectral Mesh Saliency

Data: A mesh M and v denotes a vertex in M
Result: Saliency map $S(v)$
begin
- Compute a simplified mesh $\hat{M}$ containing m vertices using QSlim;
- Apply Gaussian filters on $\hat{M}$ to generate a bank of smoothed meshes $\hat{M}(t_s)$ using different scale parameters $t_s, s = 1, 2, \ldots, 5$ (see Eq. (10) and (11));
- Generate another bank of smoothed meshes $\hat{M}(k(i)t_s)$ using dynamic scale $k(i)t_s$ where $k(i), i = 1, 2, \ldots, m$ is computed via Eq.(17);
- **for** $s \leftarrow 1$ **to** 5 **do**
  - Calculate the saliency maps for $\hat{M}(k(i)t_s)$ and $\hat{M}(t_s)$ respectively using the method in Section 2.1;
  - Calculate $\hat{\mathcal{S}}(i,t_s)$ as the absolute difference of the two saliency maps;
  - Obtain the scale saliency map $\tilde{\mathcal{S}}(v,t_s)$ by mapping $\hat{\mathcal{S}}(i,t_s)$ to M using the method in Section 3.3;
- Add all scale saliency maps to obtain $\tilde{\mathcal{S}}(v)$ and smooth $\tilde{\mathcal{S}}(v)$;
- Output saliency $S(v) = \log \tilde{\mathcal{S}}(v)$

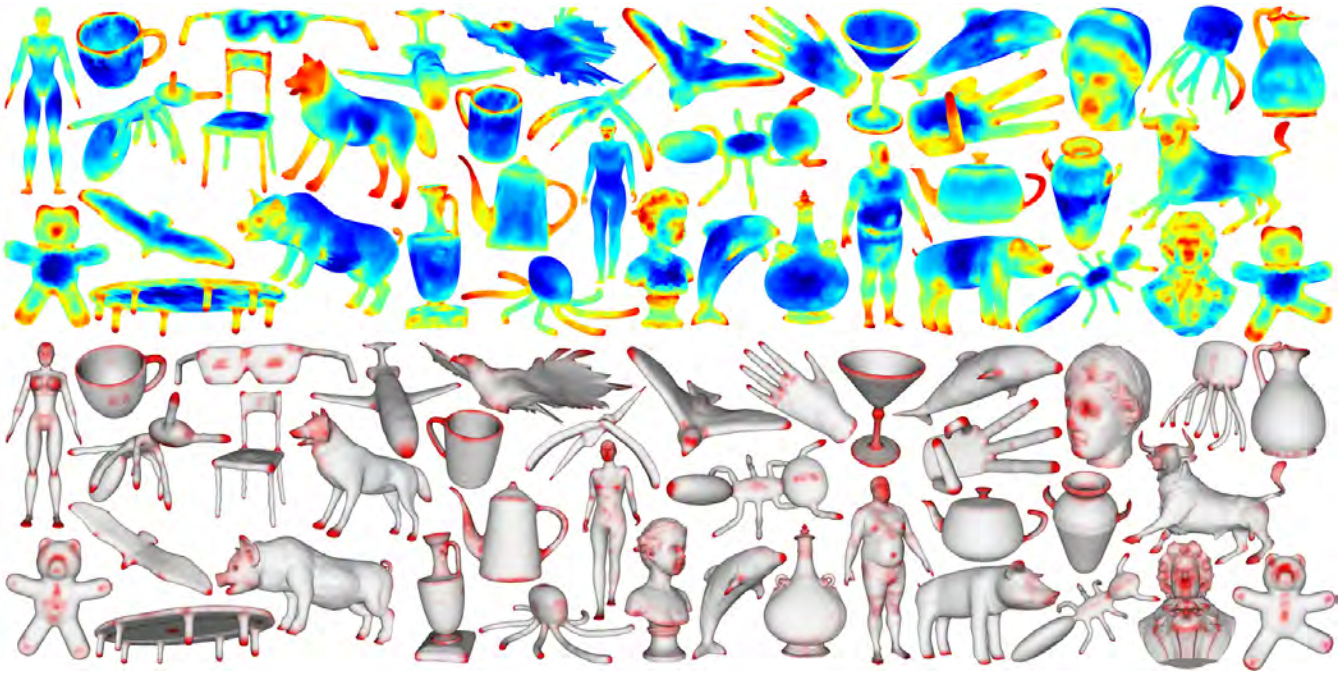

Fig. 12. A gallery of spectral mesh saliency. The upper part of this figure shows our results and the lower part shows corresponding pseudo ground truths provided in [Chen et al. 2012]

$\hat{p}$ on the simplified mesh. As well as providing computational efficiency, saliency mapping in this way presents results in accordance with the visual organisation rule [Koffka 1955], which suggests that people perceive visual components as organised patterns or wholes, instead of many different parts. This implies that a saliency map should be coarse-grained, and a group of neighbouring points representing some salient feature should have the same or similar saliency. Such an effect can be provided by using various centre-surround operators [Itti et al. 1998; Lee et al. 2005; Castellani et al. 2008] or directly using patches as primitives for analysis [Bruce and Tsotsos 2006; Alexe et al. 2010; Goferman et al. 2010]; here it arises from the simplification and mapping scheme.

However, simplification can lead to a result in which the saliency is too obviously discretised over patches with jumps in values between patches, as shown in the left image in Fig. 11. Using more points can alleviate this problem (see the middle image of Figure. 11), but at the cost of significantly greater computational expense. Instead, we smooth the saliency map using several iterations of simple Laplacian smoothing: see the right image in Fig. 11.

The final saliency map is produced by performing a logarithmic mapping after summing the scale saliency maps at all spatial scales and smoothing. Overall, then, our spectral mesh saliency approach is summarised in Algorithm 1.

4. EXPERIMENTAL RESULTS AND ANALYSIS

We have tested our spectral mesh saliency method using various meshes obtained from several different databases (including the Stanford 3D Scanning Repository, the Princeton Segmentation Benchmark [Chen et al. 2009], Cyberware 3D Models, the Watertight Track of the 2007 SHREC Shape-based Retrieval Contest, and the Minolta Range Image Database [Campbell and Flynn 1998]). In this section, we show images depicting mesh saliency visually, we provide comparisons with other saliency methods, and we demonstrate running time. The aim of this section is to demonstrate that our method can detect saliency on a variety of meshes effectively and efficiently. All experiments used a Dual Core 2.4GHz CPU with 4GB RAM.

4.1 Visual comparisons

The Watertight Track of the 2007 SHREC Shape-based Retrieval Contest contains 400 meshes distributed equally among 20 object categories (human, cup, glasses, airplane, ant, chair, octopus, table, teddy, hand, plier, fish, bird, spring, armadillo, bust, mechanical part, bearing, vase, and four-legged animal). We use the data from [Chen et al. 2012][1] as pseudo ground truth; it was produced from data collected in a large-scale online user study using a regression model trained with a leave-one-out procedure based on meshes of the same class (InClass regression). As demonstrated in [Chen et al. 2012], it is extremely difficult to obtain real ground truth for saliency. Firstly, different people have different ideas of what it means to be perceptually important if simply asked to 'indicate important regions'. Secondly, it is difficult to ask the question in a way that does not lead them to a particular kind of answer.

Fig. 12 shows a gallery of our saliency detection results together with corresponding pseudo ground truths. It can be seen that our salient regions are largely consistent with the pseudo ground truths although there are several exceptions such as the asymmetries which appear around the legs of the two octopuses, the two pigs, and the chair. Some of the legs of these models are more salient than others while such asymmetries do not exist in the ground truth data. This is because globally, these legs are not spatially symmetric

[1] http://points.cs.princeton.edu/supplement_sig12/InClass/index.html

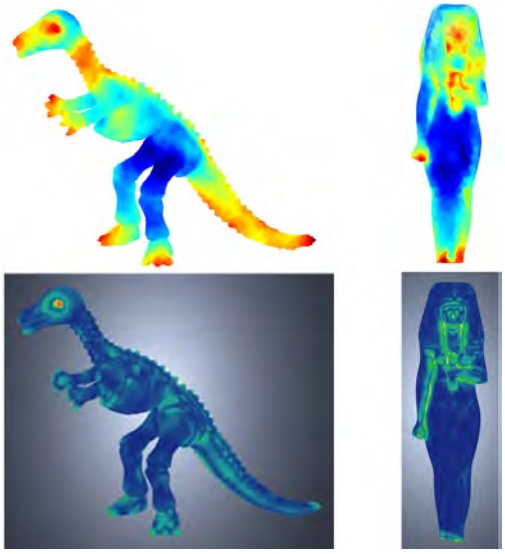

Fig. 13. Saliency detected by our method (top) and the method in [Lee et al. 2005] (bottom). Our results are less influenced by frequent local changes of curvature: our method detects larger continuous regions, rather than smaller, more isolated ones.

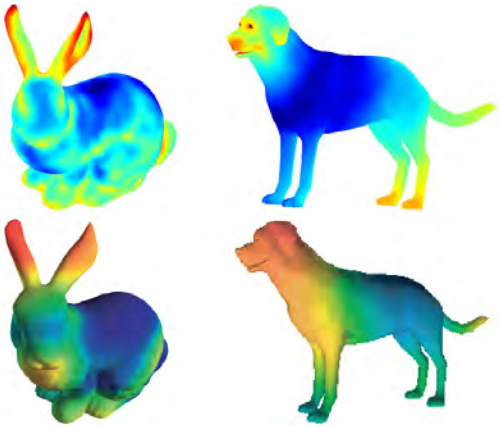

Fig. 14. Saliency detected by our method (top) and the method in [Shilane and Funkhouser 2007] (bottom). Note differences in the facial features.

Fig. 15. Saliency detected by our method (top) and the method in [Leifman et al. 2012] (bottom). Note differences in the eyes and the feet of the angel model, and the eyes of the horse model.

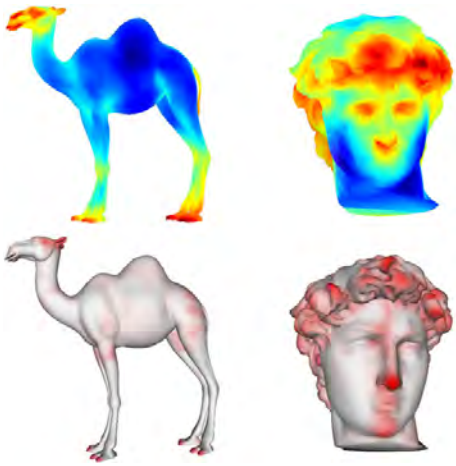

Fig. 16. Saliency detected by our method (top) and the method in [Chen et al. 2012] (bottom). Note differences in facial features such as eyes, nose and mouth of the camel model, eyes of the David's head model.

even if they share similar local shapes. For the octopuses, the global positions of their legs are obviously asymmetric (see Fig. 10 for a comparison where the legs are symmetric). The most salient leg of the octopus at the top-right corner bends most, which at least represents some sort of computational distinction although it is inconsistent with the ground truth which is purely semantic (see Fig. 26 for a more typical example showing such inconsistency between the computational saliency and the ground truth involving high-level significant semantic cues). For the pigs, the back legs are in both cases slightly more salient than the front legs, while for the chair, the opposite is true. But if we only compare the two front (or back) legs of the chair (or each pig), they are almost equally salient, since they have not only the same local geometry, but also global symmetry with respect to each other. For the chair, the two front legs are actually asymmetric with respect to the two back legs, due to the presence of the back; obviously the front and back legs are not globally symmetric for the pigs. However, our method performs symmetrically on strictly symmetric objects, such as the legs and feet of the human in the top-left corner, the ears and feet of the toy bear in the bottom-left corner, the wings of the birds in the top centre and bottom left, and the octopus in Fig. 10, for example.

From these observations, we can see that the log-Laplacian spectrum provides global shape characterisation and thus the detected saliency is closely related to the global structure of the shape, even if such a characterisation is suppressed in the ground truth because of high-level cues. In general, our method tends to find facial features, limb-like structures, shape extremities and geometric protrusions. As [Chen et al. 2012] only provides 2D rendered views of the results, here we can only perform a visual comparison; quantitative assessments are made in Section 5.

We next compare our results to some produced by competing state-of-the-art methods. Because their implementations are usu-

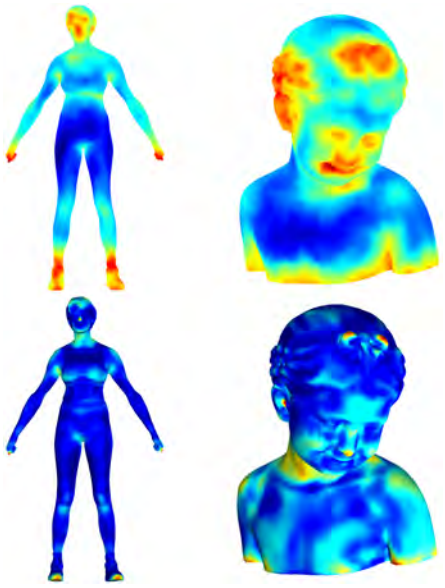

Fig. 17. Saliency detected by our method (top) and the method in [Song et al. 2013] (bottom). Note differences in the facial areas.

Fig. 18. Saliency detection for the Hippo model (containing 23K vertices) using simplified meshes containing different numbers of vertices. Left: 1000 vertices; Middle: 2000 vertices; Right: 3000 vertices.

Table I. Run Times for Computing Mesh Saliency

Model	Number of vertices	Simplification time	Saliency detection time
Foot	10K	0.3s	22.9s
Hippo	23K	0.6s	23.4s
Teapot	26K	0.8s	23.7s
Horse	49K	1.5s	24.4s
Dinosaur	56K	1.9s	24.9s
Isis	188K	7.1s	26.5s
Angel	237K	8.6s	27.0s
Dragon	3.6M	221.3s	43.2s
Lucy	14M	1023.8s	56.1s

ally unavailable, and some methods require tuning of parameters or specific learning data, we used our methods to compute saliency for the same models used in the papers where the methods were reported. Fig. 13 compares our results with those from [Lee et al. 2005]. Our results are less influenced by local changes in curvature when they happen frequently. Consequently, our method detects large and continuous salient regions. For instance, the tail of the Dinosaur and the facial region of the Isis are considered salient by our algorithm, whereas saliency is indicated in a rather disjointed manner by [Lee et al. 2005]. Our method detects saliency in a more global and coherent manner.

Fig. 14 compares our results with those of [Shilane and Funkhouser 2007]. Note however that their goals are different: to find regions that distinguish a shape from objects in a different class, while we aim to locate the salient regions within an object. Thus, while in [Shilane and Funkhouser 2007] the entire head of the dog is marked, we detect facial features, such as the eyes, the nose, and the mouth. In the bunny, both methods mark the ears as the most salient regions. However, our method also captures some meaningful secondary saliency such as the eyes of the bunny. In [Shilane and Funkhouser 2007], the secondary saliency corresponds to the chest.

Fig. 15 compares our results with those of [Leifman et al. 2012]. Note how our method captures the eyes of the angel and the horse while their method fails to detect these regions. In a visual scene that includes faces, humans have a fundamental bias to consider eyes [Henderson et al. 2005; Pelphrey et al. 2002]. Also, the feet of the angel model were found to be interesting by our method, but are not distinctive according to [Leifman et al. 2012].

Fig. 16 compares our results with those of the OutClass regression in [Chen et al. 2012]. Their approach fails to mark certain important facial features such as the eyes, nose and mouth of the camel, and the eyes of David's head, while they are captured by our method. In addition, our method detects the tail of the camel as a salient feature while their method does not.

Fig. 17 compares our results with those of [Song et al. 2013]. For the human, the competing method tends to detect shape extremities such as the hands, the feet, and the nose. However, it fails to detect most of the important facial region. For the bust of the girl, which does not contain significant shape extremities, the detected saliency is dispersed: it fails to correctly locate salient features such as the eyes and the mouth although it detects the facial region to some degree.

4.2 Degree of simplification

In our method, an important parameter which in principle balances accuracy and computational cost is the number of vertices in the simplified mesh (see Section 3.3). Fig. 18 shows that as long as the number of retained points after simplification is large enough, it does not have a significant effect on mesh saliency. This justifies use of simplification to reduce the computational cost. It can also be understood as a demonstration of the psychological claim that in pre-attentive vision, humans tend to ignore the fine details of an object, which correspond to the highest frequency parts in the spectrum [Oliva et al. 2006]. An advantage of employing mesh simplification as the first step is that we can efficiently detect saliency for meshes containing a large number of vertices.

4.3 Computational cost

The complexity of mesh smoothing for constructing the scale space is $O(m)$ where m is the number of vertices of the simplified mesh. Detecting saliency at each scale depends on eigendecomposition of the Laplacian, which is $O(m^3)$. The saliency mapping is based on closest point search and thus has complexity $O(nlog(m))$ where n is the number of vertices of the original mesh. Because of the initial simplification, the computational time of our spectral mesh saliency method increases only slowly as the number of vertices of the input meshes increases, as shown in Table I. For a mesh containing a large number of vertices such as the Stanford Dragon and Lucy, shown in Fig. 19, the total time is dominated by the simplification process. We have employed the classic QSlim method [Garland and Heckbert 1997] for mesh simplification, but there are certainly more efficient simplification methods that could be used instead.

Fig. 19. Saliency detection for meshes containing a large number of vertices. Left: saliency detection for the Dragon model (3.6M vertices); Right: saliency detection for the Lucy model (14M vertices)

5. APPLICATIONS

Mesh saliency is of broad interest since it can potentially improve the perceived quality of results of many mesh processing applications. Integrating mesh saliency into other algorithms is usually fairly straightforward, and typically can be done by using saliency as a weight map. The classic application of saliency-guided mesh simplification was first presented by [Lee et al. 2005]. We present quantitative comparisons through the root mean square (RMSE) error and the MESH error [Aspert et al. 2002] to demonstrate that using saliency to guide simplification improves the results.

We also investigate other applications of mesh saliency where conventionally little attention has been paid to perception-inspired methods. We use saliency for mesh segmentation, using a Markov random field-based method. We also show how saliency-guided segmentation can be used for scan integration, improving results both qualitatively and quantitatively relative to earlier methods.

5.1 Saliency-guided mesh simplification

To incorporate saliency information into mesh simplification, following [Lee et al. 2005], we weight the quadric errors used in QSlim [Garland and Heckbert 1997] by the saliency computed by our method. Fig. 20 demonstrates that the saliency-guided simplification method preserves important surface details in and around salient regions by retaining more points in such regions. The saliency weighting ensures that these salient regions are well-preserved while other regions are greatly simplified.

Fig. 20 qualitatively demonstrates that saliency guided simplification methods have significant advantages in terms of preserving local surface details in perceptually salient regions. We also performed a quantitative comparison between our method and Lee's saliency-guided simplification [Lee et al. 2005] by measuring the RMSE and the MESH error [Aspert et al. 2002] between the original mesh and the simplified mesh, in the expectation that these measures would reflect the visual and geometrical differences between them. The highly cited MESH evaluation method is an improved version of the well-known Metro [Cignoni et al. 1998] method. Like Metro, it also evaluates the distance between meshes but works more efficiently. Although neither of these tools measures how well salient areas are preserved, they are useful tools when comparing the results of two different ways of simplifying meshes based on saliency—we can see to what extent preserving saliency has had a deleterious effect on faithfulness to the original mesh.

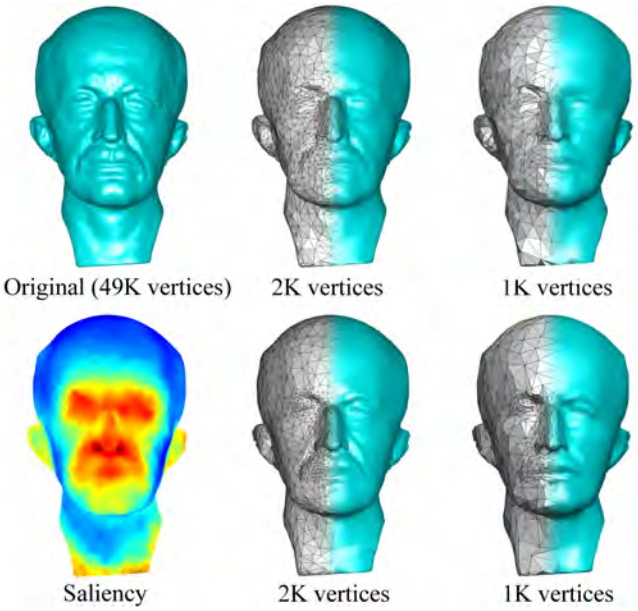

Fig. 20. Simplification results for the MaxPlanck model. Top: models simplified using unweighted QSlim. Bottom: simplification weighted by saliency. Important features such as the eyes, the nose and the mouth are better preserved in the latter case.

The results of the quantitative comparison are shown in Figs. 21 and 22. We tested eight meshes using three different simplification rates: 50%, 80% and 95%. As the simplification rate increases, the RMSE and MESH errors become larger for both methods, as expected. However, the errors of our method are consistently lower than those of Lee's method: the simplified meshes produced by our method are better approximations of the original meshes than those produced by Lee's method.

Most existing mesh simplification methods are designed to minimise a certain error function, which usually leads these methods to try to minimise error values used in measures such as RMSE, Metro and MESH. Strategies that do not directly use such error-driven mechanisms usually result in larger measured errors. For instance, since the saliency-based weight used in our simplification method changes the original order of edge contraction determined to be the order for the minimisation of quadric error in the QSlim method, the RMSE and MESH errors of saliency-guided simplification methods are usually larger than those of the QSlim method. In turn, such a change results in a better visual appearance of the perceptually important regions in the simplified mesh as demonstrated in Fig. 20. However, an interesting finding revealed by Figs. 21 and 22, in comparison to Lee's saliency-guided simplification method, and without any extra mechanism dedicated to minimising a certain error function, our method consistently produces smaller distance-based errors. The reason lies in the neighbourhood consistency of saliency computed by our method, causing groups of neighbouring points to share similar saliency: neighbouring points are more likely to be treated in the same way by the saliency weighting. In the QSlim method [Garland and Heckbert 1997], contraction occurs between two neighbouring points and the order of contraction is dependent on the quadric value Q at each point in the neighbourhood. A significant change in the contraction order can be caused if the saliency-based weighting varies within a neighbourhood. Our method changes the contraction order of QS-

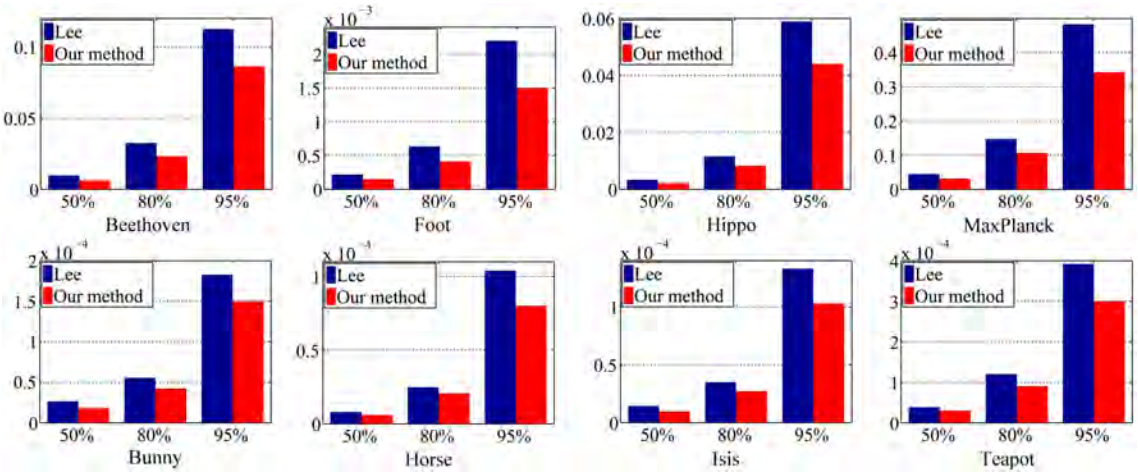

Fig. 21. RMSE errors measured for various test meshes, with different simplification rates, using our method and Lee's method [Lee et al. 2005]. Note that the magnitudes of these errors are different because the mean edge lengths (average interpoint distances) of these meshes are different.

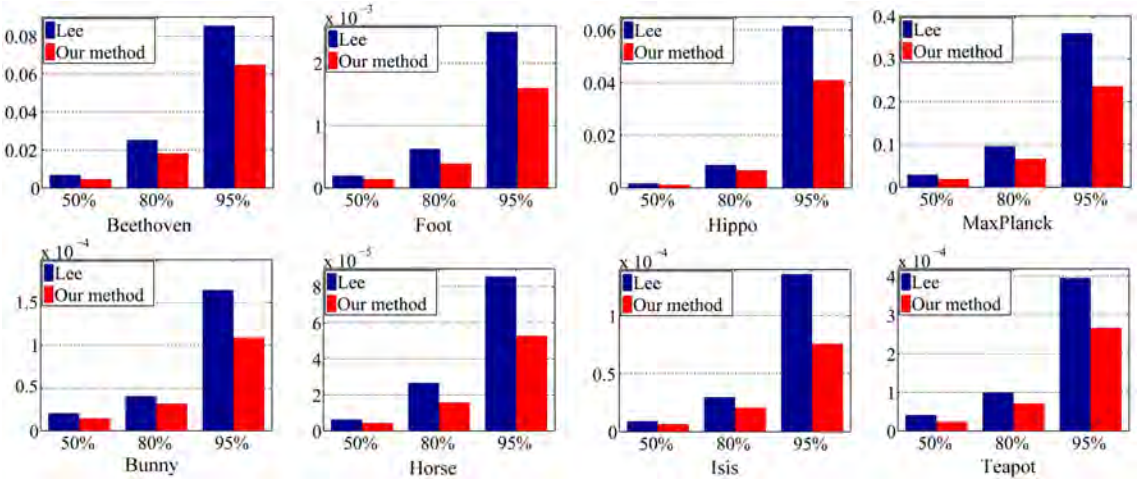

Fig. 22. MESH errors measured for various test meshes, with different simplification rates, using our method and Lee's method [Lee et al. 2005]. Note that the magnitudes of these errors are different because the mean edge lengths (average interpoint distances) of these meshes are different.

lim (the optimal order for the minimisation of a distance-based error) less than Lee's method, as, within a region, points tend to have similar weights. Therefore, as shown in Fig. 21 and 22, the RMSE and MESH errors of our method are generally lower than those for Lee's method.

5.2 Saliency-guided mesh segmentation

In many cases, the aim is to decide whether a point is salient or not, rather than to determine specific saliency values. In Lee's saliency-guided simplification, this is answered through thresholding: a point whose saliency is greater than a threshold is considered a salient point. In [Shilane and Funkhouser 2007], a similar thresholding scheme is employed for saliency-guided shape matching. However, such a thresholding scheme is problematic: points within generally salient regions may have low individual saliency values and vice versa. For example, some points on the top of the lid of the Teapot in Fig. 23 have low saliency, but the lid of a teapot should be a salient region as a whole, as shown in Fig. 12. We thereby propose a novel method to take this into account when using saliency to guide segmentation, the aim being to partition the mesh into consistent and well-defined salient and non-salient regions.

Our method uses a Markov random field (MRF) to impose a consistency constraint on neighbouring points. Such neighbourhood consistency is much stronger than the consistency caused by Gaussian smoothing or mesh simplification during saliency detection. It ensures that a point is labeled as salient no matter how low its saliency value is, if all its neighbours are salient.

A natural idea is to label each point as salient or not-salient using the MRF. Unfortunately, this idea does not work well, as it causes many salient points with high saliency values to be wrongly labeled as non-salient points, presumably because most points on a mesh are non-salient, and imposition of neighbourhood consistency tends to propagate assignment of the dominant non-salient label. To overcome this drawback, we use a set of five labels from the scale indices $\{s\} = \{1, 2, 3, 4, 5\}$. We define a label assignment $\mathbf{s} = \{s_p, \forall p \in M\}, s_p \in \{1, 2, 3, 4, 5\}$. The MRF energy function

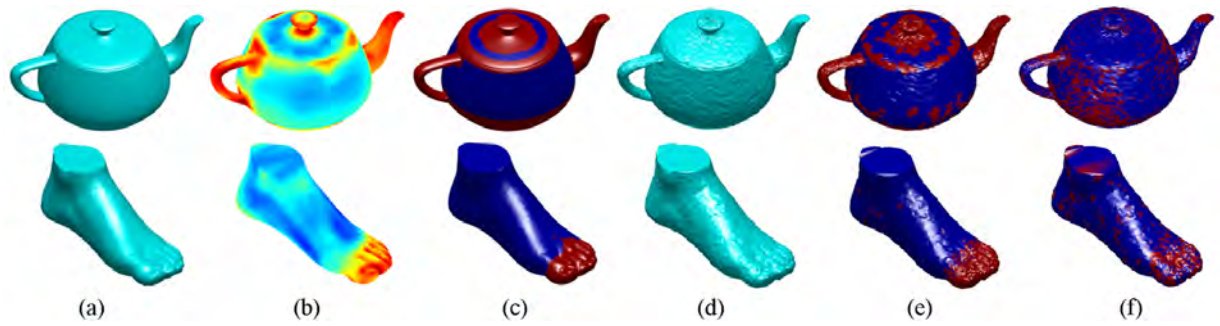

Fig. 23. Saliency-guided mesh segmentation for models with and without noise. Red regions are salient regions and blue regions are non-salient. (a) Original models. (b) Saliency. (c) MRF-based salient segmentation for original models. (d) Noisy models. (e) MRF-based salient segmentation for noisy models. (f) Thresholding-based salient segmentation [Lee et al. 2005] for noisy models.

can be expressed as:

$$E(s) = \sum_{p \in M} E_p(s_p) + \beta \sum_{p} \sum_{q \in \mathcal{N}_p(s)} E_{pq}(s_p, s_q) \tag{20}$$

where β is a weighting parameter and the $\mathcal{N}_p(s)$ denotes the neighbourhood of p at scale s as used in Gaussian filtering in Section 3. In Eq. (20), the observation term $E_p(s_p)$ is a one-point cost function associated with the state (label) that we are most likely to observe at point p, defined as the difference between the saliency corresponding to a certain scale and the largest saliency at p:

$$E_p(s_p) = \left| \tilde{M}_p(s_p) - \max_s \tilde{M}_p(s) \right|, \quad s = 1, 2, 3, 4, 5 \tag{21}$$

where $\tilde{M}_p(s)$ denotes the saliency value of a vertex p at scale s. The label $\hat{s}_p = \arg\max_s(\tilde{M}_p(s))$ always produces the lowest one-point labeling cost at p in such a model.

The compatibility term $E_{pq}(s_p, s_q)$ captures consistency between neighbouring points. It can be regularised by general and scene-specific knowledge. For instance, the smoothness prior is widely used in 2D vision applications: it suggests intensity varies smoothly in a neighbourhood, and usually increases the robustness of labeling. Scanning noise, occlusions, outliers and unreliable triangulations can lead to unreliable saliency detection, turning a point in a non-salient region into a salient point or vice versa. This consistency constraint often corrects such issues by encouraging similarity of labels at adjacent points:

$$E_{pq}(s_p, s_q) = \begin{cases} 1 & s_p \neq s_q, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

The MRF is inferred via graph cuts [Boykov et al. 2001]. There are only two alternative outcomes for each vertex: (i) it fully obeys the observation term, i.e. labels the vertex with the scale at which it is most salient, or (ii) it rejects it and assigns a different label. Typically, most points are assigned the same label and these points comprise the non-salient regions. All other points comprise the salient regions.

Sample segmentation results are shown in Figs. 23(a)–(c). Note how some points on the top of the lid of the teapot with low saliency values are converted into salient points; some points in and around the junction of the handle and the main body of the teapot with high saliency values are converted into non-salient points.

To demonstrate the robustness of our saliency-guided segmentation method, we added Gaussian noise to the original meshes and then compared outputs of our method and of thresholding-based segmentation guided by Lee's mesh saliency. As shown in Fig. 23(d)-(f), our approach can still distinguish salient regions in the presence of a considerable amount of noise. For the teapot, the salient regions produced by our method remain intact while the thresholding-based method only generates fragments across the surface. Similarly, for the foot, our method produces a complete and continuous salient region despite the presence of noise. In contrast, Lee's method fails to recognise meaningful salient regions. Lacking a mechanism to impose neighbourhood consistency, Lee's method trusts the per-point saliency values too much and fails in the presence of noise, when such values are unreliable.

5.3 Saliency-guided scan integration

Scan integration is a crucial technique for constructing a complete 3D surface model from multiple scans. Earlier scan integration techniques based on volumetric methods [Curless and Levoy 1996; Dorai and Wang 1998; Sagawa et al. 2005] or mesh methods [Rutishauser et al. 1994; Sun et al. 2003; Turk and Levoy 1994] can only cope with well-registered scans. When the registration error has similar magnitude to the scanning resolution, these methods often work poorly [Zhou and Liu 2008]. Recently, more robust techniques such as k-means clustering-based integration [Zhou and Liu 2008], FCM integration [Zhou et al. 2009] and an MRF-based method [Paulsen et al. 2010] have been developed which can tolerate a certain degree of registration error and scanning noise. In this section, we propose a saliency-guided method to integrate multiple scans comprsing meshes with holes and boundaries, and show how it significantly improves the integration.

Given two registered 3D scans, P_1 and P_2, we first employ the saliency-guided segmentation proposed in Section 5.2 to partition each of them into salient and non-salient regions. Then, two different schemes are used to integrate points in salient and non-salient regions respectively. In this way, our method redistributes registration errors within each scan in such a way as to ensure that salient regions suffer less from registration errors. Compare this to global registration which redistributes registration errors uniformly across scans.

5.3.1 *Integration in non-salient regions.* Firstly, the overlapping and non-overlapping areas of both P_1 and P_2 are efficiently detected: a point in one scan is deemed to belong to the overlapping area if its distance to the closest (corresponding) point in the other scan is within a threshold; otherwise it belongs to the non-overlapping area.

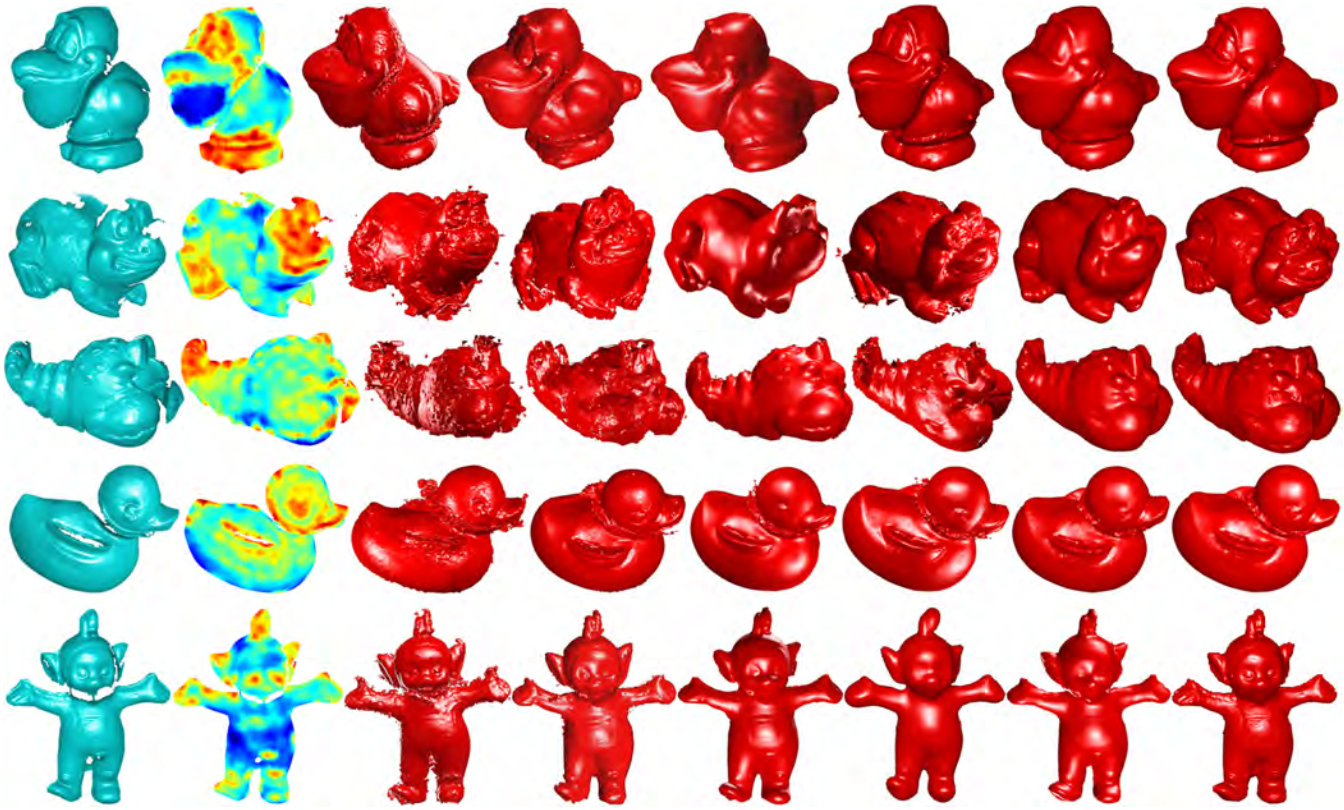

Fig. 24. Rows: Integration results for the Bird, Frog, Lobster, Duck and Teletubby scans. From left to right: one of the input scans (ground truth), saliency map of that input scan, volumetric method [Dorai and Wang 1998], mesh-based method [Sun et al. 2003], FCM [Zhou et al. 2009], k-means clustering [Zhou and Liu 2008], MRF-based method [Paulsen et al. 2010], our saliency-guided method

Next, we compute the normals for the points in the overlapping areas using the method in [Zhou et al. 2009]. We set S_1 and S_2 to be the points in the non-overlapping areas of P_1 and P_2 respectively. We then compute a point set S_{overlap} which represents the integrated points from the overlapping areas. To bring the corresponding points closer to each other, each point $\mathbf{P}$ in an overlapping area is shifted along its normal $\mathbf{N}$ towards its corresponding point $\mathbf{P}^*$ by half of its distance to $\mathbf{P}^*$. A sphere with radius $r = 1.5R$, where R is the average edge length of the scan, is then constructed, centered at each such shifted point from P_2. If other points fall into this sphere, then their original unshifted points are retrieved. The average positions of these unshifted points is then computed and returned. The set of all such positions forms the point set S_{overlap}. The integration result in the non-salient region is $P_{\text{non-salient}} = S_{\text{overlap}} \cup S_1 \cup S_2$.

This strategy compensates for registration errors, as corresponding points are moved closer to each other. It does not alter the tangential spread of the overlap, as points are moved along their normals.

5.3.2 *Integration in salient regions.* To achieve accurate integration, integration in salient regions does not rely on point normals, which are typically unreliable in the presence of noise. Instead, we use the *iterative closest point* (ICP) algorithm to reposition points in salient regions, to reduce the errors caused by inaccurate registration. Although ICP is a classical method to register entire scans, it has also been used for local registration [Brown and Rusinkiewicz 2007]. In our method, ICP is merely applied to points in salient regions. Doing so has four advantages. Firstly, registering the whole dataset is more problematic as it usually includes noise such as outliers or clutter. ICP applied to local salient regions is less likely to suffer from noise, as we are now dealing with a small number of points. Secondly, the initial transformation between neighbouring scans is usually a good enough initial estimate for refinement. Using ICP for local refinement is more likely to produce a reliable result (than when using it for overall registration). Thirdly, using local ICP offers more accurate registration at salient points; this is desirable as salient points are visually more important than the non-salient points. It essentially provides a desired error distribution which typically leads to a visually better integration. Fourthly, local ICP is more efficient than global ICP, as fewer points are involved.

In detail, for points in salient regions, we first detect the overlapping area using the same scheme as for integration of non-salient regions. Then, we employ the ICP algorithm to align points from both overlapping areas, starting from the initial correspondence found. This process not only repositions some points from P_1, but also provides updated correspondences between points from P_1 and points from P_2 in the overlapping areas. We simply integrate each pair of corresponding points by averaging to obtain the integrated point set S_{overlap}. Points in S_1 and S_2, i.e. in the salient non-overlapping areas, remain unchanged. The integration result in the salient region is again $P_{\text{salient}} = S_{\text{overlap}} \cup S_1 \cup S_2$.

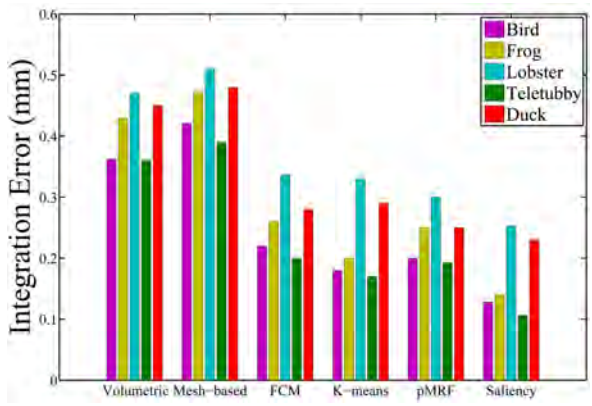

Fig. 25. Integration errors for different integration methods and different datasets.

Finally, we obtain the integrated point cloud $P_{\text{integrated}} = P_{\text{non-salient}} \cup P_{\text{salient}}$. If there are further scans, we next combine $P_{\text{integrated}}$ and the next input scan P_3. After all input scans have been integrated through this procedure, a single integrated point set is obtained. We may then triangulate the final integrated points using methods such as the power crust [Amenta et al. 2001] to make a surface if desired.

A visual comparison of different integration methods is shown in Fig. 24. The saliency-guided method produces noticeably better integration in visually important regions such as the eyes, mouth and wings of the bird, the eyes, fingers and pocket (on the chest) of the Teletubby, and the toes, eyes, and mouth of the frog. Since these regions are ones to which human attention is drawn, the integration result is perceived to be better overall—the errors have been moved to less salient regions where they are less noticeable.

Analysing the results produced by different methods in more detail, we first observe that the volumetric method fails to produce a clean surface model. The mesh-based method and the k-means clustering approach produce improved surface models but also sometimes generate fragments (around e.g. the frog's toes, the Teletubby's ears, the lobster's eyes, and the duck's neck and mouth). The pairwise MRF and FCM methods produce clean surfaces, but ones which tend to suffer from oversmoothing. Fig. 25 shows a quantitative comparison through integration errors [Zhou and Liu 2008] which measures the average squared Euclidean distance between the integrated points and their closest points in the input range scans. Our saliency-guided method performs best in terms of integration error.

6. CONCLUSIONS AND FUTURE WORK

This paper has presented a novel method for detecting mesh saliency, a measure that tries to be in accord with human perception. Unlike previous methods which operate in the spatial domain, we capture information corresponding to mesh saliency in the frequency domain. Previous methods relying on center-surround local operators tend to capture local saliency, while the proposed spectral mesh saliency method outputs a saliency map which captures globally salient, primary features. We have also considered the underlying reasons why spectral mesh analysis is useful for saliency detection.

We have demonstrated how incorporating the proposed mesh saliency can both visually and quantitatively improve the results of several graphics applications such as mesh simplification, mesh segmentation and scan integration. Note that we do not claim that our saliency measure is superior to other measures such as mesh curvature or shape index in all respects—just that spectral mesh saliency is another useful tool. We hope our work will inspire further investigation into frequency-based techniques for mesh saliency detection.

Fig. 26. Limitations. Some high level saliency is not captured by our method.

Future work will focus on how to incorporate high level cues into saliency detection. This is motivated by the limitations of the current framework. As shown in Fig. 26, we fail to detect the breasts as salient in a woman model. A challenge for combining such high level cues of human perception with a low-level computational model is that some of them are inconstant, typically depending on the gender and the age of people involved in the user study. For the glasses, our method does not capture the central points of interest in each lens suggested by [Chen et al. 2012]. Ultimately, however, such high level cues cannot be captured by a geometric approach alone, as they rely on semantic information. Human users expect to see eyes at the centers of the lenses, as eye contact is an important medium of communication between humans; breasts provide an easy clue to help distinguish whether a stranger is male or female.

We are preparing some psychological experiments to further investigate the striking similarities of mesh saliency computed via spectral processing and human provided results. It will also be interesting to explore further novel applications of mesh saliency such as range image registration and shape from shading.

ACKNOWLEDGMENTS

The support of the Higher Education Funding Council for Wales to the One Wales Research Institute of Visual Computing is gratefully acknowledged.

REFERENCES

ALEXE, B., DESELAERS, T., AND FERRARI, V. 2010. What is an object? In *Proc. CVPR*. 73–80.

AMENTA, N., CHOI, S., AND KOLLURI, R. 2001. The power crust. In *Proc. the Sixth ACM Symposium on Solid Modeling*. 249–260.

ASPERT, N., SANTA-CRUZ, D., AND EBRAHIMI, T. 2002. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proc. ICME*.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23,* 11, 1222–1239.

BROWN, B. AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26,* 3.

BRUCE, N. AND TSOTSOS, J. 2006. Saliency based on information maximization. In *Proc. NIPS*.

CAMPBELL, R. AND FLYNN, P. 1998. A www-accessible 3d image and model database for computer vision research. *Empirical Evaluation Methods in Computer Vision*, 148–154.

CASTELLANI, U., CRISTANI, M., FANTONI, S., AND MURINO, V. 2008. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum (Eurographics 2008) 27,* 2, 643–652.

CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28,* 3, 73.

CHEN, X., SAPAROV, A., PANG, B., AND FUNKHOUSER, T. 2012. Schelling points on 3D surface meshes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

CIGNONI, P., CORSINI, M., AND RANZUGLIA, G. 2008. Meshlab: an open-source 3d mesh processing system. *ERCIM News (73)-http://meshlab.sourceforge.net/*, 45–46.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: measuring error on simplified surfaces. *Computer Graphics Forum 17,* 2, 167–174.

CURLESS, B. AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH 1996*. 303–312.

DORAI, C. AND WANG, G. 1998. Registration and integration of multiple object views for 3d model construction. *IEEE Trans. Pattern Anal. Mach. Intell. 20,* 1, 83–89.

FEIXAS, M., SBERT, M., AND GONZÁLEZ, F. 2009. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception (TAP) 6,* 1, 1.

GAL, R. AND COHEN-OR, D. 2006. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics 25,* 1, 130–150.

GARLAND, M. AND HECKBERT, P. 1997. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co., 209–216.

GOFERMAN, S., ZELNIK-MANOR, L., AND TAL, A. 2010. Context-aware saliency detection. In *Proc. CVPR*. 2376–2383.

GUY, G. AND MEDIONI, G. 1997. Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3d data. *IEEE Trans. Pattern Anal. Mach. Intell. 19,* 11, 1265–1277.

HENDERSON, J., WILLIAMS, C., AND FALK, R. 2005. Eye movements are functional during face learning. *Memory & cognition 33,* 1, 98–106.

HOU, X. AND ZHANG, L. 2007. Saliency detection: A spectral residual approach. In *Proc. CVPR*. Ieee, 1–8.

HOWARD, I. 2002. *Seeing in depth*. University of Toronto Press.

HOWLETT, S., HAMILL, J., AND O'SULLIVAN, C. 2005. Predicting and evaluating saliency for simplified polygonal models. *ACM Transactions on Applied Perception (TAP) 2,* 3, 286–308.

HUANG, H. AND ASCHER, U. 2008. Fast denoising of surface meshes with intrinsic texture. *Inverse Problems 24*, 034003.

INTRILIGATOR, J. AND CAVANAGH, P. 2001. The spatial resolution of visual attention. *Cognitive psychology 43,* 3, 171–216.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 20,* 11, 1254–1259.

KIM, Y., VARSHNEY, A., JACOBS, D., AND GUIMBRETIERE, F. 2010. Mesh saliency and human eye fixations. *ACM Transactions on Applied Perception (TAP) 7,* 2.

KOCH, C. AND POGGIO, T. 1999. Predicting the visual world: silence is golden. *nature neuroscience 2*, 9–10.

KOENDERINK, J. 1984. The structure of images. *Biological cybernetics 50,* 5, 363–370.

KOFFKA, K. 1955. *Principles of Gestalt Psychology*. Routledge & Kegan Paul.

LANCASTER, P. AND TISMENETSKY, M. 1985. *The theory of matrices: with applications*. Academic Pr.

LEE, C., VARSHNEY, A., AND JACOBS, D. 2005. Mesh saliency. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

LEIFMAN, G., SHTROM, E., AND TAL, A. 2012. Surface regions of interest for viewpoint selection. In *Proc. CVPR (oral)*.

LÉVY, B. AND ZHANG, H. R. 2010. Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses*. SIGGRAPH '10. ACM, New York, NY, USA, 8:1–8:312.

LINDEBERG, T. 1994. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics 21,* 1-2, 225–270.

LOWE, D. 2004. Distinctive image features from scale-invar-iant keypoints. *International Journal of Computer Vision 60,* 2, 91–110.

MANTIUK, R., MYSZKOWSKI, K., AND PATTANAIK, S. 2003. Attention guided mpeg compression for computer animations. In *Proceedings of the 19th spring conference on Computer graphics*. ACM, 239–244.

OLIVA, A. AND TORRALBA, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision 42,* 3, 145–175.

OLIVA, A., TORRALBA, A., AND SCHYNS, P. 2006. Hybrid images. In *ACM Transactions on Graphics (Siggraph)*. Vol. 25. ACM, 527–532.

PAULSEN, R., BÆRENTZEN, J., AND LARSEN, R. 2010. Markov random field surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 16,* 4, 636–646.

PAULY, M., KEISER, R., AND GROSS, M. 2003. Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum 22,* 3, 281–289.

PAULY, M., KOBBELT, L., AND GROSS, M. 2006. Point-based multiscale surface representation. *ACM Transactions on Graphics (TOG) 25,* 2, 177–193.

PELPHREY, K., SASSON, N., REZNICK, J., PAUL, G., GOLDMAN, B., AND PIVEN, J. 2002. Visual scanning of faces in autism. *Journal of autism and developmental disorders 32,* 4, 249–261.

RAHTU, E., KANNALA, J., SALO, M., AND HEIKKILÄ, J. 2010. Segmenting salient objects from images and videos. In *Proc. ECCV*.

RUDERMAN, D. 1994. The statistics of natural images. *Network: computation in neural systems 5,* 4, 517–548.

RUTISHAUSER, M., STRICKER, M., AND TROBINA, M. 1994. Merging range images of arbitrarily shaped objects. In *Proc. CVPR 1994*.

SAGAWA, R., NISHINO, K., AND IKEUCHI, K. 2005. Adaptively merging large-scale range data with reflectance properties. *PAMI 27*, 392–405.

SHILANE, P. AND FUNKHOUSER, T. 2007. Distinctive regions of 3d surfaces. *ACM Transactions on Graphics 26,* 2, 7.

SONG, R., LIU, Y., MARTIN, R., AND ROSIN, P. 2013. 3d point of interest detection via spectral irregularity diffusion. *The Visual Computer 29,* 6-8, 695–705.

SUN, Y., PAIK, J., KOSCHAN, A., AND ABIDI, M. 2003. Surface modeling using multi-view range and color images. *Int. J. Comput. Aided Eng. 10*.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 351–358.

TREISMAN, A. AND GELADE, G. 1980. A feature-integration theory of attention. *Cognitive psychology 12,* 1, 97–136.

TURK, G. AND LEVOY, M. 1994. Zippered polygon meshes from range images. In *Proc. SIGGRAPH 1994*. 311–318.

VAN DER SCHAAF, A. AND VAN HATEREN, J. 1996. Modelling the power spectra of natural images: statistics and information. *Vision research 36,* 17, 2759–2770.

VON LUXBURG, U. 2007. A tutorial on spectral clustering. *Statistics and Computing 17,* 4, 395–416.

WALTHER, D. AND KOCH, C. 2006. Modeling attention to salient proto-objects. *Neural Networks 19,* 9, 1395–1407.

WOLFE, J. 1994. Guided search 2.0 a revised model of visual search. *Psychonomic bulletin & review 1,* 2, 202–238.

YEE, H., PATTANAIK, S., AND GREENBERG, D. 2001. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Transactions on Graphics (TOG) 20,* 1, 39–65.

ZHANG, H., VAN KAICK, O., AND DYER, R. 2010. Spectral mesh processing. *Computer graphics forum 29,* 6, 1865–1894.

ZHANG, J., ZHENG, J., WU, C., AND CAI, J. 2012. Variational mesh decomposition. *ACM Trans. Graph. 31,* 3 (June), 21:1–21:14.

ZHOU, H. AND LIU, Y. 2008. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition 41,* 1, 152–175.

ZHOU, H., LIU, Y., LI, L., AND WEI, B. 2009. A clustering approach to free form surface reconstruction from multi-view range images. *Image and Vision Computing 27,* 6, 725–747.

Received XXXX 2012; accepted XXXX XXXX