Putting Local Features on a Manifold

Marwan Torki Ahmed Elgammal
Department of Computer Science
Rutgers University, New Brunswick, NJ, USA
{mtorki,elgammal}@cs.rutgers.edu

Abstract

Local features have proven very useful for recognition. Manifold learning has proven to be a very powerful tool in data analysis. However, manifold learning application for images are mainly based on holistic vectorized representations of images. The challenging question that we address in this paper is how can we learn image manifolds from a punch of local features in a smooth way that captures the feature similarity and spatial arrangement variability between images. We introduce a novel framework for learning a manifold representation from collections of local features in images. We first show how we can learn a feature embedding representation that preserves both the local appearance similarity as well as the spatial structure of the features. We also show how we can embed features from a new image by introducing a solution for the out-of-sample that is suitable for this context. By solving these two problems and defining a proper distance measure in the feature embedding space, we can reach an image manifold embedding space.

1. Introduction

Visual recognition is a fundamental and challenging computer vision task. In recent years there have been tremendous interest in the computer vision community on recognition-related problems, such as object categorization, localization, discovering object categories, recognizing generic objects from different views, *etc*.
There are two main motivation for this paper:

1) The importance of local features: local appearance-based descriptors, such as SIFT [13], Geometric Blur [2], KAS [7], *etc*., have proven to be very successful in generic object recognition problems [14]. Such highly discriminative features can be successfully used for recognition even without any shape (structure) information, *e.g*. [14]. There has been also a lot of interest recently on encoding local geometry on top of local features, which was shown to improve recognition, *e.g*. [8, 20].

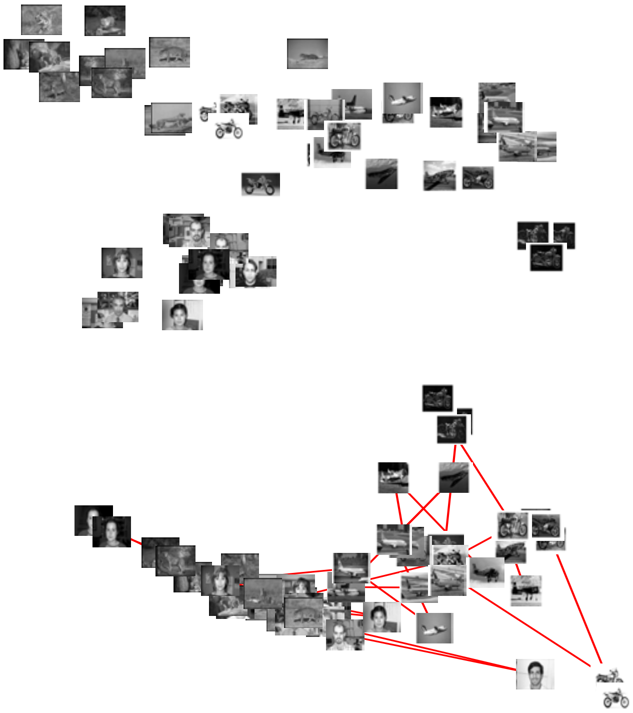

Figure 1. Example Embedding result of samples from four classes of Caltech-101. Top: Embedding using our framework using 60 Geometric Blur local features per image. The embedding reflects the perceptual similarity between the images. Bottom: Embedding based on Euclidean image distance (no local features, image as a vector representation). Notice that Euclidean image distance based embedding is dominated by image intensity, i.e., darker images are clustered together and brighter images are clustered.

2) Manifold Learning: Learning image manifold has been shown to be quite useful for learning within class variability, learning appearance manifolds from different views [15], learning activity and pose manifolds for activity recognition [5] and tracking, *etc*.

However, these two important directions do not quite intersect. Almost all the prior applications of image mani-

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

fold learning, whether linear or nonlinear, have been based on holistic image representations where images are represented as vectors, ***e.g.*** the seminal work of Murase and Nayar [15]. Alternatively, manifold learning can be done on local features if we can establish full correspondences between these features in all images, which explicitly establish a vector representation of all the features. For example, Active Shape Models (ASM) [3] and alike algorithms use specific landmarks that can be matched in all images. Obviously it is not possible to establish such full correspondences between all features in all images in generic object recognition context.

On the other hand, vectorized representations of local features based on histograms, ***e.g.*** bag of words alike representations, cannot be used for learning image manifolds since, theoretically, histograms are not vector spaces.

Another alternative for learning image manifolds is to learn the manifold in a metric space where we can learn a similarity metric between images (from local features). Once such a similarity metric is defined, any manifold learning technique can be used. Since we are interested in problems such as learning within class variability manifolds, view manifolds, or activity manifolds, the similarity kernel should reflect both the local features' appearance affinity and the spatial structure similarity in a smooth way to be able to capture the topology of the underlying image manifold without distorting it. Such similarity kernel should be also robust to clutter. There have been a variety of similarity kernels based on local features, ***e.g.*** pyramid matching kernel [8], string kernels, *etc*. However, to the best of our knowledge, none of these existing similarity measures were shown to be able to learn a smooth manifold.

The challenging question that we address in this paper is how can we learn image manifolds from a punch of local features in a smooth way such that we can capture the feature similarity and spatial arrangement variability between images. If we can answer this question, that will open the door for explicit modeling of within class variability manifolds, modeling objects' view manifolds, modeling activity manifolds, all from local features.

The contribution of this paper is: we introduce a novel framework for learning a manifold representation from collections of local features in images. We first show how we can learn a feature embedding representation that preserves both the local appearance similarity as well as the spatial structure of the features. We also show how we can embed features from a new image by introducing a solution for the out-of-sample that is suitable for this context. By solving these two problems and defining a proper distance measure in the feature embedding space, we can reach an image manifold embedding space. Fig. 1-top shows an example embedding of sample images from four classes of the Caltech101 dataset [12] where the manifold was learned from local features detected on each image. As can be noticed, all images contain significant amount of clutter, yet the embedding clearly reflects the perceptual similarity between images as we might expect. This obviously cannot be achieved using holistic image vectorization, as can be seen in Fig. 1-bottom, where the embedding is dominated by similarity in image intensity. To the best of our knowledge, this cannot be achieved with any existing similarity measure on local features.

In the experiment section we show several applications of the proposed framework on object categorization and localization.

2. Feature Embedding Space

We are given K images, each is represented with a set of feature points. Let us denote such sets by, $X^1, X^2, \cdots X^K$ where $X^k = \{(x_1^k, f_1^k), \cdots, (x_{N_k}^k, f_{N_k}^k)\}$. Each feature point (x_i^k, f_i^k) is defined by its spatial location, $x_i^k \in \mathbb{R}^2$, in its image plane and its appearance descriptor $f_i^k \in \mathbb{R}^D$, where D is the dimensionality of the feature descriptor space[1]. For example, the feature descriptor can be a SIFT [13], GB [2], etc. Notice that the number of features in each image might be different. We use N_k to denote the number of feature points in the k-th image. Let N be the total number of points in all sets, i.e., $N = \sum_{k=1}^K N_k$.

We are looking for an embedding for all the feature points into a common embedding space. Let $y_i^k \in \mathbb{R}^d$ denotes the embedding coordinate of point (x_i^k, f_i^k), where d is the dimensionality of the embedding space, ***i.e.***, we are seeking a set of embedded point coordinates $Y^k = \{y_1^k, \cdots, y_{N_k}^k\}$ for each input feature set X^k. The embedding should satisfy the following two constraints

- The feature points from different point sets with high feature similarity should become close to each other in the resulting embedding as long as they do not violate the spatial structure.
- The spatial structure of each point set should be preserved in the embedding space.

To achieve a model that preserves these two constraints we use two data kernels based on the affinities in the spatial and descriptor domains separately. The spatial affinity (structure) is computed within each image and is represented by a weight matrix $\mathbf{S}^k$ where $\mathbf{S}_{ij}^k = K_s(x_i^k, x_j^k)$ and $K_s(\cdot,\cdot)$ is a spatial kernel local to the k-th image that measures the spatial proximity. Notice that we only measure intra-image spatial affinity, no geometric similarity is measured across images. The feature affinity between image p and q is represented by the weight matrix $\mathbf{U}^{pq}$ where

[1] Throughout this paper, we will use superscripts to indicate an image and subscripts to indicate point index within that image, ***i.e.***, $\boldsymbol{x}_i^k$ denotes the location of feature i in the k-th image.

$\mathbf{U}^{pq}_{ij} = K_f(f^p_i, f^q_j)$ and $K_f(\cdot,\cdot)$ is a feature kernel that measures the similarity in the descriptor domain between the i-th feature in image p and the j-th feature in image q. Here we describe the framework given any spatial and feature weights in general and later in this section we will give specific details on which kernels we use.

Let us jump ahead and assume an embedding can be achieved satisfying the aforementioned spatial structure and the feature similarity constraints. Such an embedding space represents a new Euclidean "Feature" space that encodes both the features' appearance and the spatial structure information. Given such an embedding, the similarity between two sets of features from two images can be computed within that Euclidean space with any suitable set similarity kernel. Moreover, unsupervised clustering can also be achieved in this space.

Given the above stated goals, we reach the following objective function on the embedded points Y, which need to be minimized

$$\Phi(Y) = \sum_k \sum_{i,j} \|y^k_i - y^k_j\|^2 \mathbf{S}^k_{ij} + \sum_{p,q} \sum_{i,j} \|y^p_i - y^q_j\|^2 \mathbf{U}^{pq}_{ij}, \tag{1}$$

where k, p and $q = 1, \cdots, K$, $p \neq q$, and $\|\cdot\|$ is the L2 Norm. The objective function is intuitive; the first term preserves the spatial arrangement within each set, since it tries to keep the embedding coordinates y^k_i and y^k_j of any two points x^k_i and x^k_j in a given point set close to each other based on their spatial kernel weight $\mathbf{S}^k_{ij}$. The second term of the objective function tries to bring close the embedded points y^p_i and y^q_j if their feature similarity kernel $\mathbf{U}^{pq}_{ij}$ is high.

This objective function can be rewritten using one set of weights defined on the whole set of input points as:

$$\Phi(Y) = \sum_{p,q} \sum_{i,j} \|y^p_i - y^q_j\|^2 \mathbf{A}^{pq}_{ij}, \tag{2}$$

where the matrix $\mathbf{A}$ is defined as

$$\mathbf{A}^{pq}_{ij} = \begin{cases} \mathbf{S}^k_{ij} & p = q = k \\ \mathbf{U}^{pq}_{ij} & p \neq q \end{cases} \tag{3}$$

where $\mathbf{A}^{pq}$ is the pq block of $\mathbf{A}$.

The matrix $\mathbf{A}$ is an $N \times N$ weight matrix with $K \times K$ blocks where the pq block is of size $N_p \times N_q$. The k-th diagonal block is the spatial structure kernel $\mathbf{S}^k$ for the k-th set. The off-diagonal pq block is the descriptor similarity kernels $\mathbf{U}^{pq}$. The matrix $\mathbf{A}$ is symmetric by definition since diagonal blocks are symmetric and since $\mathbf{U}^{pq} = \mathbf{U}^{qp^T}$. The matrix $\mathbf{A}$ can be interpreted as a weight matrix between points on a large point set where all the input points are involved in this point set. Points from a given image are linked be weights representing their spatial structure $\mathbf{S}^k$; while nodes across different data sets are linked by suitable weights representing their feature similarity kernel $\mathbf{U}^{pq}$. Notice that the size of the matrix $\mathbf{A}$ is linear in the number of input points.

We can see that the objective function Eq. 2 reduces to the problem of Laplacian embedding [16] of the point set defined by the weight matrix $\mathbf{A}$. Therefore the objective function reduces to

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}^T\mathbf{D}\mathbf{Y}=I} tr(\mathbf{Y}^T\mathbf{L}\mathbf{Y}), \tag{4}$$

where $\mathbf{L}$ is the Laplacian of the matrix $\mathbf{A}$, i.e., $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal matrix defined as $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. The $N \times d$ matrix $\mathbf{Y}$ is the stacking of the desired embedding coordinates such that,

$$\mathbf{Y} = \left[y^1_1, \ldots, y^1_{N_1}, y^2_1, \ldots, y^2_{N_2}, \ldots y^K_1, \ldots, y^K_{N_K}\right]^T$$

The constraint $\mathbf{Y}^T\mathbf{D}\mathbf{Y} = \mathbf{I}$ removes the arbitrary scaling and avoids degenerate solutions [16]. Minimizing this objective function is a straight forward generalized eigenvector problem: $\mathbf{L}y = \lambda\mathbf{D}y$. The optimal solution can be obtained by the bottom d nonzero eigenvectors. The required N embedding points Y are stacked in the d vectors in such a way that the embedding of the points of the first point set will be the first N_1 rows followed by the N_2 points of the second point set, and so on.

2.1. Intra-Image Spatial Structure

The spatial structure weight matrix $\mathbf{S}^k$ should reflect the spatial arrangement of the features in each image k. In general, it is desired that the spatial weight kernel be invariant to geometric transformations. However, this is not always achievable.

One obvious choice is a kernel based on the Euclidean distances between features in the image space, which would be invariant to translation and rotation. Instead we use an affine invariant kernel based on subspace invariance [23]. Given a set of feature points from an image at locations $\{x_i \in \mathbb{R}^2, i = 1, \cdots, N\}$, we can construct a configuration matrix

$$\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_N] \in \mathbb{R}^{N\times 3}$$

where $\mathbf{x}_i$ is the homogenous coordinate of point x_i. The range space of such configuration matrix is invariant under affine transformation. It was shown in [23] that an affine representation can be achieved by QR decomposition of the projection matrix of $\mathbf{X}$, *i.e.*

$$\mathbf{QR} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$$

The first three columns of $\mathbf{Q}$, denoted by $\mathbf{Q}'$, gives an affine invariant representation of the points. We use a Gaussian kernel based on the Euclidean distance in this affine invariant space, *i.e.*,

$$K_s(x_i, x_j) = e^{-\|q_i - q_j\|^2/2\sigma^2}$$

where q_i, q_j are the i-th and j-th rows of $\mathbf{Q}'$

2.2. Inter-Image Feature Affinity

The feature weight matrix $\mathbf{U}^{pq}$ should reflect the feature-to-feature similarity in the descriptor space between the p-th and q-th sets. An obvious choice is the widely used affinity based on a Gaussian kernel on the squared Euclidean distance in the feature space, i.e.,

$$\mathbf{G}_{ij}^{pq} = e^{-\left\|f_i^p - f_j^q\right\|^2/2\sigma^2}$$

given a scale σ. Another possible choice is a soft correspondence kernel that enforces the exclusion principle based on the Scott and Longuet-Higgins algorithm [18]. We used such a kernel with the objective function in Eq 1 for feature matching in [21].

3. Solving the out-of-sample problem

Given the feature embedding space learned from a collection of training images and given a new image represented with a set of features $X^\nu = \{(x_i^\nu, f_i^\nu)\}$, it is desired to find the coordinates of these new feature points in the embedding space. This is an out-of-sample problem, however it is quite challenging. Most of out-of-sample solutions [1] depends on learning a nonlinear mapping function between the input space and the embedding space. This is not applicable here since the input is not a vector space, rather a collection of points. Moreover, the embedding coordinate of a given feature depends on all the features in the new image (because of the spatial kernel). The solution we introduce here is inspired by the formulation in [24][2]. For clarity, we show how to solve for the coordinates of the new features of a single new image. The solution can be extended to embed any number of new images in batches in a straightforward way.

We can measure the feature affinity in the descriptor space between the features of the new image and the training data descriptors using the feature affinity kernel defined in Sec 2. The feature affinity between image p and the new image is represented by the weight matrix $\mathbf{U}^{\nu,p}$ where $\mathbf{U}_{ij}^{\nu,p} = K_f(f_i^\nu, f_j^p)$. Similarly, the spatial affinity (structure) within the new image can be encoded with the spatial affinity kernel. The spatial affinity (structure) of the new image's features is represented by a weight matrix $\mathbf{S}^\nu$ where $\mathbf{S}_{ij}^\nu = K_s(x_i^\nu, x_j^\nu)$. Notice that, consistently, we do not measure any inter geometric similarity between images, we only encode intra-geometric constraints within each image.

We have a new embedding problem in hand. Given the sets $X^1, X^2, \cdots X^K, X^\nu$ where the first K sets are the training data and X^ν is the new set, we need to find embedding coordinates for all the features in all the sets, i.e., we need find $\{y_i^k\} \cup \{y_j^\nu\}$, $i = 1, \cdots, N_k$ and $k = 1, \cdots, K$, $j = 1, \cdots, N_\nu$ using the same objective function in Eq 1[3]. *However, we need to preserve the coordinates of the already embedded points.* Let $\hat{y}_i^k$ be the original embedding coordinates of the training data. We now have a new constraint that we need to satisfy

$$y_i^k = \hat{y}_i^k, \text{for } \ i = 1, \cdots, N_k, k = 1, \cdots, K$$

.

Following the same derivation in Sec 2, and adding the new constraint, we reach the following optimization problem in $\mathbf{Y}$

$$\begin{array}{ll} \min & tr(\mathbf{Y}^T \mathbf{L} \mathbf{Y}) \\ s.t. & y_i^k = \hat{y}_i^k, i = 1, \cdots, N_k, k = 1, \cdots, K \end{array} \tag{5}$$

where

$$\mathbf{Y} = \left[y_1^1, \ldots, y_{N_1}^1, \ldots y_1^K, \ldots, y_{N_K}^K, y_1^\nu, \ldots, y_{N_\nu}^\nu\right]^T$$

where $\mathbf{L}$ is the laplacian of the $(N + N_\nu) \times (N + N_\nu)$ matrix $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}^{train} & \mathbf{U}^{\nu^T} \\ \mathbf{U}^\nu & \mathbf{S}^\nu \end{pmatrix} \tag{6}$$

where $\mathbf{A}^{train}$ is defined in Eq 3 and $\mathbf{U}^\nu = [\mathbf{U}^{\nu,1} \cdots \mathbf{U}^{\nu,K}]$ Notice that the constrain $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I}$, which was used in Eq 4 is not needed anymore since the equality constraints avoid the degenerate solution.

Unlike the problem in Eq 4, which is quadratic programming with quadratic constraints that can be solved by as an eigenvalue problem, the problem in Eq 5 is a quadratic programming with linear equality constraints. It was shown in [24] that this problem can be divided into d subproblems (one in each embedding dimension), each of which is a QP with $N + N_\nu$ variables, N of which are known.

3.1. Populating the Embedding Space

The out-of-sample framework is essential not only to be able to embed features from a new image for classification purpose, but also to be able to embed large number of images with large number of features. The feature embedding framework in Sec 2 solves an Eigenvalue problem on a matrix of size $N \times N$ where N is the total number of features

[2] We are not using the approach in [24] for coordinate propagation, we are only using a similar optimization formulation.

[3] In this case the sets indices k, p, and $q = 1, \cdots K + 1$, to include the new set

in all training data. Therefore, there is a computational limitations on the number of training images and the number of features per image that can be used. Given a large training data, we use a two a step procedure to establish a comprehensive feature embedding space:

1. Initial Embedding: Given a small subset of training data with a small number of features per image, solve for an initial embedding using Eq 4.

2. Populate Embedding: Embed the whole training data with a larger number of features per image, one image at a time by solving the out-of-sample problem in Eq 5

4. From Feature Embedding to Image Manifold Embedding

The embedding achieved in Sec 2 is an embedding of the features where each image is represented by a set of coordinates in that space. This Euclidean space can be the basis to study image manifolds. All we need is a measure of similarity between two images in that space. There are a variety of similarity measures that can be used. For robustness, we chose to use a percentile-based Hausdorff based distance to measure the distance between two sets of features from two images, define as

$$H(X^p, X^q) = \max\{\overset{l\%}{\max_j} \min_i \|y_i^p - y_j^q\|, \overset{l\%}{\max_i} \min_j \|y_i^p - y_j^q\|\} \tag{7}$$

where l is the percentile used. In all the experiments we set the percentile to 50%, ***i.e.***, the median. Since this distance is measured in the feature embedding space, it reflects both feature similarity and shape similarity. Once a distance measure between images is defined, any manifold embedding techniques, such as MDS [4], LLE [17], Laplacian Eigen maps [16], *etc*., can be used to achieve an embedding of the image manifold where each image is represented as a point in that space. We call this space "Image-Embedding" space and denote its dimensionality by d_I to disambiguate it from the "Feature-Embedding" space with dimensionality d.

5. Experimental Results

In all experiments we used the Geometric Blur features (GB) [2]. It was shown in [20] that adding spatial information, geometric features, such as GB, outperform other features. This has been also confirmed with our experiments. In all experiments we set the dimensionality of the feature embedding space to be equal to the minimum number of features per image used in the initial embedding. In all experiments with SVM, a linear kernel was used.

	traning/test splits			
Classifier	1/5	1/3	1/2	2/3
Feature embedding - SVM	74.25	80.29	82.85	87.02
Image Manifold - SVM	80.85	84.96	88.37	91.27
Feature embedding - 1-NN	70.90	74.13	77.49	79.63
Image Manifold - 1-NN	71.93	75.29	78.26	79.34

Table 1. Shape dataset: Average accuracy for different classifier setting based on the proposed representation

5.1. Recognition: Shape

We used the "Shape" dataset [20] to experiment with the proposed approach. The Shape dataset contains 10 classes (cup, fork, hammer, knife, mug, pan, pliers, pot, sauce pan and scissors), with a total of 724 images. The dataset exhibits large within-class variation and moreover there are similarity between classes, ***e.g.*** mugs and cups; saucepans and pots. We used 60 images (6 samples per class chosen randomly) to learn the initial feature embedding of dimensionality 60. Each image is represented using 60 GB feature descriptor. The initial feature embedding is then expanded using out-of-sample to include all the training images with 120 features per images. It is very hard to visualize the feature embedding space, however, the image manifold embedding can be visualized. Fig. 2 shows the resulting image embedding using the first two dimensions. We can easily notice how different objects are clustered in the space. There are many interesting structures we can notice in the embedding, ***e.g.*** mugs and cups are close to each other.

To evaluate the recognition accuracy using the proposed approach, we used different training/testing random splits with 1/5, 1/3, 1/2, 2/3 for training. We used 10 times cross validation and we report the average accuracy. We evaluated four different classifiers based on the proposed representation: 1) Feature-embedding with SVM, 2) Image embedding with SVM, 3) Feature embedding with 1-NN classifier, 4) Image-embedding with 1-NN classifier. Table 1 shows the results for the four different classifier settings. We can clearly notice that a manifold-based classifier enhances the results over a feature-based classifier

In [20] the Shape dataset was used to compare the effect of modeling feature geometry by dividing the object's bounding box to 9 grid cells (localized bag of words) in comparison to geometry-free bag of words. Results were reported using SIFT [13], GB [2], and KAS [7] features. Table 2 shows the reported accuracy in [20] for comparison. All reported results are based on 2:1 ratio for training/testing split. Unlike [20] where bounding boxes are used both in training and testing, we do not use any bounding box information since our approach does not assume a bounding box for the object to encode the geometry and yet get better result.

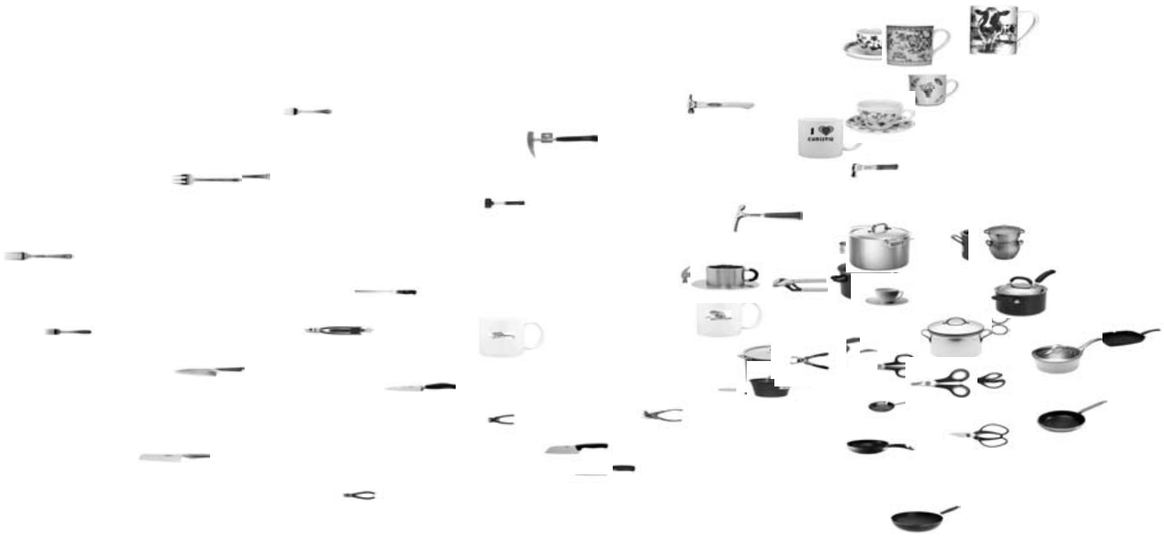

Figure 2. Manifold Embedding for 60 samples from Shape dataset using 60 GB local features per image

Accuracy %			
Feature used	SIFT	GB	KAS
Our approach	-	91.27	-
bag of words (reported by [20])	75	69	65
Localized bag of words ([20])	88	86	85

Table 2. Shape dataset: Comparison with reported results

5.2. Recognition-Caltech 101

The recognition accuracy of the proposed approach was evaluated using subsets of the Caltech-101 dataset [12]. To make it easier to compare to reported results, we used three different subsets of Caltech-101 that are typically used for evaluation: Caltech-4-I (faces, airplanes, motorbikes, leopard) as used in [19, 22, 9], Caltech-4-II (faces, airplanes, motorbikes, cars-rear) as used in [6, 10], Caltech-6 (faces, airplanes, motorbikes, cars-rear, ketch, watches) as used in [6, 10]. In all cases we used 60 geometric blur features per image. We used 12 images per class to achieve the initial feature embedding of dimensionality 60. The whole data set is then embedded using out-of-sample. The image manifold embedding is then constructed using a Haussdorff distance (Eq. 7). Table 3 shows the recognition accuracy using different number of training data and three different classifiers: FE-SVM: Feature embeding space SVM classifier, IE-SVM: Image manifold embedding SVM classifier, and FE-1-NN: Feature embedding space first nearest neighbor classifier. In all cases, the images are used without any bounding box knowledge.

As can be consistently noticed, even a simple 1-NN classifier based on the proposed feature representation gives a superior result. It is also noticeable that we achieve very good results with as little as 5 training samples per class. As can be predicted, the image manifold embedding did nor perform better than just using the feature embedding at smaller training sets (< 30). This is expected since a large number of images are needed to construct a useful manifold. It can be noticed also that the improvement gained by embedding the image manifold in this case is less than what was achieved with the "Shape" dataset (Table 1). This is also expected since, unlike "Shape" dataset, Caltech101 dataset contains lots of clutter besides the objects.

To visualize the obtained manifold, we show in Fig. 1 the embedded image manifold (first two dimensions) obtained after the initial feature embedding (12 images per class, 60 features per image). Using the whole data set we can achieve a more comprehensive embedding of all images. This is shown in Fig. 3 for both Caltech-4-II (2559 images) and Caltech-6 subsets (2912 images). In these example we used MDS to achieve the embedding using the Haussdorff metric (Eq 7) in the embedded feature space. The figure shows the embedding in the first two dimensions where each image is represented by a point. In both cases, we can notice that the classes are well clustered in the space, even though we are only showing only two dimensional embedding.

5.3. Object Localization

The goal of this experiment is to evaluate the robustness of the proposed approach to clutter in the context of object

	# traning images					
	size	5	10	30	50	100
Classifier: FE-SVM						
Caltech-4-I	2233	92.93	95.53	97.54	97.83	98.69
Caltech-4-II	2559	95.92	96.74	98.35	98.57	98.84
Caltech-6	2912	88.16	94.45	96.67	97.14	98.08
Classifier: IE-SVM						
Caltech-4-I	2233	87.46	94.98	97.65	98.14	98.73
Caltech-4-II	2559	86.01	96.73	98.35	98.69	98.84
Caltech-6	2912	82.63	93.77	96.99	97.73	98.42
Classifier: FE-1-NN						
Caltech-4-I	2233	91.57	94.39	96.41	97.22	98.11
Caltech-4-II	2559	95.25	96.03	97.38	98.01	98.45
Caltech-6	2912	89.097	92.60	94.83	95.65	96.99

Table 3. Caltech-101 dataset: Average accuracy with different training sizes. FE-SVM: Feature embedding space SVM classifier, IE-SVM: Image manifold embedding SVM classifier, and FE-1-NN: Feature embedding space first nearest neighbor classifier.

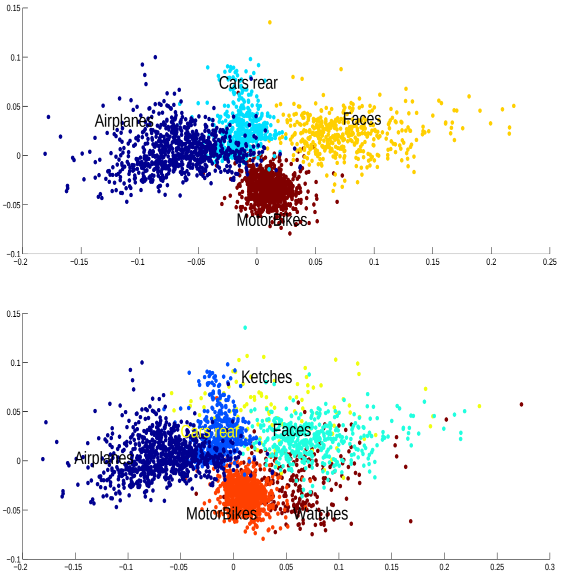

Figure 3. Manifold Embedding for all images in Caltech-4-II, Caltech-6. Only first two dimensions are shown.

localization. Many approaches that encode feature geometry are based on a bounding box, *e.g.* [20, 8]. Our approach does not require such constraint and is robust to the existence of heavy visual clutter. Therefore, it can be use in localization as well as recognition.

We used Caltech-4-I data (as defined above) for evaluation. In this case we learned the feature embedding from all the four classes, using only 12 images per class. For evaluation we used 120 features in each query image and embed them by out-of-sample. The object is localized by finding the top 20% features closer to the training data (by computing feature distances in the feature embedding space.)

Class	TPR	FPR	BBHR	BBMR
Airplanes	98.08%	1.92%	100%	0/800
Faces	68.43%	31.57%	96.32%	16/435
Leopards	76.81%	23.19%	98%	4/200
Motorbikes	99.63%	0.37%	100%	0/798

Table 4. Object localization results - Caltech101-4

Table 4 shows the results in terms of the True Positive Ratio (TPR): the percentage of localized features inside the bounding box, and False Positive Ratio (FPR), Bounding Box Hit Ratio (BBHR), the percentage of images with more than 5 features localized (a metric defined in [11]), and Bounding Box Miss Ratio (BBMR).

5.4. Visualizing Objects View Manifold

COIL data set [15] has been widely used in holistic recognition approaches where images are represented by vectors [15]. This is a relatively easy data set where object view manifold can be embedded using PCA using the whole image as a vector representation [15]. It has also been used extensively in Manifold learning literature, also using whole image as a vector representation. We use this data to validate that our approach can really achieve an embedding that is topologically correct using local features and the proposed framework. Fig 4 shows two examples of the resulting view manifold embedding. In this example we used 36 images with 60 GB features per image. The figure clearly shows an embedding of a closed one dimensional manifold in a two-dimensional embedding space. To the best of our knowledge, there is no previously reported results that successfully embed this kind of manifolds using local features.

5.5. Conclusion

In this paper we introduced a framework that enables the study of image manifolds from local features. We introduced an approach to embed local features based on their inter-image similarity and their intra-image structure. We also introduced a relevant solution for the out-of-sample problem, which is essential to be able to embed large data sets. Given these two components we showed that we can embed image manifolds from local features in a way that reflects the perceptual similarity and preserves the topology of the manifold. Experimental results showed that the framework can achieve superior results in recognition and localization. Computationally, the approach is very efficient. The initial embedding is achieved by solving an eigenvalue problem which is done offline. Incremental addition of images, as well as solving out-of-sample for a query image is done in a time that is negligible to the time needed by the feature detector per image.

There are many interesting open questions that we plan to investigate including, theoretical and empirical studies to

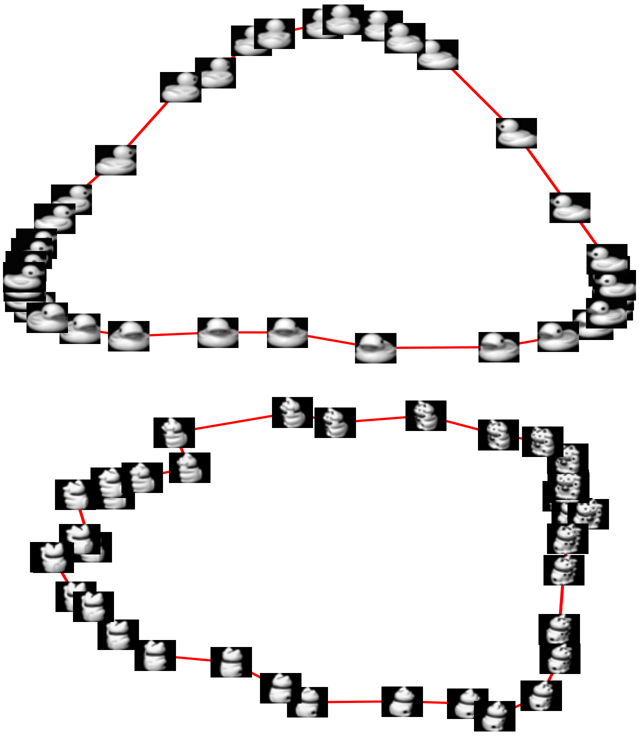

Figure 4. Examples of view manifolds learned from local features

understand how to control the embedding to be biased towards enforcing rigidity vs. enforcing descriptor similarity. In this paper both terms in the objective function have the same weight. We also will investigate the application of the approach to view estimation, facial expression analysis, and activity recognition.

Acknowledgments: This research is partially funded by NSF CAREER award IIS-0546372.

References

[1] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS 16*, 2004. 4

[2] A. C. Berg. *Shape Matching and Object Recognition*. PhD thesis, University of California, Berkeley, 2005. 1, 2, 5

[3] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *CVIU*, 61(1):38–59, 1995. 2

[4] T. Cox and M. Cox. *Multidimentional scaling*. Chapman & Hall, 1994. 5

[5] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, volume 2, pages 681–688, 2004. 1

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, pages 264–271, 2003. 6

[7] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51, 2008. 1, 5

[8] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465 Vol. 2, Oct. 2005. 1, 2, 7

[9] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006. 6

[10] A. D. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *ICCV*, 2005. 6

[11] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *CVPR*, 2008. 7

[12] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 106(1):59–70, April 2007. 2, 6

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 5

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 2005. 1

[15] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995. 1, 2, 7

[16] P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003. 3, 5

[17] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Sciene*, 290(5500):2323–2326, 2000. 5

[18] G. Scott and H. Longuett-Higgins. An algorithm for associating the features of two images. *The Royal Society of London*, 1991. 4

[19] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their localization in images. In *ICCV*, pages I: 370–377, 2005. 6

[20] M. Stark and B. Schiele. How good are local features for classes of geometric objects. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007. 1, 5, 6, 7

[21] M. Torki and A. Elgammal. One-shot multi-set non-rigid feature-spatial matching. In *CVPR*, 2010. 4

[22] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*, 2006. 6

[23] Z. Wang and H. Xiao. Dimension-free afne shape matching through subspace invariance. *CVPR*, 2009. 3

[24] S. Xiang, F. Nie, Y. Song, C. Zhang, and C. Zhang. Embedding new data points for manifold learning via coordinate propagation. *Knowl. Inf. Syst.*, 19(2):159–184, 2009. 4