Takahiko Furuya and Ryutarou Ohbuchi, Diffusion-on-Manifold Aggregation of Local Features for Shape-based 3D Model Retrieval, *accepted as oral paper*, *Proc. ACM International Conference on Multimedia Information Retrieval* (*ICMR*) 2015, Shanghai, China.

Diffusion-on-Manifold Aggregation of Local Features for Shape-based 3D Model Retrieval

Takahiko Furuya
University of Yamanashi
4-3-11 Takeda, Kofu-shi
Yamanashi-ken, 400-8511, Japan
+81-55-220-8570
g13dm003AT yamanashi.ac.jp

Ryutarou Ohbuchi
University of Yamanashi
4-3-11 Takeda, Kofu-shi
Yamanashi-ken, 400-8511, Japan
+81-55-220-8570
ohbuchiAT yamanashi.ac.jp

ABSTRACT
Aggregating a set of local features has become one of the most common approaches for representing a multi-media data such as 2D image and 3D model. The success of Bag-of-Features (BF) aggregation [2] prompted several extensions to BF, that are, VLAD [12], Fisher Vector (FV) coding [22] and Super Vector (SV) coding [34]. They all learn small number of codewords, or representative local features, by clustering a set of large number of local features. The set of local features extracted from a media data (e.g., an image) is encoded by considering distribution of features around the codewords; BF uses frequency, VLAD and FV uses displacement vector, and SV uses a combination of both. In doing so, these encoding algorithms assume linearity of feature space about a codeword. Consequently, even if the set of features form a non-linear manifold, its non-linearity would be ignored, potentially degrading quality of aggregated features. In this paper, we propose a novel feature aggregation algorithm called *Diffusion-on-Manifold* (*DM*) that tries to take into account, via diffusion distance, structure of non-linear manifold formed by the set of local features. In view of 3D shape retrieval, we also propose a local 3D shape feature defined for oriented point set. Experiments using shape-based 3D model retrieval scenario show that the DM aggregation results in better retrieval accuracy than the existing aggregation algorithms we've compared against, that are, VLAD, FV, and SV, etc.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information filtering. I.2.10 [**Vision and Scene Understanding**]: 3D/stereo scene analysis.

General Terms
Algorithms, Experimentation.

Keywords
Feature aggregation, content-based retrieval, manifold learning, manifold ranking, diffusion distance, local feature, bag-of-features, sparse coding, fisher vector, super vector, 3D shape, 3D oriented point set.

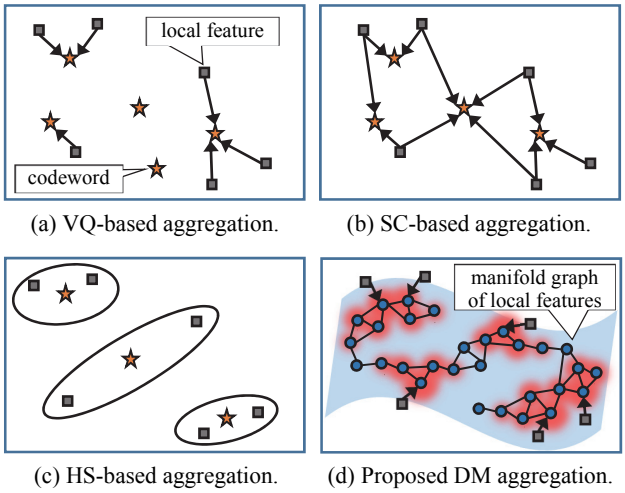
ICMR '15, June 23 - 26, 2015, Shanghai, China
Copyright 2015 ACM 978-1-4503-3274-3/15/06…$15.00
http://dx.doi.org/10.1145/2671188.2749380

1. INTRODUCTION
Majority of state-of-the-art shape-based 3D model retrieval (3DMR) algorithms represent a 3D model as a set of local features extracted from its 3D geometry or 2D appearances [4, 5, 6, 14, 15, 17, 19, 20]. Using a set of local features with an appropriate aggregation algorithm yields a 3D shape descriptor robust against articulation or global deformation of 3D models. Aggregation of local feature also make comparison among a pair of 3D models much more efficient than comparing local features individually.

Bag-of-Features (BF) [2] (Figure 1a) is first-of-a-kind local feature aggregation algorithm and is widely used to realize classification, segmentation, annotation, or *retrieval* of 2D images or 3D models. BF first learns a set of codewords by clustering the set of local features extracted from all the 3D models in a training set database. Each of a set of local features extracted from a 3D model is vector-quantized into its nearest codeword, and the codewords are accumulated into a histogram to become an aggregated feature vector for the 3D model. The vector quantization (VQ) process of the BF throws away a lot of information, e.g., relative positions or global arrangement of the set of local features. Consequently, the quality of aggregated feature suffers.

Several refinements to BF aggregation have been proposed [12, 16, 22, 23, 29, 30, 31, 34] to remedy this issue. They use either sparse coding (SC) or higher-order statistics (HS) to reduce information loss due to VQ. Both SC-based and HS-based algorithms use a number of codewords much smaller than the one for BF, and complement VQ with additional information. The Local Coordinate (LC) coding [31], Locality-constrained Linear (LL) coding [29], and localized soft-assigment coding [16] sparsely encode each local feature as weighted linear sum of neighboring codewords (Figure 1b). The Vector of Locally Aggregated Descriptor (VLAD) [12], Fisher Vector (FV) coding [22], and Super Vector (SV) coding [34] all try to encode local features by using higher-order statistics around the codewords (Figure 1c). The FV aggregates the set of local features by using displacement of mean and variance between the local features and the codewords, each of which is defined as a multivariate normal distribution derived by GMM clustering. The VLAD, which can be considered as an approximation of the FV, uses displacement of mean only.

These SC-based and HS-based algorithms aggregate a set of local features more accurately than a VQ-based approach, i.e., BF. However, most of them still can't handle non-linear distribution of local features in feature space. Small number of codewords essentially ignores local geometry of feature distribution. Extensions employed by the SC-based and HS-based algorithms listed above won't be able to completely capture non-linear nature of distribution of a set of local features in its feature space.



(a) VQ-based aggregation. (b) SC-based aggregation.

(c) HS-based aggregation. (d) Proposed DM aggregation.

Figure 1. The proposed Diffusion-on-Manifold (DM) algorithm (d) aggregates a set of local features by relevance diffusion on the manifold graph of local features to generate more accurate aggregated features than VQ-based (a), SC-based (b), and HS-based (c) approaches.

In this paper, we propose a novel local feature aggregation algorithm called *Diffusion-on-Manifold* (*DM*) that exploits structure of potentially non-linear manifold of local features. Figure 1 illustrates the proposed algorithm along with the existing algorithms for feature aggregation. Previous aggregation algorithms learn a set of cluster centers, that are, codewords, from a large number (e.g., 250k) of local features in a training set. During training stage, our algorithm generates a graph representing potentially non-linear manifold of all the training set features. Given a query, e.g., a 3D model, unseen local features extracted from it are vector quantized to their respective nearest neighbors in the manifold graph. Then, using these local features of the query as sources, relevance values are diffused over the manifold graph. After diffusion, relevance values from *all* the features on the manifold graph are aggregated into a feature vector for the query. The proposed DM aggregation algorithm may be regarded as an extension of LL coding [29] to non-linear manifold.

In view of 3DMR, in addition to the feature aggregation algorithm, the choice of low-level local shape feature to be aggregated is equally important. Local features for 3D models can be classified into two groups; local 3D geometric feature [8, 13, 17, 20, 24, 27] and local 2D visual feature [4, 5, 6, 19]. At this point, the latter has an advantage in retrieval accuracy. Many state-of-the-art 3DMR algorithms employ local 2D visual features extracted densely from images rendered from multiple viewpoints of 3D model. For example, top two finishers in the SHape REtrieval Contest (SHREC) 2014 Large-scale Comprehensive 3D shape retrieval (SH14LC) track used aggregation of local 2D visual features [15]. Despite their success, a 2D visual feature has its drawback; it is incapable of describing internal structures of 3D models. To capture internal structure of 3D models represented as surfaces, voxels, or point set, an accurate and efficient local 3D geometrical feature is desired.

We thus propose a local 3D geometric feature called *Position and Orientation Distribution* (*POD*) for 3D oriented point set. To accurately describe a geometry represented by distribution of a set of oriented points enclosed in a local region of a 3D model, POD encodes the oriented point sets by using SV coding.

An experimental evaluation of proposed DM aggregation using multiple local features and multiple benchmark databases shows that it significantly outperforms existing SC-based and HS-based aggregation methods. Furthermore, POD features aggregated by using DM yields retrieval accuracy comparable to state-of-the-art 3DMR algorithms using local 2D visual features.

Contributions of this paper can be summarized as follows.

- Proposition of Diffusion-on-Manifold (DM) aggregation of local features. DM exploits non-linear manifold structure of local features for more accurate aggregation.
- Proposition of a local 3D geometric feature for oriented point set. POD feature performs comparably in retrieval accuracy to state-of-the-art local 2D visual features for 3DMR.

The rest of this paper is structured as follows. We will describe related work in the next section. The proposed algorithms are described in Section 3. Empirical evaluation of the proposed algorithms will be presented in Section 4, followed by conclusion and future work in Section 5.

2. RELATED WORK

2.1 Aggregation of Local Features

The BF [2] is first-of-a-kind feature aggregation algorithm and was first introduced to 3DMR by Liu et al. [17]. The BF counts the number of vector-quantized local features to generate frequency histogram of codewords. Retrieval accuracy of BF-aggregated features are insufficient since positional information of local features in their feature space is lost due to VQ.

To more accurately aggregate the set of local features, several SC-based aggregation algorithms and HS-based aggregation algorithms have been proposed. The ScSPM [30], LC coding [31], LL coding [29], and localized soft-assignment coding [16] are grouped in SC-based approach. They sparsely encode each local feature as weighted linear sum of neighboring codewords. The VLAD [12], VLAT [23], FV [22], and SV [34] are grouped in HS-based approach. They all try to encode local features by using higher-order statistics around the codewords. They compute displacement vector between the mean of local features and the codeword, possibly with some additional information. In addition to displacement vector of mean, FV accumulates displacement vector with respect to variance of local features and SV accumulates frequency of codewords as with the BF. VLAT encodes local features around the codeword by using their covariance in addition to displacement vector of mean.

These SC-based approaches and HS-based approaches produce higher accuracy than BF in 2D image classification [1] or 3DMR [5]. However, accuracy of these methods may be still insufficient since they ignore structure of non-linear manifold of local features. The FV can essentially capture non-linearity of features due to its use of multivariate Gaussian distributions as codewords. But, in practice, it is difficult to approximate the structure of feature manifold by using small number (e.g., tens to hundreds) of multivariate Gaussians.

2.2 Manifold Learning

Manifold learning is used in many applications such as information retrieval, classification, clustering, or segmentation, to improve their accuracies. It learns better distance metric in the feature space by analyzing the structure of non-linear distribution of features, i.e., manifold. Typically, the manifold is represented as a graph where each node corresponds to a feature of a multi-media data (e.g., 3D model), and each edge indicates similarity between two features.

For information retrieval, several diffusion distance-based ranking algorithms have been proposed [3] to improve retrieval accuracy. One of the representative algorithms is the Manifold Ranking (MR) by Zhou et al. [33]. The MR diffuses relevance from source node(s), which correspond to given query(s), over the manifold graph. Retrieval ranking is generated base on relevance values diffused on the manifold.

While most of studies using manifold learning apply it on the feature space of multi-media data such as 3D model, only a few studies apply it on the local feature space. Tao et al. [26] computed the MR on the manifold of local 3D geometric features extracted from a 3D model for mesh saliency detection. Torki et al. [28] and Zhu et al. [35] reduced dimensionality of local features by eigen-analyzing the manifold graph of local 2D visual features for accurate feature matching. In this paper, we utilize the manifold of local features for feature aggregation.

2.3 3D Local Feature for Oriented Point Set

Recently, 3D oriented point set has been becoming one of the most used 3D shape representation due to proliferation of inexpensive 3D range scanners. To compare or recognize 3D models represented as oriented point set, many local 3D geometric features for oriented point set have been proposed [8, 13, 20, 24, 27].

Given an oriented point set of a 3D model, a set of *Sphere-Of-Interest* (*SOI*), in which a local feature is computed, is densely sampled. SPFH [24] describes the SOI by angular statistics of pairs of oriented points to form a 125-dim. feature vector. LSF [20] uses distances among the oriented points within the SOI in addition to angular statistics to generate 625-dim. feature vector. Spin Image (SI) [13] projects a set of oriented points in the SOI into a cylindrical coordinate and counts frequency of points for each region of the coordinate. SI feature is a 153-dim. vector. SPFH, LSF, and SI are invariant against 3D rotation of SOIs. RoPS [8] describes density of the point set within the SOI. It first normalizes 3D rotation of the SOI by using PCA. The rotation-normalized SOI is then rendered from multiple viewpoints to generate a set of 2D images. For each rendered image, RoPS computes first- and second-order moments to generate 135-dim. feature vector.

3. PROPOSED ALGORITHM

3.1 Overview of the Algorithm

For accurate 3DMR, we propose a novel feature aggregation algorithm called *Diffusion on Manifold* (*DM*) and a novel local 3D geometric feature *Position and Orientation Distribution* (*POD*).

The DM algorithm aggregates a set of local features extracted from a 3D model into a feature vector by using relevance diffusion on a manifold graph of local features (Figure 1d). As a pre-processing for DM aggregation, the manifold graph is generated from a set of large number of training local features. However, the manifold graph may suffer from *burstiness* of local features [11], which is a phenomenon where certain types of local features appear more frequently due to repetitive structures in 3D models or small-scale features that capture too primitive shapes such as flat surfaces. These "bursty" local features are thought to form dense clusters on the manifold. In such a case, relevance from bursty local features would diffuse only within the dense clusters. As a result, the aggregated features would become inaccurate since they are dominated by relevance values diffused within the bursty clusters.

To alleviate this burstiness problem, we perform weighting of nodes on the manifold graph. We use L_p-norm IDF weighting algorithm by Zheng et al. [32] to assign smaller weights to bursty nodes and to assign larger weights to non-bursty nodes. These node weights are used in relevance diffusion on the manifold graph for more accurate aggregation.

The POD feature describes distribution of oriented points within a SOI by using SV coding. The SOI is divided into multiple cells by using a regular grid. For each cell, positions of the oriented points within the cell are encoded by using SV, and normals of the oriented points are described by their covariance. A set of POD features is densely extracted from a 3D model, and is aggregated by the DM algorithm for accurate 3D model comparison.

3.2 Diffusion-on-Manifold Aggregation

3.2.1 Aggregating Local Features

Given a set of local features extracted from a 3D model, DM algorithm aggregates the set of local features into a feature vector per 3D model by relevance diffusion on the manifold graph of local features. The manifold graph is represented as a sparse matrix **P**, whose construction process will be described in Section 3.2.2.1. The matrix **P** has a size $N_t \times N_t$ where N_t is the number of nodes, i.e., training local features, on the manifold graph (e.g., N_t=250K).

The DM algorithm first generates an N_t-dim. source vector **y** for relevance diffusion. For each local feature **x** of a 3D model, **x** is vector-quantized into its nearest node (i.e., nearest training local feature) and the nearest node is set as a source for relevance diffusion. We use a *k*d-forest having 20 *k*d-trees to efficiently and accurately search nearest nodes. The source value y_n for the node n in the source vector **y** is computed by using the following equation;

$$y_n = w_n N_n \tag{1}$$

where N_n is the number of local features vector-quantized into the node n, and w_n is the weight for the node n. An algorithm for computing w_n will be described in Section 3.2.2.2.

We then compute relevance diffusion on the manifold graph to generate the aggregated feature vector for the 3D model. Relevance is diffused from the multiple sources defined by the source vector **y** over the manifold graph **P**. We use the following iterative form of relevance diffusion used in the Manifold Ranking algorithm [33].

$$\mathbf{f}(t+1) = \alpha \mathbf{f}(t)\mathbf{P} + (1-\alpha)\mathbf{y} \tag{2}$$

The N_t-dim. vector **f** is initialized by the source vector, i.e., **f**(0)=**y**. The number of iteration T determines range of relevance diffusion on the manifold. As we will show in the experiments, small T (e.g., T=5) is sufficient to obtain high retrieval accuracy. α=[0, 1) is a regularization parameter for relevance diffusion. Since the matrix **P** is sparse, computing Equation (2) is quite efficient.

After relevance diffusion, the N_t-dim. vector **f**(T) is normalized by using the similar method to the existing feature aggregation methods such as FV or VLAD. **f**(T) is power-normalized by taking square-root of each element of **f**(T). Then, the power-normalized **f**(T) is normalized by its L2-norm to generate the aggregated feature vector for the 3D model. Similarity between a pair of two DM-aggregated features is computed by using Cosine similarity.

3.2.2 Pre-processing for DM Aggregation

3.2.2.1 Constructing a Manifold Graph

In this section, we describe the method for constructing the manifold graph **P** used for DM aggregation. We first randomly sub-sample the set of N_t (e.g., N_t=250K) training local features from the set of local features extracted from all the 3D models in the database. A sparse affinity matrix **W** having size $N_t \times N_t$ is generated by connecting the training local features which are close to each other in the local feature space. The element $W(i, j)$ of **W**, which

indicates the similarity between the local feature i and the local feature j, is computed by using the following equation;

$$W(i,j)=\begin{cases}\exp\left(-d(i,j)/\sigma\right) & if\ \ i\neq j\ \text{and}\ j\in kNN(i)\\ 0 & otherwise\end{cases} \quad (3)$$

where $kNN(i)$ is a set of k nearest neighbor local features of a local feature i. Using smaller k (e.g., k=5) makes the affinity matrix $\mathbf{W}$ sparser, resulting in faster aggregation. $d(i,j)$ is L1 distance, whose range is normalized in [0,1], between a pair of two local features. σ is a scaling parameter for distance. We fix σ to 0.5 in this paper.

The affinity matrix $\mathbf{W}$ is then normalized to generate the probability transition matrix $\mathbf{P}$ as $\mathbf{P}=\mathbf{D}^{-1}\mathbf{W}$ where $\mathbf{D}$ is a diagonal matrix whose i-th diagonal element is $D(i,i)=\sum_{j=1}^{N_t}W(i,j)$. The element $P(i,j)$ in $\mathbf{P}$ indicates the transition probability of relevance from the local feature i to the local feature j. $\mathbf{P}$ represents the manifold graph of training local features and is used for DM aggregation.

3.2.2.2 *Weighting Nodes on a Manifold Graph*

To alleviate the burstiness problem of local features, we compute a weight for each of N_t nodes on the manifold graph $\mathbf{P}$. We use L_p-norm IDF weighting algorithm [32]. It assigns smaller weights to bursty nodes and assigns larger weights to non-bursty nodes.

We perform clustering on the set of N_t training local features. For efficiency, we use the Extremely Randomized Clustering Tree (ERC-Tree) algorithm [7]. The ERC-Tree algorithm recursively splits the local feature space until the number of training local features within the cluster becomes less than S_{min}. We use S_{min}=40 in this paper. The weight w_c for the cluster c is computed by using the following equation;

$$w_c=\log\left(1+N_M\Big/\sum\nolimits_{m\in M_c}f_{m,c}^{p}\right) \quad (4)$$

where N_M is the number of 3D models in the database, M_c is a set of 3D models containing local features that belong to the cluster c, and $f_{m,c}$ denotes the frequency of local features of the 3D model m within the cluster c. p is a parameter. The weight w_c is assigned to all the training local features within the cluster c.

Since the ERC-Tree is a randomized clustering algorithm, accuracy of node weights due to a single tree would be insufficient. To obtain reliable node weights, we perform ERC-Tree clustering for N_{tree} times and take an average of N_{tree} weights produced by N_{tree} trees to generate final weight w_n for the node n. We use N_{tree}=200 trees in this paper. The set of N_t weights for all the N_t nodes is used for generating the source vector $\mathbf{y}$ (Equation 1) for DM aggregation.

3.2.3 *Reducing Dimensions of Aggregated Features*

Since DM aggregation tends to produce high-dimensional (e.g., N_t=250K dim.) vector, comparing DM-aggregated features is costly. To accelerate feature comparison, we reduce the dimensionality of DM-aggregated feature vectors by using Kernel PCA (KPCA) with dot kernel as with [5]. To train the KPCA, we use a set of DM-aggregated features extracted from 5,000 randomly selected 3D models in the database. The number of reduced dimensions N_d is less than 100 for most cases, which is much smaller than the dimensionality of the DM-aggregated feature vectors. Comparison among dimension-reduced features is done by Cosine similarity.

3.3 POD Feature for 3D Oriented Point Set

3.3.1 *Generating Oriented Point Set*

The POD feature extraction algorithm expects 3D model defined as oriented point set. For a 3D model defined as polygonal mesh, we first convert it into oriented point set by sampling its surfaces. We use the algorithm by Osada et al. [21] for converting a polygonal model into an oriented point set. The algorithm randomly and uniformly samples points on the surfaces of the 3D polygonal model. Each point is associated with the normal vector of the triangle on which the point is sampled. In this paper, we sample N_p=3,000 oriented points per 3D model. Oriented point set of the 3D model is scaled to fit a sphere having diameter 1.

3.3.2 *Extracting Local Features*

Given an oriented point set of the 3D model, we densely extract a set of POD features from the oriented point set (Figure 2). For each oriented point, we define a *Sphere-Of-Interest* (*SOI*) whose radius is R. For robustness against scale change of the local 3D shapes, we use multi-scale SOIs. For each SOI, R is selected randomly from a range [r-v, r+v]. For example, if we use r=0.5 and v=0.2, R is randomly selected from a range [0.3, 0.7].

To achieve invariance against 3D rotation of local 3D shapes, we normalize orientation of the SOIs. For each SOI, we perform PCA on coordinates of oriented points within the ROI to obtain principle axes in the rotated coordinate system. We then disambiguate the directions, or signs, of the principle axes by using the method similar to SHOT [27] or RoPS [8]. That is, the sign s_1 for the first principal axis $\mathbf{e}_1$ is computed as $s_1=sign\left(\sum_{i=1}^{n_p}(\mathbf{p}_i-\mathbf{p})\cdot\mathbf{e}_1\right)$ where n_p is the number of oriented points within the SOI, $\mathbf{p}_i$ is the position of i-th oriented point, and $\mathbf{p}$ is the center of the SOI. The sign s_3 for the third principal axis $\mathbf{e}_3$ is computed in the same manner as s_1. The second principal axis $\mathbf{e}_2$ is computed as $\mathbf{e}_2=s_3\mathbf{e}_3\times s_1\mathbf{e}_1$ where $\times$ denotes cross product of two vectors.

After orientation normalization, we extract a POD feature from the rotated SOI. We first compute a bounding box of the SOI, which may be a cuboid. And we normalize the cuboid bounding box by transforming it into a cube. We then split the normalized bounding box into $N_1\times N_2\times N_3$ cuboid cells by using a regular grid. N_1 (N_2, N_3) is the number of cells along the first (second, third) principal axis. In this paper, we set $N_1>N_2>N_3$ to make the POD feature compact. That is, we put finer grids along the principal axis having larger variance of oriented point distribution. We use $N_1\times N_2\times N_3$=4×2×1.

For each cell c, we describe the distribution of oriented points. The position of oriented points within the cell is described by using SV coding [34]. We count the number of oriented points n_c within the cell c and pool displacements of the n_c oriented points from the center of the cell c. The normals of oriented points within the cell is described by their 3×3 covariance matrix $\mathbf{C}$. These statistics are concatenated to form a 10-dimensional feature vector for the cell c; $\mathbf{f}_c=\left[\beta\cdot\sqrt{n_c},\ \left(1/\sqrt{n_c}\right)\mathbf{d}_c,\ \mathbf{v}_c\right]$ where $\mathbf{d}_c$ is the 3-dim. displacement vector pooled within the cell c, and $\mathbf{v}_c$ is the 6-dim. vector whose elements are upper triangle elements of the covariance matrix $\mathbf{C}$.

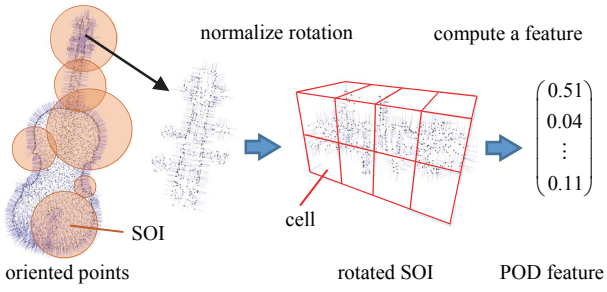


Figure 2. Extracting POD features from an oriented point set.

β is a balancing parameter between the frequency and the two vectors $\mathbf{d}_c$ and $\mathbf{v}_c$. We set β=0.001 in this paper.

The set of 10-dimensional vectors $\mathbf{f}_c$ computed for all the $N_1 \times N_2 \times N_3$ cells is concatenated to form a POD feature vector for the SOI. If we use $N_1 \times N_2 \times N_3$=4×2×1, the POD feature has 80 dim., which is more compact than other existing local 3D geometric features such as SI (153 dim.), LSF (625 dim.), FPFH (125 dim.), RoPS (135 dim.), and SHOT (320 dim.).

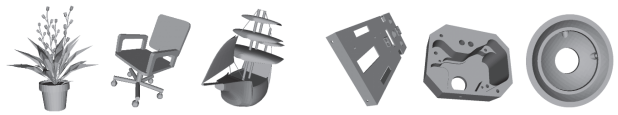
4. EXPERIMENTS AND RESULTS

4.1 Experimental Setup

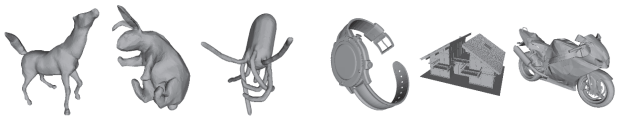
4.1.1 Benchmark Databases

To evaluate accuracy and efficiency of the proposed algorithms, we use four benchmark databases; the Princeton Shape Benchmark (PSB) [25], the Engineering Shape Benchmark (ESB) [10], the SHREC 2011 Non-Rigid watertight meshes dataset (SH11NR) [14], and the SHREC 2014 Large-scale Comprehensive 3D shape retrieval dataset (SH14LC) [15]. Figure 3 shows examples of 3D models as well as the number of 3D models and the number of categories contained in the datasets. The PSB and SH14LC contain generic, rigid 3D shapes. The ESB consists of mechanical CAD models. The SH11NR has articulated, non-rigid 3D shapes.

For all the benchmarks, a 3D model in the database is used as a query and remaining 3D models are used as retrieval targets. We use Mean Average Precision (MAP) [%] as an index of retrieval accuracy. For the SH11NR, to make local feature more robust against articulation, 3D models are deformed to their canonical forms by using a method by Jain et al. [9] prior to feature extraction.



(a) PSB (N_M =907, N_C =92) (b) ESB (N_M =867, N_C =45)



(c) SH11NR (N_M =600, N_C =30) (d) SH14LC (N_M=8,987, N_C=171)

Figure 3. Benchmark databases used in the experiments (N_M : number of 3D models, N_C : number of categories).

4.1.2 Feature Aggregation

We compare the DM aggregation algorithm against five existing feature aggregation methods; BF [2], LL [29], FV [22], VLAD (VL) [12], and SV [34]. For SV, we use two variants of SV; one is *k*SV which learns codewords by using *k*-means clustering, and another is gSV [5] which learns codewords by GMM clustering and assigns each local feature to multiple codewords according to posterior probabilities that the local features belong to the codewords.

For BF, we use ERC-Tree clustering to generate about 30K codewords. For LL, we use *k*-means to learn 8K codewords. The number of codewords for FV, VL, *k*SV, and gSV are determined so that dimensionality of aggregated feature vector becomes about 300K. For DM, we use N_t=250K training local features. The aggregated vectors are power-normalized and are then L2-normalized. We use Cosine similarity to generate ranking results.

4.1.3 Local Features

We compare the POD against five local features; SI [13], LSF [20], and RoPS [8] as local 3D geometric feature, and DSIFT [4, 5] and MO1SIFT [5] as local 2D visual feature. Table 1 summarizes parameters for DM aggregation used for PSB.

Table 1. Parameters used for DM aggregation (PSB).

	POD	DSIFT	MO1SIFT	LSF	SI	RoPS
k	5	5	10	5	5	5
α	0.1	0.01	0.5	0.1	0.1	0.1
p	1.5	1.5	2.5	1.5	2.0	1.5

For local 3D geometric features, i.e., POD, SI, LSF, and RoPS, we sample N_p=3,000 oriented points from surfaces of a 3D model. The parameters for radius of SOIs are set to r=0.5, v=0.4 for PSB, ESB, and SH14LC and r=0.1, v=0.1 for SH11NR.

For local 2D visual features, i.e., DSIFT and MO1SIFT, a 3D model is rendered from 42 viewpoints spaced uniformly in solid angle. For DSIFT, We densely and randomly extract a set of 300 SIFT [18] features from a rendered image, resulting in about 13,000 SIFT features per 3D model. To reduce bursty SIFT features, we use scale-weighted sampling of DSIFT [19]. Specifically, for PSB, ESB, and SH14LC, we set the scale-weighting parameter W [19] to 0.25, which means more SIFT features are sampled at larger scales on the image. On the other hand, for SH11NR, we use W=4.0 meaning that more SIFT features are extracted from smaller scales. MO1SIFT rotates the rendered image to 16 different orientations and describes each rotated image by a global SIFT feature. Since we use 42 views for rendering, 42×16=672 MO1SIFT features are extracted per 3D model.

4.2 Experimental Results

4.2.1 Effectiveness of DM Aggregation

Figure 4 compares retrieval accuracies of the feature aggregation algorithms for the PSB. We use POD as a local feature. In Figure 4, we plot MAP scores by varying the number of training local features N_t for DM aggregation or the number of codewords for other aggregation algorithms. The DM aggregation significantly outperforms other feature aggregation algorithms. Aggregating local features considering the non-linear manifold structure of local features is effective for higher retrieval accuracy.

Table 2, 3, 4, and 5 summarizes retrieval accuracies of feature aggregation algorithms for the PSB, ESB, SH11NR and SH14LC, respectively. For benchmarks containing rigid shapes, i.e., PSB, ESB, SH14LC, the DM aggregation performs the best among the seven feature aggregation algorithms for most local features. DM aggregation is effective for both local 3D geometric features and local 2D visual features.

On the other hand, for SH11NR having non-rigid shapes, the DM aggregation and its competitors produce similar MAP scores for most cases. We speculate that a key factor behind high accuracy of DM aggregation is "smoothness" of the manifold graph of local features. That is, if local features extracted from 3D models distribute smoothly or continuously in the local feature space, relevance would diffuses appropriately on the manifold graph of local features. For PSB, ESB, and SH14LC, SOIs are highly overlapped each other since we used SOIs having large radius. In such a case, local features, whose sampled positions on 3D model surfaces are near each other, can also located near in the local feature space with high possibility. Therefore, the manifold graph of the large-scale local features becomes smooth and DM aggregation works well for PSB, ESB, and SH14LC. On the other

hand, the manifold graph for SH11NR might not be smooth since SOIs with small radius don't overlap each other. Consequently, improvement of accuracy due to DM aggregation would be limited for the SH11NR.

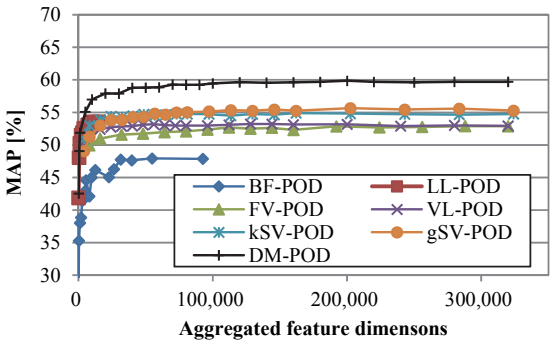


Figure 4. Comparison of feature aggregation methods (PSB).

Table 2. MAP [%] of feature aggregation (PSB).

	POD	DSIFT	MO1SIFT	LSF	SI	RoPS
BF	47.8	54.0	51.9	33.0	38.1	40.5
LL	53.1	57.6	56.5	33.8	41.0	46.4
FV	52.9	61.7	54.2	36.7	44.9	46.3
VL	52.9	60.9	50.6	38.3	45.5	42.7
*k*SV	54.8	61.3	49.6	40.1	47.8	46.8
gSV	55.3	63.8	53.2	40.2	**49.6**	48.4
DM	**60.1**	**64.7**	**61.1**	**41.4**	47.3	**50.4**

Table 3. MAP [%] of feature aggregation (ESB).

	POD	DSIFT	MO1SIFT	LSF	SI	RoPS
BF	52.1	53.6	49.1	47.1	48.3	47.2
LL	53.0	56.0	53.8	50.7	50.9	49.1
FV	54.4	57.8	52.2	52.8	51.7	48.3
VL	49.8	57.5	47.3	49.7	46.5	44.0
*k*SV	52.4	58.7	47.6	51.8	47.1	49.9
gSV	54.0	**59.2**	47.9	52.6	50.9	49.8
DM	**57.2**	59.0	**57.4**	**54.3**	**54.1**	**51.1**

Table 4. MAP [%] of feature aggregation (SH11NR).

	POD	DSIFT	MO1SIFT	LSF	SI	RoPS
BF	87.3	94.3	75.7	87.5	85.8	89.0
LL	94.7	97.0	82.4	93.8	94.2	94.2
FV	**96.7**	95.6	74.7	93.0	95.2	94.9
VL	95.9	96.8	73.1	95.9	95.8	95.2
*k*SV	96.1	**97.2**	74.5	96.0	95.9	**95.6**
gSV	96.5	97.0	75.5	**96.2**	95.4	93.3
DM	95.8	96.8	**89.2**	95.7	**96.9**	93.8

Table 5. MAP [%] of feature aggregation (SH14LC).

	POD	DSIFT	MO1SIFT	LSF	SI	RoPS
BF	39.6	38.5	36.5	30.9	32.7	35.8
LL	44.7	41.7	39.4	31.4	34.6	39.2
FV	44.8	44.1	36.6	35.7	38.6	38.6
VL	44.0	43.4	34.9	34.5	37.1	35.5
*k*SV	45.2	44.1	35.5	36.2	38.9	38.2
gSV	45.3	**45.5**	36.3	37.1	**40.2**	40.1
DM	**50.7**	44.9	**45.5**	**38.5**	39.3	**42.4**

Table 6 shows effectiveness of node weighting on manifold graph for the PSB. In Table 6, "without weighting" indicates accuracies of DM aggregation without node weighting, that is, node weights w_n in Equation 1 is fixed to 1, and "with weighting" shows accuracies of DM aggregation with node weighting. We can observe that node weighting improves retrieval accuracy. Decreasing weights for densely distributed nodes on manifold graph alleviates the problem of local feature burstiness to generate more accurate aggregated feature vectors.

Table 6. Effectiveness of node weighting (MAP [%] for PSB).

	DM-POD	DM-DSIFT	DM-MO1SIFT
without weighting	57.8	61.4	60.8
with weighting	60.1	64.7	61.1

Figure 5 plots retrieval accuracies against the number of iterations T for relevance diffusion on the manifold graph. Interestingly, for DM-POD and DM-DSIFT, accuracies saturate at T=1. While DM-MO1SIFT has a slight peak at around T=5. For larger T, retrieval accuracies are almost unchanged. This is because, presumably, we use small number of neighbors k for constructing manifold graphs (k=5 for POD and DSIFT, k=10 for MO1SIFT as shown in Table 1) to make feature aggregation efficient. In such a case, the manifold graph would consist of multiple sub-manifolds which have almost no connections among them. Relevance diffuses only within a sub-manifold and thus retrieval accuracy remains unchanged if we use larger T.

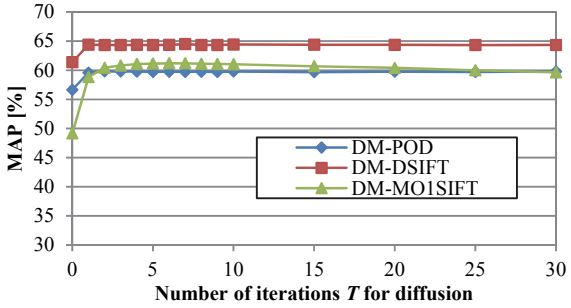


Figure 5. Number of iterations for relevance diffusion and retrieval accuracy (PSB).

Figure 6 plots retrieval accuracies against dimensionality of KPCA-processed feature for the PSB. For all the three features, i.e., DM-POD, DM-DSIFT, and DM-MO1SIFT, MAP score has a peak at around 100 dimensions. As the dimensionality of DM-aggregated feature is 250K, dimension reduction by KPCA down to about 100 dimensions significantly improve computational efficiency of feature comparison among 3D models.

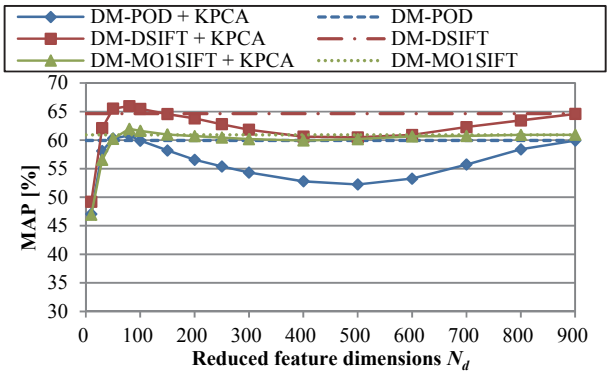


Figure 6. Reduced dimensions and retrieval accuracy (PSB).

4.2.2 Effectiveness of POD Feature

Figure 7 compares retrieval accuracy of POD feature against other existing local 3D geometric features, i.e., SI, LSF, and RoPS, for the PSB. We use DM algorithm to aggregate a set of local features. In Figure 7, the POD performs the best among the four local features. Describing distribution of oriented points within SOI by using SV coding works well for 3DMR.

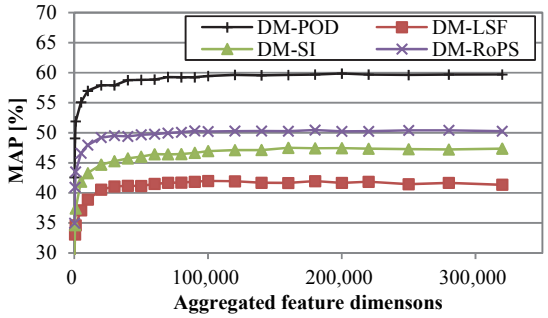


Figure 7. Retrieval accuracies for 3D local features (PSB).

Next, we evaluate impact of radius for SOIs on retrieval accuracy. Figure 8 plots MAP scores, for the PSB, against the parameter r of SOIs. For single scale POD, we fix another parameter v for SOI radius to 0. For multi-scale POD, we set v to values described in Section 4.1.3. We use the DM for aggregating the set of POD features. For the PSB, accuracies have peaks at around r=0.3 to 0.5 and multi-scale POD shows higher MAP than single-scale POD. For SH11NR, on the other hand, the peaks exist at r=0.1 and single-scale POD and multi-scale POD performs comparably well. The articulated models in SH11NR prefer local features extracted from smaller radius for robustness against articulation. On the other hand, the rigid models in the PSB prefers SOIs having more global and more diverse radius.

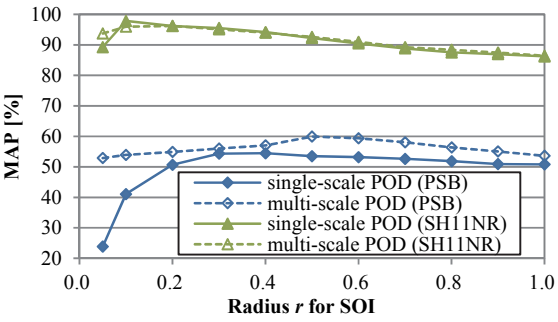


Figure 8. Feature scale and retrieval accuracy (PSB).

4.2.3 Efficiency

In this section, we evaluate efficiency of DM aggregation and POD feature. We use a PC having two *Intel Xeon E5-2650V2* CPUs and 256GB DRAM. All the programs run on a single thread.

Table 7 compares computation times for DM, gSV, and VL with respect to their pre-processing time and feature aggregation time. We sample N_p=3,000 POD features per 3D model. For VL and gSV, pre-processing means learning 3,000 codewords. For DM, pre-processing includes manifold graph construction and node weighting on the manifold graph. The DM aggregation shows the shortest pre-processing time as it doesn't require costly clustering such as k-means or GMM clustering of local features. On the other hand, the DM is slightly slower than the gSV for aggregating N_p=3,000 POD features per 3D model. However, 0.66 [s] for aggregating the set of local features of the 3D model is still acceptable for efficient 3DMR.

Table 7. Computation time [s] of feature aggregation.

algorithm	pre-processing	feature aggregation / model
VL	213.4	0.09
gSV	2876.2	0.50
DM	167.2	0.66

Table 8 shows computation time per query for the SH14LC. In table 8, column "Feat." is a time for extracting a set of N_p=3,000 POD features from a query 3D model, "Agg." is a time for aggregating the set of POD features of the query by using DM, "KPCA" is a time for reducing dimensionality of DM-aggregated feature of the query from N_f=250K dimensions down to N_d=50 dimensions, and "Sim." is a time for computing similarities among the query and 8,987 3D models in the database. The proposed DM-POD takes only about 1.5 [s] to query the SH14LC, which is one of the largest dataset for 3DMR.

Table 8. Computation time [s] per query for the SH14LC.

algorithm	Feat.	Agg.	KPCA	Sim.	total
DM-POD	0.75	0.66	0.03	0.01	1.45

4.2.4 Comparison with Other Retrieval Algorithms

Table 9 compares retrieval accuracy of the proposed DM-POD algorithm with five state-of-the-art 3DMR algorithms; SV-DSIFT [5], LL-MO1SIFT [5], ZFDR [15] and DBSVC [15]. Of the five algorithms, ZFDR combines both 2D visual feature and 3D geometric feature, and others extract 2D visual feature. We also evaluate effectiveness of distance metric learning using Manifold Ranking (MR) [33]. In table 9, "without MR" indicates accuracy without MR, that is, feature vectors for 3D models are directly compared using fixed similarity metric such as Cosine similarity to generate ranking results. And "with MR" shows accuracy using MR algorithm, that is, ranking results are generated by relevance diffusion on the manifold graph of 3D model features.

The proposed DM-POD shows the highest accuracy when MR isn't used. The new local 3D geometric feature combined with the DM aggregation significantly outperforms other 2D view-based retrieval algorithms. On the other hand, when MR is applied, DM-POD shows MAP comparable to that of the DBSVC, which is the best performing algorithm among the SH14LC track entries [15].

Table 9. MAP [%] for the SH14LC.

algorithm	without MR	with MR
SV-DSIFT [5]	46.4	53.1
LL-MO1SIFT [5]	39.9	46.5
ZFDR [15]	38.7	
DBSVC [15]	44.6	54.1
DM-POD (proposed)	**49.5**	**54.3**

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed *Diffusion-on-Manifold* (*DM*), a novel aggregation algorithm for local features, and evaluated it in a shape-based 3D model retrieval (3DMR) setting. The DM exploits, via manifold graph of features and diffusion distance, non-linear manifold structure of local features for more accurate aggregation. Existing feature aggregation algorithms such as Bag-of-Features [2], VLAD [12], Fisher Vector coding [22], Super Vector (SV) coding [34], on the other hand, often fail to capture non-linearity in feature space due to their use of small number of codewords.

We also proposed *Position and Orientation Distribution* (*POD*), a local 3D geometrical feature for 3D oriented point set. It describes distribution of oriented points by using SV coding to generate compact yet accurate local feature.

In experiments, the DM aggregation showed higher accuracy than the existing aggregation. The POD feature also outperformed existing local 3D geometric features for oriented point. As a future work, we will evaluate effectiveness of the DM aggregation on 2D image retrieval setting.

6. ACKNOWLEDGMENTS

This research is supported by *JSPS Grant-in-Aid for Scientific Research on Innovative Areas* #26120517 and *JSPS Grants-in-Aid for Scientific Research* (*C*) #26330133.

7. REFERENCES

[1] Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A. 2011. The devil is in the details: an evaluation of recent feature encoding methods, *British Machine Vision Conference* (*BMVC*) 2011.

[2] Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C. 2004. Visual Categorization with Bags of Keypoints, *Proc. ECCV 2004 workshop on Statistical Learning in Computer Vision*, 59–74.

[3] Donoser, M., Bischof, H. 2013. Diffusion Processes for Retrieval Revisited, *Proc. CVPR 2013*, 1320–1327.

[4] Furuya T., Ohbuchi. R. 2009, Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features, *Proc. ACM CIVR* 2009, Article No. 26.

[5] Furuya, T., Ohbuchi, R. 2014. Fusing Multiple Features for Shape-based 3D Model Retrieval, *Proc. British Machine Vision Conference* (BMVC) *2014*.

[6] Gao, Y., Dai, Q. 2014. View-Based 3D Object Retrieval: Challenges and Approaches, *IEEE MultiMedia*, **21**(3), 52–57.

[7] Geurts, P., Ernst, D., Wehenkel, L. 2006. Extremely randomized trees, *Machine Learning*, **63**(1), 3–42.

[8] Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J. 2013. Rotational Projection Statistics for 3D Local Surface Description and Object Recognition, *IJCV*, **105**(1), 63–86.

[9] Jain V., Zhang, H. 2006. Robust 3D Shape Correspondence in the Spectral Domain, *Proc. SMI 2006*.

[10] Jayanti, S., Kalyanaraman, Y., Iyer, N., Ramani, K. 2006. Developing an engineering shape benchmark for CAD models, *Proc CAD*, **38**(9), 939–953.

[11] Jégou, H., Douze, M., Schmid, C. 2009. On the burstiness of visual elements, *Proc. CVPR 2009*, 1169–1176.

[12] Jégou, H., Douze, M., Schmid, C., P. Perez. 2010. Aggregating local descriptors into a compact image representation, *Proc. CVPR 2010*, 3304–3311.

[13] Johnson, A.E., Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes, *Pattern Analysis and Machine Intelligence*, **21**(5), 433–449.

[14] Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D. 2011. SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes, *Proc. EG 3DOR 2011*, 79–88.

[15] Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Chen, Q., Chowdhury, N. K., Fang, B., Furuya, T., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A. 2014. Large Scale Comprehensive 3D Shape Retrieval, *Proc. EG 3DOR 2014*, 131–140.

[16] Liu, L., Wang, L., Liu, X. 2011. In defense of soft-assignment coding, *Proc. ICCV 2011*, 2486–2493.

[17] Liu, Y., Zha, H., Qin, H. 2006. Shape Topics: A Compact Representation and New Algorithms for 3D Partial Shape Retrieval, *Proc. CVPR 2006*, 2025–2032.

[18] Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, **60**(2), 91–110.

[19] Ohbuchi, R., Furuya, T. 2009. Scale-Weighted Dense Bag of Visual Features for 3D Model Retrieval from a Partial View 3D Model, *Proc. ICCV 2009 workshop on Search in 3D and Video* (*S3DV*), 63–70.

[20] Ohkita, Y., Ohishi, Y., Furuya, T., Ohbuchi, R. 2012. Non-rigid 3D Model Retrieval Using Set of Local Statistical Features, *Proc. ICME 2012 Workshop on Hot Topics in 3D Multimedia*, 593–598.

[21] Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D. 2002. Shape Distributions. *ACM Trans. on Graphics*, **21**(4), 807–832.

[22] Perronnin, F., Sánchez, J., Mensink, T. 2010. Improving the fisher kernel for large-scale image classification, *Proc. ECCV 2010*, Part IV, 143–156.

[23] Picard, D., Gosselin, P.-H. 2011, Improving Image Similarity with Vectors of Locally Aggregated Tensors, *Proc. ICIP 2011*, 669–672.

[24] Rusu, R. B., Blodow, N., Beetz, M. 2009. Fast Point Feature Histograms (FPFH) for 3D registration, *Proc. ICRA 2009*, 3212–3217.

[25] Shilane, P., Min, P., Kazhdan, M., Funkhouser, T. 2004. The Princeton Shape Benchmark, *Proc. SMI 2004*, 167–178.

[26] Tao, P., Cao, J., Li, S., Liu, X., Liu, L. 2015. Mesh saliency via ranking unsalient patches in a descriptor space, *Computers & Graphics*, Volume 46, 264–274.

[27] Tombari, F., Salti, S., Stefano, L. D. 2010. Unique signatures of histograms for local surface description, *Proc. ECCV 2010*, 356–369.

[28] Torki, M., Elgammal, A. 2010. Putting Local Features on a Manifold, *Proc. CVPR 2010*, 1743-1750.

[29] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong. Y. 2010, Locality-constrained Linear Coding for Image Classification, *Proc. CVPR 2010*, 3360–3367.

[30] Yang, J., Yu, K., Gong, Y., Huang, T. 2009, Linear spatial pyramid matching using sparse coding for image classification, *Proc. CVPR 2009*, 794–1801.

[31] Yu, K., Zhang, T., Gong, Y. 2009. Nonlinear Learning using Local Coordinate Coding, *Proc. NIPS 2009*.

[32] Zheng, L., Wang, S., Liu, Z., Tian, Q. 2013. Lp-norm IDF for Large Scale Image Search, *CVPR 2013*, 1626–1633.

[33] Zhou, D., Weston, J., Gretton, A., Bousquet, O., Schölkopf, B. 2004. Ranking on Data Manifolds, *Proc. NIPS 2004*.

[34] Zhou, X., Yu, K., Zhang, T., Huang, T.S. 2010. Image Classification using Super-Vector Coding of Local Image Descriptors, *Proc. ECCV* 2010, 141–154.

[35] Zhu, G., Wang, Q., Yuan, Y., Yan, P. 2013. SIFT on manifold: An intrinsic description, *Neurocomputing*, **113**(30), 227–233.