

# Sparse approximation of 3D shapes via spectral graph wavelets

Ming Zhong · Hong Qin

Published online: 10 May 2014  
© Springer-Verlag Berlin Heidelberg 2014

**Abstract** This paper investigates the compressive representation of 3D meshes and articulates a novel sparse approximation method for 3D shapes based on spectral graph wavelets. The originality of this paper is centering on the first attempt of exploiting spectral graph wavelets in the sparse representation for 3D shape geometry. Conventional spectral mesh compression employs the eigenfunctions of mesh Laplacian as shape bases. The Laplacian eigenbases, generalizing the Fourier bases from Euclidean domain to manifold, exhibit global support and are neither efficient nor precise in representing local geometry. To ameliorate, we advocate an innovative approach to 3D mesh compression using spectral graph wavelets as dictionary to encode mesh geometry. In contrast to Laplacian eigenbases, the spectral graph wavelets are locally defined at individual vertices and can better capture local shape information in a more accurate way. Nonetheless, the multiscale spectral graph wavelets form a redundant dictionary as shape bases, therefore we formulate the compression of 3D shape as a sparse approximation problem that can be readily handled by powerful algorithms such as orthogonal matching pursuit. Various experiments demonstrate that our method is superior to the existing spectral mesh compression methods.

**Keywords** Sparse approximation · Spectral graph wavelets · Mesh compression

---

M. Zhong (✉) · H. Qin  
Department of Computer Science,  
Stony Brook University, Stony Brook, NY 11794, USA  
e-mail: mzhong@cs.stonybrook.edu

H. Qin  
e-mail: qin@cs.stonybrook.edu

## 1 Introduction and motivation

In recent years, sparse approximation techniques have become increasingly popular in various fields of signal processing and analysis, especially in image processing and computer vision. Its widespread applications include data compression, signal denoising, shape recognition, etc. Most of the existing works are focusing mainly on the regular domain of 2D images, while approximating/compressing signals defined over graphs or manifolds are much less investigated due to the irregularity of underlying domains. This paper takes an initiative to explore the challenging problem of sparse approximation of discrete 3D shapes for compressive shape representation.

The fundamental idea of sparse approximation is to estimate a signal as the linear combination of a very few vectors selected from a large number of candidate vectors serving as bases. These candidate vectors, also called *atoms*, constitute a set called the *dictionary*. With a given dictionary, the signal is encoded as the coefficients w.r.t. the selected atoms and can be easily reconstructed because of sparsity. The rationale of sparse approximation is that most meaningful high-dimensional signals must have some intrinsic structures or patterns, which can be exploited for efficient representation in a lower-dimensional subspace. Because of the versatility of acquired signals, dictionaries are typically redundant or overcomplete, allowing great freedom and flexibility in design to accommodate a sparse set of atoms that can better capture the signals' intrinsic characteristics.

Conventional Fourier analysis decomposes a signal into mutually independent components with the multiscale and orthogonal Fourier bases. Compression is achieved by discarding certain number of high-frequency Fourier coefficients. This scheme has been transplanted to mesh compression, using the eigenbases of mesh Laplacian, i.e., the mani-

fold harmonic basis (MHB), as the Fourier bases [7]. The key disadvantage of Fourier compression is that the Fourier bases are only localized in the frequency domain yet having global support in the spatial domain, and thus are not efficient in encoding local signal information. A popular and powerful solution is to use wavelet bases, which are functions localized in both location and frequency and can capture local signal information in a more compact and efficient way.

In this paper, we propose to use the spectral graph wavelets (SGW), pioneered by Hammond et al. [6], for mesh compression. To the best of our knowledge, we believe that it is the first attempt to exploit the SGW in sparse representation with a unique application in 3D geometric compression. The SGW has many attractive properties such as spatial localization, being smooth, multiscale, and shape-aware, and being flexible and versatile for 3D shapes of arbitrary topology and complicated geometry, hence is well suited for encoding shapes with many local details. We employ the SGW as shape bases to construct redundant dictionary with multiscale wavelets centered around each vertex, and employ the simultaneous orthogonal matching pursuit (S-OMP) algorithm to find a sparse coding of the original shape geometry. This paper's primary contributions are hinging upon the unique integration of the spectral graph wavelets (SGW) and sparse representation and its powerful application in 3D shape compression.

Through our extensive experiments, we wish to demonstrate that our compression method outperforms the MHB-based Fourier compression in terms of compression quality at different compression ratio settings. Since our sparse shape approximation framework is independent of any data-specific dictionary design, other formulations of bases or dictionaries, as well as other powerful sparse approximation algorithms, can all be migrated into our system with very little extra workload. Therefore, we are expecting further computational improvement in compression performance in the near future.

## 2 Brief background review

### 2.1 Sparse approximation

Harmonic analysis techniques such as Fourier transform and wavelet transform have achieved tremendous success in compressing images, audio, and video signals. Prominent applications include the JPEG [19] and JPEG2000 [16] image compression standards, which are based on 2D discrete cosine transform and discrete wavelet transform, respectively. The key idea of harmonic compression is to decompose the original signal into a set of harmonic basis and reduce the representation size by discarding coefficients that correspond to much less-noticeable signal components.

While traditional harmonic analysis oftentimes uses orthogonal basis, such as the Fourier basis, recent years have witnessed the increasing popularity of sparse approximation methods, which enable the utility of redundant or overcomplete dictionaries for signal representation. From the dictionary of elementary signals, a small subset that can best capture the structure of the input signal is selected, and the input signal is approximated by a linear combination of the selected signals.

One of the most commonly used approaches to sparse approximation is the greedy pursuit method. The central idea is to iteratively refine a sparse solution in a greedy manner. More specifically, in each iteration one or more atoms of the dictionary are chosen and the corresponding coefficients are modified such that the greatest improvement in approximation quality can be achieved. Representative greedy pursuit algorithms include matching pursuit (MP) [12], orthogonal matching pursuit (OMP) [13], and simultaneous orthogonal matching pursuit (S-OMP) [17]. In particular, the S-OMP is suitable for solving the simultaneous sparse approximation problem where the input signal has multiple correlated channels and the same subset of atoms is to be used for every channel.

### 2.2 Graph wavelets and mesh compression

A 3D mesh can be expressed as the connectivity information of the mesh topology plus the 3D mesh coordinates. The mesh connectivity defines the domain of coordinate functions and has several efficient, lossless coding [5, 15]. To compress the mesh coordinates, traditional Fourier and wavelet compression techniques for images cannot be directly applied, since 3D meshes generally do not have a fixed regular graph structure. Consequently, there is no universally feasible Fourier basis and the dictionary should be derived from specific object's graph topology. In [7], Karni and Gotsman employed the mesh Laplacian eigenbases to encode the mesh geometry, and the compression is achieved by discarding high-frequency coefficients. Later, Karni and Gotsman extended the spectral compression method by using fixed eigenbases derived from a six-regular mesh to approximate the eigenbases of the non-regular input meshes, avoiding the cost of Laplacian decomposition on the decoder side [8].

It is attractive to be able to define wavelet transform directly on 3D shapes without the need of parameterization. Various schemes of manifold wavelets have been proposed via different approaches [1]. Diffusion wavelets, introduced by Coifman and Maggioni [2], use diffusion as a scaling tool to achieve multiscale analysis. Wavelet and scaling functions are constructed by repeatedly applying a diffusion operator  $T$  on the graph or manifold space. After applying dyadic powers of  $T$  at each scale, a localized orthogonalization procedure is performed to yield nested approximation spaces, and then

wavelets are produced by locally orthogonalizing vectors spanning the difference of these approximation spaces. The derived diffusion wavelets are orthogonal, compact, and multiscale in nature, and have been employed in 3D mesh compression in [11]. In [4, 14], tree-based, data-adaptive wavelet transforms are developed for high-dimensional Euclidean data sets and weighted graphs, under the assumption that the data have a rich geometrical structure that can be captured by a hierarchical tree.

One particularly interesting type of graph wavelets is the spectral graph wavelet (SGW), proposed by Hammond et al. [6]. SGW functions are locally defined in the vicinity of each point by their spectral representations using the Laplacian basis, and the scaling is carried out in the frequency domain. Because of its attractive and powerful properties such as spatial localization, multiscale, and geometry awareness, SGW has already been adopted as a tool for shape analysis applications such as mesh segmentation [10] and shape retrieval [10]. To the best extent of our knowledge, our current work is the first attempt to employ the SGW in the task of 3D mesh compression.

### 3 Manifold harmonic basis and spectral graph wavelet

#### 3.1 Mesh Laplacian and manifold harmonic basis

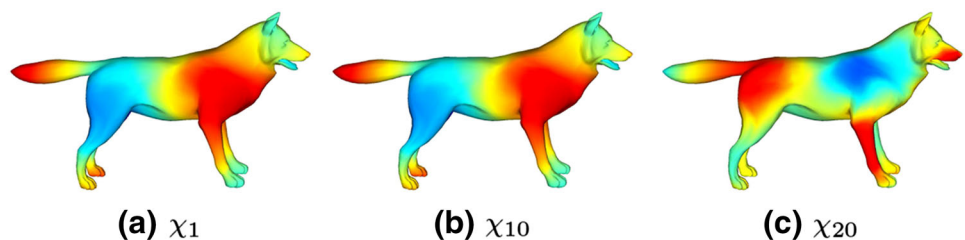
Consider a 3D mesh represented as a graph  $M = (V, E)$  with vertices  $V$  and edges  $E$ , where  $V = \{v_1, v_2, \dots, v_n\}$ . A vector-valued function  $\mathbf{f} : V \rightarrow \mathbb{R}^c$  defined on  $V$  can be represented as an  $n \times c$  matrix, where the  $i$ -th row represents the function value at  $v_i$ .

The discrete Laplace operator of a mesh, or mesh Laplacian, is an analog of the continuous Laplace–Beltrami operator defined on manifold surface, and is fundamental to a vast number of geometry processing applications [20]. The Laplace matrix of mesh  $M$  can be defined by the result of applying it to a function  $\mathbf{f}$  defined on  $V$ :

$$(L\mathbf{f})_i = \frac{1}{a_i} \sum_{j \in N(i)} w_{ij}(\mathbf{f}_i - \mathbf{f}_j), \tag{1}$$

where  $N(i)$  denotes the index set of the one-ring neighbors of  $v_i$ , and  $a_i$  are the masses associated with each vertex and  $w_{ij}$  represents the weight of each edge.

**Fig. 1** Visualization of the Laplacian eigenfunctions (MHB). From left to right, the first, tenth, and twentieth eigenfunctions are highlighted



Depending on the choice of  $a_i$  and  $w_{ij}$ , mesh Laplacian may have many different forms and can be classified as either *combinatorial* or *geometric* [21]. A combinatorial Laplacian is determined solely by the connectivity of the mesh. A geometric Laplacian, on the other hand, takes into account both the topological and geometric information.

Although a geometric Laplacian affords much more precise description of the mesh geometry, it is not a feasible choice in mesh compression applications because the geometric information is unknown on the decoder side. On the other hand, a combinatorial Laplacian can be easily reconstructed in the decoder size since the mesh connectivity can be efficiently encoded and transmitted independent of the geometric coordinates. In this work, we use the graph Laplacian defined as

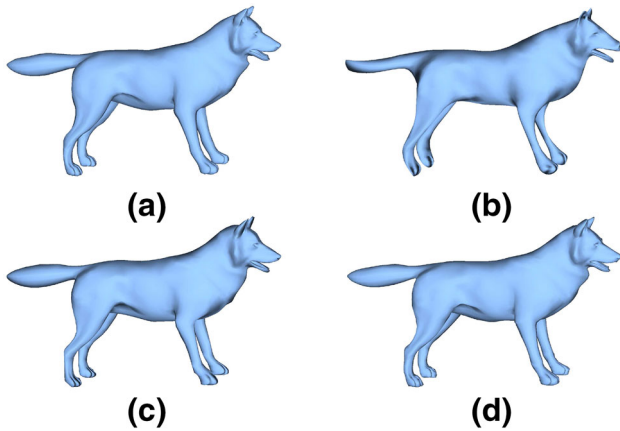
$$L_{ij} = \begin{cases} 1 & \text{if } j \in N(i), \\ -d(i) & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where  $d(i)$  represents the valence of  $v_i$ .

The graph Laplacian is symmetric and negative semi-definite, and by solving the eigenvalue problem  $L_s \chi_k = -\lambda_k \chi_k$  we obtain the eigensystem  $\{\lambda_k, \chi_k\}_{k=0}^{n-1}$ , where  $\{\lambda_k, \chi_k\}$  denotes the  $k$ -th eigenvalue and eigenfunction. According to the spectral theorem, the eigenfunctions  $\{\chi_k\}$  form a complete and orthonormal basis, called the *Laplacian eigenbasis*, or, in the context of shape analysis, the manifold harmonic basis (MHB) [18]. Figure 1 visualizes the MHB on an example mesh. It is straightforward to see that the values of MHB oscillate between negative and positive on the surface, and the larger the associated eigenvalues, the more frequent the oscillation becomes, similar to the behaviors of regular Fourier basis functions in the Euclidean domain.

#### 3.2 Spectral mesh approximation via MHB

The MHB can be employed to define the graph Fourier transform, also known as the *manifold harmonic transform* (MHT), which converts a function between spatial domain and frequency domain. Any  $f \in L^2(M)$  can be expanded by MHB as



**Fig. 2** Spectral approximation [7] of a 3D wolf model containing 4,344 vertices. **a** The original mesh. **b** Reconstruction using 100 eigenbases. **c** Reconstruction using 300 eigenbases. **d** Reconstruction using 1,000 eigenbases

$$f = \sum_{k=0}^{n-1} \hat{f}_k \phi_k = \sum_{k=0}^{n-1} \langle f, \phi_k \rangle \phi_k, \tag{3}$$

in which  $\hat{f}_k$  is the  $k$ -th MHT coefficient of  $f$ .

The MHT is the theoretical foundation of the spectral mesh compression proposed by Karni and Gotsman [7]. Viewing the Euclidean mesh coordinates  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{z}$  as functions defined on vertices, the basic idea of spectral compression is to compute the MHT of the coordinate function and then truncate out certain number of high-frequency coefficients. Take the  $x$ -coordinate function  $\mathbf{x}$  as an example. The original coordinates can be perfectly recovered as in Eq. (3) if all  $n$  MHT coefficients  $\{\hat{x}_0, \dots, \hat{x}_{n-1}\}$  are used. If we only retain the first  $n' < n$  coefficients, the reconstructed  $x$ -coordinate function  $\mathbf{x}' = \sum_{k=0}^{n'-1} \hat{x}_k \phi_k$  is a low-pass-filtered version of  $\mathbf{x}$  and can be regarded as an acceptable approximation. The reconstructed mesh is smooth and the overall appearance tends to be very similar to the original mesh, since low-frequency components, which correspond to large-scale shape information, are prioritized to be preserved, and our visual observations tend to be more forgiving to the loss of high-frequency information.

Figure 2 shows an example result of spectral approximation using different number of MHB.

### 3.3 Spectral graph wavelets

The spectral graph wavelets (SGW), as described in [6], are expressed as bivariate kernel functions expanded on the MHB

$$\Psi_t(i, j) = \sum_{k=0}^{n-1} g(t\lambda_k) \chi_k(i) \chi_k(j), \tag{4}$$

where  $g$  is the real-valued wavelet generator function and  $t$  is the scale parameter. The generator function  $g$  modulates

the spectral wavelets in the frequency domain, and should satisfy the admissibility condition

$$C_g = \int_0^\infty \frac{g^2(x)}{x} dx < \infty \tag{5}$$

and  $g(0) = 0$ . The  $i$ -th row of  $\Psi_t$

$$\psi_{t,i}(\cdot) = \sum_{k=0}^{n-1} g(t\lambda_k) \chi_k(i) \chi_k(\cdot) \tag{6}$$

is the spectral wavelet spatially localized at  $v_i$ , and in the frequency domain, localized at scale  $t$ .

In practice, the scale parameter  $t$  is discretized. The spectral graph wavelets  $\psi_{t,i}$  are near orthogonal to  $\chi_k$  for  $\lambda_k$  near 0, i.e., low-frequency eigenbasis, for any discrete scale  $t$ . Hence, to better capture low-frequency signal information, [6] also introduced the *spectral scaling functions* which have similar constructions with SGW but act like low-pass filters

$$\Phi_t(i, j) = \sum_{k=0}^{n-1} h(t\lambda_k) \chi_k(i) \chi_k(j), \tag{7}$$

in which the scaling generator function  $h$  should satisfy  $h(0) > 0$  and  $h(x) \rightarrow 0|_{x \rightarrow \infty}$ .

Suppose we compute the spectral wavelets at  $J$  different scales  $\{t_1, t_2, \dots, t_J\}$ , the constructed SGW then comprises  $(J + 1) \times n$  functions in  $\mathbb{R}^n$ . In this paper, we adopt the same formulation of wavelet and scaling functions used in [6] with the generator function

$$g(x) = \begin{cases} x^2 & \text{if } x < 1 \\ -5 + 11x - 6x^2 + x^3 & \text{if } 1 \leq x \leq 2 \\ 4x^{-2} & \text{if } x > 2. \end{cases} \tag{8}$$

The  $J$  scales are selected to be logarithmically equally spaced between the minimum scale  $t_J = 2/\lambda_{\max}$  and the maximum scale  $t_1 = 40/\lambda_{\max}$ , where  $\lambda_{\max}$  is the upper bound of the Laplacian eigenvalues. For the scaling function, the generator is  $h(x) = \gamma \exp(-(\frac{20x}{0.6\lambda_{\max}})^4)$ , in which  $\gamma = h(0)$  equals the maximum value of  $g$ .

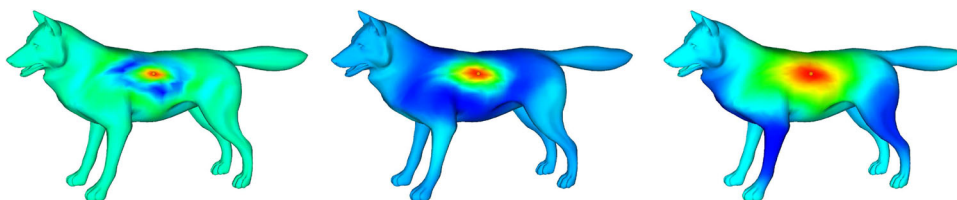
Figure 3 visualizes multiscale spectral graph wavelets on a 3D mesh. It may be noted that, the values of wavelets are attenuated and oscillating on the mesh, and wavelets with a larger scale have a wider oscillating window.

## 4 Mesh compression via sparse approximation

In Sect. 3.2, we have shown that the mesh coordinates can be transformed into the frequency domain via MHT using the Laplacian eigenbasis, and compression can be



**Fig. 3** Visualization of spectral graph wavelets localized at the same point but with different scales. From *left to right*, the spectral wavelets at scale 1, 3, and 5 are highlighted



achieved by trimming out a user-specified number of high-frequency coefficients. The main drawbacks of this naive and simple low-pass spectral compression method are: (1) it innately favors low-frequency information while most high-frequency geometric details are compromised; (2) the Laplacian eigenbasis, which serve as the compression dictionary, all have global support and therefore are not efficient in encoding local geometry.

In this paper, we propose to use SGW to construct a redundant dictionary. Because of its powerful property of spatial localization, the multiscale SGW functions are much more efficient in representing local mesh geometry around individual vertices than the Laplacian eigenbasis. Since the size of SGW dictionary is much larger than the number of mesh vertices, we employ powerful sparse approximation algorithms to find a compact representation which selects the most appropriate basis in the procedure.

### 4.1 Sparse approximation of mesh coordinates

For mesh  $M$ , let  $\mathbf{D}$  be a dictionary of  $L^2(M)$  containing  $m$  normalized basis vectors. The dictionary can be written as a  $n \times m$  matrix  $\mathbf{D} = (\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_m)$ , where  $\mathbf{a}_i \in \mathbb{R}^{n \times 1}$ . Our aim is to approximate function  $\mathbf{y} \in L_2(M)$  with a linear combination of the atoms in  $\mathbf{D}$ , expressed in the matrix form as  $\hat{\mathbf{y}} = \mathbf{D}\mathbf{x} = \sum_{i=1}^m x_i \mathbf{a}_i$ . Here, the vector  $\mathbf{x} \in \mathbb{R}^m$  is the coefficient representation of the input signal  $\mathbf{y}$  w.r.t. the dictionary  $\mathbf{D}$ .

An effective compression of the original signal  $\mathbf{y}$  requires the number of elementary signals that participate in the linear combination to be small and the reconstructed result  $\hat{\mathbf{y}}$  to be as close to  $\mathbf{y}$  as possible. In principle, the number of non-zero elements of the coefficient vector  $\mathbf{x}$ , denoted by the pseudo-norm  $\|\mathbf{x}\|_0$ , should satisfy  $\|\mathbf{x}\|_0 \ll n$  to achieve significant reduction in storage. Fixing the number of participating atoms in the sparse approximation to be  $n'$ , the problem to produce the optimal sparse representation  $\mathbf{x}$  can be formulated as:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \text{ subject to } \|\mathbf{x}\|_0 = n'. \tag{9}$$

If the input signal has  $c$  channels, denoted as a  $n \times c$  matrix  $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_c)$ , the coefficient representation should be a  $m \times c$  matrix  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_c)$  satisfying  $\mathbf{Y} \approx \mathbf{D}\mathbf{X}$ . We may either treat each channel (column) of  $\mathbf{Y}$  independently

and select different subsets of participating atoms for each channel, or use the same subset for all the channels and minimize the combined approximation errors. The latter one is called the *simultaneous sparse approximation* problem

$$\begin{aligned} &\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ &\text{subject to} \\ &\begin{cases} \text{support}(\mathbf{x}_1) = \dots = \text{support}(\mathbf{x}_c) \\ \|\mathbf{x}_1\|_0 = \dots = \|\mathbf{x}_c\|_0 = n', \end{cases} \end{aligned} \tag{10}$$

where  $\|\cdot\|_F$  is the Frobenius norm, and  $\text{support}(\mathbf{x}_i)$  denotes the index set of non-zero elements in  $\mathbf{x}_i$ .

The vertex coordinates of a mesh can be treated as a three-channel signal  $p(\mathbf{v}) = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$ . Since the three coordinate functions are correlated, it is preferable to formulate the mesh compression as the simultaneous sparse approximation problem. Determining the optimal solution to Eq. (10) is NP-hard, but we can find approximate solutions using greedy pursuit algorithms such as the *simultaneous orthogonal matching pursuit* (S-OMP) [17]. The key idea is to iteratively select from the dictionary a new atom that has the best correlation with the residual shape, and then project the original mesh onto the space spanned by the selected atoms to obtain a new approximate shape. Please refer to Algorithm 1 for details.

We may also adopt the *simultaneous matching pursuit* (S-MP) algorithm, which can be viewed as a simplification of S-OMP. The main difference from S-OMP is the omission of the step to update all existing coefficients by orthogonal projection. If all atoms in  $\mathbf{D}$  are mutually orthogonal (e.g., the Fourier dictionary), S-MP and S-OMP will produce exactly the same result.

### 4.2 Dictionary design strategies

The key to effective sparse approximation is the selection of elementary functions that form the dictionary. A natural choice is the MHB dictionary composed entirely of Laplacian eigenbasis  $\{\chi_0, \chi_2, \dots, \chi_{n-1}\}$ . The MHB functions have global support and multiple frequencies, making the MHB dictionary a good choice for encoding a shape when global, periodical, and symmetric information is prioritized to be preserved. In addition, since MHB is orthogonal basis, we can replace orthogonal matching pursuit (OMP) with the

**Algorithm 1** S-OMP on 3D mesh coordinates.**Input:**

- 3D mesh coordinates  $\mathbf{S} \in \mathbb{R}^{n \times 3}$ .
- Dictionary  $\mathbf{D} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ ,  $\mathbf{a}_i \in \mathbb{R}^3$ .
- The number of atoms to be selected  $n'$ .

**Initialization:**

- The initial index set of selected atoms  $\Lambda_0 = \emptyset$ .
- The initial residual  $\mathbf{R}_0 = \mathbf{S}$ ;
- The iteration counter  $t = 1$ .

**Procedure:**

(1) Find an index  $i_t$  of  $\mathbf{D}$  that satisfies

$$i_t = \arg \max_{j \notin \Lambda_{t-1}} \sum_{k=1}^3 |(\mathbf{R}_{t-1} \mathbf{e}_k, \mathbf{a}_j)|,$$

where  $\mathbf{e}_k$  denotes the  $k$ -th canonical basis vector in  $\mathbb{R}^3$ .

(2) Set  $\Lambda_t = \Lambda_{t-1} \cup \{i_t\}$ .

(3) Compute the coefficient matrix  $\mathbf{C}_t$  by solving the least-square problem

$$\mathbf{C}_t = \arg \min_{\mathbf{X}} \|\mathbf{S} - \mathbf{D}\mathbf{X}\|_2^2$$

subject to  $\text{support}(X) = \Lambda_t$ .

(4) Calculate the new approximation and residual:

$$\begin{aligned} \widehat{\mathbf{S}}_t &= \mathbf{D}\mathbf{C}_t, \\ \mathbf{R}_t &= \mathbf{S} - \widehat{\mathbf{S}}_t. \end{aligned}$$

(5) Stop if  $t = n'$ . Otherwise, increment  $t$  and go to (1).

**Output:**

- The index set of selected atoms  $\Lambda_{n'}$ .
- Final coefficient matrix  $\mathbf{C}_{n'}$ .

much faster matching pursuit (MP) and the approximation results will be the same.

However, MHB dictionary is very inefficient in capturing non-periodical, local details due to the global support. It is desirable to have a dictionary with a class of functions that have local support but are still smooth and multiscale. In this work, we propose to use normalized multiscale SGW, as described in Sect. 3.3, as atoms to construct the dictionary

for sparse approximation. The SGW dictionary has several advantages:

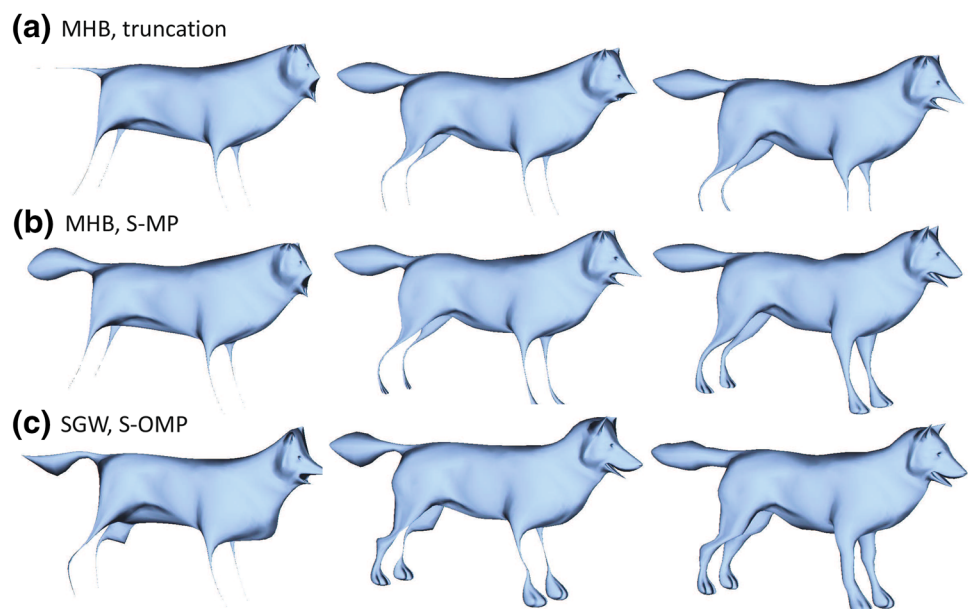
- The SGW atoms are compact and localized at vertices, suitable for encoding local geometric features.
- The SGW atoms can cover multiple scales, enabling the efficient representation of both small-scale and large-scale shape information in the vicinity of each vertex.
- The computation of SGW from MHB is straightforward, and can be done on the decoder side provided the mesh connectivity is known.

On the flip side, the SGW are less efficient than MHB for encoding global shape structures. Moreover, since SGW functions always have extreme values at their origin vertices, a mesh reconstructed from SGW atoms may exhibit unpleasant protrudes at vertices where selected SGW are centered, which can be further ameliorated by constructing a dictionary that contains both MHB and SGW. The mixed dictionary potentially inherits the advantages of both waveforms, at the cost of increased dictionary size.

The SGW or SGW+MHB dictionary are non-orthogonal and redundant, forcing us to use costly sparse approximation methods such as S-OMP. The enlarged dictionary also increase the storage requirements for the dictionary themselves and for the sparse coefficient representation (see Sect. 4.3).

Figure 4 shows an example of sparse approximation results using three methods: (1) spectral mesh compression via MHB, as described in Sect. 3.2; (2) S-MP with the MHB dictionary; (3) S-OMP with the SGW dictionary. In this example, the S-MP method produces higher-quality shape

**Fig. 4** Comparison of the approximation results of three different approximation methods. *Top row* spectral compression by truncating MHB coefficients. *Second row* S-MP approximation with MHB bases. *Bottom row* S-OMP approximation with SGW bases. For each method, from *left to right*, the number of participating bases are 20, 50, and 100, respectively



approximation than the naive low-pass spectral approximation using the same MHB dictionary. Adopting the multi-scale SGW dictionary in place of MHB further improves the approximation results. In particular, the SGW dictionary is more effective in preserving local geometric features, while MHB-based approximations have the tendency to smooth out some body parts such as the wolf's legs when the number of participating bases is small.

### 4.3 Compression ratio and analysis

Now, we analyze the compression ratio of the simultaneous sparse representation using a simple coding scheme. Assume the dictionary  $D$  is known in advance on both the encoder and decoder sides. The sparse  $m \times 3$  coefficient matrix  $\mathbf{X}$  contains  $n'$  non-zero rows, and can be conveniently expressed by  $3n'$  non-zero values and a vector of size  $n'$  specifying the indices of corresponding atoms. For a dictionary containing  $m$  atoms, each index occupies  $\lceil \log_2 m \rceil$  bits, and the total cost to store the index vector is  $n' \lceil \log_2 m \rceil$ . If the sparsity of  $\mathbf{X}$ , namely, the ratio of non-zero elements, is greater than  $1/\lceil \log_2 m \rceil$ , it would be more efficient to represent the non-zero positions with a bit vector of size  $m$ . Assuming each signal element in  $\mathbf{Y}$  takes up  $k$  bits, and each coefficient in  $\mathbf{X}$  requires  $k'$  bits to store, the storage size of the original 3D coordinates is then  $3nk$  bits. The effective compression ratio is

$$\Gamma = \frac{3n'k' + \min(m, n' \lceil \log_2 m \rceil)}{3nk} \tag{11}$$

Assume that the dictionary contains  $m = \alpha n$  atoms, and both the coordinates and coefficients are stored in single-precision  $k = k' = 32$ , the compression ratio is then

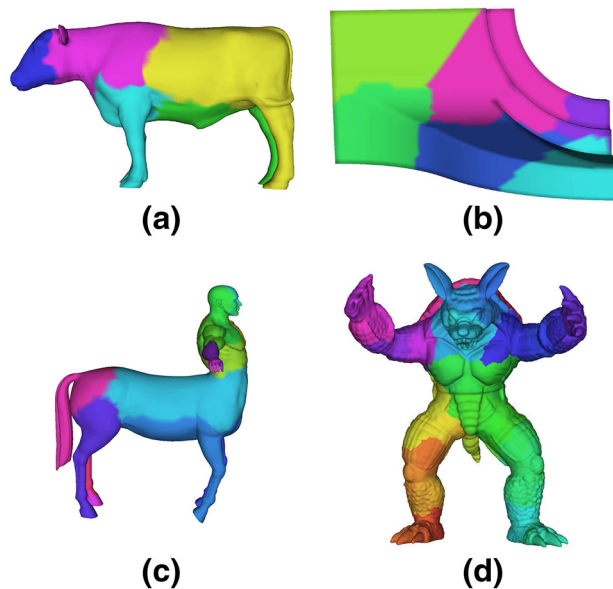
$$\Gamma = \frac{n'}{n} + \min\left(\frac{\alpha}{96}, \frac{n' \lceil \log_2 \alpha n \rceil}{96n}\right) \tag{12}$$

In comparison, the compression ratio of the coefficient truncation method introduced in Sect. 3.2 is simply  $n'/n$  with  $n'$  coefficients, since there is no need to store the indices of non-zeros.

From Eq. (12), we can easily see that enlarging the dictionary (larger  $\alpha$ ) increases the overhead ratio for a given mesh. In addition, when the coefficient matrix is very sparse, the overhead ratio becomes smaller for larger meshes, since  $\log_2 \alpha n$  increases slower than  $n$ .

### 4.4 Mesh partitioning

The most time-consuming part of S-OMP is to compute the maximum inner product between the residual and available atoms, which costs  $O(mn)$  in each iteration. In principle, the required number of iterations  $n'$  and the size of dictionary



**Fig. 5** Models used in our experiments and their partitioning. **a** Cow 4,315 vertices. **b** Fandisk 6,475 vertices. **c** Centaur 15,768 vertices. **d** Armadillo 172,974 vertices

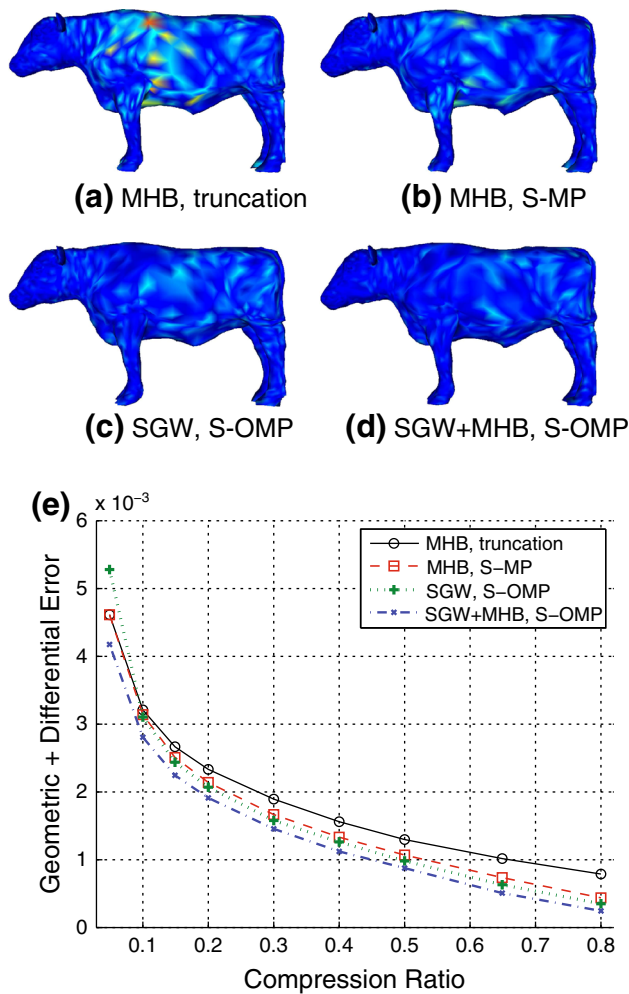
$m$  are linearly proportional to the mesh size  $n$ , hence the total time complexity is  $O(n^3)$ , which is unacceptable for very large meshes. In addition, all the dictionaries we use are constructed from the eigenvectors of mesh Laplacian, but the full Laplacian eigendecomposition of a large mesh is very time consuming and can be numerically instable. Hence, when the input mesh is very large, it is necessary to perform graph partitioning and carry on the compression algorithm on each individual sub-mesh. As suggested in [7], we use the METIS package [9] which implements several fast graph partitioning algorithms.

On the other hand, as mentioned in Sect. 4.3, the overhead for storing the indices of selected atoms is smaller when the mesh size becomes larger. Moreover, increasing the number of sub-meshes also increases the occurrences of unpleasant artifacts along sub-mesh boundaries. Thus, a tradeoff needs to be made between the compression time and quality. In our implementation, a large mesh is decomposed into a number of patches containing approximately equal number of vertices, with the maximum patch size set to be 1,000.

## 5 Experimental results

### 5.1 Evaluation method

To evaluate the effectiveness of lossy mesh compression methods, we adopt the mesh comparison metric proposed in [7] to measure the errors between the original mesh geometry and approximate ones. Let  $M_1$  and  $M_2$  be two meshes to be compared, both containing  $n$  vertices, and  $\mathbf{v}_1^1$  and  $\mathbf{v}_1^2$



**Fig. 6** Comparison of mesh compression performance for the cow model. **a–d** Show the reconstructed meshes at 20% compression ratio and visualize each vertex's positional error comparing with the original model. **e** Shows how the compression errors change with the compression ratio

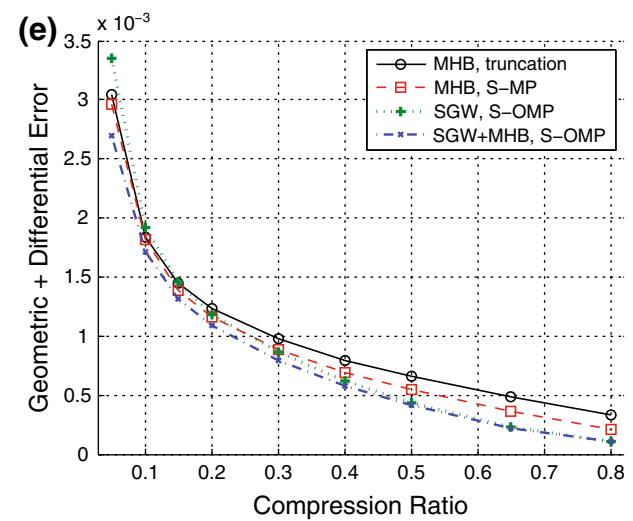
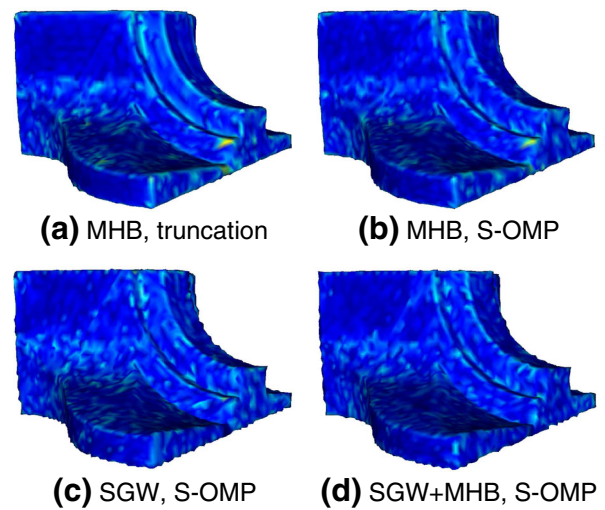
denote the 3D coordinates of the  $i$ -th vertex in  $M_1$  and  $M_2$ , respectively. The *geometric error* between  $M_1$  and  $M_2$  is

$$\|M_1 - M_2\|_g = \sum_{i=1}^n \frac{1}{n} \|\mathbf{v}_i^1 - \mathbf{v}_i^2\|_2. \quad (13)$$

To better capture visual closeness such as smoothness, [7] introduces another metric which measures the errors after applying the geometric Laplacian to mesh coordinates, i.e., transforming the absolute coordinates to differential coordinates

$$GL(\mathbf{v}_i) = \mathbf{v}_i - \frac{\sum_{j \in N(i)} l_{ij}^{-1} \mathbf{v}_j}{\sum_{j \in N(i)} l_{ij}^{-1}}, \quad (14)$$

where  $l_{ij}$  represents the edge length between  $v_i$  and  $v_j$ . The *differential error* between  $M_1$  and  $M_2$  is then defined as



**Fig. 7** Comparison of mesh compression performance for the fandisk model. **a–d** Show the reconstructed meshes at 20% compression ratio and visualize each vertex's positional error comparing with the original model. **e** Shows how the compression errors change with the compression ratio

$$\|M_1 - M_2\|_d = \sum_{i=1}^n \frac{1}{n} \|GL(\mathbf{v}_i^1) - GL(\mathbf{v}_i^2)\|_2. \quad (15)$$

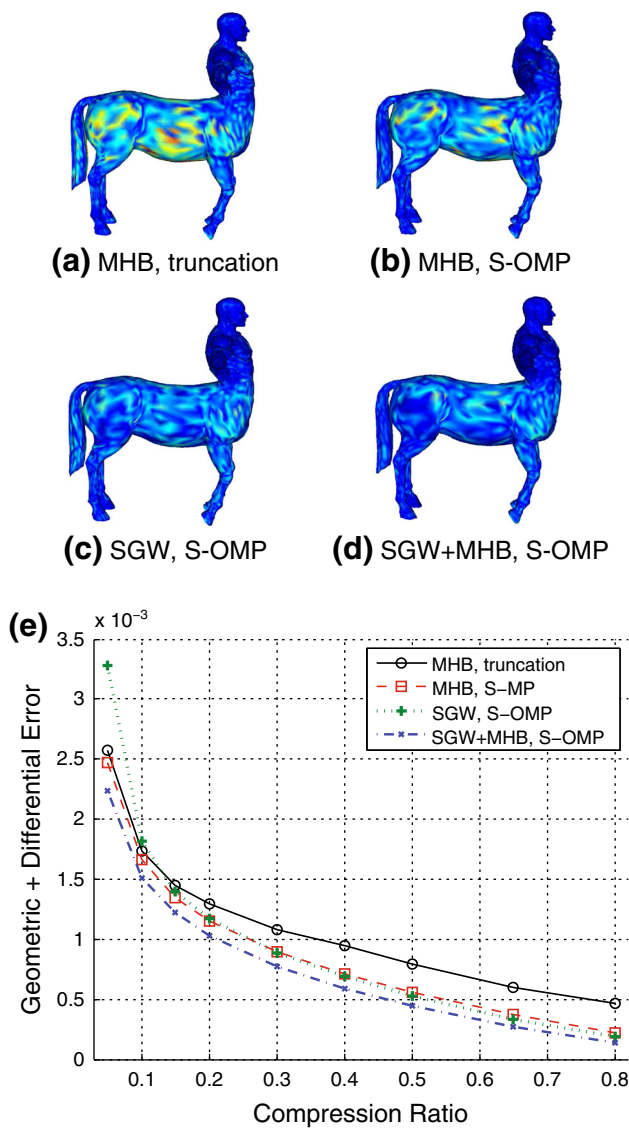
The final error metric is the average of geometric error and differential error

$$\|M_1 - M_2\| = \frac{1}{2} (\|M_1 - M_2\|_g + \|M_1 - M_2\|_d). \quad (16)$$

## 5.2 Compression performance

In all our tests, we compare the compression performance between the classical spectral compression method based on MHB coefficient truncation [7] and our sparse approximation compression method employing three different dictionaries: (1) S-MP with the MHB-only dictionary, (2) S-OMP with the

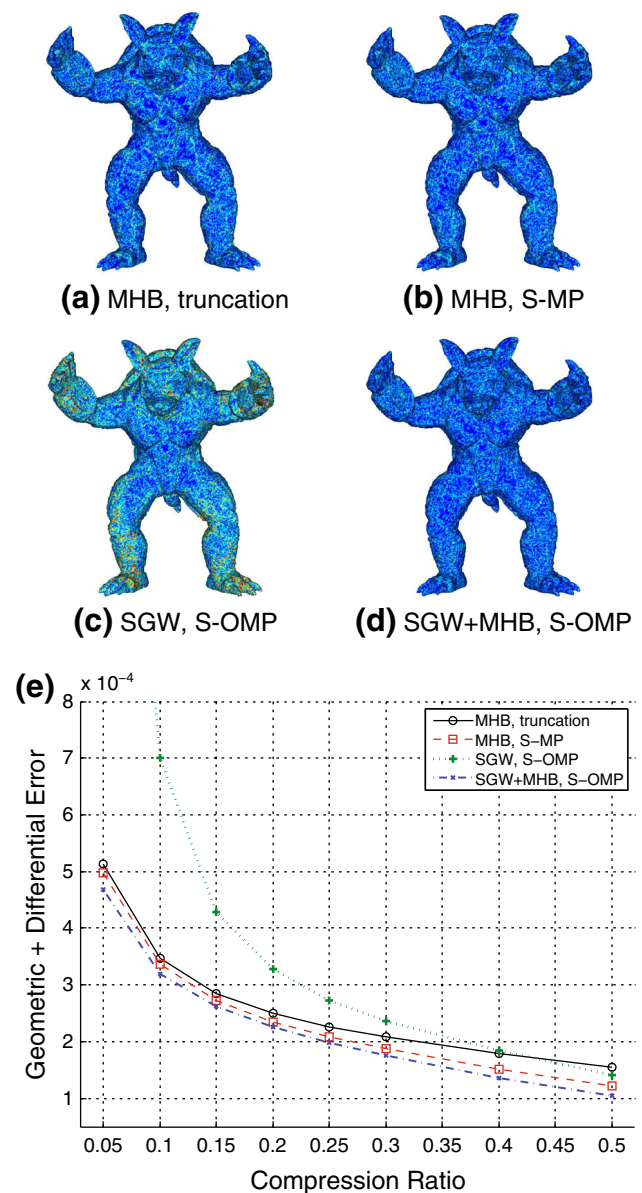




**Fig. 8** Comparison of mesh compression performance for the centaur model. **a–d** Show the reconstructed meshes at 20% compression ratio and visualize each vertex’s positional error comparing with the original model. **e** Shows how the compression errors change with the compression ratio

SGW-only dictionary and (3) S-OMP with the SGW+MHB dictionary.

Figure 5 shows the original meshes and their partitioning used in our experiments, including a “cow”, a “fandisk”, a “centaur”, and a “armadillo” model. All meshes are scaled to have unit surface area. The evaluation results are shown in Figs. 6, 7, 8, and 9, respectively. For each 3D mesh, we compute the approximation at specified compression ratios in the ranges between 5 and 80%. The overall compression quality is measured by the combined geometric and differential error (see Eq. (16)) w.r.t. the original mesh. Table 1 documents the compression errors and timing of S-OMP with SGW+MHB dictionaries, and compares the errors with the MHB coefficient truncation method.



**Fig. 9** Comparison of mesh compression performance for the armadillo model. **a–d** Show the reconstructed meshes at 15% compression ratio and visualize each vertex’s positional error. Comparing **c** and **d**, it is obvious that the SGW+MHB dictionary produces much smaller positional error than the SGW-only dictionary. **e** shows how the compression errors change with the compression ratio

From the experimental results, we see that, with a properly chosen dictionary, simultaneous sparse approximation can generate higher-fidelity mesh compression than the MHB truncation method at the same compression ratio. The SGW functions are a viable choice for efficient mesh approximation, but the performance of a SGW-only dictionary degenerates significantly when the required compression ratio is small or the mesh is large, which is especially evident in the armadillo model (Fig. 9). A dictionary combining SGW and MHB overcomes this deficiency, and its performance

**Table 1** Statistics of compression errors and running times using the S-OMP algorithm with SGW+MHB dictionary (on a machine with quad-core 2.4 GHz processor and 16 GB RAM)

Model (#vertices)	Ratio (%)	Error	Error decrease (%)	Timing (s)
Cow (4,315)	20	1.91e-3	18.0	3.4
	40	1.12e-3	27.5	10.5
Fandisk (6,475)	20	1.09e-3	11.6	7.2
	40	5.74e-4	28.1	18.9
Centaur (15,768)	20	1.03e-3	20.3	19.6
	40	5.89e-3	37.7	50.3
Armadillo (172,974)	20	2.26e-3	9.4	178.0
	40	1.76e-3	15.7	354.5

The “Error decrease” column shows the decrease of compression error compared with the MHB truncation method

in mesh approximation is consistently superior to the MHB truncation method.

## 6 Conclusion and future work

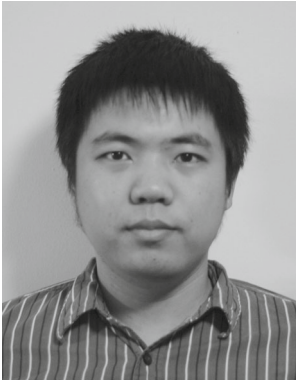
In this paper, we have developed an algorithm for sparse approximation of 3D shapes. We employed the SGW to construct the redundant dictionary of shape bases, and used simultaneous orthogonal matching pursuit to seek a sparse representation of the input mesh. The use of spatially localized wavelets makes our algorithm very suitable and powerful for better approximating shapes with many local and fine geometric features. Through comprehensive experiments, we have demonstrated the superiority of our algorithm for approximating complex 3D objects at different compression ratio settings towards sparse representation.

As for the immediate future work, we plan to investigate other improved formulations of graph wavelets as basis vectors to enhance the expressive power of dictionary. For example, it is more desirable to have data-specific, anisotropic wavelets that are adaptive to shape features such as sharp corners and edges, which can further facilitate more efficient and sparse representation of shape geometry. We also plan to explore faster sparse approximation algorithms such as stage-wise orthogonal matching pursuit (StOMP) [3] to arrive at better time performance.

**Acknowledgments** This research is supported in part by National Science Foundation of USA (Grant Nos. IIS-0949467, IIS-1047715, and IIS-1049448), and National Natural Science Foundation of China (Grant Nos. 61190120, 61190121, 61190125).

## References

1. Antoine, J.P., Roşca, D., Vandergheynst, P.: Wavelet transform on manifolds: old and new approaches. *Appl. Comput. Harmon. Anal.* **28**(2), 189–202 (2010)
2. Coifman, R.R., Maggioni, M.: Diffusion wavelets. *Appl. Comput. Harmon. Anal.* **21**(1), 53–94 (2006)
3. Donoho, D., Tsai, Y., Drori, I., Starck, J.L.: Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **58**(2), 1094–1121 (2012)
4. Gavish, M., Nadler, B., Coifman, R.R.: Multiscale wavelets on trees, graphs and high dimensional data: theory and applications to semi supervised learning. In: *Proceedings of ICML*, pp. 367–374 (2010)
5. Gumhold, S., Straßer, W.: Real time compression of triangle mesh connectivity. In: *Proceedings of SIGGRAPH*, pp. 133–140 (1998)
6. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.* **30**(2), 129–150 (2011)
7. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: *Proceedings of SIGGRAPH*, pp. 279–286 (2000)
8. Karni, Z., Gotsman, C.: 3D mesh compression using fixed spectral bases. In: *Proceedings of Graphics, Interface*, pp. 1–8 (2001)
9. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **20**(1), 359–392 (1998)
10. Li, C., Hamza, A.B.: A multiresolution descriptor for deformable 3D shape retrieval. *Visual Comput.* **29**(6–8), 513–524 (2013)
11. Mahadevan, S.: Adaptive mesh compression in 3D computer graphics using multiscale manifold learning. In: *Proceedings of ICML*, pp. 585–592 (2007)
12. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**(12), 3397–3415 (1993)
13. Pati, Y.C., Rezaifar, R., Krishnaprasad, P.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 (1993)
14. Ram, I., Elad, M., Cohen, I.: Redundant wavelets on graphs and high dimensional data clouds. *IEEE Signal Process. Lett.* **19**(5), 291–294 (2012)
15. Rossignac, J.: Edgebreaker: connectivity compression for triangle meshes. *IEEE Trans. Visual Comput. Graphics* **5**(1), 47–61 (1999)
16. Skodras, A., Christopoulos, C., Ebrahimi, T.: The JPEG 2000 still image compression standard. *IEEE Signal Process. Mag.* **18**(5), 36–58 (2001)
17. Tropp, J.A., Gilbert, A.C., Strauss, M.J.: Algorithms for simultaneous sparse approximation part I: Greedy pursuit. *Signal Process.* **86**(3), 572–588 (2006)
18. Vallet, B., Lévy, B.: Spectral geometry processing with manifold harmonics. *Comput. Graphics Forum* **27**(2), 251–260 (2008)
19. Wallace, G.K.: The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.* **38**(1), 18–34 (1992)
20. Wardetzky, M., Mathur, S., Kälberer, F., Grinspun, E.: Discrete Laplace operators: no free lunch. In: *Proceedings of 5th Eurographics Symposium on Geometry Processing*, pp. 33–37 (2007)
21. Zhang, H., Van Kaick, O., Dyer, R.: Spectral mesh processing. *Comput. Graphics Forum* **29**(6), 1865–1894 (2010)



**Ming Zhong** is a Ph.D. candidate in Department of Computer Science, Stony Brook University, USA. He received his B.S. degree from University of Science and Technology of China in 2010. His research interests include visual computing, shape analysis, shape correspondence, geometric modeling, and sparse representation.



**Hong Qin** is a professor of Computer Science in Department of Computer Science at Stony Brook University, USA. He received his B.S. degree and his M.S. degree in Computer Science from Peking University, China. He received his Ph.D. degree in Computer Science from the University of Toronto. During his years at the University of Toronto (UofT), he received UofT Open Doctoral Fellowship. He was also a recipient of NSF CAREER Award

from the National Science Foundation (NSF), Honda Initiation Award, and Alfred P. Sloan Research Fellow by the Sloan Foundation. Currently, he serves as an associate editor for *The Visual Computer*, *Graphical Models*, and *Journal of Computer Science and Technology*. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer aided geometric design, human-computer interaction, visualization, and scientific computing.