CrossMark

ORIGINAL ARTICLE

Full and partial shape similarity through sparse descriptor reconstruction

Lili Wan[1] · Changqing Zou[2,3] · Hao Zhang[3]

Abstract We introduce a novel approach to measuring similarity between two shapes based on *sparse reconstruction* of shape descriptors. The main feature of our approach is its applicability in situations where either of the two shapes may have moderate to significant portions of its data missing. Let the two shapes be A and B. Without loss of generality, we characterize A by learning a sparse dictionary from its local descriptors. The similarity between A and B is defined by the error incurred when reconstructing B's descriptor set using the basis signals from A's dictionary. Benefits of using sparse dictionary learning and reconstruction are twofold. First, sparse dictionary learning reduces data redundancy and facilitates similarity computations. More importantly, the reconstruction error is expected to be small as long as B is similar to A, regardless of whether the similarity is full or partial. Our proposed approach achieves significant improvements over previous works when retrieving non-rigid shapes with missing data, and it is also comparable to state-of-the-art methods on the retrieval of complete non-rigid shapes.

Keywords Full and partial shape similarity · Incomplete shapes · Shape retrieval · Sparse dictionary learning · Sparse reconstruction

✉ Lili Wan
   llwan@bjtu.edu.cn; lili.wan@gmail.com

✉ Changqing Zou
   aaronzou1125@gmail.com

   Hao Zhang
   haoz@sfu.ca

[1] Institute of Information Science, Beijing Jiaotong University, Beijing, China

[2] Hengyang Normal University, Hengyang, China

[3] Simon Fraser University, Burnaby, Canada

1 Introduction

One of the recurring questions in shape analysis is how to deal with missing data. Methods designed to handle complete shapes may still be applicable if the missing data are insignificant and the shape interpolation remains effective. However, with significant amounts of missing data, the shape analysis problem becomes quite challenging, as testified by previous works on incomplete shape retrieval [1] and segmentation [2].

In this paper, we are interested in the fundamental problem of measuring similarity between two 3D shapes. More specifically, we seek an approach that is effective at handling moderate to significant amounts of missing data in either or both non-rigid shapes.

Some shape similarity measures rely on one or more *global* shape descriptors [3,4]. However, by design, these global descriptors are unlikely to be suitable for highly incomplete shapes. *Local* shape descriptors, including the well known shape context [5] and heat kernel signature (HKS) [6], encode geometry at or from the perspective of a point over a shape's surface. These local descriptors are commonly pooled to form a global descriptor by some statistical strategies such as bag-of-words [7–9] and sparse coding [10,11]. As a result, the statistics are collected over the entire shape, decreasing the accuracy of shape similarity when the input shapes have missing data. In this work, we

propose a novel approach to organizing the entire set of local descriptors properly for two shapes, incomplete or not, to achieve a sensible comparison between them.

Sparse dictionary learning aims at finding a set of basis elements which compose a dictionary. With the dictionary, each input signal can be represented as a sparse linear combination of these basis elements. Sparse dictionary learning has achieved great success in signal processing and image processing [12]. In recent years, it has also attracted some researchers in the field of 3D shape analysis [10,11,13].

Our key observation is that local descriptors of a shape are largely redundant because the descriptors on nearby vertices are very close to each other. Sparse dictionary learning is especially appropriate to deal with this kind of information [14]. Each local descriptor is regarded as a signal, and a dictionary, including a set of basis signals, can therefore be learned. The dictionary is capable of reconstructing all the given signals by sparse linear combinations of these basis signals. Then, we make a connection between the shape similarity measure and sparse reconstruction of local descriptors. More specifically, for comparing two shapes A and B, we characterize A, without loss of generality, as a set of basis signals from A's dictionary, and sparsely reconstruct B's local descriptors. The similarity between A and B is, therefore, defined using these reconstruction errors. An overview of our approach is shown in Fig. 1.
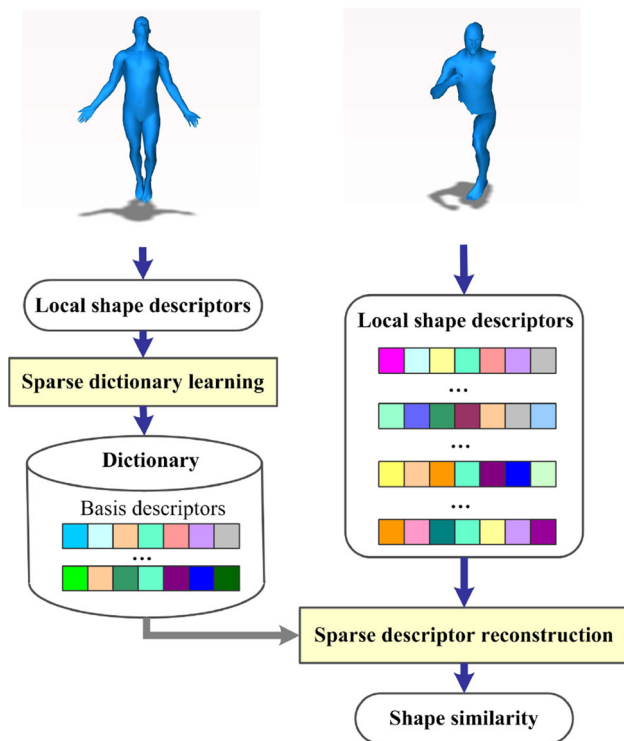

Fig. 1 Overview of our approach to computing the similarity between a complete shape (*left*) and an incomplete 3D shape (*right*) via sparse descriptor reconstruction

Our approach requires local shape descriptors that are insensitive to pose changes and missing shape parts. To achieve this, we modify the computation of HKS. And meanwhile, the dimension of the descriptor is chosen to fit for sparse dictionary learning.

The solution of shape similarity measure can be greatly beneficial to incomplete shape retrieval. First, dictionaries are computed for each shape in the dataset. Next, for a query, the shape similarities can be obtained using these dictionaries, respectively, to reconstruct its local shape descriptors. Such a retrieval application may be needed in practice when a modeler wants to create a new 3D shape via part composition and needs to search for one or more missing parts for a partially created shape, which is incomplete. In 3D model reconstruction amid significant missing data, a partial reconstructed shape, which is again incomplete, may be used to query a database for data-driven model completion. For these scenarios, the shapes in the database are expected to be more complete than the queries.

The problem of retrieving incomplete articulated (non-rigid) shapes has been addressed in Dey et al.'s work [1]. They rely on detecting and matching critical points to measure shape similarity. These critical points are HKS maxima, and their HKSs are named as persistent heat signatures (PHS). However, when large parts of a shape are missing, the detection of critical points will be easily impacted.

Differing from the aforementioned work, we propose a novel approach to measuring shape similarity based on sparse reconstruction of local descriptors for non-rigid incomplete shape retrieval. Experimental results show the effectiveness of our method. The contributions of our approach are twofold:

– Our method of computing local descriptors can maintain invariance under non-rigid deformations and also tolerate the missing parts of a shape to some extent.
– Our measure of shape similarity, which is defined from the perspective of sparse reconstruction of local shape descriptors, can be applied for two shapes which can be complete or not. The reason is that similar shapes with similar local descriptors can share the same dictionary, and the reconstruction error would be insensitive to the missing parts of a shape.

2 Related work

The literature on shape descriptors, shape matching, and shape retrieval is vast. In this section, we only cover methods that are most closely related to our work. We refer the readers to a number of surveys on these topics, including [15–17].

Spectral descriptors for non-rigid shapes In numerous non-rigid shape analysis tasks, the spectral descriptors achieve

state-of-the-art performance. Jain and Zhang [18] define an affinity matrix based on geodesic distances and take the spectrum of the matrix as a global descriptor. Many researchers study shape descriptors based on the spectrum of the Laplace–Beltrami operator on the surface. Due to the intrinsic nature of the Laplace–Beltrami operator, its spectrum is isometry-invariant. Reuter et al. [19] propose a Shape-DNA descriptor in which a shape is described using the Laplace–Beltrami spectrum (eigenvalues). Heat diffusion has recently been paid much attention according to its suitability for non-rigid shape analysis. The well-known heat kernel signature (HKS) [6] is a local shape descriptor based on heat diffusion. Wave kernel signature (WKS) [20] also carries a physical interpretation, and a quantum mechanics equation is used to replace the heat equation that gives rise to HKS. HKS and WKS are both invariant with respect to isometric transformations.

Non-rigid shape matching and partial shape matching are two hotspots of 3D shape analysis. Our aim is to solve the matching problem of shapes which are non-rigid and also have missing parts. It is more challenging than a single non-rigid shape matching problem, since missing shape parts may influence the Laplace–Beltrami operator obtained from the global shape.

Sparse coding for 3D shape retrieval Sparse coding is usually combined with dictionary learning. For shape retrieval, dictionary learning is used to replace the clustering process of the bag-of-words framework, and can be performed in an unsupervised [10] or supervised [11] scheme. Then, for a shape, its local descriptors are sparsely coded, and the resulting sparse coefficients are integrated to form a global shape descriptor. Besides local descriptors, the samples (signals) for training can be patch features. In Liu et al.'s work [13], each shape is over-segmented into a set of patches, and patch words are learned via sparse coding from all the patch features. Boscaini and Castellani [21] propose to exploit sparse coding for two retrieval applications: non-rigid shape retrieval and partial shape retrieval. Their partial shape retrieval is different from our incomplete shape retrieval. The best matches in their method are partly similar to the query, that is to say, some parts of the shapes might be dissimilar to the query. In contrast, we discuss a specific application aimed at solving the non-rigid and incomplete problem together. In our work, the best matches would be overall similar to the query, which may have pose changes and missing parts. Consequently, our work differs from [21] in the local descriptors and shape similarity measure.

Despite some retrieval applications via sparse coding, to our knowledge, sparse coding has not yet been used for incomplete non-rigid shape retrieval.

Partial matching Partial shape matching is appropriate for comparing shapes with significant variability and missing

data. Shape retrieval and correspondence are its typical applications. For shape retrieval, partial shape matching is applied to compute shape similarity [22,23]. To solve this problem, many methods detect and match feature points characterized by local shape descriptors. Gal and Cohen-Or [22] extract and store a set of salient regions for each model. Dey et al. [1] detect critical points based on the HKS descriptors. Itskovich and Tal [24] integrate feature point similarity and segment similarity for partial matching. Kaick et al. [25] propose a bilateral approach, where a local shape descriptor is defined by exploring the region of interest from the perspective of two points instead of one point. Quan and Tang [26] present a local shape descriptor called local shape polynomials (LSP), which is based on the evolution pattern of geodesic isocontour's length.

Another way is to encode the topological information as a graph for partial matching [27,28]. Biasotti et al. [27] present a structural shape descriptor, by which the structure and the geometry are coupled for recognizing similar parts among shapes. Tierny et al. [28] match partial 3D shapes via Reeb pattern unfolding.

Some methods involve shape segmentation to investigate meaningful parts of an object [29–31]. Toldo et al. [29] utilize bag-of-words to cluster the shape descriptors of segmented regions. Shapira et al. [30] define a similarity measure between two parts based on their geometry and context. Ferreira et al. [31] propose a part-in-whole matching method.

In our work, we focus on computing shape similarity between two non-rigid shapes, which may have missing shape parts, via sparse reconstruction of local descriptors.

3 Incomplete HKS (I-HKS)

Local shape descriptors, which are expected to maintain consistency between non-rigid shapes and their incomplete versions, are crucial to measure the shape similarity. Spectral descriptors are invariant under isometric transformations. However, these descriptors may vary when a shape misses some parts. Therefore, in this section, we will analyze two well-known spectral descriptors to deduce which is less sensitive to missing parts.

3.1 Preliminary

HKS and WKS are notable local descriptors using the spectral decomposition of the Laplace–Beltrami operator associated with a shape, and are widely used in numerous non-rigid shape analysis tasks.

Heat diffusion is an elegant mathematical tool with a good physical interpretation, which is the foundation of HKS. The heat kernel is used to describe the process of heat diffusion on a Riemannian manifold. Given a unit heat source at a point

x, the heat kernel $K_t(x, y)$ can be considered as the amount of heat that is transferred from x to y in time t, which can be written as [32]:

$$K_t(x, y) = \sum_{k \geq 0} e^{-\lambda_k t} \phi_k(x) \phi_k(y), \qquad (1)$$

where $0 = \lambda_0 \geq -\lambda_1 \geq -\lambda_2, \ldots$ are eigenvalues of the Laplace–Beltrami operator and $\phi_0, \phi_1, \phi_2, \ldots$ are the corresponding eigenfunctions.

Sun et al. [6] propose to take $K_t(x, x)$ as local shape descriptors, and call it HKS. For a point x, its HKS can be expressed as:

$$h(x, t) = K_t(x, x) = \sum_{k \geq 0} e^{-\lambda_k t} \phi_k^2(x). \qquad (2)$$

According to the analysis in [6], HKS has built-in advantages such as being isometry-invariant, multi-scale and robust against small perturbations.

Ovsjanikov et al. [7] present a compact representation of HKS. By sampling the HKS descriptor in time $t_i = \alpha^{i-1} t_0$, they obtain a descriptor vector $\mathbf{p}(x) = (p_1(x), \ldots, p_n(x))^T$, and the elements are

$$p_i(x) = c(x) h(x, \alpha^{i-1} t_0), \quad i = 1, \ldots, n, \qquad (3)$$

where the constant $c(x)$ is determined by $\|\mathbf{p}(x)\|_2 = 1$.

WKS is induced from quantum mechanics, which is another physical tool used to analyze non-rigid objects. From the uncertainty principle of quantum mechanics, a quantum mechanical particle's position and energy cannot be accurately determined at the same time. Thus, WKS represents the average probability of measuring a particle at a specific location by varying the energy of the particle. Let γ denote the logarithmic energy, for a vertex $x \in V$, its WKS can be computed by [20]

$$WKS(x, \gamma) = \frac{\sum_{k \geq 0} \phi_k^2(x) e^{\frac{-(\gamma - ln\lambda_k)^2}{2\sigma^2}}}{\sum_{k \geq 0} e^{\frac{-(\gamma - ln\lambda_k)^2}{2\sigma^2}}}, \qquad (4)$$

where σ is the variance of energy distributions.

3.2 HKS vs. WKS for incomplete shapes

In this section, we analyze HKS and WKS to decide which is more suitable for incomplete shape comparisons.

For a fair comparison, we make the parameter settings as consistent as possible for HKS and WKS. We take the first 100 eigenvalues and eigenfunctions to evaluate both of them. The importance of the diffusion time t to HKS is just as the energy γ to WKS. Thereby, we set t and γ to be adaptive to a shape. For each shape, we adopt its t_{min} and t_{max} to set t for the HKS, and take γ_{min} and γ_{max} to compute γ for the WKS. According to [6], we set $t_{min} = 4 \ln 10 / \lambda_1$ and $t_{max} = 4 \ln 10 / \lambda_{99}$. As in [33], we adopt $\gamma_{min} = \ln \lambda_1$ and $\gamma_{max} = \frac{\ln \lambda_{99}}{1.02}$. Then, t is uniformly sampled to get $n = 100$ values over $[t_{min}, t_{max}]$, while γ is also set to $n = 100$ values, ranging from γ_{min} to γ_{max} with linear increment $\delta = \frac{\gamma_{max} - \gamma_{min}}{99}$. As a result, t and γ are sampled using similar formulas which are $t_i = t_{min} + \frac{t_{max} - t_{min}}{99}(i - 1)$, $i = 1, \ldots, 100$ and $\gamma_i = \gamma_{min} + \frac{\gamma_{max} - \gamma_{min}}{99}(i - 1)$, $i = 1, \ldots, 100$. Additionally, the WKS has a parameter of variance σ which is set to 7δ [33].

According to the multi-scale property of the HKS, for small values of t, the $h(x, t)$ is mainly influenced by a small neighborhood of x. So we can deduce that for an incomplete shape, the HKS descriptors with small t are almost invariant, except for those points near the cutting boundaries. It can be verified by the visualization of the HKS descriptors for a human shape and its incomplete versions, as shown in Fig. 2a–e. Furthermore, the $h(x, t)$ decreases sharply as t
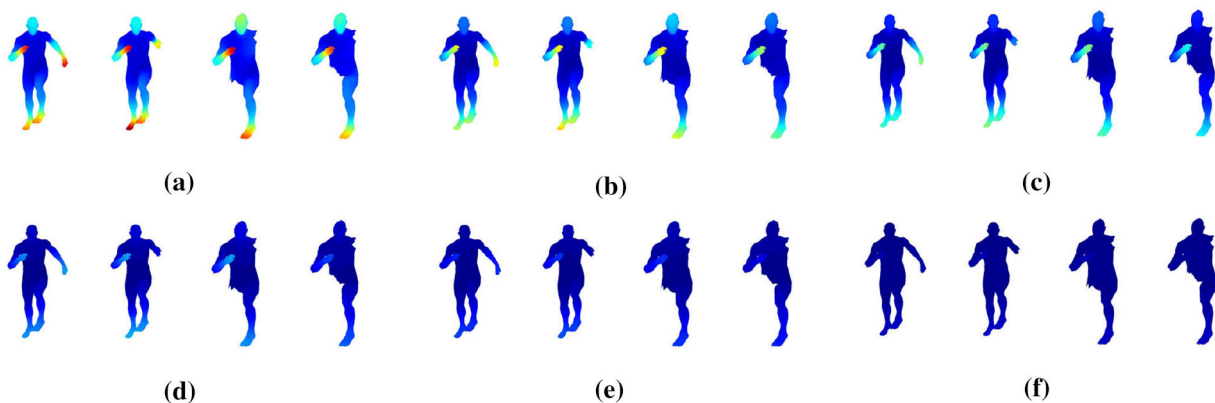

Fig. 2 Elements of the HKS descriptors mapped on the original shape (from [7]) and its incomplete versions, **a** $h(x, t_2)$, **b** $h(x, t_3)$, **c** $h(x, t_4)$, **d** $h(x, t_7)$, **e** $h(x, t_{10})$, **f** $h(x, t_{20})$. *Hotter colors* represent larger values
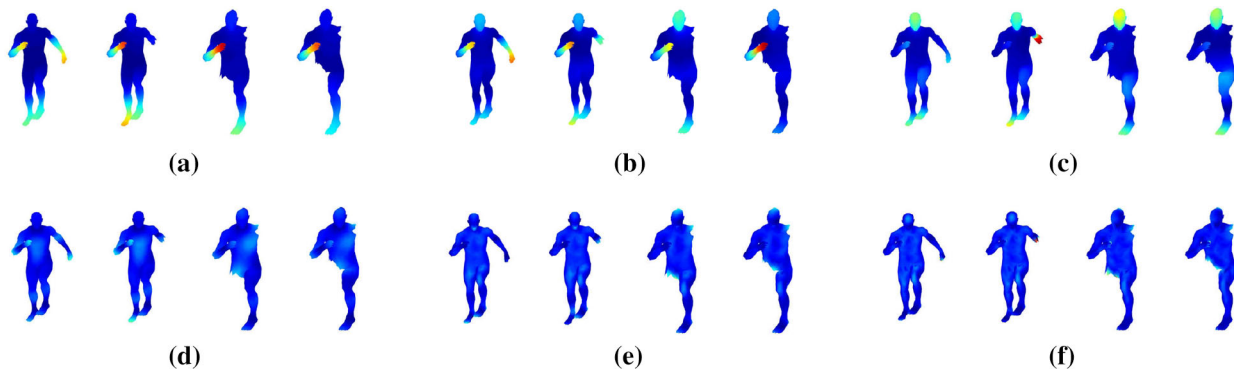
Fig. 3 Elements of the WKS descriptors mapped on the original shape and its incomplete versions, **a** $WKS(x, \gamma_{10})$, **b** $WKS(x, \gamma_{20})$, **c** $WKS(x, \gamma_{30})$, **d** $WKS(x, \gamma_{50})$, **e** $WKS(x, \gamma_{70})$, **f** $WKS(x, \gamma_{80})$. *Hotter colors* represent larger values

increases (see Fig. 2f). These two observations are the reason of our setting of the time interval in Sect. 3.3.

The WKS descriptors are visualized in Fig. 3. From it, we can have the following findings: (1) For small energies, the WKS descriptors change significantly in some regions far from the boundaries (see Fig. 3a–c); (2) For larger energies, most of the WKS descriptors are relatively small (see Fig. 3d–f). As known from [20], the WKS descriptors of small energies are induced by the global geometry. Based on this property, the missing parts influence the global geometry, and then result in the variation of the WKS descriptors. So it can explain our first finding. The WKS descriptors of large energies have good local attributes, but the small values are not good for discrimination.

Besides, we can discuss this problem from another perspective. Based on the analysis in [33,34], the HKS descriptor can be seen as a collection of low-pass filters, while the responses of the WKS descriptor are band-pass. However, for the WKS, the center frequencies of band-pass filters are defined by the eigenvalues which will be influenced by the missing parts. As a result, the elements of WKS, as a collection of band-pass filters, are also varied.

Based on the aforementioned analysis, we can draw a conclusion that the HKS descriptors are more suitable to be taken as local shape descriptors than the WKS descriptors for incomplete shapes.

3.3 HKS for incomplete shapes

We improve the computation of the HKS descriptors for incomplete shapes on the following aspects: (1) The descriptors are calculated on the largest connected component for a disconnected shape, while some descriptors of the boundary vertices and their 1-ring neighbors are excluded; (2) The dimension of each descriptor is chosen for sparse dictionary learning according to the dictionary size and the sparsity threshold; (3) The diffusion time scales are adaptively set for each shape, rather than some fixed values. To distinguish

the modified descriptors from the original HKS descriptors, we call them I-HKS descriptors.

The dimension of each descriptor needs to be suitable for the subsequent procedure of dictionary learning. We utilize the K-SVD algorithm for dictionary learning. The sparsity threshold should be small enough relative to the dimension of a signal, because in these circumstances the convergence can be guaranteed [35]. Therefore, the dimension n of an I-HKS descriptor cannot be too small. Meanwhile, n should be smaller than the dictionary size for designing an overcomplete dictionary. Consequently, in all the experiments of this paper, n is set to a reasonable value 10.

We use the first 100 eigenvalues and eigenfunctions to compute the I-HKS descriptors. Elements of an I-HKS descriptor with $t > t_{max}$ remain almost unchanged and those elements with $t < t_{min}$ need more eigenvalues and eigenfunctions [6]. For incomplete shape matching, small time is more appropriate for representing local attributes. Furthermore, from Fig. 2e, when $t_{10} = t_{min} + \frac{t_{max} - t_{min}}{99}(10 - 1)$, although the values of $h(x, t_{10})$ can still be used to distinguish the different points on a shape, they are indeed very small. So we choose the diffusion time from $t_{start} = t_{min}$ to $t_{end} = t_{min} + (t_{max} - t_{min})/10$. So for each 3D model, we sample n points over this time interval, and generate a logarithmically spaced vector. The time scales are then formulated as:

$$t_i = 10^{\lg t_{start} + \frac{\lg t_{end} - \lg t_{start}}{n-1}(i-1)}, \quad i = 1, \ldots, n. \quad (5)$$

Finally, all the I-HKS descriptors are normalized to unit L2 norm for the subsequent matching procedure.

4 Shape similarity

A 3D model may consist of as many as tens of thousands of vertices, and each vertex has a local shape descriptor. As a result, the set of local shape descriptors may be very

large. It is not efficient to directly compare between such huge descriptor sets. Therefore, many researchers use the bag-of-words framework to pool them into a global shape descriptor. An alternative scheme is to utilize critical points. A small set of local shape descriptors is computed at detected critical points, and the shape similarity is measured by these representative descriptors. However, the missing parts of an incomplete shape may impact both the global descriptor via bag-of-words and the detection of critical points.

Recently, the sparse dictionary learning theory has shown excellent performance in many applications. Given a set of signals, the information in this set is often largely redundant. Therefore, it is very important to determine a proper representation of the set. The aim of dictionary learning is to find a small set which is appropriate for representing all the signals in a given signal set. The signals in a learned dictionary are called basis signals. By means of the dictionary, each signal in the set can be efficiently expressed as a linear combination of basis signals, wherein the linear coefficients are sparse (most of them are zero).

From Fig. 2a, it is clear that: (1) The local descriptors of a vertex and its neighbors are very close; (2) Two symmetric parts, e.g., left and right hands, also have nearly equal local descriptors and, therefore, these descriptors are largely redundant. For a 3D model, taking its I-HKS descriptors as signals, we attempt to use the sparse dictionary learning theory to understand these signals, and formulate the shape similarity problem. Specifically, sparse dictionary learning is utilized to compute the basis descriptors for the descriptor set of each shape in the dataset. If the shape, either complete or not, is similar to the query and more complete, its dictionary can be applicable to reconstruct the query's local descriptors. We, therefore, use the reconstruction errors to measure the shape similarity between them.

4.1 Dictionary learning

In dictionary learning, researchers define multiple kinds of objective functions, and compute a dictionary by minimizing the objective function. In our application, the local descriptors of vertices vary smoothly along the surface, and thus a vertex's local descriptor can be approximately interpolated by the descriptors of its nearby vertices. Therefore, we choose the objective function with a sparsity threshold to constrain each time how many basis signals are used for interpolating a local descriptor.

For a shape S_A with N_A vertices in the database, its I-HKS set $\{f_j^A | j = 1, \ldots, N_A\}$ is computed, each of which is taken as a training signal. Let us denote its dictionary as D_A. Each signal f_j^A is expected to be approximately represented as a sparse linear combination of basis signals from D_A, which can be described as:

$$f_j^A \approx D_A \gamma_j^A \quad \text{s.t.} \quad \|\gamma_j^A\|_0 \leq T, \tag{6}$$

where γ_j^A consists of sparse coefficients and T is a sparsity threshold.

In the learning process, taking the training signal set $\{f_j^A\}$ and the dictionary size as inputs, the constrained optimization problem can be formulated as:

$$\tilde{D}_A = \min_{D_A} \frac{1}{N_A} \sum_{j=1}^{N_A} \|f_j^A - D_A \gamma_j^A\|_2^2 \quad \text{s.t.} \quad \|\gamma_j^A\|_0 \leq T. \tag{7}$$

The K-SVD algorithm [35] is widely used to solve the problem given by Eq. (7). It iteratively updates a dictionary and computes the sparse coefficients. The initial dictionary can be randomly selected from training signals. After the sparse coding with orthogonal matching pursuit (OMP), the dictionary update is performed by sequentially updating each column of the dictionary matrix using singular value decomposition (SVD) to minimize the approximation error.

4.2 Sparse reconstruction

Given a query shape S_B with N_B vertices, its I-HKS set $\{f_j^B | j = 1, \ldots, N_B\}$ is computed. Then, we use S_A's dictionary D_A to sparsely code each I-HKS descriptor f_j^B of the query S_B, and the reconstruction error is expressed as:

$$E\left(f_j^B, D_A\right) = \min \left\| f_j^B - D_A \gamma_j^B \right\|_2^2 \quad \text{s.t.} \quad \left\| \gamma_j^B \right\|_0 \leq T. \tag{8}$$

Next, we use the average reconstruction error to measure the distance between the shapes S_A and S_B, which is formulated as:

$$\text{Dist}\left(S_A, S_B\right) = \frac{1}{N_B} \sum_{j=1}^{N_B} E\left(f_j^B, D_A\right). \tag{9}$$

Each shape in the dataset has its own dictionary. Hence, after using each dictionary to, respectively, reconstruct the query's descriptors, we can compute and sort the shape similarities. In practice, we use SPAMS (SPArse modeling software) [36,37] which is an efficient optimization toolbox for solving various dictionary learning and sparse coding problems.

5 Results

In this section, we evaluate the retrieval performance of our method from two aspects: incomplete non-rigid shape

retrieval and complete non-rigid shape retrieval, and then test the running time. At last, we discuss the influence of the sparsity threshold on retrieval accuracy.

5.1 Experimental setup

To make the comparison informative with regard to the work of Dey et al. [1], we first test our method and two comparison methods on the dataset used in [1]. Then, we expand the scale of the experiment significantly by retrieving 150 incomplete shapes from the SHREC 2015 database. At last, we conduct the most challenging test where the complete versions of the query (incomplete) shapes do not exist in the database.

Dataset Two publicly available collections are used to construct the datasets for the experiments. The first collection is the PHS dataset from [1], which consists of two parts: 50 queries and a database of 300 shapes divided into 21 classes, and the second is the newest non-rigid shape retrieval benchmark: SHREC 2015 database [38], which is composed of 1200 models of 50 categories. In all, we have the following three datasets for experiments:

- *Dataset 1: PHS queries + PHS database* This is the dataset used in [1]. The queries are 32 incomplete and 18 complete shapes, and the database contains complete and incomplete shapes.
- *Dataset 2: Generated incomplete shapes + SHREC 2015 database* The database only contains complete shapes, so we manually generate 150 incomplete shapes (three per class) as the queries, which appear in three incomplete strength levels numbered 1–3. Some of them are shown in Fig. 4. The corresponding complete versions of these queries are in the database.

  Since these incomplete shapes are manually made, we know their corresponding complete versions, and thus can evaluate the incomplete strength quantitatively. The missing rate of an incomplete shape S_{incom} relative to its complete version S_{com} is defined as:

$$\mathrm{Mrate}(S_{com}, S_{incom}) = \frac{A_{com} - A_{incom}}{A_{com}}, \qquad (10)$$

  where A_{com} and A_{incom} are the surface areas of S_{com} and S_{incom}.

  The incomplete shapes in level 1 are made by deleting a part from complete shapes. Then, the shapes in level 2 and 3 are created based on the incomplete shapes one level below. The missing parts are variable in size, so the missing rates are different for these incomplete shapes. Therefore, we evaluate the missing rates of each incomplete level using the averages, which are, respectively, 10.56, 19.41 and 27.75 %.
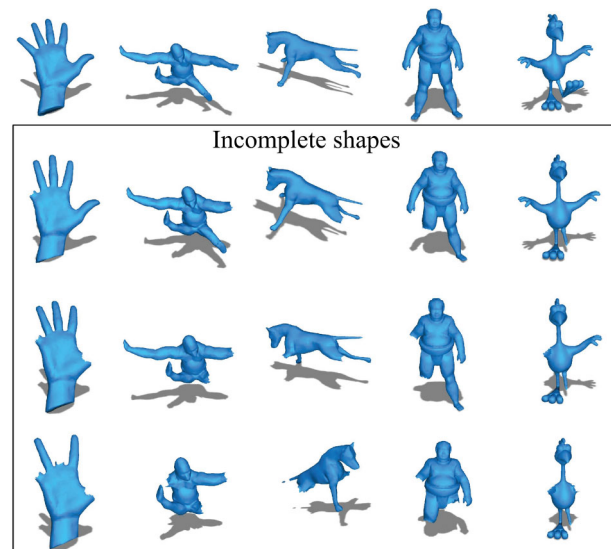

Fig. 4 Examples of generated incomplete shapes. *First row* shows complete shapes, and the *other rows*, respectively, show incomplete shapes in strength 1, 2 and 3

- *Dataset 3: PHS incomplete queries + SHREC 2015 database* This is the most challenging among the three datasets, because the corresponding complete versions of queries are not in the database. The database is the same as that of Dataset 2. The query set is a subset of that of Dataset 1. Since we are concerned with the shape matching involving incomplete shapes, 24 queries of incomplete shapes are chosen after excluding some queries whose classes are not in the database.

Parameters For each model, our I-HKS descriptors are computed using the first 100 eigenvalues and eigenfunctions of the Laplace–Beltrami operator. The dimension n of each I-HKS descriptor is 10, and the selection of time scales is introduced in Sect. 3.3. During dictionary learning, the dictionary size is fixed to 12, and the number of iterations is set to 1000. The sparsity threshold T for dictionary learning and sparse coding is set to the same value. If not specifically stated, the sparsity threshold T is set to 2 for the dictionary learning and sparse coding.

Assessment criteria We utilize the Top-k hit rate [1] to evaluate the performance of incomplete shape retrieval. If a query shape and one of its top k matches are from the same class, there is a Top-k hit. The Top-k hit rate is the percentage of the Top-k hits with respect to the number of query shapes. An ideal score is 100 %, and higher scores represent better results. In addition, we evaluate our method for complete shape retrieval based on the following five quantitative measures (see [39] for details): nearest neighbor (NN), first tier (FT), second tier (ST), e-measure (E), and discounted cumulative gain (DCG). For all of them, higher values are better.

5.2 Evaluations of incomplete shape retrieval

We compare our method with two competitive shape retrieval methods: persistent heat signature (PHS) [1], and heat Kernel signature (HKS) [7]. We choose these two methods because PHS represents a state-of-the-art technique for incomplete non-rigid shape retrieval, and HKS is a representative spectral method for complete non-rigid shape retrieval.

For the PHS method, all the parameter settings are the same as [1]. The time unit τ is set to 0.0002; the first 8 eigenvalues and eigenfunctions are used to compute the HKS function; 15 feature points are detected for each model using the HKS function at time 5τ; 15 different time scales are chosen to compute a 15D feature vector for each feature point, and the time scales are $t = \alpha * \tau$ with α varying over 5, 20, 40, 60, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000.

For the HKS method, we use the first 100 eigenvalues and eigenfunctions as [7] to compute the HKS descriptors. To be adaptive for the model scales in our datasets, the time scales are changed to $t = \alpha^{i-1} t_0$ with $t_0 = 0.006$, $\alpha = 2$ and $i = 1, \ldots, 6$. The resulting HKS descriptors are all 6D vectors. To obtain geometric words, for Dataset 1, all local descriptors collected from the database are used for K-means Clustering, while for Datasets 2 and 3, local descriptors from 50 models (1 per class) are selected. The number of words is 64 for Dataset 1 and 192 for Datasets 2 and 3 which have more classes.

Table 1 shows the Top-3 and Top-5 hit rates on Dataset 1. PHS performs better than HKS for the queries of incomplete shapes, and HKS achieves better performance than PHS for complete shapes as the queries. However, our method has better performance than PHS and HKS in these two circumstances. Since the database has both incomplete and complete shapes, we conclude that our method can also deal with the shape matching between a pair of complete or incomplete shapes. Figure 5 shows the top five matches for some queries.

Next, we assess our method under different incomplete strengths on Dataset 2. Table 2 shows the Top-3 and Top-5 hit rates. Each row shows the hit rates using the queries of the specified incomplete strength. Our method performs better than PHS and HKS for these three cases. From Table 2, we can conclude that our retrieval method can deal with the incomplete queries with the missing rates up to 30 %.

Table 1 Top-3/Top-5 hit rates on Dataset 1

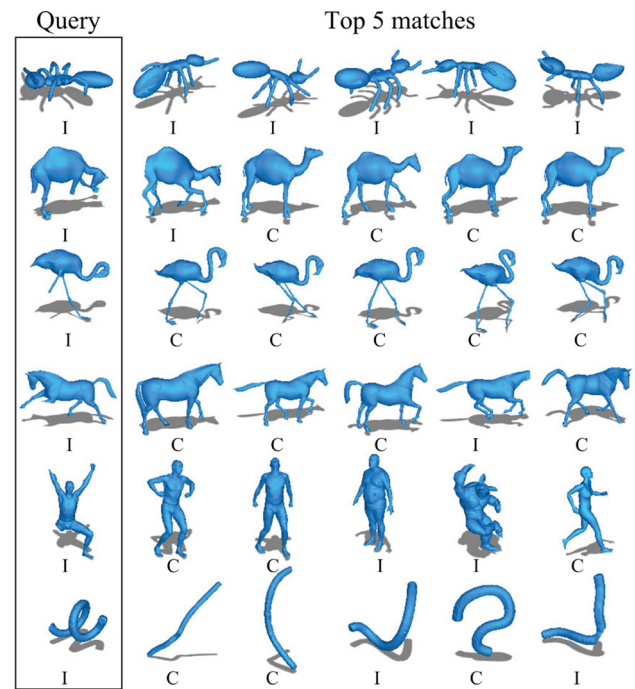| #Queries	Ours	PHS	HKS
32 Incompl.	91 %/94 %	88 %/91 %	56 %/63 %
18 Compl.	100 %/100 %	78 %/83 %	83 %/89 %
50 Total	94 %/96 %	84 %/88 %	66 %/72 %

Fig. 5 Query models and top five matches returned for each query on Dataset 1. *Letters* C and I indicate complete and incomplete models, respectively

Table 2 Top-3/Top-5 hit rates on Dataset 2

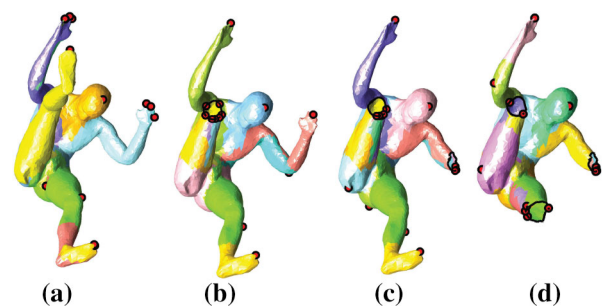| Strength	Ours	PHS	HKS
1	94 %/94 %	78 %/78 %	48 %/54 %
≤ 2	87 %/88 %	65 %/67 %	36 %/41 %
≤ 3	74 %/76 %	51 %/56 %	30 %/34 %

Fig. 6 Critical points detected by the PHS method. In **a**, a complete shape is shown together with its critical points (*red dots*), and its incomplete versions with an increasing incomplete strength are, respectively, illustrated in (**b**), (**c**) and (d). The *black curves* are the boundaries of shapes

To analyze why the PHS method fails for some incomplete queries, we design an experiment to investigate the detection of critical points, using a complete shape and its incomplete versions from Dataset 2. From Fig. 6, we can find when one or more parts are missing, some critical points move to the cut

boundary. Impacted by the variations of critical points, the retrieval result of the PHS method is, therefore, poor (see Fig. 7a), while our method still has good retrieval performance (see Fig. 7b).

We finally evaluate our method on a more challenging dataset where the queries and the database come from two different shape collections. Table 3 shows the Top-3 and Top-5 hit rates on Dataset 3. Although the performance of the three methods is worse than theirs on Dataset 1 and 2, our method achieves the best results and the hit rates are still acceptable.

5.3 Evaluations of complete shape retrieval

In this section, we test our method for complete shape retrieval using the SHREC 2015 non-rigid database. The settings of our method and PHS in this experiment are the same as in the experiments of incomplete shape retrieval. The
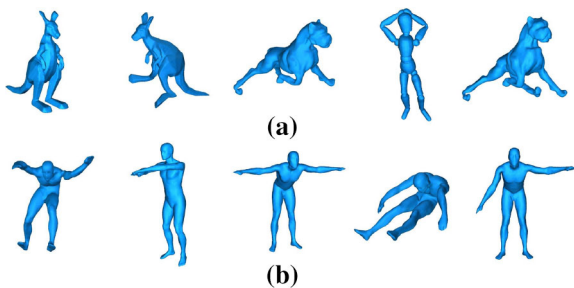

(a)

(b)

Fig. 7 Comparison of retrieval results on Dataset 2 between **a** PHS and **b** ours, using the shape shown in Fig. 6b as the query

Table 3 Top-3/Top-5 hit rates on Dataset 3

#Queries	Ours	PHS	HKS
24 Incompl.	71 %/71 %	33 %/46 %	17 %/25 %

results of our method and PHS are presented in Table 4. For comparison, we also show the best runs of each group taking part in the SHREC 2015 non-rigid track. Details of their methods can be found in [38]. Our method is better than PHS, and comparable to state-of-the-art methods on complete shape retrieval.

5.4 Running time

We test the running time of three algorithms using the SHREC 2015 non-rigid database. All the experiments in this section are carried out using MATLAB R2010b on a laptop with a 2.5 GHz dual-core 4-thread CPU and 8.00 GB RAM. For the PHS algorithm, we use the authors' implementation available on the web to compute the PHS descriptors, and implement their matching algorithm according to the description in [1]. For the HKS algorithm, we use the code provided by [40] to compute the HKS descriptors only omitting λ_0 and ϕ_0, due to the fact that λ_0 is theoretically 0 and ϕ_0 is a constant vector [7], and implement the subsequent steps: obtaining geometric words through K-means, pooling the HKS descriptors to a global descriptor, and measuring the shape similarity.

We present the pre-processing time of the three algorithms in Table 5. Each entry in the Ours column shows the time for computing the I-HKS descriptors and training a dictionary for the model. Each entry in the HKS column shows the time for computing the HKS descriptors and pooling them to a global descriptor. Our algorithm is the slowest, but it is comparable to PHS when the number of vertices is over 14,000. For our algorithm, the training time of each model is quite similar, because it is mainly determined by the number of iterations.

The retrieval time (not including the feature extraction of a query shape) is shown in Table 6. HKS has the fastest retrieval

Table 4 Quantitative evaluations of complete shape retrieval on the SHREC 2015 non-rigid database

Method	NN	FT	ST	E	DCG
SV-LSF_kpaca50	1.0000	0.9972	0.9997	0.8357	0.9997
HAPT_run1	0.9983	0.9657	0.9821	0.8150	0.9919
SPH_SparseCoding_1024	0.9975	0.9568	0.9696	0.8047	0.9885
CompactBoFHKS-10D	0.9842	0.8714	0.9082	0.7465	0.9582
EDBCF_NW	0.9775	0.7931	0.8839	0.7076	0.9431
FVF-WKS	0.9767	0.8225	0.8945	0.7242	0.9518
SID-4	0.9767	0.7188	0.8213	0.6482	0.9200
SG_L1	0.9725	0.7596	0.8143	0.6597	0.9192
Ours	0.9658	0.7572	0.8323	0.6692	0.9227
SNU_2	0.8992	0.5657	0.6706	0.5181	0.8335
TSASR1024	0.8958	0.5316	0.5960	0.4722	0.7975
PHS	0.8342	0.3803	0.4764	0.3572	0.7182
Multi-feature	0.4508	0.1864	0.2625	0.1846	0.5259

Table 5 Pre-processing time on the SHREC 2015 database (s)

Model	#v	Ours	PHS	HKS
T684	2959	1.4 + 15.7	2.1	0.7 + 1.2
T784	5971	1.9 + 15.8	5.1	1.6 + 2.3
T470	9999	2.9 + 15.8	9.8	2.3 + 3.9
T837	14,718	4.6 + 15.8	18.8	3.3 + 5.4

HKS additionally needs 244.7 s to obtain geometric words via K-means clustering

Table 6 Retrieval time on the SHREC 2015 database (s)

Query	#v	Ours	Ours(PC)	PHS	HKS
T684	2959	5.8	3.2	0.8	0.5
T784	5971	10.7	5.5	0.8	0.5
T470	9999	17.8	10.8	0.8	0.5
T837	14,718	25.8	13.4	0.8	0.5

speed. The retrieval time of PHS algorithm is nearly equal for any query, and so is the retrieval time of HKS algorithm. This happens because the computation of their shape similarity is performed on already extracted feature vectors or sets, and is irrelevant to the vertex number of a query shape. The retrieval time of our algorithm increases with the number of vertices as we use all the I-HKS descriptors for sparse coding. To accelerate our algorithm, we utilize Matlab parallel computing, and the results are shown in the Ours(PC) column. It is clear from the results that parallelism can reduce the retrieval time. However, our algorithm is still slower than PHS and HKS. Although more time is needed, the accuracy improves using our algorithm as shown in Sect. 5.2.

5.5 Discussions on parameter settings

In this section, we study two parameters in the stage of sparse dictionary learning. They are the dictionary size and sparsity threshold.

First, we use different dictionary sizes for incomplete shape retrieval on Dataset 2. In this experiment, we fix the sparsity threshold T to 2. The hit rate results are shown in Table 7. From it, we can see that the hit rates only have slight

Table 7 Top-3/Top-5 hit rates versus the dictionary size on Dataset 2

Strength	Dictionary size		
	12	24	48
1	94 %/94 %	92 %/94 %	92 %/92 %
≤ 2	87 %/88 %	85 %/88 %	84 %/86 %
≤ 3	74 %/76 %	73 %/76 %	71 %/73 %
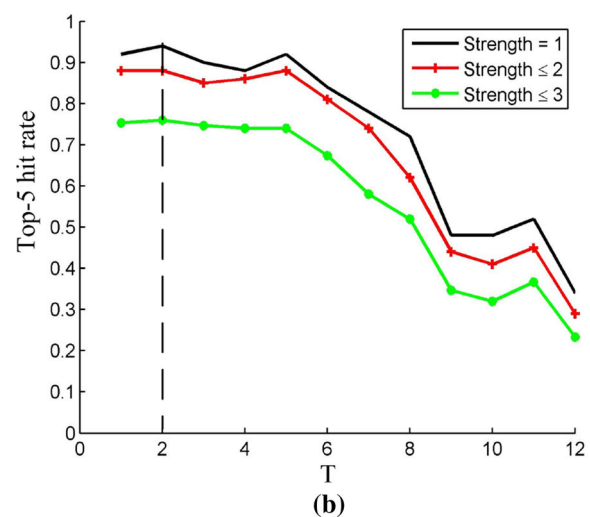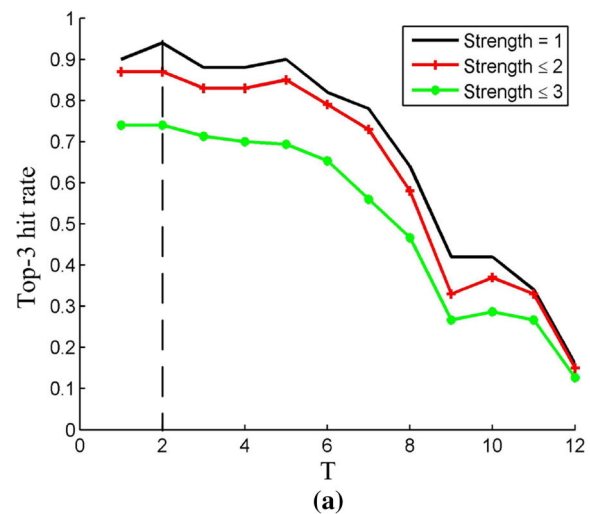
Fig. 8 Hit rates versus different T on Dataset 2, **a** Top-3 hit rates, **b** Top-5 hit rates

variations as the dictionary size increases. We thus prefer to use 12 as the final dictionary size.

Second, we examine the role of the sparsity threshold T on Dataset 2. The "Hit rates vs. T" curves of our method are presented in Fig. 8. From Fig. 8, we can deduce that T is a very important parameter to our retrieval method since T has a great influence on the retrieval accuracy. The top-3 and top-5 hit rates for the queries of different incomplete strengths all reach their maximum values when T is set to 2. It tells why we choose 2 as the final setting of T in the retrieval experiments. When T is much less than the dictionary size, only a small number of basis signals are used to reconstruct each local shape descriptor. All the queries of different incomplete strengths, thus, have high retrieval accuracies. However, when T is larger than six, more basis signals are involved. The retrieval accuracies decrease sharply. Surely, the retrieval accuracies also decrease when the incomplete strength increases.

5.6 Influence of different missing parts

To study the influence of missing different parts, we manually generate incomplete shapes for two models from the SHREC 2015 database. They are a deer and a chicken model indexed as "T578" and "T802". We, respectively, delete two horns and four legs of the deer shape, and two feet and two wings of the chicken shape. Figure 9 shows the original shapes and their corresponding incomplete shapes, with missing rates given beneath each shape.

We then use four incomplete shapes as queries to conduct the experiment. The retrieval results are shown in Fig. 10. The deer without horns in Fig. 10a has two wrong matches: a horse and a dog, while the deer without feet has only one wrong match: a centaur. In Fig. 10b, the chicken without feet has three wrong matches: two birds and a watch, while the chicken without wings has no wrong match. The incorrect
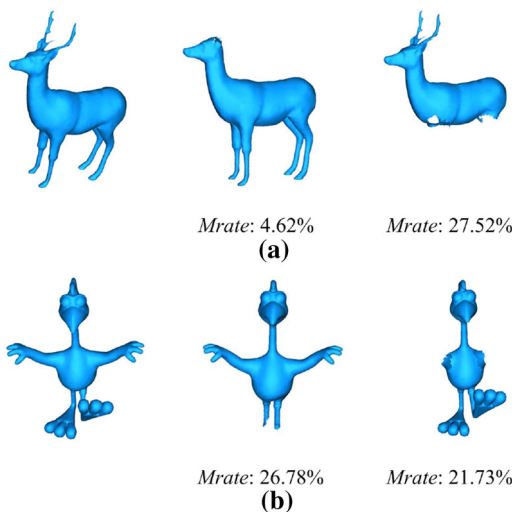

Mrate: 4.62%   *Mrate*: 27.52%
(a)

Mrate: 26.78%   *Mrate*: 21.73%
(b)

Fig. 9 Original and incomplete shapes, **a** Deer, **b** Chicken
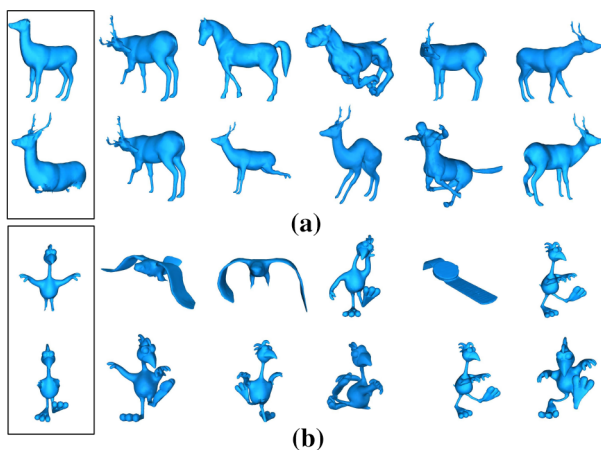

(a)

(b)

Fig. 10 Query models and top 5 matches on Dataset 2, **a** Deer, **b** Chicken

matches are reasonable. For example, a deer without horns sure looks like a horse or a dog, and a chicken without feet is quite similar to a bird. From the experiment of retrieving deer shapes, we can see that horns are more indispensable than legs. When the horns are deleted, even the missing rate is only 4.62%, the retrieval results are greatly influenced. From the experiment of retrieving chicken shapes, we can find that feet play a more important role than wings, although their missing rates are close. Consequently, we can deduce that different parts of a shape may have different importance in the similarity measure.

6 Conclusion, limitation, and future work

We propose a novel approach to measuring shape similarity based on sparse reconstruction of local descriptors. Differing from the previous work of detecting and matching critical points, we characterize each shape in the database by a learned dictionary, and define the shape similarity using the dictionary to reconstruct the query's local descriptors under a sparse constraint. We also modify the computation of HKS for dealing with non-rigid incomplete shapes. Experimental results show that the proposed method has achieved significant improvements on retrieving non-rigid shapes amid missing data, and is comparable to some complete shape retrieval approaches.

Our current retrieval approach has several limitations that leave room for improvement. One major limitation is that our modified local descriptors cannot completely solve the problem of missing parts. From the experiments, the retrieval accuracies of our method also decrease with the increasing incomplete strength, and our retrieval method can have improved performance for incomplete queries with missing rates up to 30%. Our method is not suitable for some datasets [41,42] composed of range scans. It is because our I-HKS descriptors are still computed using the Laplace–Beltrami operator of a whole shape which usually misses more than a half for a range scan. However, our approach is not restricted to a particular local shape descriptor, and the I-HKS descriptors are possible to be replaced by other descriptors in the future.

Second, the query shape is assumed to be connected. If the input shape is disconnected but with some large connected components, then the retrieval can simply be conducted on the largest component.

In the end, we assume that the boundary regions are easy to detect. While this assumption often holds when a complete model is being cut, in practice, particularly for partial surface reconstruction, boundary detection is not always an easy task.

3D object retrieval based on partial shape queries remains an open problem. For future work, we would like to deal with incomplete point clouds, incomplete topology-varying

man-made shapes, etc. With the progress of local shape descriptors, it may perhaps be practical to apply them to those more complex cases in retrieving incomplete shapes.

References

1. Dey, T., Li, K., Luo, C., Ranjan, P., Safa, I., Wang, Y.: Persistent heat signature for pose-oblivious matching of incomplete models. Comput. Gr. Forum **29**(5), 1545–1554 (2010)
2. Kaick, O.V., Fish, N., Kleiman, Y., Asafi, S., Cohen-OR, D.: Shape segmentation by approximate convexity analysis. ACM Trans. Gr. **34**(1), 4:1–4:11 (2014)
3. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Trans. Gr. **21**(4), 807–832 (2002)
4. Vranić, D.V.: DESIRE: A composite 3D-shape descriptor, In: Proceedings of IEEE international conference on multimedia and expo, pp. 962-965, IEEE (2005)
5. Körtgen, M., Park, G.J., Novotni, M., Klein, R.: 3D shape matching with 3D shape contexts. In: The 7th central European seminar on computer graphics vol. 3, pp. 5–17 (2003)
6. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. Comput. Gr. Forum **28**(5), 1383–1392 (2009)
7. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: geometric words and expressions for invariant shape retrieval. ACM Trans. Gr. **30**(1), 1:1–1:20 (2011)
8. Lavoué, G.: Combination of bag-of-words descriptors for robust partial shape retrieval. Vis. Comput. **28**(9), 931–942 (2012)
9. Zou, C., Wang, C., Wen, Y., Zhang, L., Liu, J.: Viewpoint-aware representation for sketch-based 3D model retrieval. IEEE Signal Process. Lett. **21**(8), 966–970 (2014)
10. Wan, L., Li, S., Miao, Z.J., Cen, Y.G.: Non-rigid 3D shape retrieval via sparse representation, Pacific Graphics Short Papers, pp. 11–16, Eurographics Association (2013)
11. Litman, R., Bronstein, A., Bronstein, M., Castellani, U.: Supervised learning of bag-of-features shape descriptors using sparse coding. Comput. Gr. Forum **33**(5), 127–136 (2014)
12. Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, Berlin (2010)
13. Liu, Z., Bu, S., Han, J.: Locality-constrained sparse patch coding for 3D shape retrieval. Neurocomputing **151**, 583–592 (2015)
14. Tosic, I., Frossard, P.: Dictionary learning. IEEE Signal Process. Mag. **28**(2), 27–38 (2011)
15. Bimbo, A.D.: Content-based retrieval of 3d models. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **2**(1), 20–43 (2006)
16. Tangelder, J.W., Veltkamp, R.C.: A survey of content based 3D shape retrieval methods. Multimed. Tools Appl. **39**(3), 441–471 (2008)
17. van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D.: A survey on shape correspondence. Comput. Gr. Forum **30**(6), 1681–1707 (2011)
18. Jain, V., Zhang, H.: A spectral approach to shape-based retrieval of articulated 3D models. Comput. Aided Des. **39**(5), 398–407 (2007)
19. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace–Beltrami spectra as 'Shape-DNA' of surfaces and solids. Comput. Aided Des. **38**(4), 342–366 (2006)
20. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In: IEEE international conference on computer vision workshops, pp. 1626–1633, IEEE (2011)
21. Boscaini, D., Castellani, U.: A sparse coding approach for local-to-global 3D shape description. Vis. Comput. **30**(11), 1233–1245 (2014)
22. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. ACM Trans. Gr. **25**(1), 130–150 (2006)
23. Funkhouser, T., Shilane, P.: Partial matching of 3D shapes with priority-driven search. In: Proceedings of the fourth eurographics symposium on geometry processing, pp. 131–142 (2006)
24. Itskovich, A., Tal, A.: Surface partial matching and application to archaeology. Comput. Gr. **35**(2), 334–341 (2011)
25. van Kaick, O., Zhang, H., Hamarneh, G.: Bilateral maps for partial matching. Comput. Gr. Forum **32**(6), 189–200 (2013)
26. Quan, L., Tang, K.: Polynomial local shape descriptor on interest points for 3D part-in-whole matching. Comput. Aided Des. **59**, 119–139 (2015)
27. Biasotti, S., Marini, S., Spagnuolo, M., Falcidieno, B.: Sub-part correspondence by structural descriptors of 3D shapes. Comput. Aided Des. **38**(9), 1002–1019 (2006)
28. Tierny, J., Vandeborre, J.P., Daoudi, M.: Partial 3D shape retrieval by Reeb pattern unfolding. Comput. Gr. Forum **28**(1), 41–55 (2009)
29. Toldo, R., Castellani, U., Fusiello, A.: The bag of words approach for retrieval and categorization of 3D objects. Vis. Comput. **26**(10), 1257–1268 (2010)
30. Shapira, L., Shalom, S., Shamir, A., Cohen-Or, D., Zhang, H.: Contextual part analogies in 3D objects. Int. J. Comput. Vis. **89**(2–3), 309–326 (2010)
31. Ferreira, A., Marini, S., Attene, M., Fonseca, M., Spagnuolo, M., Jorge, J., Falcidieno, B.: Thesaurus-based 3D object retrieval with part-in-whole matching. Int. J. Comput. Vis. **89**(2–3), 327–347 (2010)
32. Jones, P., Maggioni, M., Schul, R.: Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. Proc. Natl. Acad. Sci. **105**(6), 1803–1808 (2008)
33. Aubry, M., Schlickewei, U., Cremers, D.: Pose-consistent 3D shape segmentation based on a quantum mechanical feature descriptor. In: Proceedings of the 33rd international conference on pattern recognition, pp. 122–131, Springer, Berlin (2011)
34. Litman, R., Bronstein, A.: Learning spectral descriptors for deformable shape correspondence. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 171–180 (2014)
35. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
36. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th annual international conference on machine learning, pp. 689–696, ACM (2009)
37. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res. **11**, 19–60 (2010)
38. Lian, Z., Zhang, J., Choi, S., ElNaghy, H., El-Sana, J., Furuya, T., Giachetti, A., Guler, R.A., Lai, L., Li, C., Li, H., Limberger, F.A., Martin, R., Nakanishi, R.U., Neto, A.P., Nonato, L.G., Ohbuchi, R., Pevzner, K., Pickup, D., Rosin, P., Sharf, A., Sun, L., Sun, X., Tari, S., Unal, G., Wilson, R.C.: Shrec'15 track: Non-rigid 3d shape retrieval. In: Eurographics workshop on 3D object retrieval, pp. 107–120, Eurographics Association (2015)
39. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Proceedings of the shape modeling international, pp. 167–178, IEEE (2004)
40. Kokkinos, I., Bronstein, M.M., Yuille, A.: Dense scale-invariant descriptors for images and surfaces, Technical report. (2012)

41. Sipiran, I., Meruane, R., Bustos, B., et al.: A benchmark of simulated range images for partial shape retrieval. Vis. Comput. **30**(11), 1293–1308 (2014)
42. Godil, A., Dutagaci, H., Bustos, B. et al.: Range scans based 3D shape retrieval. In: Proceedings of the eurographics workshop on 3D object retrieval, pp. 153–160. Eurographics Association (2015)

Hao Zhang co-directs the GrUVi Lab at Simon Fraser University, Canada, where he is a full professor in the School of Computing Science. He received his Ph.D. from the Dynamic Graphics Project (DGP), Department of Computer Science, University of Toronto in 2003 and M.Math. and B.Math. degrees from the University of Waterloo. His research area is computer graphics with a focus on geometry modeling and processing, shape analysis, and 3D content creation.

Lili Wan is an associate professor in the Institute of Information Science at Beijing Jiaotong University, China. She received her doctoral degree in Computer Application from Beihang University in 2007 working on 3D shape retrieval. From September 2014 to September 2015, she was a visiting researcher at the GrUVi Lab at Simon Fraser University (SFU), Canada. Her current research interests are 2D and 3D shape analysis.

Changqing Zou received B.E. degree from Harbin Institute of Technology, China, in 2005, and M.E. degree from Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, China, in 2008, and PHD degree at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China, in 2015. He is now a post doctoral fellow at Simon Fraser University, Canada. He is also an assistant professor at Hengyang Normal University, China. His research interests include computer vision and computer graphics.