Multiscale Representation of 3D Surfaces via Stochastic Mesh Laplacian

Ran Song

Centre for Secure, Intelligent and Usable Systems, School of Computing, Engineering and Mathematics, University of Brighton, UK

Liping Wang*

School of Information Science and Engineering, Shandong Normal University, Jinan, China

Abstract

Multiscale representation of a 3D surface mesh is a useful tool to understand a mesh both locally and globally. One method is to analyse eigenvalues and eigenvectors of some matrix which represents a discrete operator (e.g. the Laplacian) taking into account the topological and/or geometric structure of the input mesh. However, eigendecomposition is computationally expensive, making this method intractable for meshes containing more than a few thousand vertices. To overcome this problem, we present a novel method for multiscale mesh representation which avoids solving an eigenproblem, based on the proposed stochastic mesh Laplacian. We present the complete algorithm and the theoretical analysis of the stochastic mesh Laplacian. In the experiments, we compare our method with several state-of-the-art approaches to demonstrate its advantages over popular frameworks such as spectral mesh processing, heat diffusion and wavelets. The utility of the method is demonstrated via applications in mesh saliency and interest point detection.

Keywords: Multiscale representation, stochastic matrix, Laplacian, mesh saliency.

1. Introduction

Multiscale representation of manifolds and graphs, in particular 3D surface meshes, provides an analytical tool to capture and interpret information conveying local and global shape structures. It provides a basis for state-of-the-art approaches to a wide range of graphics applications including mesh saliency [1, 2], 3D feature detection [3, 4], mesh segmentation [5, 6], mesh completion [7], 3D face recognition [8] and shape retrieval [9, 10]. It also has promise for applications in manifold approximation, learning and dimensionality reduction [11].

Recent years have witnessed a variety of methods for multiscale mesh representation, with a leading class of approaches based on eigenvalues and eigenvectors of the Laplacian and its generalisations [4, 11, 12, 13, 8]. The eigenvectors form a set of basis functions which uniquely define a vector space in $\mathbb{R}^m$, where m denotes the number of vertices in the mesh. A multiscale representation can be obtained by first projecting the original mesh onto these basis functions in $\mathbb{R}^m$ and then retaining different number of coefficients. However, computing the eigenvalues and eigenvectors of a matrix has a computational complexity of $O(m^3)$ and thus is usually very expensive for a large matrix. Aiming to overcome this problem, we present a novel framework for multiscale mesh representation based on the proposed stochastic mesh Laplacian. Compared to previous model-driven, top-down methods [14, 15, 16], it promises greater flexibility and simplicity as we do not need to design a specific model fitting the problem in each case. Furthermore, since the stochastic mesh Laplacian is locally-supported and discriminative, it can effectively preserve local geometric features. This is an appealing property for many applications such as mesh saliency and interest point detection.

In summary, the three specific contributions of this paper are:

- A novel and efficient method for representing meshes at multiple scales based on the stochastic mesh Laplacian;
- A sketched theoretical analysis of the properties of stochastic Laplacian for multiscale mesh representation;
- Demonstration of uses of the proposed method in applications including mesh saliency and interest point detection.

2. Related work

In this section, we briefly review previous work on multiscale mesh representation, including spectral mesh processing, heat diffusion, wavelets and point-based methods.

Spectral mesh processing. As summarised in [13, 17], mesh processing in the spectral domain was inspired by use of the Fourier transform for 2D images. Explicitly, the eigenvectors of the discrete Laplacian used for spectral processing behave analogously to the basis functions of the discrete Fourier transform and can be regarded as its extension to irregular domains [18]. To obtain a multiscale description, an eigendecomposition of the Laplacian is performed and then the mesh is projected onto the eigenvectors. These eigenvectors can be viewed as manifold harmonics and form an orthonormal basis in the square-integrable Hilbert space. In a multiscale representation based on spectral mesh processing, small-scale mesh information is captured by eigenvectors corresponding to large eigenvalues while large-scale information is captured by eigenvectors corresponding to small (but non-zero) eigenvalues.

Heat diffusion. The generating equation of a linear multiscale representation is the linear heat diffusion equation [19], so a multiscale representation of a manifold can be obtained by solving an appropriate heat diffusion problem. The fundamental solution of the heat diffusion equation is known as the heat kernel. Formally, it can be expressed as $h_t(p, q) = \sum_{f=1}^{m} e^{-\lambda_f t}\psi_f(p)\psi_f(q)$,

*Corresponding author

Email addresses: r.song@brighton.ac.uk (Ran Song), wangliping19872011@gmail.com (Liping Wang*) which can be physically regarded as the amount of heat transferred from a vertex p to another vertex q in time t given a unit heat source at p. λ_f and ψ_f are the fth eigenvalue (in ascending order of size) and corresponding eigenvector of the Laplacian. The heat kernel has quite a few nice properties [4]. Among them, the multiscale property reflects the ability of the heat kernel to capture local or global shape information over small or large t respectively, making heat diffusion and the heat kernel a popular tool in a wide range of applications, including mesh fairing [20], data representation and dimensionality reduction [21], shape segmentation [22] and shape matching [23]. A concise heat kernel signature (HKS) is proposed in [4] as a temporal domain restriction of the heat kernel on a manifold for efficient computation. It also demonstrated that the HKS preserves all of the information about intrinsic geometry.

Wavelets. Both spectral mesh processing and heat diffusion require the inefficient eigendecomposition of the Laplacian. Also, multiscale representations produced by the two methods are not locally supported: the eigenvectors of the Laplacian are determined by the whole manifold and its global properties (e.g. topological invariants). This leads to poor suboptimal results in applications requiring an operator with good localisation property such as mesh denoising, sampling, saliency detection and interest point detection.

For these two reasons, wavelets on meshes have been extensively studied for more than two decades since the publication of [24]. On one hand, wavelets can facilitate multiscale analysis on meshes by dilating mother wavelet, which is potentially more efficient than solving the Laplacian eigenproblem; on the other hand, wavelets provide powerful localisation in both spectral and spatial domains, making them better suited to representing functions whose nature varies in different surface regions. Conventional wavelets-based methods for multiscale mesh representation are built upon explicit mesh subdivision schemes [14, 15, 16]. However, an equally-fine subdivision is computationally costly, and its construction is typically model-driven. Hence, in recent years, more flexible wavelet tools, diffusion wavelets [11, 25, 26, 27], have been proposed. The basic idea is to use powers of a diffusion operator on a manifold to capture multiscale information in both spatial and spectral domains. Compared to conventional wavelets on meshes, the focus shifts from dilation groups acting 'geometrically' in the space, to diffusion semigroups acting on functions in the space [11]. Wavelet-based multiscale representation of 3D surface meshes have been applied to mesh saliency [27], interest point detection [27], surface denoising [28] and mesh optimisation [29].

However, a general difficulty in employing wavelets for graphics applications is the design of the wavelet generating kernel, since it is typically task-oriented, with different wavelet generating kernels having different properties. In contrast, the framework proposed in this paper is generic, as well as being easily implemented, making it widely applicable. Moreover, the proposed method is significantly more efficient than a state-of-the-art wavelet-based method as shown later in our experiments.

Point-based methods. The progressive point set surfaces proposed in [30] constructed a multiscale point-based surface representation by progressively refining a smooth base surface in a coarse-to-fine manner. [31] gave a multiscale surface representation scheme based on point samples, implemented via a smoothing operator and a decomposition operator. The smoothing operator generated successively smoother approximations to the surface while the decomposition operator provided detail preservation. [32] provided a multiscale representation based on interpolation of scattered points. It can generate implicit solids that can be used in applications such as surface morphing and carving.

The point-based methods usually require that the density of points is highly uniform to make sure that the implicit connectivity information can be correctly exploited. So compared with the previous three categories of methods, this requirement limits its applications.

3. Multiscale mesh representation via stochastic mesh Laplacian

The concept of multiscale representation of a 3D surface mesh can be understood in two ways: spatially or spectrally. In the spatial domain, it can be interpreted as a shape description representing a given mesh at different levels of details, explicitly encoding the geometry of local surface areas in different sizes. In the spectral domain, the multiple levels essentially correspond to different frequency bands in the spectrum of the mesh. High frequencies represent fine details while low frequency bands capture coarse details. Global and topological structures in geometric data can be effectively revealed in the spectral domain [13].

As pointed out in [31], a multiscale representation describes a surface at different levels of details without any reference to a particular sampling distribution, which suggests its difference from another popular concept, multiresolution representation. A multiresolution representation refers to a set of surface approximations with varying sampling resolution (i.e., number of vertices), thus describing a surface at potentially different levels of coarseness. For multiscale representation, the sampling resolution might remain the same (e.g., a mesh hierarchy produced by repeatedly applying a Laplacian filter to smooth the mesh). It is difficult to clarify the rigorous mathematical definition for multiscale mesh representation since as a widely used semantic term, its concept often slightly varies in different graphics applications.

3.1. Problem statement

The main target of this paper is to present a framework of multisacle mesh representation which can be used for efficient and effective multiscale analysis on a 3D surface mesh. The problem can be defined as below.

Given a mesh M, we seek a sequence of meshes representing M at multiple scales and their corresponding coefficients. The meshes at multiple scales form a hierarchy of surface approximations, delivering multiple levels of surface details based on M. And, the original M can be reconstructed from any of the meshes using the coefficients.

Our method is composed of two major components, one is the stochastic mesh Laplaican. The other is the formulation of a multiplicative function α for a scale invariant representation. In the following, we first elaborate the two components separately and then propose the complete algorithm integrating them, followed by the sketched analysis on its nice properties.

3.2. Stochastic mesh Laplaican

The Laplacian matrix of a mesh is based on the discretisation of a continuous Laplacian (e.g., the Laplace-Beltrami operator) defined mathematically for a smooth manifold using some weighted sum of adjacent vertices [33]. If a mesh M contains m vertices $P_1, \ldots, P_m$, in its simplest form, the Laplacian matrix can be computed as:

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

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 (connectivity). Ideally, a multiscale representation should not only describe the topological structure of the mesh, but also encode its local geometric details. To incorporate local small-scale geometric information, the adjacency matrix is weighted by the distances between neighbouring vertices:

$$W(i, j) = \frac{1}{\left\|P_i - P_j\right\|^2} A(i, j); \tag{3}$$

This leads to the mesh Laplacian

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

Note that there are other ways to compute the geometric weights for a mesh Laplacian, such as the uniform Laplacian [34] where $W(i, j) = 1$ and the popular cotangent Laplacian [20] where $W(i, j) = \cot\phi_1 + \cot\phi_2$ and ϕ_1 and ϕ_2 are the two angles opposite to the edge linking i and j in the two triangles that have this edge in common. We generate the weights using the distances between neighbouring vertices in Eq. (3) since it is consistent with our computation of the scale invariance function elaborated in Section 3.3, which accelerates the implementation of the algorithm as a whole.

In this work, we propose two novel steps. First, we compute $\widehat{L}$ composed of the absolute values of the elements of the mesh Laplacian: $\widehat{L}_{ij} = \left|L_{ij}\right|$. And second, we normalise $\widehat{L}$ subsequently so that the sum of each row is 1

$$\mathcal{L} = \left[\widehat{L}\right]_r \tag{5}$$

where $[\cdot]_r$ denotes the row-based normalisation.

Since $\mathcal{L}$ is now a stochastic matrix, by a slight abuse of notation, we name $\mathcal{L}$ *stochastic mesh Laplacian* simply to differ it from other mesh Laplacians. Note that researchers have developed different approximations of the real Laplacian (i.e. Laplace-Beltrami operator) used in the heat diffusion equation [4]. These approximations have their own advantages and disadvantages. For example, the popular cotangent Laplacian is known to be a good approximation of the surface normal [34] but does not converge in general [4]. Also, as revealed in [34], it tends to flatten a feature vertex into the surface defined by its 1-ring neighbouring vertices. This is highly undesirable in a multiscale representation where we expect that features can always be well preserved. In this work we shall show that the proposed stochastic mesh Laplacian has a desired property of preserving features in a multiscale mesh representation.

3.3. Scale invariant representation

As pointed by Koenderink [19], the generating equation of a linear multiscale representation is the linear heat diffusion equation. Let $S^{(k)} = (P^{(k)}, T)$ denote the representation of a 3D surface mesh at a particular scale k where $P^{(k)}$ is the set of vertices at scale k with $P^{(1)} = P$ representing the vertices of the original mesh and T is the set of triangular faces. $P^{(k)}$ can be obtained by solving an appropriate heat diffusion problem:

$$\frac{\partial P^{(k)}}{\partial k} - \eta\Delta P^{(k)} = 0, \tag{6}$$

where η is the diffusion constant and Δ denotes the Laplace operator. $\Delta P^{(k)}$ can be computed by a finite difference approximation to $\partial P^{(k)}/\partial k$, using the difference of nearby scales at αk and k:

$$\eta\Delta P^{(k)} = \frac{\partial P^{(k)}}{\partial k} \approx \frac{P^{(\alpha k)} - P^{(k)}}{\alpha k - k} \tag{7}$$

where α is a multiplicative factor. According to Lindeberg [35], the normalisation of the Laplacian is required for scale invariance. Therefore, considering Eq. (7), we have

$$(\alpha - 1)k\eta\Delta P^{(k)} = k\Delta P^{(k)}, \quad \text{so} \quad \alpha = \frac{1}{\eta} + 1. \tag{8}$$

In the diffusion equation Eq. (6), η is the diffusivity, which depends on the density of the material. A 3D surface mesh has different local densities of vertices, which can be reflected by the distance between two neighbouring vertices. We thus formulate α by relating density to the average of the normalised distances between a vertex and its 1-ring neighbours.

$$\alpha(i) = \frac{1}{\eta} + 1 = \frac{cn}{\sum_{j \in \mathcal{N}(i)} \left\|P_i - P_j\right\|} + 1 \tag{9}$$

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. α is a fixed value for each vertex and does not change in each iteration of our algorithm.

Algorithm 1: Multiscale mesh representation via stochastic mesh Laplacian

```
Data: A mesh $M$ represented by a $m \times 3$ vertex matrix $P$
      where each row denotes a vertex and a face matrix $T$
      where each row denotes a triangular face, written as
      $(P, T)$
Result: A sequence of $S^{(k)}$ representing $M$ at multiple scales
        and a sequence of corresponding coefficients $C^{(k)}$.
begin
    Compute the scale invariance function $\alpha$;
    Compute $\widehat{L}$ and then the stochastic mesh Laplacian $\mathcal{L}$,
    input the number of scales as $K$ and a precision $\lambda$;
    Initialise $S^{(1)} = (P^{(1)}, T)$ where $P^{(1)} = P$;
    Initialise $C^{(1)} = 0_{m,3}$;
    Initialise $F = \mathcal{L}$;
    for $k \leftarrow 2$ to $K$ do
        Update $F$ by matrix multiplication $F = F\widehat{L}$;
        Generate the new stochastic Laplacian $\mathcal{F} = [F]_r$;
        for $i \leftarrow 1$ to $m$ do
            for $j \leftarrow 1$ to $m$ do
                if $\mathcal{F}(i, j) < \lambda$ then
                    $\mathcal{F}(i, j) = 0$
        Update the stochastic matrix through multiplying
        each of its rows by $\alpha$ to achieve the scale invariance:
        $F_i = \alpha \circ \mathcal{F}_i$ where $\circ$ denotes the Schur product;
        $F = [F]_r$;
        $P^{(k)} = FP$;
        $C^{(k)} = P^{(k-1)} - P^{(k)}$;
        $S^{(k)} = (P^{(k)}, T)$;
```

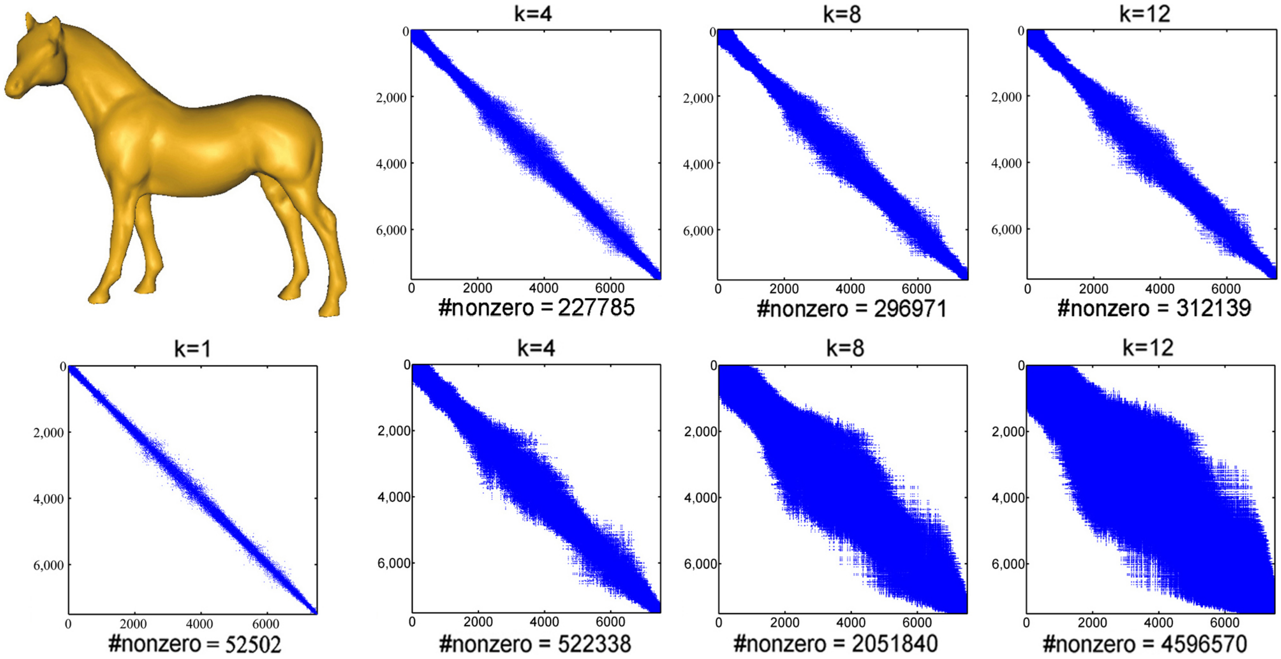


Figure 1: The stochastic mesh Laplacians of a horse mesh containing 7502 vertices. Top row: from left to right, the original mesh, the stochastic mesh Laplacians produced with the thresholding ($\lambda = 0.001$) at the scales of $k = 4, 8$ and 12; Bottom row: from left to right, the initial stochastic mesh Laplacian $\mathcal{L}$ (with the scale of $k = 1$), the stochastic mesh Laplacians produced without the thresholding at the scales of $k = 4, 8$ and 12.

3.4. The complete algorithm

We explicitly show our algorithm in Algorithm 1. It is based on the iterative updates of the stochastic mesh Laplacian through matrix multiplication.

The row-based normalisation in Algorithm 1 guarantees that the vertex matrix is updated by a stochastic matrix in each iteration, which in fact describes the transitions of a Markov chain. It is known that for a stochastic matrix $\mathcal{L}$, $\mathcal{L}_{ij}$ denotes the probability of the one-step transition from the point P_i to one of its neighbouring points P_j. Therefore, the k-th stochastic Laplacian produced through a series of $k-1$ matrix multiplications gives the k-step transition probability. Since the matrix of stochastic Laplacian is sparse and here we also set a precision value λ (set to 0.001 in this work) to rule out small entries, most transitions are prohibitive (transition probability equals zero). And, only the transitions within a neighbourhood is available because it is constructed based on the adjacency matrix. Hence, the k-step transition actually defines a connected k-ring neighbourhood while all other connectivities/paths are prohibited. Also, the row-based normalisation after multiplying a row of the stochastic matrix by α makes the algorithm numerically stable since we used a fixed λ to suppress the small entries of the matrix. Note that as computed in Eq. (9), α is always larger than 1. So if we just keep multiplying a row by α in each iteration but do not normalise it, the fixed threshold λ will be pointless after several iterations since the entries of the stochastic matrix will become too large numerically.

When we perform the multiplication $P^{(k)} = FP$ in Algorithm 1, we essentially perform a displacement in a k-ring neighbourhood for each vertex. Such a vertex displacement leads to the loss of local details and a hierarchy of meshes with multiple levels of details is formed. We shall analyse the vertex displacement in details in Section 3.8. On the other hand, if we record the displacements for all vertices as coefficients, we can reconstruct the original mesh using the coefficients from the coarsest scale.

Although Algorithm 1 contains some heuristic steps, they lead to several desired properties of the stochastic Laplacian. In the following, we shall present some brief theoretical analysis of the properties in the hope that the behaviour of the stochastic Laplican can be well understood.

3.5. Computational complexity

This section provides an analysis for the computational complexity of our method. According to Algorithm 1, the computational complexity depends on the multiplication of the sparse matrices $F = F\widehat{L}$.

Proposition 1. *Given that m is the number of vertices in the original mesh, K is the total number of scales and n denotes the number of non-zero elements in the adjacency matrix, the naive computational complexity of the algorithm is not greater than $O((K-1)mn)$.*

Proof 1. *Given that at some scale, there are f_i non-zero elements in the i-th column of the stochastic Laplacian F (assuming that it contains n' non-zeros in total) and l_i non-zero elements in the i-th row of $\widehat{L}$ respectively, the naive complexity of the matrix multiplication at that scale can be computed as*

$$O = \sum_i f_i l_i \tag{10}$$

Considering $\sum_i f_i \leq n'$, $\sum_i l_i \leq n$ (since the sparseness of $\widehat{L}$ is the same as that of the adjacency matrix) and $0 \leq f_i, l_i \leq m$, we have

$$O = \sum_i f_i l_i \leq \sum_i f_i m \leq mn' \tag{11}$$

and similarly

$$O = \sum_i f_i l_i \leq \sum_i n l_i \leq mn \tag{12}$$

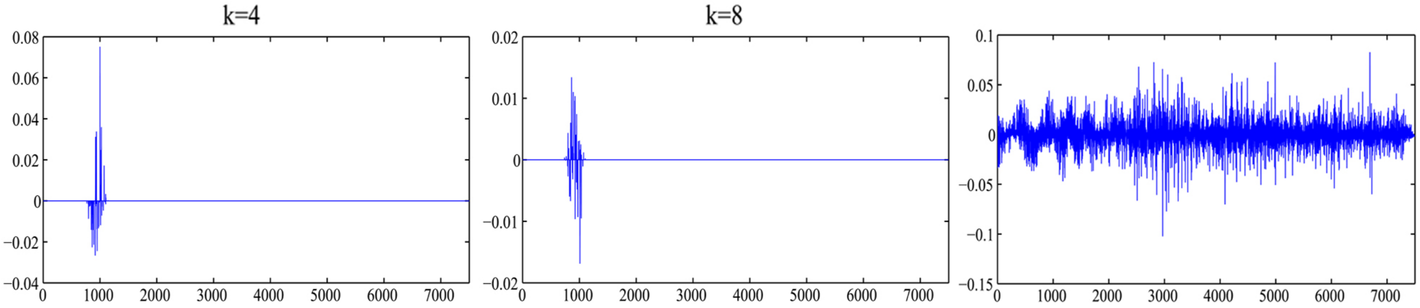


Figure 2: Basis functions produced by our method and the eigendecomposition of the mesh Laplacian. Left: the 1000th basis function at the scale of $k = 4$; Middle: the 1000th basis function at the scale of $k = 8$; Right: the 1000th eigenvector of the mesh Laplacian

According to Algorithm 1, due to the dilation effect produced by the matrix multiplications (see Fig. 1), the sparseness of the stochastic Laplacian F is lower than that of $\mathcal{L}$ which has the same sparseness with the adjacency matrix. Hence $n < n'$ and the computational complexity of producing multiscale representation with K scales is not greater than $O((K-1)mn)$.

Eq. (12) can be understood in this way: each element of $\widehat{L}$ is multiplied by at most m elements from F and we only need to consider the n non-zero elements in $\widehat{L}$. Thus the exact complexity of the proposed method also depends on the sparseness of the stochastic Laplacian F in each iteration. This lies our motivation for introducing the thresholding which accelerates our method significantly. As shown in Fig. 1, with the thresholding delivered by the precision λ, the sparseness of the stochastic Laplacians does not decrease too much at large scales. In a sharp contrast, the number of non-zero elements increases rapidly without applying the proposed thresholding scheme. In short, the efficiency of our method depends on both the number of the vertices and their connectivity.

In comparison, the naive computational complexity of spectral mesh presentation is $O(m^3)$ due to the computationally expensive eigendecomposition. The Gaussian-based methods [1, 36] need to iteratively identify the neighbours of each vertex by nearest neighbour search and then apply a convolution over the neighbourhood. Assuming it computes at K scales and on average the identified neighbourhood includes c vertices, the naive complexity of the Gaussian-based methods is $O(Kcm^2)$. The point-based methods have similar complexity with the Gaussian-based methods since they also identify and process a number of neighbourhoods based only on Euclidean distance. The computational complexity of wavelets-based methods also depends on both the number of vertices and their connectivity. According to [27], the complexity for wavelets-based methods is $O(mn^{2^{K-1}})$. Furthermore, since the wavelet functions are typically dilated by the dyadic powers of some local operators, the wavelet is much less sparse than the stochastic mesh Laplacian at the same scale. Thus in practice, as we can see later in the experiment section, our method is significantly faster than the wavelets-based methods.

3.6. Localisation property

To understand the nice localisation property of the stochastic mesh Laplacian, we analyse the spectral processing method in a comparative manner.

For spectral mesh processing, given that P is the vertex matrix of a mesh including m vertices and E is the matrix whose columns are the m normalised eigenvectors of the Laplacian matrix of the mesh, P can be expressed as

$$P = EC \tag{13}$$

where C denotes the matrix composed of spectral coefficients. Because the columns of E are normalised, C can be computed as

$$C = E^T P \tag{14}$$

where E^T is the transpose of E. Note that in continuous domain, such coefficients $c_i(j) = \left\langle e_j, P \right\rangle$ where c_i is the i-th row of C and the basis e_j is the j-th column of E actually denote the manifold harmonic transform (MHT) [37] which for a function f, is defined as the inner product of a basis function ϕ and f: $H_\phi(j) = \left\langle \phi_j, f \right\rangle$ where j is related to the 'frequency' (corresponding to the index of the eigenvalues of the Laplacian in the spectral mesh processing). A multiscale representation of the mesh is constructed by using the k ($k < m$) leading eigenvectors,

$$C^{(k)} = E^{(k)^T} P \tag{15}$$

where $E^{(k)}$ is the matrix composed of the k eigenvectors corresponding to the k smallest eigenvalues of the Laplacian matrix.

In spectral mesh processing, the eigenvectors of the Laplacian can be interpreted as a set of basis functions which define a Hilbert space. Essentially, a multiscale mesh representation is a sequence of projections of the original mesh in multiple scale spaces defined by multiple sets of basis functions.

In our method, according to Algorithm 1, we have

$$C^{(k)} = P^{(k-1)} - P^{(k)} = (F^{(k-1)} - F^{(k)})P. \tag{16}$$

Thus like $E^{(k)^T}$ in Eq. (15), $F^{(k-1)} - F^{(k)}$ in Eq. (16) can also be interpreted as the basis functions which define a scale space. There is no guarantee that the basis functions $\{\varphi_{k,i} | i = 1, 2, \ldots, m\}$, which are the rows of the matrix $F^{(k-1)} - F^{(k)}$, are orthogonal. But since the matrix is highly sparse and we perform row-based normalisation at each iteration which suppresses the magnitudes of the non-zero elements, it can be validated experimentally that the basis functions are generally 'highly orthogonal' (their inner products are zero or very close to zero). Note that the eigenvectors of the Laplacian matrix are also not strictly orthogonal since the discretization of a continuous Laplacian leads to the loss of its strict symmetry. Even so, the past two decades have witnessed a considerable amount of work using spectral mesh processing and applications based on it, including mesh compression [38, 39], smoothing [18, 40], segmentation [5, 6], quadrangulation [41] and shape matching [42]. As shown in previous papers, loss of strict orthogonality has no significant impact on the multiscale representation.

In fact, it is impossible to construct a basis which is simultaneously fully orthogonal and locally supported [14]. A set of locally-supported basis functions make it possible to associate

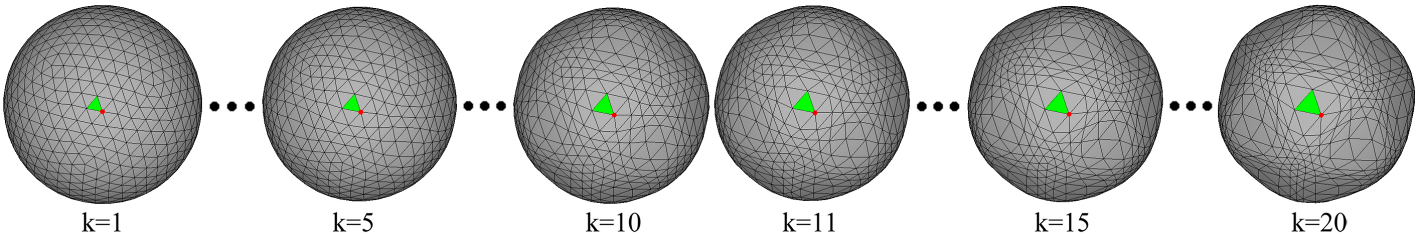


Figure 3: Applying our method on a sphere mesh where $k = 1$ corresponds to the original sphere mesh.

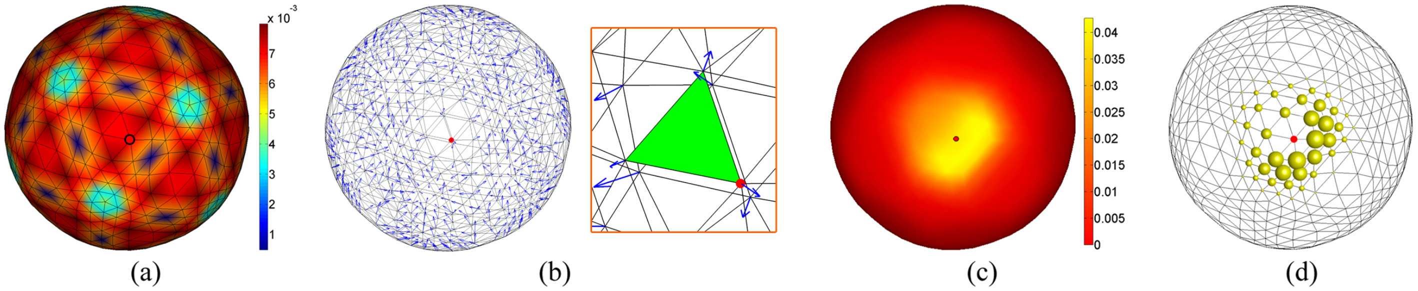


Figure 4: Vertex displacement. (a) Colour map of vertex displacement: warm colour denotes that the vertex moves a large distance from $S^{(10)}$ to $S^{(11)}$. (b) Vector map of vertex displacement: each arrow represents a vector of vertex displacement from $S^{(10)}$ to $S^{(11)}$; the subfigure shows that the marked vertex in $S^{(10)}$ will move to the bottom right and the marked triangle will be stretched in $S^{(11)}$. (c) The stochastic mesh Laplacian ($k = 10$) applied at the marked vertex (d) A discrete visualisation of the stochastic mseh Laplacian where the sizes of the yellow balls are proportional to the corresponding entries of the stochastic mesh Laplacian.

local features captured in the spectral domain with local geometric features observed in the spatial domain. Thus in many graphics applications such as saliency detection, mesh segmentation and interest point detection, a nice localisation property is more desired due to the need of accurate feature localisation. Compared with the Laplacian eigenvectors which are in general highly non-localised, being global Fourier modes (the sinusoidal behaviour of the eigenvectors of the 1D discrete Laplacian can be seen in Fig. 4 in [13]), the nice localisation of the basis functions produced by our method allows to reinterpret most of the operations in the spectral domain in a geometric fashion. An intuitive comparison is shown in Fig. 2. It can be seen that instead of oscillating across the entire domain like the eigenvectors of the Laplacian, the basis functions produced by our method are locally supported.

3.7. Admissibility

Admissibility is a vital property for wavelet transform, as it ensures that the transform can be recovered. Hou and Qin [27] demonstrated that admissible wavelets facilitate mesh processing, generating nice results in applications such as mesh saliency, interest point detection and surface filtering. In [27], the admissibility condition in manifold space is defined as below:

Definition 1. ***Admissibility condition**: On manifold with bounded geometry, a basis function φ is* admissible, *or equivalently satisfies the* admissibility condition, *if $\sum_{j=0}^{\infty} \frac{|H_\varphi(j)|^2}{j} < \infty$, where H_φ denotes MHT of φ.*

Based on Definition 1, we can prove the following proposition:

Proposition 2. *The basis functions $\{\phi_{k,i} | i = 1, 2, \ldots, m\}$ produced by our method, which are the rows of the matrix $F^{(k-1)} - F^{(k)}$, are admissible.*

Proof 2. *According to Algorithm 1, each row $f_i^{(k)}$ of $F^{(k)}$ is normalised to 1. Thus each basis function $\varphi_{k,i}$ has a zero mean*

$$\sum_{j=1}^{m} \varphi_{k,i}(j) = \sum_{j=1}^{m} f_i^{(k-1)}(j) - \sum_{j=1}^{m} f_i^{(k)}(j) = 0. \tag{17}$$

Hence, it vanishes at zero frequency in its MHT,

$$H_{\varphi_{k,i}}(0) = 0. \tag{18}$$

Also, since the basis function $\varphi_{k,i}$ is locally-supported, it has limited bandwidth in its MHT. Assume that its upper frequency is J. We then have,

$$H_{\varphi_{k,i}}(j) = 0, \textit{ for } j > J. \tag{19}$$

Hence, we have

$$\sum_{j=0}^{\infty} \frac{|H_\varphi(j)|^2}{j} = \sum_{j=1}^{J} \frac{|H_\varphi(j)|^2}{j} < \infty, \tag{20}$$

and hence, $\varphi_{k,i}$ is admissible.

Since the basis functions $\varphi_{k,i}$ satisfy the admissibility condition, we can use a method similar to [27] for the reconstruction of the original mesh. In [27], a function (e.g., the vertices of a 3D surface mesh) can be rapidly reconstructed by aggregating a series of K levels of wavelet coefficients $\{\mathcal{W}_f(k)\}_{k=1}^{K}$ and a scaling coefficients at the coarse level $S_f(K)$,

$$f(x) = S_f(0, x) = S_f(K, x) + \sum_{k=1}^{K} \mathcal{W}_f(K, x). \tag{21}$$

In this work, the vertex matrix P of the original mesh $S = (P, T)$ can be reconstructed from a series of coefficients $C^{(k)}$ and the mesh representation at the coarsest level, formulated as below

$$P = P^{(1)} = P^{(K)} + \sum_{k=1}^{K} C^{(k)} \tag{22}$$

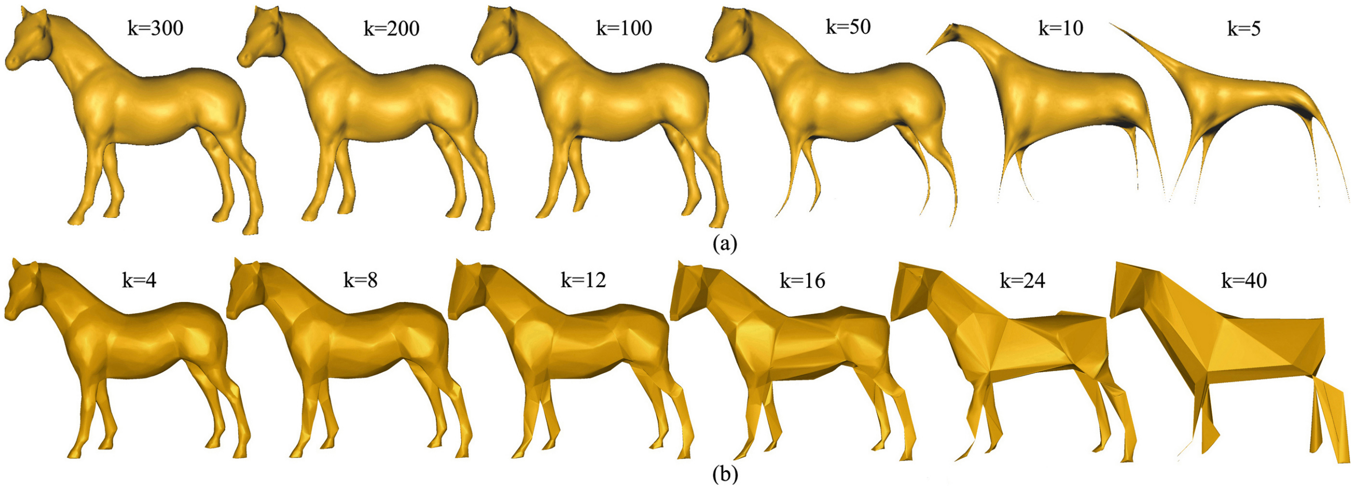


Figure 5: A comparison of two multiscale representations for the horse mesh shown in Fig. 1. (a) The multiscale representation produced by spectral mesh processing using k leading eigenvectors (courtesy of [13]) (b) the multiscale representation produced by the stochastic mesh Laplacian where k denotes the scale.

3.8. Discriminability

The discriminability of a multiscale mesh representation is the capacity that it can discriminate the distinctive vertices in the mesh. It is delivered through discriminative vertex displacement. Ideally, distinctive vertices should remain fixed in place or only moves slightly, while other vertices can be subject to large displacement. Note that neither the uniform Laplacian nor the widely used cotangent Laplacian using a weighting scheme different from Eq. (3) can achieve this as shown and analysed in [34], which typically results in the local features excessively smoothed since a feature vertice tends to move towards the surface defined by its 1-ring neighbouring vertices. By contrast, we shall take a sphere mesh as an example to demonstrate that the heuristics such as the weighting scheme introduced in Eq. (3) for computing the stochastic mesh Laplacian leads to the desired vertex displacement, which guarantees that the proposed multiscale representation based on it is discriminative and thus good at preserving local geometric features.

The original sphere is shown in the leftmost part of Fig. 3. It can be observed that all vertices on the sphere are geometrically equivalent since they have the same curvature. However, they are not topologically equivalent. Some have 5 neighbours while most have 6 neighbours. Such topological difference might not be easily observed on the original mesh but it becomes rather visible on the representations of large scales: the vertex density around the distinctive vertices (the ones with 5 neighbours) is significantly higher when $k = 20$ since they only suffer from small displacements while other vertices move a lot towards them.

To study such vertex displacement, we mark a vertex in red and one of its neighbouring triangles in green as shown in Fig. 3. It can be seen that the green triangle keeps stretching when k is increasing. More specifically, we show vertex displacement between the representations $S^{(10)}$ and $S^{(11)}$ in Fig. 4(a) and (b). Fig. 4(a) shows the colour map of the magnitude of vertex displacement. Intuitively, vertices suffer from different magnitudes of displacement. The distinctive vertices suffer from small displacement while most of the others experience large displacement. In (b), we can see that from $S^{(10)}$ to $S^{(11)}$, the marked vertex will move to the bottom right and the marked triangle will be stretched by looking at the directions of the displacements of its three vertices. Fig. 4(c) and (d) further illustrate what really controls magnitude and direction of vertex displacement. (c) visualises the stochastic mesh Laplacian of scale $k = 10$ applied at the marked vertex in a continuous mode where the colour corresponds to the entries of the stochastic mesh Laplacian, which, as we mentioned in Section 3.4, denote the probabilities of displacements within a 10-ring neighbourhood. Note that the neighbourhood shown here is actually smaller due to the thresholding. (d) visualises the same thing in a discrete mode where each yellow ball denotes the probability of a displacement which starts from the marked vertex and points to the centre (also a mesh vertex) of the yellow ball. Generally, the balls distant from the marked vertex are very small, which means their corresponding vertices barely have an effect on the marked vertex. Thus introducing a precision threshold λ that completely eliminates the effect of extremely small non-zero probabilities has almost no impact on the method but speeds it up since the thresholding makes the matrix of stochastic mesh Laplacian more sparse. It can be seen that the outcome of the combined effect of all yellow balls is a displacement which moves the marked vertex to the bottom right as the bottom right balls are generally larger than the top left ones. Such a combined effect is essentially a weighted averaging in a neighbourhood where the probabilities can be interpreted as the weights assigned to the neighbours.

4. Experiments and applications

We run experiments for evaluating the proposed method using various test meshes and record its time performance.

4.1. Comparisons with state-of-the-art methods

In this section, we compare our results to the ones produced by other existing state-of-the-art methods mentioned in Section 2. Because their implementations are usually unavailable, for direct and fair comparisons, we implement our method on the same meshes used in these papers.

Fig. 5 compares our result with the one produced by spectral mesh processing [13]. Spectral mesh processing produces a representation retaining the global structure of the shape while discarding details. In contrast, while the multiscale representation based on the stochastic mesh Laplacian also discards details, it retains some important local features such as the head and the feet of the horse, even at a large scale.

Figs. 6–8 compare our method with the diffusion-based method [20], the wavelets-based method [27] and the point-based

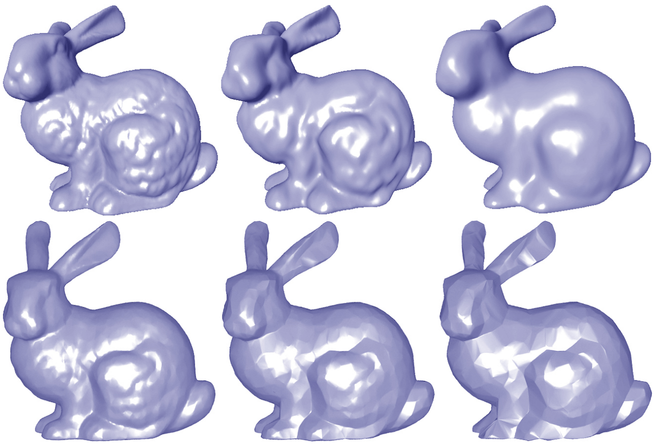


Figure 6: Two multiscale representations of the Stanford Bunny used in [20]. Top: the result of the diffusion-based method [20] with varying time parameters; the leftmost figure shows the original mesh. Bottom: the result of our method with $k = 3, 9, 15$ from left to right.

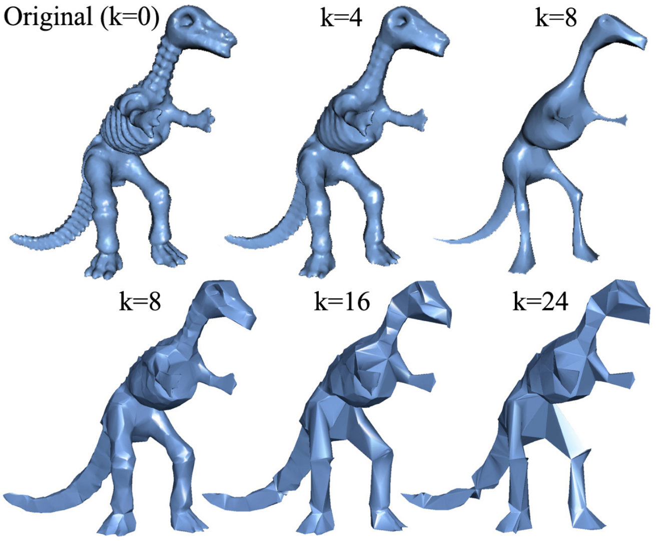


Figure 7: Two multiscale representations of the dinosaur mesh used in [27]. Top: the result of the admissible diffusion wavelets [27] where k is the dyadic power used to dilate the wavelet; Bottom: the result of our method.

method [32] respectively. These figures further demonstrate that our method has an appealing property on the preservation of points representing important features. As shown in Fig. 6, compared to the diffusion-based method using the cotangent Laplacian, our method based on the stochastic mesh Laplacian better preserves the ear of the bunny and the geometrically important grooves that disjoint its leg from the main body. The results produced by the wavelets-based method shown in Fig. 7 appear analogous to the ones produced by the spectral mesh processing shown in Fig. 5 where at a sufficiently large scale, the mesh representation suffers from a uniform contraction from shape extremities towards a certain point somewhere in the middle of the mesh. In comparison, our method better preserves the shape extremities such as the forefeet of the dinosaur. Fig. 8 shows that the point-based method is apparently equivalent to a repeated smoothing operation throughout the surface. And this smoothing seems not discriminative and thus not feature-preserving. In comparison, important features such as the eyes of the squirrel are still recognisable in our mesh representation at a large scale.

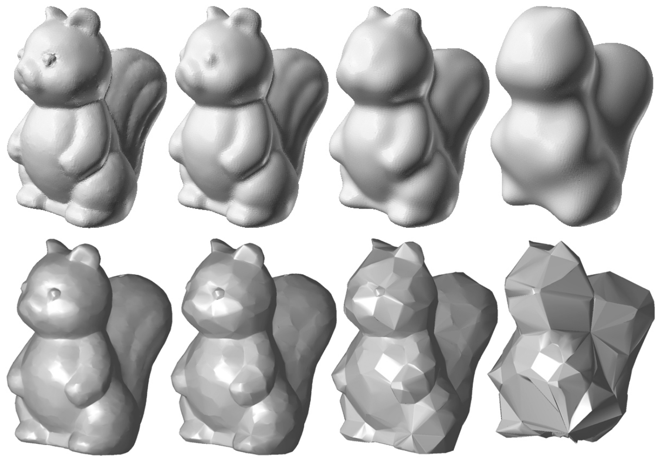


Figure 8: Two multiscale representations of the squirrel mesh used in [32]. Top row: the result of the point-based method [32] where the leftmost figure shows the original mesh; Bottom row: the result of our method where we used $k = 3, 6, 12, 24$ respectively from left to right.

Table 1: Time performance of our method where K denotes the total number of the scales.

Meshes	#Vertices	K	Time (sec)
Horse (Fig. 5)	7502	12	2.18
Bunny (Fig. 6)	35947	8	3.24
Dinosaur (Fig. 7)	14053	10	2.82
Squirrel (Fig. 8)	9995	10	1.65
Girl bust (Fig. 10)	15516	8	1.46
Angel (Fig. 10)	40000	6	2.18
Foot (Fig. 10)	10010	10	1.96
Max Planck (Fig. 10)	49132	6	1.64
Cow (Fig. 10)	11610	10	1.82
Camel (Fig. 10)	9757	6	0.25
Gargoyle (Fig. 12)	10002	6	0.38

Figs. 5 and 7 demonstrate that our method mitigates the shrinkage effect of mesh compared with spectral mesh processing and wavelets-based method. Shrinkage is a typical problem caused by mesh smoothing methods. Our method is not free from shrinkage, which can be observed by noticing the vertex displacement shown in Fig. 4(b). However, we can recognise a comparatively weak degree of shrinkage in Figs. 5–8. This is because in our method, some mesh points which represent important features suffer from very small displacement while other points move towards them. The feature points with small displacement are usually corners and thus visually sustain the mesh since the shrinkage at important features are usually more perceptually significant to human visual system.

4.2. Efficiency

As analysed in Section 3.5, the proposed method is very efficient. As shown in Figs. 5–8, it discards surface details rapidly at small scales while much more slowly at large scales. Hence, in practice, we can set K a relatively small number without significant deterioration of the performance, which also saves computational time by avoiding the multiplication of large matrices with decreasing sparseness. The time performance of the proposed method using a list of 3D meshes shown in this paper is documented in Table. 1 where to compare our method with previ-

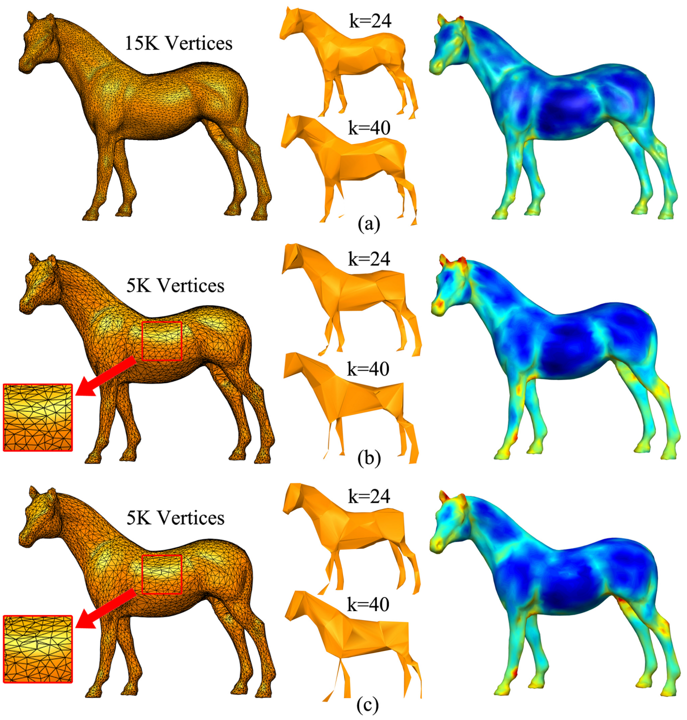


Figure 9: Multiscale representations of meshes with different numbers of vertices and triangulations. Left column: the test meshes; Middle column: multiscale mesh representations produced by the stochastic Laplacian where *k* denotes the scale; Right column: the corresponding mesh saliency. (a) A horse mesh containing 15K vertices; (b) A horse mesh containing 5K vertices generated by the QSlim simplification [43]; (c) A horse mesh containing 5K vertices with a different triangulation generated by a MeshLab's built-in simplification method.

ous methods as fair as possible, all experiments used an old dual core, 2.4GHz, 3.25GB RAM PC intentionally. Please compare Table. 1 with Table.2 of [44], Table.1 of [1], Table.1 of [36] (the column of Mutiscale descriptor) and Table.2 of [27]. In particular, by comparing our method with one state-of-the-art method [27] where experiments were implemented under a hardware environment similar to ours, we can claim that our method is much faster (by at least an order of magnitude). For example, in [27], producing an 8-scale representation of the dinosaur mesh took 35.06 seconds while using our method on the same mesh to generate a 10-scale representation just took 2.82 seconds.

Interestingly in Table. 1, compared to the MaxPlanck mesh, the Angel has a smaller number of vertices while taking longer running time. The same phenomenon can also be found at the Foot compared to the Cow. These phenomena demonstrate our analysis in Section 3.5: the speed of our method depends on both the number of vertices and their connectivity which are sometimes inconsistent (e.g, fewer vertices but more edges).

In general, our method is fast enough to afford rapid shape analysis, which benefits a wide range of applications such as mesh saliency, interest point detection, mesh simplification, mesh segmentation, point sampling, etc. In this paper, we give more details on the applications of mesh saliency and interest point detection. In these applications, the total number of scales K is set to 8 and the precision λ is set to 0.001.

4.3. Reliability

Fig. 9 demonstrates that the proposed multiscale mesh representation is reasonably insensitive to the number of mesh vertices and triangulation. It can be seen that the important local features such as the head and the feet of the horse are always well preserved at various scales. Such reliability is desired since usually the scale of a feature is not dependent on the number of vertices or triangulation but mainly semantic. Consequently, as shown in the right column of Fig. 9, the saliency (a perceptual measure that we shall introduce in the next section) maps of the 3 meshes are highly consistent.

Algorithm 2: Saliency detection

```
Data: A mesh $M$ represented by a vertex matrix $P$ and a
      face matrix $T$
Result: A saliency map $I$
begin
    Compute the scale invariance function $\alpha$;
    Compute $\widehat{L}$ and then the stochastic Laplacian $\mathcal{L}$ and
    input the number of scales $K$ and a precision value $\lambda$;
    $F^{(1)} = \mathcal{L}$;
    for $k \leftarrow 2$ to $K$ do
        $F^{(k)} = F^{(k-1)}\widehat{L}$;
        $\mathcal{F} = \left[F^{(k)}\right]_r$;
        for $i \leftarrow 1$ to $m$ do
            for $j \leftarrow 1$ to $m$ do
                if $\mathcal{F}(i, j) < \lambda$ then
                    $\mathcal{F}(i, j) = 0$
        $F_i = \alpha_i \mathcal{F}_i$;
        $F^{(k)} = [F]_r$;
        $\mathcal{D}^{(k)} = \left|F^{(k)} - F^{(k-1)}\right|$;
        Calculate the difference map as $D^{(k)} = \sum_r \mathcal{D}^{(k)}$
        where $\sum_r$ denotes the row-based summation;
    $\mathcal{I} = \sum_{k=2}^{K} D^{(k)}$;
    $I = \log(\mathcal{I})$
```

4.4. Application 1: mesh saliency

Mesh saliency, first proposed in [1], is a measure of regional importance in accordance with human perception. Various multiscale representation methods [1, 27, 36] have been employed for computing mesh saliency. In these methods, saliency is captured by calculating the differences between adjacent scales and then aggregating them. We follow this scheme and propose an algorithm shown in Algorithm 2 for saliency detection based on our multiscale representation. In this algorithm, we calculate $\mathcal{I}$ as the aggregation of the difference maps at all scales excluding the first one in order to avoid being affected by noise perturbation. We finally output the saliency map I by computing the logarithm of $\mathcal{I}$ merely for a better visualisation effect. Some results using a collection of meshes are shown in Fig. 10.

As summarised from previous papers [1, 27, 45, 46], facial regions, especially the eyes, are typically salient. Also, shape extremities are often regarded as saliency by humans [45, 46]. Thus for quadruped animals, the feet are usually salient. The results in Fig. 10 demonstrate that the stochastic mesh Laplacian can be effectively used for mesh saliency over a variety of objects.

Fig. 11 shows a comparison between our method based on the stochastic mesh Laplacian and two popular alternatives where we replace the stochastic mesh Laplaican with the uniform Laplacian [34] and the cotangent Laplacian [20] respectively in Algorithm 2 for computing mesh saliency. Our method detects some

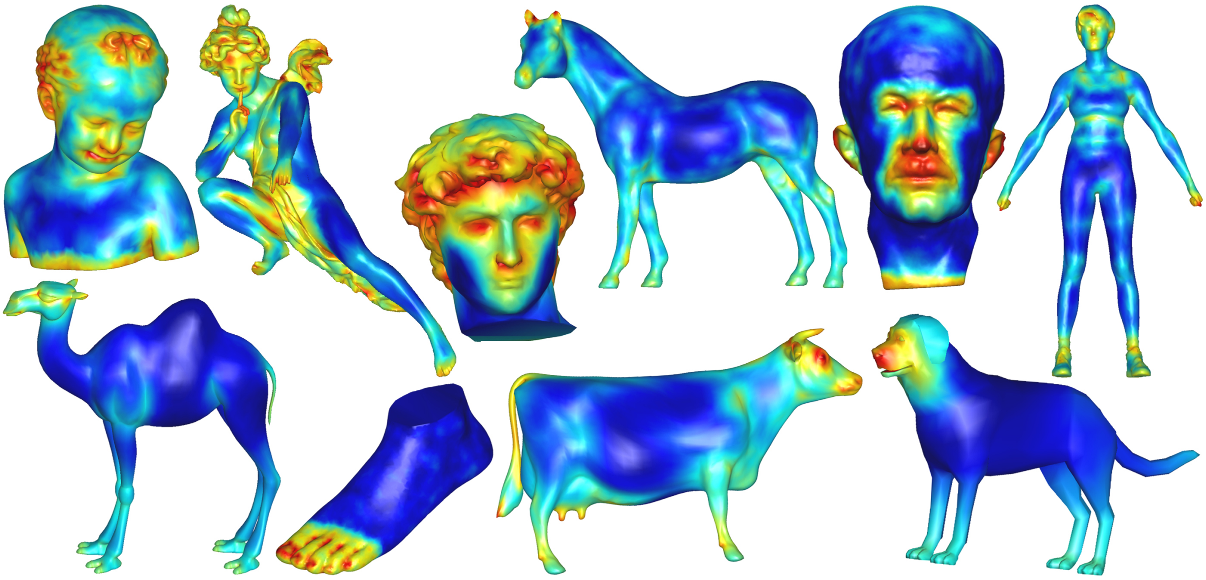


Figure 10: Saliency detection on meshes where warm colour indicates high saliency.

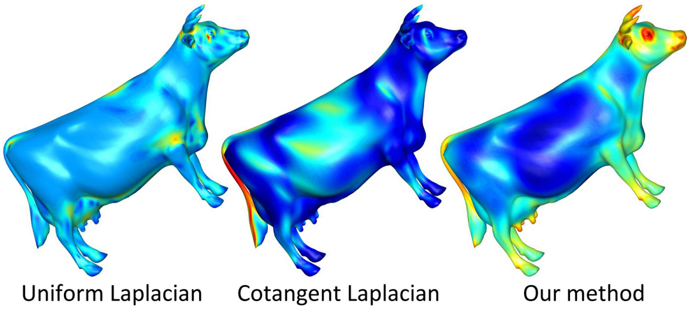


Figure 11: Results of saliency detection using uniform Laplacian [34], cotangent Laplacian[20] and the proposed method respectively.

salient features such as the eyes, the horns and the mouth of the cow while the alternatives fail to detect any meaningful local features. It demonstrates that the proposed stochastic mesh Laplacian is superior to both the uniform and the cotangent Laplacians when applied in the detection of mesh saliency. This further validates our claim made in Section 3.8 that the stochastic mesh Laplacian is good at capturing local geometric features.

4.5. Application 2: Interest point detection

We also employ stochastic mesh Laplacian method for interest point detection. Acquiring ground truth data for this task is easier than saliency since people can just click on the points that they are interested in on a 3D surface mesh. With such ground truth data, we can implement quantitative evaluations and comparisons for competing methods.

Detection of interest points on a 3D surface itself is a fundamental problem in computer vision and graphics. Interest points are often interpreted as local extrema of a scalar field of saliency, with the help of multiscale mesh representation [2, 47, 48]. It can be directly incorporated with the proposed algorithm for saliency detection. More specifically, we first detect the saliency for a mesh using Algorithm 2. Then, we find the maximum saliency I_{max} and compute the standard deviation of saliency I_{std}. I_{max} is normalised by I_{std} as $I'_{max} = I_{max}/I_{std}$. Next, we detect the interest points as the points with both high *global interest* and high *local interest*. A mesh point is of interest if (i) its saliency is not less than $T_{global}I'_{max}$ where T_{global} is a threshold factor fixed to 0.4 in our experiments and (ii) its saliency is maximal over its 1-ring neighbourhood.

We perform our method on the publicly available benchmark [49] and show some visual results in Figs. 12-14. This benchmark also provides human-generated ground truth (interest points manually selected by humans) and the results of 6 comparing methods, namely 3D-Harris [50], 3D-SIFT [48], HKS [4], mesh saliency [1], salient points [2] and SD-corners [47]. With these data and the 3 metrics false negative error (FNE), false positive error (FPE) and weighted miss error (WME) proposed in [49], a quantitative comparison can be conducted.

Fig. 15 gives FNE, FPE and WME graphs with respect to localisation error tolerance r. A mesh point is considered to be 'correctly detected' as an interest point if its geodetic distance to its closest ground truth interest point is not larger than r. Normally, as more points of interest are captured, more false positives are detected although that usually corresponds to a lower FNE. And, if a method tends to mark fewer points of interest, it results in a lower FPE, at the cost of a higher FNE. An algorithm gets a low WME if it manages to detect a point that is frequently voted for by human subjects. Thus it measures the ability of a method to detect the most salient points. In contrast, FNE and FPE treat all ground truth interest points equally. Therefore, an ideal method should keep FNE, FPE and WME all low. From Fig. 15 we can see that our method achieves the lowest FNE and WME while SD-corners mostly achieves the second. For the FPE, our method still has a good performance and in particular, it mostly outperforms SD-corners.

4.6. Limitation

Visually, the multiscale representations produced by the stochastic mesh Laplacian show local coplanarity at large scales, leading to a 'piecewisely planar' effect. This is due to the vertex displacement explained in detail in Section 3.8. Some vertices representing local geometric features remain fixed in place while some other vertices are subject to a large displacement towards them and thus some large planar areas containing few vertices occur. This may limit applications such as mesh compres-

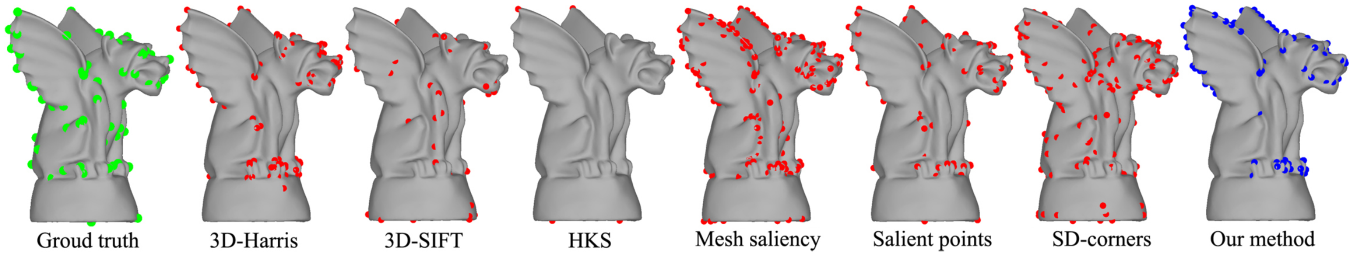


Figure 12: Points of interest detected on the gargoyle mesh by various approaches.

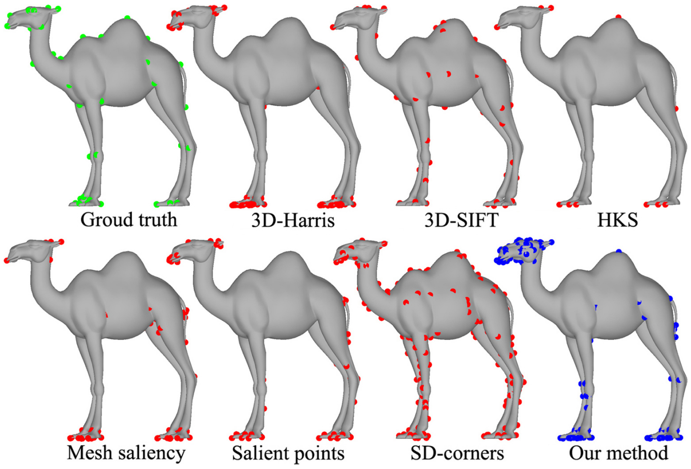


Figure 13: Points of interest detected on the camel mesh by various approaches.

sion [38, 39] and mesh smoothing [18, 40] where some alternatives such as the representation based on cotangent Laplacian might work well as shown in [40].

5. Conclusions

We have developed a novel method for constructing multiscale representation of meshed surfaces. The method is based on iteratively updating the matrix of the proposed stochastic mesh Laplacian. Overall, the stochastic mesh Laplacian is constructed in a bottom-up manner, which starts from the one corresponding to the original mesh and expands as the scale increases. By theoretically analysing the proposed method and relating it to the competing methods, we have revealed its desired properties which potentially benefit a wide range of graphics applications. Through the experiments using a variety of test meshes, we demonstrated the effectiveness and the efficiency of the proposed method. It is much faster than solving the global eigen-system and some wavelets-based techniques.

Compared with most existing methods for multiscale mesh representation, our approach using the stochastic Laplacian is potentially of broad interest in many graphics applications. Although we only showcase two applications with special emphasis on the detection of geometric features in this work, the proposed approach could have a greater impact in the community. This is because the nice properties of the method empower it to have the unique characteristic of multiscale processing towards seeking local features, while retaining gross shape/structure globally. To further demonstrate the impact of our method, future research endeavours will focus on investigating whether it can be used in more applications, with special emphasis on surface filtering, viewpoint selection and shape deformation.

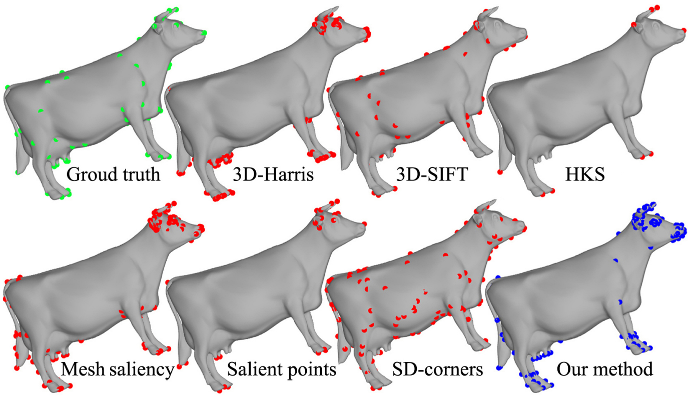


Figure 14: Points of interest detected on the cow mesh by various approaches.

References

[1] C. Lee, A. Varshney, D. Jacobs, Mesh saliency, in: Proc. SIGGRAPH, 2005.

[2] U. Castellani, M. Cristani, S. Fantoni, V. Murino, Sparse points matching by combining 3d mesh saliency with statistical descriptors, in: Proc. Eurographics, 2008, pp. 643–652.

[3] A. Zaharescu, E. Boyer, K. Varanasi, R. Horaud, Surface feature detection and description with applications to mesh matching, in: Proc. CVPR, 2009, pp. 373–380.

[4] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, in: Proc. SGP, 2009, pp. 1383–1392.

[5] R. Liu, H. Zhang, Mesh segmentation via spectral embedding and contour analysis, Computer Graphics Forum 26 (3) (2007) 385–394.

[6] J. Zhang, J. Zheng, C. Wu, J. Cai, Variational mesh decomposition, ACM Transactions on Graphics 31 (3) (2012) 1–14.

[7] X. Wang, J. Hu, D. Zhang, L. Guo, H. Qin, A. Hao, Multi-scale geometry detail recovery on surfaces via empirical mode decomposition, Computers & Graphics 70 (2018) 118–127.

[8] H. Fadaifard, G. Wolberg, R. Haralick, Multiscale 3d feature extraction and matching with an application to 3d face recognition, Graphical Models 75 (4) (2013) 157–176.

[9] A. Bronstein, M. Bronstein, L. Guibas, M. Ovsjanikov, Shape google: Geometric words and expressions for invariant shape retrieval, ACM Transactions on Graphics 30 (1) (2011) 1.

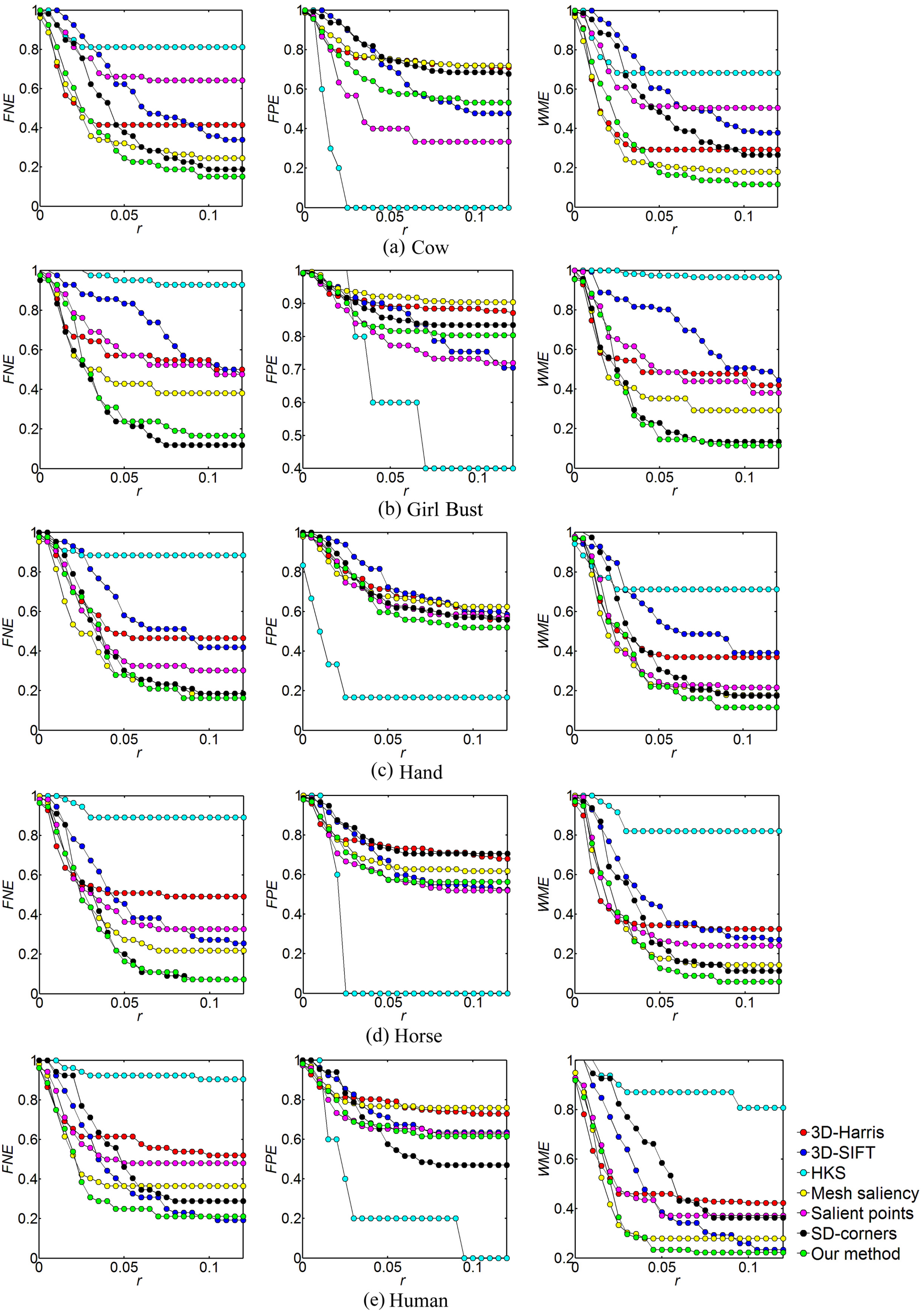
[10] J. Xie, G. Dai, F. Zhu, E. K. Wong, Y. Fang, Deepshape: Deep-learned shape descriptor for 3d shape retrieval, IEEE transactions on pattern analysis and machine intelligence 39 (7) (2017) 1335–1345.

[11] A. D. Szlam, M. Maggioni, R. R. Coifman, J. C. BremerJr, Diffusion-driven multiscale analysis on manifolds and graphs: Top-down and bottom-up constructions, in: Optics & Photonics, 2005.

[12] S. Mahadevan, M. Maggioni, Value function approximation with diffusion wavelets and laplacian eigenfunctions, Advances in neural information processing systems 18 (2006) 843.

[13] B. Lévy, H. R. Zhang, Spectral mesh processing, in: SIGGRAPH Courses, 2010, pp. 8:1–8:312.

[14] M. Lounsbery, T. D. DeRose, J. Warren, Multiresolution analysis for surfaces of arbitrary topological type, ACM Transactions on Graphics 16 (1) (1997) 34–73.



(a) Cow

(b) Girl Bust

(c) Hand

(d) Horse

(e) Human

Figure 15: Quantitative comparisons of various methods for interest point detection using various test meshes.

[15] I. Guskov, W. Sweldens, P. Schröder, Multiresolution signal processing for meshes, in: Proc. SIGGRAPH, 1999, pp. 325–334.

[16] S. Valette, P. Prost, Wavelet-based multiresolution analysis of irregular surface meshes, IEEE Transactions on Visualization and Computer Graphics 10 (2) (2004) 113–122.

[17] O. Sorkine, Differential representations for mesh processing, Computer Graphics Forum 25 (4) (2006) 789–807.

[18] G. Taubin, A signal processing approach to fair surface design, in: Proceedings of SIGGRAPH, ACM, 1995, pp. 351–358.

[19] J. Koenderink, The structure of images, Biological cybernetics 50 (5) (1984) 363–370.

[20] M. Desbrun, M. Meyer, P. Schröder, A. H. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, in: Proc. SIGGRAPH, 1999, pp. 317–324.

[21] S. S. Lafon, Diffusion maps and geometric harmonics, Ph.D. thesis, Yale University (2004).

[22] F. De Goes, S. Goldenstein, L. Velho, A hierarchical segmentation of articulated bodies, Computer graphics forum 27 (5) (2008) 1349–1356.

[23] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, G. Sapiro, A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching, International Journal of Computer Vision 89 (2) (2010) 266–286.

[24] E. J. Stollnitz, A. D. DeRose, D. H. Salesin, Wavelets for computer graphics: a primer. 1, Computer Graphics and Applications, IEEE 15 (3) (1995) 76–84.

[25] M. Maggioni, J. C. Bremer Jr, R. R. Coifman, A. D. Szlam, Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs, in: Proc. SPIE Wavelet XI, Vol. 5914, 2005.

[26] R. R. Coifman, M. Maggioni, Diffusion wavelets, Applied and Computational Harmonic Analysis 21 (1) (2006) 53–94.

[27] T. Hou, H. Qin, Admissible diffusion wavelets and their applications in space-frequency processing, IEEE Transactions on Visualization and Computer Graphics 19 (1) (2013) 3–15.

[28] B. Dong, Q. Jiang, C. Liu, Z. Shen, Multiscale representation of surfaces by tight wavelet frames with applications to denoising, Applied and Computational Harmonic Analysis 41 (2) (2016) 561–589.

[29] R. Florea, A. Munteanu, S.-P. Lu, P. Schelkens, Wavelet-based L_∞ semi-regular mesh coding, IEEE Transactions on Multimedia 19 (2) (2017) 236–250.

[30] S. Fleishman, D. Cohen-Or, M. Alexa, C. T. Silva, Progressive point set surfaces, ACM Transactions on Graphics 22 (4) (2003) 997–1011.

[31] M. Pauly, L. Kobbelt, M. Gross, Point-based multiscale surface representation, ACM Transactions on Graphics 25 (2) (2006) 177–193.

[32] Y. Ohtake, A. Belyaev, H.-P. Seidel, A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions, in: Proc. Shape Modeling International, 2003, pp. 153–161.

[33] T. K. Dey, P. Ranjan, Y. Wang, Convergence, stability, and discrete approximation of laplace spectra, in: Proc. ACM-SIAM Symposium on Discrete Algorithms, 2010, pp. 650–663.

[34] A. Nealen, T. Igarashi, O. Sorkine, M. Alexa, Laplacian mesh optimization, in: SIGGRAPH, ACM, 2006, pp. 381–389.

[35] T. Lindeberg, Scale-space theory: A basic tool for analyzing structures at different scales, Journal of applied statistics 21 (1-2) (1994) 225–270.

[36] J. Wu, X. Shen, W. Zhu, L. Liu, Mesh saliency with global rarity, Graphical Models.

[37] B. Vallet, B. Lévy, Spectral geometry processing with manifold harmonics, in: Computer Graphics Forum, Vol. 27, 2008, pp. 251–260.

[38] Z. Karni, C. Gotsman, Spectral compression of mesh geometry, in: Proc. SIGGRAPH, 2000, pp. 279–286.

[39] M. Ben-Chen, C. Gotsman, On the optimality of spectral compression of mesh data, ACM Transactions on Graphics 24 (1) (2005) 60–80.

[40] B. Kim, J. Rossignac, Geofilter: Geometric selection of mesh filter parameters, Computer Graphics Forum 24 (3) (2005) 295–302.

[41] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, J. C. Hart, Spectral surface quadrangulation, in: Proc. SIGGRAPH, 2006, pp. 1057–1066.

[42] M. Reuter, F.-E. Wolter, N. Peinecke, Laplace–beltrami spectra as 'shape-dna' of surfaces and solids, Computer-Aided Design 38 (4) (2006) 342–366.

[43] M. Garland, P. Heckbert, Surface simplification using quadric error metrics, in: Proc. SIGGRAPH, 1997, pp. 209–216.

[44] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans, Reconstruction and representation of 3d objects with radial basis functions, in: Proc. SIGGRAPH, ACM, 2001, pp. 67–76.

[45] Y. Kim, A. Varshney, D. Jacobs, F. Guimbretiere, Mesh saliency and human eye fixations, ACM Transactions on Applied Perception 7 (2) (2010) 12:1–12:13.

[46] G. Leifman, E. Shtrom, A. Tal, Surface regions of interest for viewpoint selection, IEEE transactions on pattern analysis and machine intelligence 38 (12) (2016) 2544–2556.

[47] J. Novatnack, K. Nishino, Scale-dependent 3d geometric features, in: Proc. ICCV, 2007, pp. 1–8.

[48] A. Godil, A. Wagan, Salient local 3d features for 3d shape retrieval, in: Proc. SPIE, 2011.

[49] H. Dutagaci, C. Cheung, A. Godil, Evaluation of 3d interest point detection techniques via human-generated ground truth, The Visual Computer 28 (2012) 901–917.

[50] I. Sipiran, B. Bustos, Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes, The Visual Computer 27 (11) (2011) 963–976.