# Learning Semantic Visual Vocabularies Using Diffusion Distance

Jingen Liu
Computer Vision Lab
University of Central Florida
liujg@cs.ucf.edu

Yang Yang
Computer Vision Lab
University of Central Florida
yyang@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

## Abstract

*In this paper, we propose a novel approach for learning generic visual vocabulary. We use diffusion maps to automatically learn a semantic visual vocabulary from abundant quantized midlevel features. Each midlevel feature is represented by the vector of pointwise mutual information (PMI). In this midlevel feature space, we believe the features produced by similar sources must lie on a certain manifold. To capture the intrinsic geometric relations between features, we measure their dissimilarity using diffusion distance. The underlying idea is to embed the midlevel features into a semantic lower-dimensional space. Our goal is to construct a compact yet discriminative semantic visual vocabulary.*

*Although the conventional approach using k-means is good for vocabulary construction, its performance is sensitive to the size of the visual vocabulary. In addition, the learnt visual words are not semantically meaningful since the clustering criterion is based on appearance similarity only. Our proposed approach can effectively overcome these problems by capturing the semantic and geometric relations of the feature space using diffusion maps. Unlike some of the supervised vocabulary construction approaches, and the unsupervised methods such as pLSA and LDA, diffusion maps can capture the local intrinsic geometric relations between the midlevel feature points on the manifold. We have tested our approach on the KTH action dataset, our own YouTube action dataset and the fifteen scene dataset, and have obtained very promising results.*

## 1. Introduction

In the field of computer vision, bag of features (BOF) is receiving increasing attention due to its simplicity and surprisingly good performance on object, scene, and action recognition problems. The underlying idea is that a variety of statistical cues are present in images and videos, such as color or edge patterns and local structural elements [7, 15, 18], which can be effectively used for recognition. Inspired by the success of the bag of words (BOW) approach in text categorization [12], computer vision researchers have recently discovered the connection between local patches in images/videos and words in documents. In the BOW text representation, a document is represented as a histogram of words. In order to employ the BOW to represent an image or video, we need to quantize the local patches into *visual words*. The *k*-means algorithm is commonly used to construct an initial visual vocabulary due to its simplicity. However, it has two major drawbacks, the first being that the quality of the visual vocabulary is sensitive to the vocabulary size [26]. In general, thousands of visual words are used to obtain better performance on a relatively large dataset. But this vocabulary may contain a large amount of information redundancy. On the other hand, since the clustering criterion is only based on the appearance similarity, *k*-means is unable to capture the semantic relation between the features. This semantic relationship is useful for image and video understanding.

Several attempts have been made to bring the semantic information into visual vocabularies. We can categorize these attempts into two major classes: the supervised and unsupervised approaches. The supervised approaches use either local patch annotation [27] or image/video annotation [9, 13, 17, 28, 30] to guide the construction of a semantic visual vocabulary. Specifically, Vogel *et al.* [27] construct a semantic vocabulary by manually associating the local patches to certain semantic concepts such as "*stone*", "*sky*", "*grass*", etc. The obvious drawback is that this approach is infeasible due to the large amount of manual labor required. Yang *et al.* [30] proposed unifying the vocabulary construction with classifier training, and then encoding an image by a sequence of *visual bits* that constitute the semantic vocabulary. Another interesting work utilizes randomized clustering forests to train a visual semantic vocabulary [17]. The classification trees are built first, but instead of using them for classification, the authors assign a *visual word* label to each leaf, which is how a semantic visual vocabulary is constructed. In addition, several other works [9, 13, 16, 28] use mutual information (MI) between the features and class labels to create the semantic vocabulary from an initial and relatively larger vocabulary quantized by the *k*-means algorithm (Hereafter, we will call the *visual words* in the initial vocabulary midlevel features in order to distinguish them from the low-level raw features and high-level semantic vocabulary features).

Some unsupervised approaches [1,2,8,14,22,24,29] were inspired by the success of the textual *topic* models in text categorization, such as pLSA [2, 22, 24, 29] and LDA [8]. Those models represent an image or video as the mixture distribution of *hidden topics* that can essentially be a se-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE 

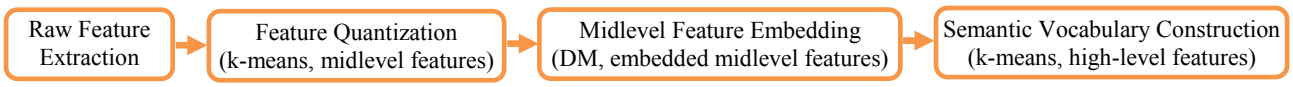



Figure 1: Flowchart of learning semantic visual vocabulary.

mantic visual vocabulary. There is a soft mapping between the *hidden topics* and the midlevel features. Liu *et al.* [14] used *maximization of mutual information* (MMI) to obtain the optimal size of the visual semantic vocabulary for action recognition. We observe that semantically similar features generally have a higher co-occurrence value in the dataset. This is the intrinsic reason that both the *topic* and MMI model can be successfully used to construct a semantic vocabulary.

Both the supervised and unsupervised approaches obtained good performance on object, scene, and action recognition. This is because the semantic visual vocabulary can capture not only the appearance similarity but also the semantic correlation between the midlevel features. We can explain this point clearly using an example in text categorization. For instance, "*pitching*", "*score*" and "*team*" can be correlated to each other by "baseball"; while "*biker*", "*wheel*", and "*ride*" may be correlated to each other by "motorcycle". Hence, we conjecture that the midlevel features produced by similar sources are apt to lie on dynamic feature manifolds. In other words, there exist strong correlations between each dimension of the features, which means the features may have a limited number of degrees of freedom.

However, very few attempts have been made to explicitly preserve the manifold geometry of the feature space when constructing the semantic visual vocabulary. In this paper, we propose to use the diffusion distance, an explicit metric that reflects the connectivity of the feature points (geometric structure between the points), to measure the semantic distance between two feature points when constructing a compact semantic vocabulary. Diffusion distance is derived from diffusion maps (DM) [5] which embeds the manifold points into a lower-dimensional space while preserving the intrinsic local geometric data structure. The diffusion process begins by organizing the data points into a weighted graph (where the weight between two feature points is the feature similarity), which is a good way to represent the complex relationships between the feature points. Once we normalize the weight matrix and also make it symmetric and positive, we can further interpret the pairwise similarities as edge flows in a Markov random walk on the graph. In this case, the similarity is analogous to the transition probability on the edge. Then utilizing the spectral analysis on the Markov matrix of the graph, we can find the dominant $k$ eigenvectors as the coordinates of the embedding space and map the feature points to the low-dimensional space while preserving their local geometric structures. In addition, by adjusting the time of the Markov chain, DM can be also used to employ multi-scale analysis on the data. This multi-scale analysis is similar to Pyramid Match Kernel (PMK) [10], which performs matching under different resolutions of the feature space. If we consider the embedding process as clustering, DM embeds semantically similar features into the same cluster (i.e. some concept). The size of the cluster or the range of the concept is defined by the diffusion time. A larger diffusion time corresponds to a bigger cluster, which means a larger range of concept. For instance, "*sport*" is on a larger scale than "*baseball*" and "*football*"; and "*baseball*" is on a larger scale than "*team*". With the multi-scale data analysis, we can match the data under different scales.

In fact, DM is one of the techniques used for manifold dimension reduction like PCA, ISOMAP [4], Laplacian Eigenmaps [3], etc. In many applications, the distances between feature points that are far apart are meaningless, so preserving the local structure is sufficient for the embedding. Unlike DM, PCA and ISOMAP are global techniques that do not preserve local geometric information of the feature space. In addition, PCA is unable to handle nonlinear manifold data points. Since the diffusion distance derived from DM uses all the paths between two points to compute the distance, it is more robust to noise than the geodesic distance (shortest path distance) used by ISOMAP. DM is very similar to Eigenmaps-based approaches. However, since the embedding coordinates are weighted eigenvectors of the graph Laplacian, DM has an explicit distance measure induced by a nonlinear embedding in the Euclidean space. Eigenmaps representation does not have any explicit metric in the embedding space. Additionally, DM can employ multi-scale analysis on the feature points by defining different time values of the random walk.

In this paper, we represent the midlevel features using Pointwise Mutual Information (PMI) [21], which is employed to measure the correlation of two variables, i.e. features and images or videos. We can consider mutual information as the expectation of PMI.

### 1.1. Overview of the diffusion maps framework

Fig. 1 shows the major steps for constructing a semantic visual vocabulary using diffusion maps. There are 4 components: extracting raw features, quantizing raw features into midlevel features using $k$-means, embedding midlevel features and constructing visual vocabulary using $k$-means. We extract local patches (cuboids) from images (videos) and represent them with the corresponding descriptors. These raw features are quantized into an initial visual vocabulary using $k$-means based on their appearance similarity. We call these quantized features midlevel features.

Then each midlevel feature is represented by a vector, where each element corresponds to PMI of the feature with a particular image or a video. Next, the midlevel features are embedded into a lower-dimensional semantic space using DM. Since the distance between any two feature points is measured by the diffusion distance, we can further apply *k*-means with diffusion distance to construct the final semantic visual vocabulary.

We have tested our proposed approach on the KTH action dataset [15], our own YouTube action dataset, and the fifteen scene dataset [20]. Very inspiring results have been obtained on them.

## 2. Diffusion maps

In this section, we describe DM and the derivation of diffusion distances, as well as the procedure for diffusion embedding.

### 2.1. Diffusion distances in a graph

Graph-based data representation is an effective way to capture the structure information of the data. The distance between two nodes is often defined as the shortest path separating them, which is also called geodesic distance. However, it is sensitive to noise and lack of structure information of the data. The diffusion distances defined in this section can overcome these shortcomings.

We construct a graph $G(\boldsymbol{\Omega}, \boldsymbol{W})$ with $n$ nodes on the midlevel feature set $\boldsymbol{\Omega}$, where $\boldsymbol{W}= \{w_{ij}(x_i, x_j)\}$ is its weighted adjacent matrix that is symmetric and positive. The definition of $w_{ij}(x_i, x_j)$ is totally application-driven, but it needs to represent the degree of similarity or affinity of two feature points. In our case, suppose that the midlevel features are on a manifold, we use a Gaussian kernel function, leading to a matrix with entries

$$w_{ij}(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \quad (1)$$

where $\sigma^2$ indicates the variance of the Gaussian. This graph G with **W** represents our knowledge of the local geometric relationships between the midlevel features.

Then, we define a Markov random walk on the feature graph G using the dynamical systems theory. It is intuitive that if two nodes are closer (more similar), they are more likely transmitted to each other. Therefore, we can treat the normalized edge weight as the transition probability between them. As a result, we form Matrix $\mathbf{P^{(1)}}= \{p_{ij}^{(1)}\}$ by normalizing matrix $\boldsymbol{W}$ such that its rows add up to 1.

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}}. \quad (2)$$

Therefore, each entry of $\boldsymbol{P}$ can be considered as the transition kernel of the Markov chain on G. In other words, $p_{ij}^{(1)}$ defines the transition probability from node $i$ to $j$ in a single time step and $\boldsymbol{P}$ defines the entire Markov chain. $\boldsymbol{P^{(1)}}$ reflects the first-order neighborhood geometry of the data. We could run random walk forward in time to capture information on larger neighborhoods by taking powers of the matrix $\boldsymbol{P}$. The forward probability matrix for $t$ time steps $\boldsymbol{P^{(t)}}$ is given by $[\boldsymbol{P^{(1)}}]^t$. The entries in $\boldsymbol{P^{(t)}}$ represent the probability of going from $i$ to $j$ in $t$ time steps.

In such a framework, a cluster is a region in which the probability of escaping this region is low. The higher the $t$, the higher the probability weight can be diffused to other points which are further away. It means the quantities in $\boldsymbol{P^{(t)}}$ reflect the intrinsic geometry of the data set defined via the connectivity of the graph in a diffusion process and the diffusion time $t$ plays the role of a scale parameter in the data analysis. Generally, larger diffusion time means lower data resolution representation.

Subsequently, we define the diffusion distance $D$ using the random walk forward probabilities $p_{ij}^{(t)}$ to relate the spectral properties of a Markov chain (its matrix and its eigen values and eigenvectors) to the geometry of the data.

$$[D^{(t)}(x_i, x_j)]^2 = \sum_{q\in\Omega} \frac{(p_{iq}^{(t)} - p_{jq}^{(t)})^2}{\varphi(x_q)^{(0)}}.$$

where $\varphi(x_q)^{(0)}$ is the unique stationary distribution which measures the density of the data points. It is defined by $\varphi(x_q)^{(0)} = \frac{d_q}{\sum_j d_j}$, where $d_q$ is the degree of node $x_q$ defined by $d_q = \sum_j p_{qj}$. Note that pairs of data points with high forward transition probability have a small diffusion distance. In other words, the diffusion distances will be small between two points if they are connected by many paths in the graph. This notion of proximity of points in the graph reflects the intrinsic geometry of the set in terms of connectivity of the data points in a diffusion process. The idea behind the diffusion distance is that it is computed on many paths through the graph. Compared to the shortest path method, diffusion distance takes into account all the evidence relating $x_i$ to $x_j$, so it is more robust to noise.

### 2.2. Diffusion maps embedding

Using the spectral theory of the random walk, matrix **P** has $n$ eigenvectors and eigenvalues such that:

$$\mathbf{P^{(t)}}\nu_s = \lambda_s \nu_s \ (s = 0,1, \dots, n-1).$$

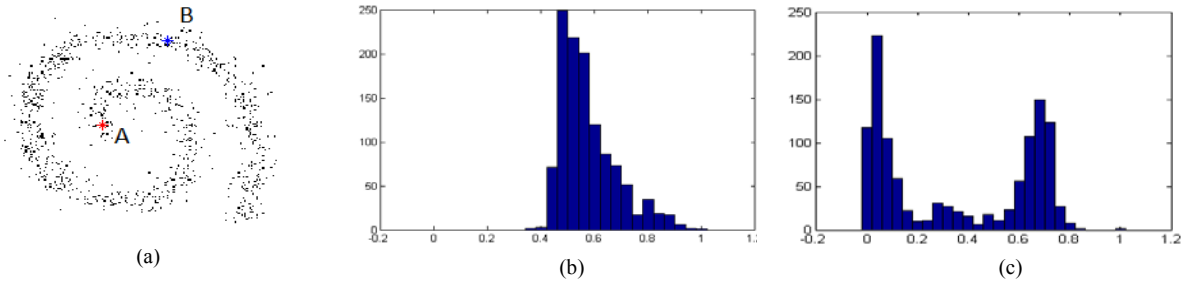
From [23], we can compute the diffusion distance using eigenvectors $\nu$ and eigenvalues $\lambda$ of **P**:

$$[D^{(t)}(x_i, x_j)]^2 = \sum_{s=1}^{n-1} (\lambda_s^t)^2 (\nu_s(x_i) - \nu_s(x_j))^2.$$

The distance can be approximated with the first $k$ eigenvectors. We only need a few terms in the above sum for certain accuracy because of the decay of the eigenvalues. The diffusion distance can then be approximated with relative precision δ using the first $k$ nontrivial eigenvectors and eigenvalues according to

$$[D^{(t)}(x_i, x_j)]^2 \simeq \sum_{s=1}^{k} (\lambda_s^t)^2 (\nu_s(x_i) - \nu_s(x_j))^2,$$

where $\lambda_k^t > \delta\lambda_1^t$. If we use the eigenvectors weighted with $\lambda$ as coordinates on the data, $D^{(t)}$ could be interpreted as the Euclidean distance in the low-dimensional space. Hence, we introduce diffusion map embedding and the




(a) (b) (c)

Figure 2: (a) Two dimensional spiral points. (b) The distribution of the diffusion distance between point A and B. (c) the distribution of the geodesic distance between point A and B.

low-dimensional representation is given by

$$\Pi_t: x_i \mapsto \{\lambda_1^t v_1(x_i) \;\; \lambda_2^t v_2(x_i) \ldots \lambda_k^t v_k(x_i)\}^T \qquad (3)$$

The diffusion map $\Pi_t$ embeds the data into a Euclidean space in which the distance is approximately the diffusion distance:

$$[D^{(t)}(x_i, x_j)]^2 \simeq \left\|\Pi_t(x_i) - \Pi_t(x_j)\right\|^2 .$$

The scaling of each eigenvector by its corresponding eigenvalue leads to a smoother mapping in the final embedding, since higher eigenvectors are attenuated. The mapping provides a realization of the graph G as a cloud of points in a lower-dimensional space, where the rescaled eigenvectors are the coordinates. The dimensionality reduction and the weighting of the relevant eigenvectors are dictated by both the diffusion time *t* of the random walk and the spectral fall-off of the eigenvalues. Diffusion maps embed the entire data set in a low-dimensional space in such a way that the Euclidean distance is an approximation of the diffusion distance. We summarize the procedure of DM in Algorithm 1.

Algorithm 1 Procedure of diffusion maps embedding.

**Objective:** Given $n$ points $\{x_i\}_{i=1}^n$ in a high dimensional space $\Omega$, embed all points into a $k$-dimensional space.

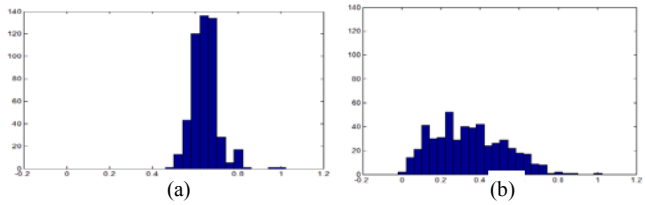
1. Construct a graph $G$ with $n$ nodes: add an edge between nodes $i$ and $j$ using Gaussian kernel.
2. Construct the weight matrix **W**: if nodes $i$ and $j$ are connected, the edge weight $w_{ij}$ is computed by Equation 1.
3. Create Markov transition matrix **P**: normalize matrix **W** using Equation 2 such that its rows add up to 1.
4. Compute Markov transition matrix $\mathbf{P^{(t)}}$ at diffusion time $t$
5. Perform eigen-decomposition on $\mathbf{P^{(t)}}$, and obtain eigenvalues $\lambda_s$ and eigenvectors $v_s$, such that $P^{(t)}v_s = \lambda_s v_s$.
6. Embed data by DM as Equation 3.

## 2.3. Robustness to noise

As aforementioned, the diffusion distance is robust to noise and small perturbations of the data. This results from the fact that the diffusion distance reflects the connectivity of nodes in the graph. In other words, the distance is computed from all the paths between two nodes s.t. all the "evidences" are considered. Although one of the paths may be affected by the noise, it contributes little to the total diffusion distance. However, the geodesic distance that is used in ISOMAP only considers the shortest path between two points, so it is sensitive to noise. Therefore, diffusion distance is more robust than geodesic distance to noise. In the following paragraphs, we want to verify this fact on a synthetic spiral [23] and the real action dataset.

We generated 1,000 instances of two-dimensional spiral with Gaussian noise (see Fig. 2). As for each instance of spiral, we construct a graph by connecting any two points whose Euclidean distance is less than a threshold δ. When constructing the adjacency matrix W, $w_{ij}$ is computed using Equation 1 for the connected points. In order to ensure the connectivity of the graph, $w_{ij}$ is set to a small number for any two non-connected points. We picked two fixed points A and B from all instances, and computed the diffusion distance and geodesic distance between them. Fig. 2 shows the distribution for the distances on all the trials. From it we can easily see the geodesic distance has much larger standard deviation than the diffusion distance. This shows that geodesic distance is more sensitive to the noise as compared to the diffusion distance.

We further verified the robustness of diffusion distance on the KTH dataset. We selected two midlevel features (A and B) that have the maximum Euclidean distance in an initial visual vocabulary with 1,000 visual words (midlevel features). Then we added Gaussian noise to the rest of the features, and repeated this procedure 500 times. For each trial, we constructed a graph as we described in section 2. The distributions of the diffusion distances and geodesic distances between midlevel features A and B are shown in Fig. 3. Although the distribution of geodesic distance is better than that of the synthetic spiral, we can still see that diffusion distance has smaller standard deviation than



(a) (b)

Figure 3: The distribution of (a) diffusion distance and (b) geodesic distance between two midlevel features.

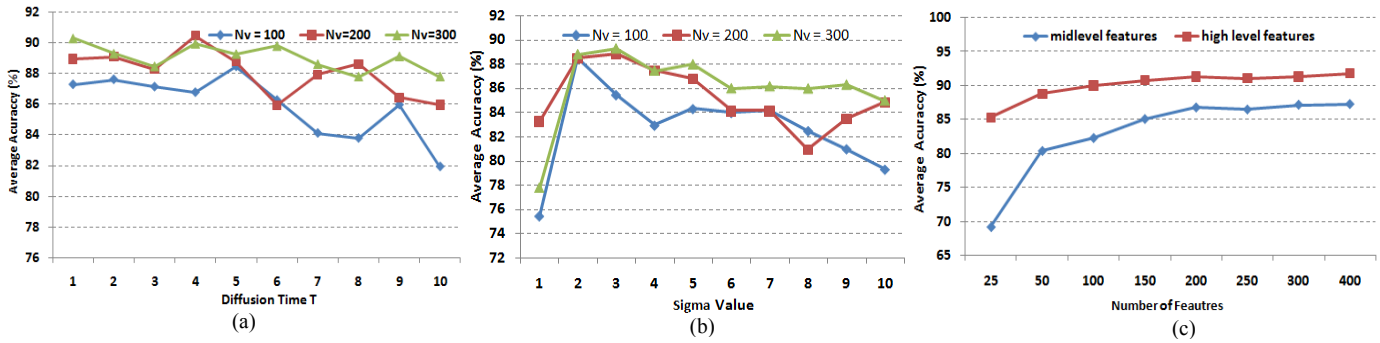



Figure 4: (a) and (b) shows the influence of diffusion time and sigma value on the recognition performance respectively. Three curves correspond to three visual vocabularies of size 100, 200 and 300 respectively. The sigma value is 3 in (a) and the diffusion time is 5 in (b); (c) The comparison of recognition rate between midlevel and high level features.

geodesic distance, which further verifies that diffusion distance is more robust.

## 3. Feature Extraction

In this section, we briefly describe the methods to extract raw features (i.e. motion features for action recognition and SIFT features for scene classification), and then how to generate and represent the midlevel features.

**Motion features for action recognition:** We use the spatiotemporal interest point detector proposed by Dollar *et al.* [7]. Compared to the 3D Harris-Corner detector [15], it produces dense features that can improve the recognition performance in most cases. It utilizes 2-D Gaussian filter and 1-D Gabor filters in spatial and temporal directions respectively. A response value is given at every position *(x, y, t)*. It produces high responses to the temporal intensity change points. The interest points are selected at the locations of local maximal responses, and 3D cuboids are extracted around them. For simplicity, we use the flat gradient vectors to describe the cuboids with PCA being utilized to reduce the descriptor dimension (e.g. 100 dimensions in our paper), which we call the gradient PCA descriptor.

**SIFT features for scene classification:** It has been shown that the dense features can achieve a better classification rate than sparse interest point features for the scene classification [31,8] problem. In this paper, we utilize dense features sampled using regular grid with space M=8 pixels. The patch size is randomly sampled between scales of 10 to 30 pixels. SIFT descriptor [18] is computed for each patch.

**Midlevel feature representation:** Once we extract the raw features (low-level features), we use *k*-means clustering to quantize these gradient PCA features or SIFT features into $C$ clusters, which are the midlevel features forming the initial vocabulary. In general, a larger $C$ value can obtain better performance. We choose $C$ equals 1,000 and 2,000 for the action dataset and scene dataset respectively. In order to construct the semantic vocabularies based on the midlevel features, we use PMI to represent the midlevel features. Suppose we have $N_t$ number of training images or videos; we compute the PMI between a training image/video $x$ and midlevel feature $y$ as

$$pmi(x;y) = \log\left(\frac{f_{xy}}{\sum_{\varepsilon} f_{\varepsilon y} \sum_{\omega} f_{x\omega}}\right),$$

where $f_{xy} = c_{xy}/N_t$, $c_{xy}$ is the number of times feature $y$ appears in image or video $x$. Then we can represent the midlevel feature $y$ in terms of an $N_t$ dimensional feature vector, and the distance between any two features $y_1$ and $y_2$ can be computed using equation 1.

## 4. Dataset and Experiments

We tested our approach on the KTH action dataset, our own YouTube action dataset, and the fifteen scene dataset. SVM with Histogram Intersection kernel is chosen as the default classifier. For the action dataset, we perform the leave one out cross validation (LOOCV) scheme, which means 24 actors or groups are used for training and the rest for testing. For the fifteen scene dataset, we randomly selected 100 images from each category for training, and the rest for testing.

### 4.1. Experiments on KTH dataset

The KTH dataset contains six actions: *boxing*, *clapping*, *waving*, *jogging*, *walking* and *running*. They are performed by 25 actors under four different scenarios. In total it contains 598 video sequences. All the following experiments are conducted on 1,000 midlevel features.

As we discussed, the DMs provide a method to represent the data at different resolutions by using varied diffusion times. Generally, high data resolution can be obtained at smaller diffusion times. Therefore, the diffusion time $t$ can affect the performance of the visual vocabulary. The three curves in Fig. 4 (a) illustrate the influence of $t$ on the action recognition rates when the size of the semantic visual vocabulary ($N_v$) is 100, 200 and 300 respectively (Here, the sigma value is 3 for all of them). It seems that higher recognition accuracy is obtained at a smaller $t$ value when the sigma is fixed. In fact, when $t$ is larger, the data resolution is lower, which may decrease the quality of the visual vocabulary. Additionally, the sigma value in Equation 1 also

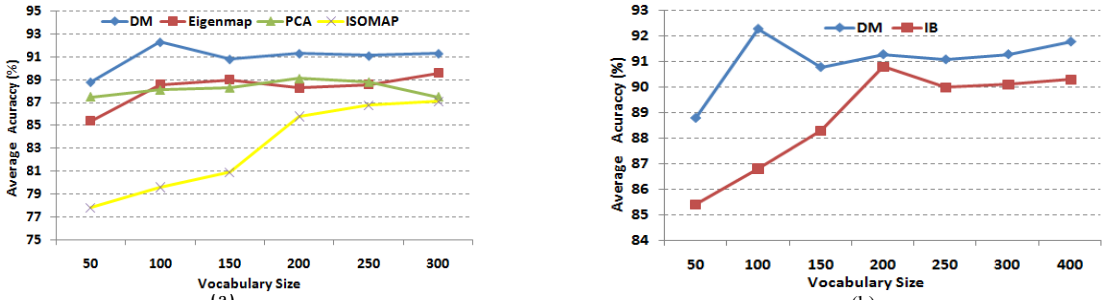



Figure 5: (a) Comparison between different manifold learning schemes. (b) Comparison between DM and IB (Information Bottleneck).

affects the recognition rate. Fig. 4 (b) shows its influence on the recognition performance when fixing the diffusion time $t$=5. The sigma value affects the recognition accuracy by influencing the decay speed of the eigenvalues of matrix $\boldsymbol{P}^{(t)}$. In general, larger sigma values perform worse when diffusion time is fixed. In the following experiments, all the results are reported with the tuned (better) parameters.

In order to verify that our learnt semantic visual vocabulary (high-level features) is more discriminative than the midlevel features, we compared the recognition rate obtained by using high-level and midlevel features under the same size. The high-level features are learnt from the 1,000 midlevel features using DM. The reported recognition rates are the best ones achieved with different diffusion times and sigma values. Fig.4 (c) shows the comparison. It is clear that high-level features can achieve much better performance than midlevel features. Particularly, the recognition rate (88.9%) with 50 features is comparable to that of 400 midlevel features. In addition, when the number of features is larger than 100, the recognition rate is over 90%, and the increase is slow with the growing number of features. It means the recognition rate is not sensitive to the number of features, which is not the case with the midlevel features. This verified the aforementioned fact that the learnt high-level features are semantically meaningful. They can largely improve the recognition efficiency without decreasing the performance for a large dataset.

We believe the features lie on some manifolds, therefore we can apply the manifold learning technique to embed them into a low-dimensional space while maintaining the data structure. We conducted a group of experiments to compare some other manifold techniques (e.g. PCA, ISOMAP, Eigenmap) to DM. We have briefly discussed the difference between them in the introduction. All of them firstly embed the midlevel features into a 100- dimensional space, and then apply $k$-means to the midlevel features to obtain $N$ clusters (high-level features). The results are shown in Fig. 5 (a) (All the techniques have been tuned to have better parameters). We can see DM can achieve improvements from about 2% to 5% in terms of recognition rate, as compared to others. Both DM and ISOMAP define an explicit metric in the embedding space (i.e. diffusion distance and geodesic distance respectively). The experiments further confirm that diffusion distance is more robust than geodesic distance.

As described earlier, the semantic high-level features are learnt by applying $k$-means clustering on the embedded midlevel features. Another way to show the effectiveness of DM embedding is to compare the recognition rate of high-level features learnt by embedded midlevel features to that of original midlevel features without embedding ($k$-means is used as a clustering for both). The results are shown in table 1. The improvements are varied from 2.7% to 4.0%.

Table 1 Performance comparisons between two vocabularies learnt from midlevel features with and without DM embedding.

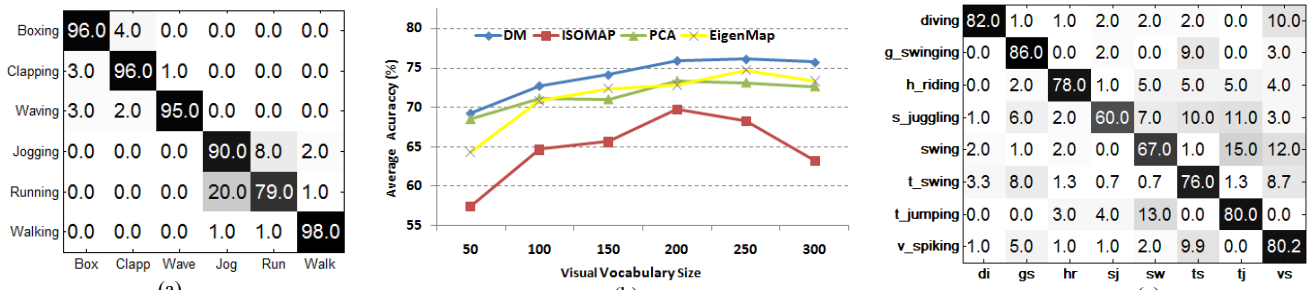
| Vocabulary Size($N_v$) | 50 | 100 | 200 | 300 |
|---|---|---|---|---|
| Embedded features | 88.8 | 92.3 | 91.3 | 91.3 |
| original features | 84.8 | 88.3 | 88.6 | 88.3 |

Information Bottleneck (IB) can also be used to learn a semantic visual vocabulary from the midlevel features [9,14]. Both IB and DM use mutual information for learning. The difference is that DM uses PMI while IB uses expectation of PMI. In addition, IB directly groups the midlevel features without embedding them into a lower-dimensional space. The performance comparisons between them are shown in Fig. 5 (b). Although the IB can achieve very comparable results to DM, the overall performance is worse than DM. We can see DM can achieve more stable performance when the number of features increases, as compared to IB.

Table 2 Performance comparison between two different midlevel feature representations: PMI vs. Frequency.

| Size($N_v$) | 50 | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|
| PMI | 88.8 | 92.3 | 90.8 | 91.3 | 91.1 |
| Freq. | 85.8 | 88.3 | 88.6 | 89.8 | 88.3 |

We believe PMI can capture the relationship between a particular midlevel feature and videos as well as other midlevel features. This is further verified by the experiments shown in Table 2. We conducted two groups of experiments. Both of them use DM to embed features into a lower-dimensional space. The difference is that one of




(a) (b) (c)

Figure 6: (a) Confusion table for KTH dataset when the size of semantic visual vocabulary is 100. The average accuracy is 92.3%. (b) Compassion between DM and other manifold learning schemes on YouTube action dataset. (c) Confusion table of YouTube dataset when the size of semantic visual vocabulary is 250. The average accuracy is 76.1%.


Figure 7: Examples of YouTube action dataset.

them uses PMI to represent the midlevel features and the other directly uses frequency to represent them.

It is very interesting to check the confusion table when the best average accuracy is obtained, see Fig. 6 (a). "*Jogging*" obtains a 90% recognition rate which is better than most exiting approaches [32]. However, *running* is easily misclassified as *jogging*. The overall average accuracy of 92.3% is much better than the average accuracy of 89.3% obtained by directly using the 1,000 midlevel features for classification. It is also a little bit better than some existing BOF-based approaches [15,29].

## 4.2. Experiments on YouTube dataset

Since the KTH dataset is relatively simple, we collected a more complex and challenging dataset based on YouTube videos and our personal video collections. Since we do not have control over the video capturing process, the dataset has the following properties: 1) a mix of still and moving cameras, 2) cluttered background, 3) variation in object scale, 4) varied viewpoint, 5) varied illumination, and 6) low resolution. This action dataset contains 8 categories: *volleyball spiking(v_spiking)*, *trampoline jumping* (*t_jumping*), *soccer juggling (s_juggling)*, *horse-back-riding (h_riding)*, *diving, swinging, golf-swinging (g_swinging),* and *tennis-swinging (t_swinging)*. Most of them share some common motions such as "jumping" and "swinging". We organize the video sequences into 25 relatively independent groups, where separate groups are either taken in different environments or by different photographers. The dataset contains 800 video sequences in total. Fig. 7 shows some examples of the YouTube dataset.

We extracted from 200 to 400 cuboids from each video, and then used *k*-means to obtain 1,000 midlevel features. All the experiments were conducted on these features. Fig. 6 (b) shows the performance comparison between DM and other manifold learning methods. It shows DM gives more stable recognition rates than other approaches with varied sizes of the vocabulary. It obtained the best result of 76.1% accuracy, which is at least about 2.4% higher than the best results obtained by others. We show its details in the confusion table in Fig. 6 (c) for the best results. We can see that several actions are misclassified as "*t_jumping*" and "*v_spiking*". The reason may be that these two actions are not uniform and share many action units with other action categories. We also noticed the best result of 76.1% is competitive to the result of 75.1% obtained by directly using the 1,000 midlevel features for recognition.

Fig. 8 shows the decay of the eigenvalues of $\boldsymbol{P}^{(t)}$ when sigma value is 14. For diffusion time $t$=2, the top 70 eigenvectors are the most significant ones, and for $t$=4, the top 10 are the most significant ones. We noticed when t is larger, very few (i.e. 20) eigenvectors can achieve good performance.

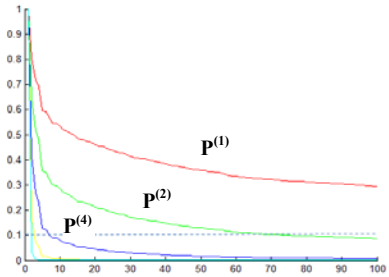



Figure 8: The decay of the eigenvalues of $\mathbf{P}^{(t)}$ on YouTube dataset when sigma is 14.

### 4.3. Experiments on Fifteen scene dataset

We further verified our framework on the fifteen scene dataset [20]. We learnt 2,000 midlevel features using *k*-means. Table 3 lists all the best results we obtained using different manifold learning. DM can perform better than the other methods, but its advantage is less obvious for this dataset as comparing with the action dataset.

Table 3 Best results (%) of different manifold learning techniques.

| | DM | ISOMAP | PCA | EigenMap |
|---|---|---|---|---|
| Average Accuracy | 74.9 | 73.5 | 73.3 | 73.1 |

By using spatial information, Lazebnik's best result was 81.4% [20]. In fact, the results we achieved in this paper are only based on bag of features without any spatial or temporal information. Due to different experiment setup, we cannot directly compare our results to theirs.

## 5. Conclusion

In this paper, we propose a novel approach for generic visual vocabulary learning. We first learnt the midlevel features (the initial visual vocabulary) using *k*-means, then use DM to embed the midlevel features into low-dimensional while maintaining the local relationships of the features. These embedded midlevel features are further clustered to reconstruct a semantically meaningful visual vocabulary. We tested our approach on three complicated datasets. The results verify that the learnt semantic visual vocabularies obtained stable performance compared to the midlevel features learnt by *k*-means. In addition, we also compared DM with other manifold learning techniques. In most cases, DM can perform better, especially for the action dataset.

## 6. Acknowledgement

This research was funded by the US Government VACE program.

## References

[1] J. Liu and M. Shah, Scene modeling using co-clustering, ICCV 2007.
[2] A. Bosch, A. Zisserman and X. Munoz. Scene classification via pLSA, ECCV 2006.
[3] M. Belkin and P. Niyogi. Laplacian Eigenmaps for dimensionality reduction and data representation. In Neural Computation, 15:1373-1396, 2003.
[4] M. Balasubramanian, E. Schwartz, et al. The Isomap algorithm and topological stability. Science, 2002.
[5] R.R. Coifman and S. Lafon, Diffusion maps, in Applied and Computational harmonic Analysis, 21:5-23, 2006.
[6] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints, ECCV 2004.
[7] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie. Behavior recognition via sparse spatiotemporal features, In VS-PETS 2005.
[8] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories, CVPR 2005.
[9] B. Fulkerson, A. Vedaldi and S. Soatto. Localizing objects with smart dictionaries, In ECCV 2008.
[10] K. Grauman and T. Darrell. Pyramid Match Kernels: discriminative classification with sets of image features. ICCV, 2005.
[11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. In JMLR, 3:1157-1182, 2003.
[12] T. Hofmann. Unsupervised learning by Probabilistic Latent Semantic Analysis. Machine Learning, 42, 177-196, 2001.
[13] W.H. Hsu and S. Chang. Visual cue cluster construction via information bottleneck principle and kernel density estimation, CIVR 2005.
[14] J. Liu and M. Shah. Learning human action via information maximization, CVPR 2008.
[15] I. Laptev,M. Marszalek, C. Schmid and B. Rozenfeld. Learning realistic human actions from movies, CVPR 2008.
[16] S. Lazebnik and M. Raginsky. Learning Nearest-Neighbor quantizers from labeled data by Information Loss Minimization, ICAIS 2007.
[17] F. Moosmann, B. Triggs and F. Jurie, Fast discriminative visual codebooks using Randomized Clustering Forests, NIPS 2006.
[18] D.G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91-110,2004.
[19] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition, ICCV 2005.
[20] S. Lazebnik, C. Schmid and J. Ponce. Beyond Bags of Features: spatial Pyramid Matching for recognizing natural scene categories, CVPR 2006.
[21] P. Pantel and D. Lin, Discovering word scenes from text, SIGKDD 2002.
[22] P. Quelhas, F. Monay, J.-M Odobez, D. Gatica-Perez, T. Tuytelaars and L. Van Gool. Modeling scenes with local descriptors and latent aspects. ICCV, 2005.
[23] S. Lafon and A. B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization, PAMI, 28:1393-1430, 2006.
[24] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman, Discovering objects and their location in images, ICCV 2005.
[25] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. ICCV 2003.
[26] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. IJCV, 62:61-81, 2005.
[27] J. Vogel and B. Schiele. Natural scene retrieval based on a semantic modeling step, CIVR 2004.
[28] J. Winn, A. Criminisi and T. Minka. Object categorization by learned universal visual dictionary, ICCV 2005.
[29] S. Wong, T. Kim, et al. Learning motion categories using both semantics and structural information, CVPR 2007.
[30] L. Yang, R. Jin, R. Sukthankar and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category reorganization, CVPR 2008.
[31] E. Nowak, B. Triggs and F. Jurie. Sampling strategies for Bag-of-Features Image Classification, ECCV, 2006
[32] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest, CVPR 2008.