Chapter 10

Learning Image Manifolds from Local Features

Ahmed Elgammal and Marwan Torki [1]

10.1 Introduction

Visual recognition is a fundamental yet challenging computer vision task. In the recent years there have been tremendous interest in investigating the use of local features and parts in generic object recognition-related problems such as, object categorization, localization, discovering object categories, recognizing objects from different views, *etc.* In this Chapter we present a framework for visual recognition that emphasizes the role of local features, the role of geometry and the role of manifold learning. The framework learns an image manifold embedding from local features and their spatial arrangement. Based on that embedding several recognition-related problems can be solved, such as object categorization, category discovery, feature matching, regression, etc. We start by discussing the role of local features, geometry and manifold learning; and follow that by discussing the challenges in learning image manifolds from local features.

1) The Role of Local Features: Object recognition based on local image features have shown a lot of success recently for objects with large within-class variability in shape and appearance [23, 39, 51, 69, 2, 8, 20, 60, 21]. In such approaches, objects are modeled as a collection of parts or local features and the recognition is based on inferring the class of the object based on parts' appearance and (possibly) their spatial arrangement. Typically, such approaches find

[1]Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA.

Figure 10.1: Example Painting of Giuseppe Arcimboldo (1527-1593). Faces are composed of parts of irrelevant objects

interest points using some operator such as corners [27] and then extract local image descriptors around such interest points. Several local image descriptors have been suggested and evaluated [41], such as Lowe's scale invariant features (SIFT) [39], Geometric Blur [7], and many others (see Sec. 10.7). Such highly discriminative local appearance features have been successfully used for recognition even without any shape (structure) information, *e.g.* bag-of-words like approaches [71, 54, 41].

2) The Role of Geometry: The spatial structure, or the arrangement of the local features plays an essential role in perception since it encodes the shape. There are no better example to show the importance of the shape in recognition over the appearance of local parts than the paintings of the Italian painter Giuseppe Arcimboldo (1527-1593). Arcimboldo is famous for painting portraits that are made of parts of different objects such as flowers, vegetables, fruits, fish, etc. Examples are shown in Figure 10.1. Human perception has no problem recognizing the faces in the paintings mainly from the shape, i.e., the arrangement of parts, rather than from the appearance of the local parts. There are many other examples that can show such a point. One argument might be that it is a matter of scale, at the right scale the local parts can become discriminative. In contrary, we believe that, at the right scale the arrangement of the local features would become discriminative and not the local feature appearance.

There is a fundamental trade-off in part-structure approaches in general: The more discriminative and/or invariant a feature is, the sparser this feature becomes. Sparse features result in losing the spatial structure. For example, a corner detector results in dense but indiscriminative features while an affine invariant feature detector like SIFT will result in sparse features that do not necessarily capture the spatial arrangement. The above trade-off shapes the research in object recognition and matching. On one extreme, are approaches such as bag-of-feature approaches [71, 54] that depend on highly discriminative features and end up with sparse features that do not represent the shape of the object. Therefore, such approaches tend to heavily depend on the feature distribution in recognition. Many researches recently have tried to include the spatial information of features, *e.g.* , by spatial partitioning and spatial histograms, *e.g.* [40, 32, 25, 55]. On the other end of the tradeoff, are approaches that focus on the spatial arrangement for recognition. They tend to use very abstract and primitive feature detectors like corner detectors, which result in dense binary or oriented features. In such cases, the correspondence between features

are established on the spatial arrangement level, typically through formulating the problem as a graph matching problem, *e.g.* [5, 61].

3) The Role of Manifold: Learning image manifolds has been shown to be quite useful in recognition, for example for learning appearance manifolds from different views [44], learning activity and pose manifolds for activity recognition and tracking [17, 65], *etc.*. Almost all the prior applications of image manifold 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 [44], or by establishing a correspondence framework between features or landmarks, *e.g.* [11].

The Manifold of Local Features:

Consider collections of images from any of the following cases or combinations of them:

- Different instances of an object class (within-class variations);
- Different views of an object;
- Articulation and deformation of an object;
- Different objects across-classes or within-class sharing a certain attribute.

Each image is represented as a collection of local features. In all these cases, both the features appearance and their spatial arrangement will change as a function of all the above-mentioned factors. Whether a feature appears in a given frame and where, relative to other features, are functions of the viewpoint of the object and/or the articulation of the object and/or the object instance structure and/or a latent attribute.

Consider in particular, the case of different views of the same object. There is an underlying manifold (or a subspace) where the spatial arrangement of the features should follow. For example, if the object is viewed from a view circle, which constitutes a one-dimensional view manifold, there should be a representation where the features and their spatial arrangement are expected to be evolving on a manifold of dimensionality at most one (assuming we can factor out all other nuisance factors). Similarly, if we consider a full view sphere, a two-dimensional manifold, the features and their spatial arrangement should be evolving on a manifold of dimensionality at most two. *The fundamental question is what is such representation that reveals the underlying manifold topology.* The same argument holds for the cases of within-class variability, articulation, and deformation, and across-class attributes; but in such cases, the underlying manifold dimensionality might not be known.

A central challenging question is how can we learn image manifolds from a bunch 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 within-class variability manifolds, objects' view manifolds, activity manifolds, attribute manifolds; all from local features.

Why manifold learning from local features is challenging :

There are different ways researchers have approached the study of image manifolds, which are not applicable here. This points out the challenges for the case of learning from local features.

1. *Image vectorization based analysis:* Manifold analysis require a representation of images in a vector space or in a metric space. Therefore, almost all the prior applications for image manifold learning, whether linear or non-linear, have been based on wholistic image representations where images are represented as vectors [44, 57, 66, 17]. Such wholisitic image representation provides a vector space representation and a correspondence frame between pixels in images.

2. *Histogram based analysis:* 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 theatrically histograms are not vector spaces. Histograms do not provide smooth transition between different images with the change in the feature-spatial structure. Extensions to the bag-of-words approach, where the spatial information is encoded in a histogram structure, *e.g.* [40, 32, 55] cannot be used for the same reasons.

3. *Land-mark based analysis:* Alternatively, manifold learning can be done on local features if we can establish full correspondences between these features in all image, which explicitly establish a vector representation of all the features. For example, Active Shape Models (ASM) [11] 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, since the same local features are not expected to be visible in all images. This is a challenge in the context of generic object recognition, given the large within-class variability. Establishing a full correspondence frame between features is also not feasible between different views of an object or different frames of an articulated motion because of self occlusion or between different objects sharing a common attribute.

4. *Kernel-based analysis:* 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, activity manifolds, the similarity kernel should reflect both the appearance affinity of local features 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 [25], string kernels [14], *etc.*. However, to the best of our knowledge, none of these existing similarity measures were shown to be able to learn a smooth manifold representation.

Framework Overview: In the following sections we present a framework for learning an image manifold representation from collections of local features in images. Section 10.2 shows how to learn a feature embedding representation that preserves both the local appearance similarity as well as the spatial structure of the features. Section 10.3 shows how to 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, an image manifold embedding space can be obtained. Section 10.5 illustrates several applications of the framework for object categorization, localization, category discovery, and feature matching.

10.2 Joint Feature-Spatial Embedding

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 $X^k = \left\{(x_1^k, f_1^k), \cdots, (x_{N_k}^k, f_{N_k}^k)\right\}$. 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. Throughout this chapter, 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. For example, the feature descriptor can be a SIFT [38], GB [7], 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 $Y^k = \left\{y_1^k, \cdots, y_{N_k}^k\right\}$ 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 $\mathbf{U}_{ij}^{pq} = K_f(f_i^p, f_j^q)$ 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.

10.2.1 Objective Function

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

$$\Phi(Y) = \sum_k \sum_{i,j} \|y_i^k - y_j^k\|^2 \mathbf{S}_{ij}^k + \sum_{p,q} \sum_{i,j} \|y_i^p - y_j^q\|^2 \mathbf{U}_{ij}^{pq}, \tag{10.1}$$

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_i^k and y_j^k of any two points x_i^k and x_j^k in a given point set close to each other based on their spatial kernel weight $\mathbf{S}_{ij}^k$. The second term of the objective function tries to bring close the embedded points y_i^p and y_j^q if their feature similarity kernel $\mathbf{U}_{ij}^{pq}$ is high.

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

$$\Phi(Y) = \sum_{p,q} \sum_{i,j} \|y_i^p - y_j^q\|^2 \mathbf{A}_{ij}^{pq}, \tag{10.2}$$

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

$$\mathbf{A}_{ij}^{pq} = \begin{cases} \mathbf{S}_{ij}^k & p = q = k \\ \mathbf{U}_{ij}^{pq} & p \neq q \end{cases} \tag{10.3}$$

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. 10.2 reduces to the problem of Laplacian embedding [45] of the point set defined by the weight matrix $\mathbf{A}$. Therefore the objective function reduces to

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}^T \mathbf{D} \mathbf{Y} = I} tr(\mathbf{Y}^T \mathbf{L} \mathbf{Y}), \tag{10.4}$$

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,

$$\mathbf{Y} = \left[y_1^1, \ldots, y_{N_1}^1, y_1^2, \ldots, y_{N_2}^2, \ldots y_1^K, \ldots, y_{N_K}^K\right]^T$$

The constraint $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I}$ removes the arbitrary scaling and avoids degenerate solutions [45]. 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.

10.2.2 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 [68]. 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 homogeneous coordinate of point x_i. The range space of such configuration matrix is invariant under affine transformation. It was shown in [68] 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 $\mathbf{Q}'$

10.2.3 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 [52], this is particularly useful for feature matching application [58] as will be discussed in section 10.5.6.

10.3 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 [6] 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 [72][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 10.2. The feature affinity between image p and the new image is represented by the weight matrix $\mathbf{U}^{\nu,p}$ where $\mathbf{U}^{\nu,p}_{ij} = K_f(f^\nu_i, f^p_j)$. 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 $\mathbf{S}^\nu_{ij} = K_s(x^\nu_i, x^\nu_j)$. 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 $\{y^k_i\} \cup \{y^\nu_j\}$, $i = 1, \cdots, N_k$ and $k = 1, \cdots, K$, $j = 1, \cdots, N_\nu$ using the same objective function in Eq 10.1; in this case the indices k, p, and $q = 1, \cdots K+1$, to include the new set. *However, we need to preserve the coordinates of the already embedded points.* Let $\hat{y}^k_i$ be the original embedding coordinates of the training data. We now have a new constraint that we need to satisfy

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

.

Following the same derivation in Sec 10.2, and adding the new constraint, we reach the following optimization problem in $\mathbf{Y}$

$$\begin{array}{ll} \min & tr(\mathbf{Y}^T \mathbf{L} \mathbf{Y}) \\ s.t. & y^k_i = \hat{y}^k_i, i = 1, \cdots, N_k, k = 1, \cdots, K \end{array} \tag{10.5}$$

where

$$\mathbf{Y} = \left[y^1_1, \ldots, y^1_{N_1}, \ldots y^K_1, \ldots, y^K_{N_K}, y^\nu_1, \ldots, y^\nu_{N_\nu}\right]^T$$

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

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}^{train} & \mathbf{U}^{\nu^T} \\ \mathbf{U}^\nu & \mathbf{S}^\nu \end{pmatrix} \tag{10.6}$$

where $\mathbf{A}^{train}$ is defined in Eq 10.3 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 10.4 is not needed anymore since the equality constraints avoid the degenerate solution.

The out of sample solution described earlier used to obtain such a function. We can achieve a closed form solution for this function given the spatial and feature affinity matrices $\mathbf{S}^\nu$, $\mathbf{U}^\nu$

$$\mathbf{Y}^\nu = (\mathbf{L}^\nu)^{-1} \mathbf{U}^\nu \mathbf{Y}^\tau \tag{10.7}$$

[2]We are not using the approach in [72] for coordinate propagation, we are only using a similar optimization formulation.

where $\mathbf{Y}^\tau$ is an $N \times d$ matrix stacking of the embedding coordinate of the training features and $\mathbf{L}^\nu$ is block of the Laplacian $\mathbf{L}$ corresponding to the spatial affinity block$\mathbf{S}^\nu$.

10.3.1 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 10.2 solves an Eigenvalue problem on a matrix of size $N \times N$ where N is the total number of features 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 10.4.
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 10.5

10.4 From Feature Embedding to Image Embedding

The embedding achieved in Sec 10.2 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

$$H_l(X^p, X^q) = \max\{\max_j^{l\%} \min_i \|y_i^p - y_j^q\|, \max_i^{l\%} \min_j \|y_i^p - y_j^q\|\} \tag{10.8}$$

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. However one problem with this distance is that it is not a metric and does not guarantee a positive semi-definite kernel. Therefore, we use this measure to compute a positive definite matrix $\mathbf{H}^+$ by computing the eigen vectors corresponding to the positive eigenvalues of the original $\mathbf{H}_{pq} = H_l(X^p, X^q)$.

Once a distance measure between images is defined, any manifold embedding techniques, such as MDS [13], LLE [48], Laplacian Eigen maps [45], *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.

10.5 Applications

10.5.1 Visualizing Objects View Manifold

The COIL data set [44] has been widely used in holistic recognition approaches where images are represented by vectors [44]. This is a relatively easy data set where the view manifold of an object can be embedded using PCA using the whole image as a vector representation [44]. 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 10.2 shows two examples of the resulting view manifold embedding. In this example we used 36 images with 60 GB features [7] per image. The figure clearly shows an embedding of a closed one-dimensional manifold in a two-dimensional embedding space.

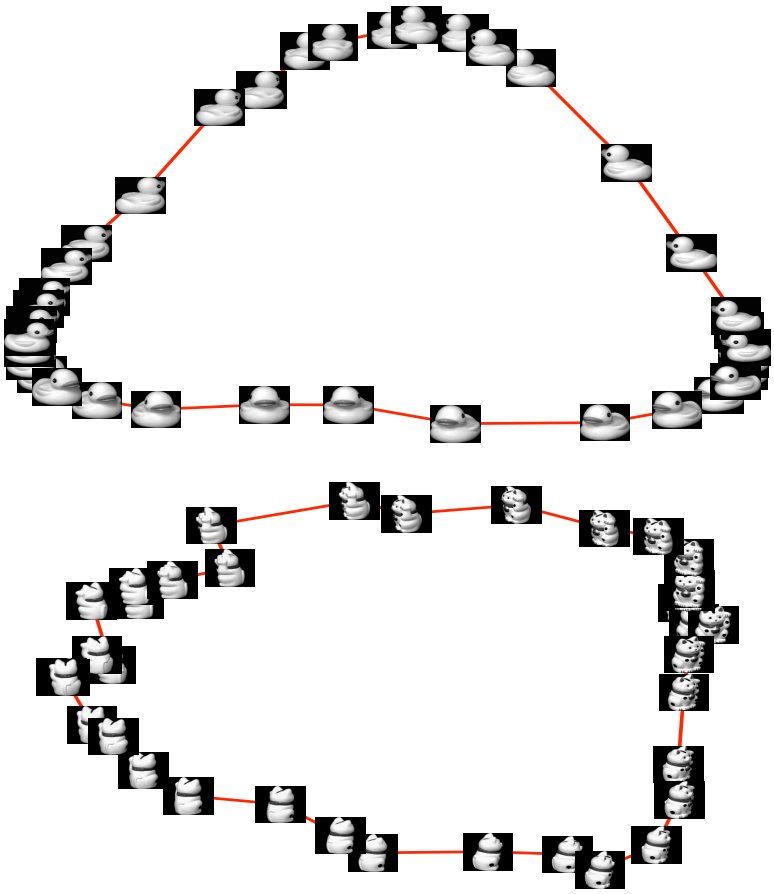

Figure 10.2: Examples of view manifolds learned from local features

10.5.2 What the Image Embedding Captures

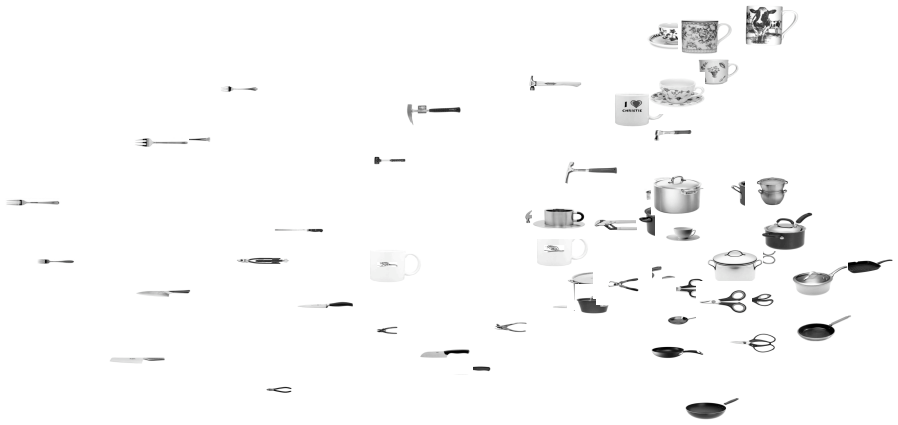

Figure 10.3: Manifold Embedding for 60 samples from Shape dataset using 60 GB local features per image

Fig. 10.3 shows the resulting image embedding space (the first two dimensions are shown) of images from the "Shape" dataset [55]. 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 local features per image. 60 images were used to learn the initial feature embedding of dimensionality 60 (6 samples per class chosen randomly). Each image is represented using 60 randomly chosen geometric blur local feature descriptor [7]. The initial feature embedding is then expanded using the out-of-sample solution to include all the training images with 120 features per images. We can notice how different objects are clustered in the space. It is clear that the embedding captures the object global shape from the local feature arrangement, *i.e.*, the spatial the global spatial arrangement is captured. There many interesting semantics that we can notice in the embedding. There are many interesting structures we can notice. We can notice that objects with similar semantic attributes are grouped together. For example, elongated objects (*e.g.* forks and knifes) are to the left, cylindrical objects (*e.g.* mugs) are to the top right, circular objects (*e.g.* pans) are to the bottom right, *i.e.*, the embedding captures shape attributes. Beyond shape, we can also notice that other semantic attributes are captured, *e.g.* metal forks, knives and other metal objects with black handles, mugs with texture, metal pots and pans *etc.*Notice that this is a two-dimensional projection of the embedding, the dimensionality of the embedding space itself is much higher. This points out that this embedding space captures different global semantic similarities between images only based on local appearance and arrangement information.

Fig. 10.4-top shows an example embedding of sample images from four classes of the Caltech101 dataset [37] where the manifold was learned from

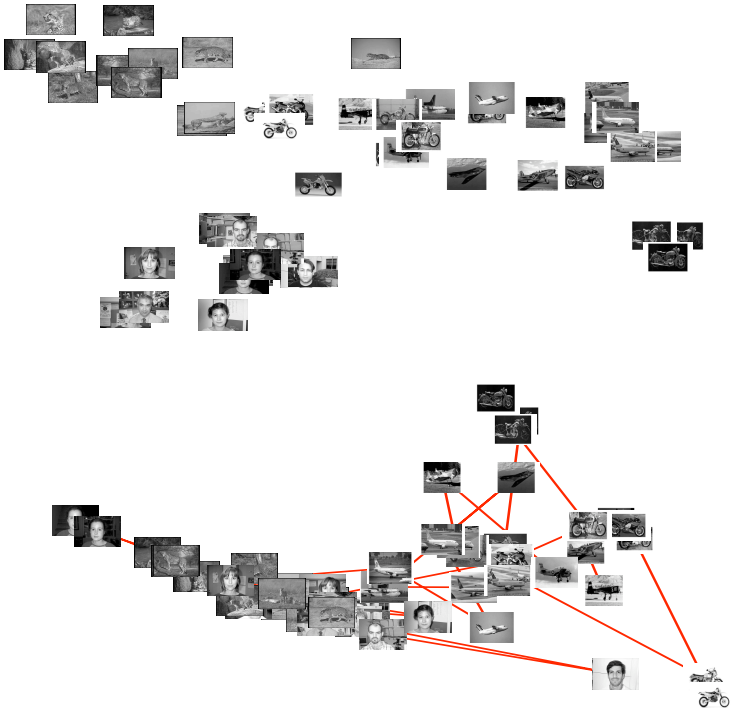

Figure 10.4: Example Embedding result of samples from four classes of Caltech-101. Top: Embedding using our framework using 60 Geometric Blur local features per image. The embedding reflects the perceptual similarity between the images. Bottom: Embedding based on Euclidean image distance (no local features, 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.

local features detected on each image. As can be noticed, the 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. 10.4-bottom, where the embedding is dominated by similarity in image intensity.

Fig 10.5 shows an embedding of four classes in Caltech-4 [37] (2880 images of faces, airplanes, motorbikes, cars-rear). We can notice that the classes are well clustered in the space, even though only the first two dimensions embedding are shown.

10.5.3 Object Categorization

We describe the object categorization problem as an application of learning the image manifold form local features with their spatial arrangement. The goal is to achieve an embedding of a collection of images to facilitate the categorization task. The resulting embedding captures both appearance and shape similarities.

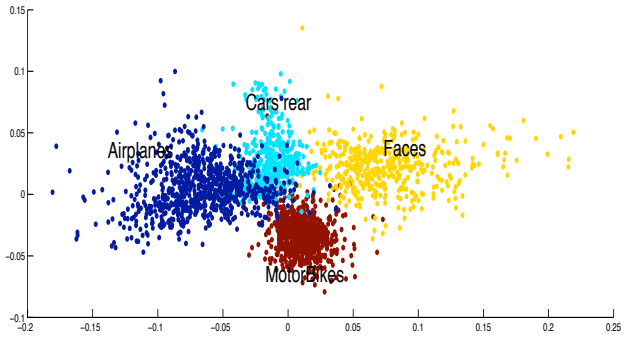

Figure 10.5: Manifold Embedding for all images in Caltech-4-II. Only first two dimensions are shown.

Using such an embedding gives more accurate results when contrasted with other state-of-the art methods.

In [59] the "Shape" dataset [55] was used to evaluate the application of the framework for object categorization based on both the feature embedding and image embedding. Different training/testing random splits were used for training with 1/5, 1/3, 1/2, 2/3 splits and 10 times cross validation and average accuracies were reported. Four different classifiers were evaluated 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 10.1 shows the results for the four different classifier settings. We can clearly notice that image manifold-based classifiers enhance the results over feature embedding-based classifiers. In [59] several other data sets were used to evaluate the performance with similar conclusion. The evaluation also showed very good recognition rates (above 90%) even with as low as 5 training images.

In [55] 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 [38], GB [7], and KAS [22] features. Table 10.2 shows the reported accuracy in [55] for comparison. All reported results are based on 2:1 ratio for training/testing split. Unlike [55] 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.

10.5.4 Object Localization

Many approaches that encode feature geometry are based on a bounding box, *e.g.* [55, 25]. Our approach does not require such a constraint and it is robust to the existence of heavy visual clutter. Therefore, it can be use in localization as well as recognition. We used Caltech-4 data {Airplane, Leopards, Faces, Motorbikes} 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

Table 10.1: Shape dataset: Average accuracy for different classifier setting based on the proposed representation

	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 10.2: Shape dataset: Comparison with reported results

Accuracy %			
Feature used	SIFT	GB	KAS
Our approach	-	91.27	-
bag of words (reported by [55])	75	69	65
Localized bag of words ([55])	88	86	85

object is localized by finding the top 20% features closer to the training data (by computing feature distances in the feature embedding space.) Table 10.3 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 [30]), and Bounding Box Miss Ratio (BBMR).

10.5.5 Unsupervised Category Discovery

Another intersting application for framework is unsupervised categroy discovery. We tested the approach for unsupervised category dicovery by following the setup by [26, 30, 34] on the same benchmark subsets of Caltech-101 dataset. Namely we use the {Airplane, Cars-rear, Faces, Motorbikes} for Caltech-4. We add the class {Watches} for Caltech-5 and the class {Ketches} for Caltech-6. The output is the classification of images according to object category. We use the clustering accuracy as our measure to evaluate the categorization process. We report the average accuracy over 40 runs.

We use NCUT spectral clustering algorithm to compute the desired clustering. Using the $\mathbf{H}^+$ matrix, we compute a weight matrix $\mathbf{W}$ as an input to the clustering algorithm. We further use the K-nearest neighbor graph on the weight matrix $\mathbf{W}$, where K is $O(log(M))$ and M is number of images in the

Table 10.3: Object localization results - Caltech101-4

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 10.4: Caltech-4,5 and 6: Average clustering accuracy, best results are shown in bold.

Categories	FE Clustering	Baseline	[30]	[34]	[26]	Baseline [34]
Caltech-4	**99.54**±0.31	96.43	98.55	98.03	86	87.37
Caltech-5	**98.59**±0.47	96.28	97.30	96.92	NA	83.78
Caltech-6	**97.48**±0.57	94.03	95.42	96.15	NA	83.53

dataset.

We randomly select $12 \times C$ random samples to form an initial embedding that is used to generate initially the common feature embedding of all features. We select 120 features per image for initial embedding and we out-of-sample 420 features (at the most) per image. This results in a common feature embedding that has $100C \times 420$ features. We chose dimensionality of the common feature embedding = 120. Table 10.4 shows comparative evaluation, the state of the art results in [30, 34]. We also show the results by using the baseline that uses feature descriptor similarity to compute $\mathbf{H}_{descriptor}$, in other words there is no spatial arrangement proximity in this $\mathbf{H}_{descriptor}$. The results show that our method is doing extremely excellent job for all the subsets Caltech-4,5 and 6. We infer from these results that the approaches that use explicit spatially consistent matching step like [30, 34] can be outperformed by using a common feature embedding space that encodes the spatial proximity and appearance similarity at the same time, which is done without an explicit matching step.

10.5.6 Multiple Set Feature Matching

Finding correspondences between features in different images plays an important role in many computer vision tasks, including stereoscopic vision, object recognition, image registration, mosaicing, structure from motion, motion segmentation, tracking, etc. [43]. Several robust and optimal approaches have been developed for finding consistent matches for rigid objects by exploiting a prior geometric constraint [63]. The problem becomes more challenging in a general setting, *e.g.* , matching features on an articulated object, deformable object, or matching between two instances (or a model to an instance) of the same object class for recognition and localization. For such problems, many researchers recently tend to use high-dimensional descriptors encoding the local appearance, (e.g. SIFT features [38]). Using such highly discriminative features makes it possible to solve for correspondences without much structure information or avoid solving for correspondences all together, which is quite popular trend in object categorization [49]. This is also motivated by avoiding the high complexity of solving for spatially consistent matches.

The framework for the joint feature-spatial embedding presented in this chapter provides a way to find consistent matches between *multiple* sets of features where both the feature descriptor similarity and the spatial arrangement of the features need to be enforced. However, the spatial arrangement of the features needs to be encoded and enforced in a relaxed manner to be able to deal with non-rigidity, articulation, deformation, and within class variation.

The problem of matching appearance features between two images in a spatially consistent way has been addressed recently (e.g. [36, 12, 10, 61]). Typi-

cally this problem is formulated as an attributed graph matching problem where graph nodes represent the feature descriptors and edges represent the spatial relations between features. Enforcing consistency between the matches led researchers to formulate this problem as a quadratic assignment problem where a linear term is used for node compatibility and a quadratic term is used for edge compatibility. This yields an NP-hard problem [10]. Even though some efficient solutions (e.g. linear complexity in the problem description length) have been proposed for such a problem [12] the problem description itself remains quadratic, since consistency has to be modeled between every pair of edges in the two graphs. This puts a huge limitation on the applicability of such approaches to handle large number of features; for example, for matching n features in two images, an edge compatibility matrix of size $n^2 \times n^2$, i.e., $O(n^4)$, needs to be computed and manipulated to encode the edge compatibility constraints. Obviously this is prohibitively complex and does not scale to handle a large number of features.

The problem of consistent matching can be formulated as an embedding problem [58] where the goal is to embed all the features in a Euclidean embedding space where the locations of the features in that space reflect both the descriptor similarity and the spatial arrangement. This is achieved through minimizing the same objective function in Eq 10.1 enforcing both the feature similarity and the spatial arrangement. A soft correspondence kernel that enforces the exclusion principle based on the Scott and Longuet-Higgins algorithm [52] is advantageous for such application. The embedding space acts as a new unified feature space (encoding both the descriptor and spatial constraints) where the matching can be easily solved. This embeding-based matching framework directly generalizes to matching multiple sets of features in one shot through solving one Eigen-value problem. An interesting point about this formulation is that the spatial arrangement for each set is only encoded within that set itself, *i.e.*, in a graph matching context no compatibility needs to be computed between the edges (no quadratic terms or higher order terms), yet we can enforce spatial consistency. Therefore, this approach is scalable and can deal with hundreds and thousands of features. Minimizing the objective function in the proposed framework can be done by solving an Eigen-value problem *which size is linear in the number of features in all images.*

Fig. 10.6 shows sample matches on motorbike images from Caltech101 [37]. Eight images were used to achieve a unified feature embedding and then pairwise matching were performed in the embedding space using the Scott and Longuet-Higgins (SLH) algorithm [52]. Extensive evaluation of the feature matching application of the framework can be found in [58]

10.6 Summary

In this chapter we presented 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

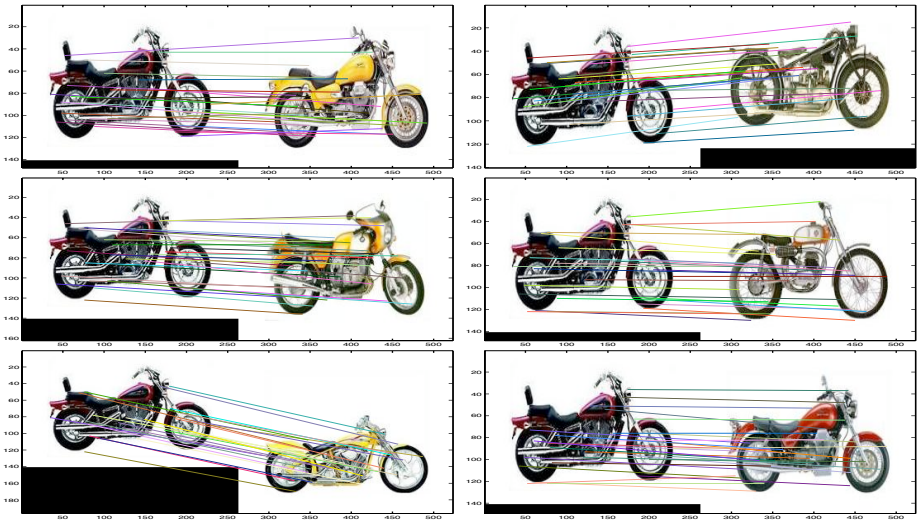

Figure 10.6: Sample Matching results on Caltech 101 Motorbike images.

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.

10.7 Bibliographical and Historical remarks

The use of local features and parts for visual recognition is rooted in the computer vision literature for long time, *e.g.* [23], however such paradigm received extensive interest in the last decade, *e.g.* [39, 51, 69, 2, 8, 20, 60, 21], and others. Several local feature descriptors have been proposed and widely used such as Lowe's scale invariant features (SIFT) [39], entropy-based scale invariant features [29, 20], Geometric Blur [7], contour based features (kAS) [22], and other local features that exhibit affine invariance, such as [3, 62, 50].

Modeling the spatial structure of an object varies dramatically in the literature of object classification. On the extreme, are approaches that totally ignore the structure and classify the object only based on the statistics of the features (parts) as an unordered set, *e.g.* bag-of-features approaches [71, 54]. Generalized Hough transform like approaches provide a way to encode spatial structure in a loose mannar [35, 46]. Similar idea was used earlier in the constellation model of Weber *et al.* [69] where part locations were modeled statistically given a central coordinate system, also in [20]. Pairwise distances and directions between parts have also been used to encode the spatial structure, *e.g.* [1]. Felzenszwalb and Huttenlocher's Pictorial structure [19] uses spring like constraints between pairs of parts as well to encode global structure. The constellation model of Weber *et al.* [69] constrains the part locations given a central coordinate system.

The seminal work of Murase and Nayar [44] showed how linear dimension-

ality reduction using PCA [28] can be used to establish a representation of an object's view and illumination manifolds. Using such representation, recognition of a query instance can be achieved by searching for the closest manifold. Such subspace analysis has been extended to decompose multiple orthogonal factors using bilinear models [57] and multi-linear tensor analysis [66].

The introduction of nonlinear dimensionality reduction techniques such as Local Linear Embedding (LLE) [48], Isometric Feature Mapping (Isomap) [56], and others [56, 48, 4, 9, 31, 70, 42], made it possible to represent complex manifolds in low-dimensional embedding spaces in ways that preserve the manifold topology. Such manifold learning approaches have been used successfully in human body pose estimation and tracking [17, 18, 65, 33].

There is a huge literature on formulating correspondence finding as a graph-matching problem. We refer the reader to [10] for an excellent survey on this subject. Matching two sets of features can be formulated as a bipartite graph matching in the descriptor space, e.g. [5], and the matches can be computed using combinatorial optimization, *e.g.* the Hungarian algorithm [47]. Alternatively, spectral decomposition of the cost matrix can yield an approximate relaxed solution, e.g. [52, 15], which solves for an orthonormal matrix approximation for the permutation matrix. Alternatively, matching can be formulated as a graph isomorphism problem between two weighted or unweighted graphs to enforce edge compatibility, e.g. [64, 53, 67]. The intuition behind such approaches is that the spectrum of a graph is invariant under node permutation and, hence, two isomorphic graphs should have the same spectrum, the converse does not hold. Several approaches formulated matching as a quadratic assignment problem and introduced efficient ways to solve it, e.g. [24, 7, 12, 36, 61]. Such formulation enforces edgewise consistency on the matching, however that limits the scalability of such approaches to a large number of features. Even, higher order consistency terms have been introduces [16]. In [10] an approach was introduced to learn the compatibility functions from examples and was found that linear assignment with such a learning scheme outperforms quadratic assignment solutions such as [12]. In [58] the approach described in this chapter was also shown to outperform quadratic assignment and without the need to resort to edge compatibilities.

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

Bibliography

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *TPAMI*, 26(11):1475–1490, 2004.

[2] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, pages 113–130, 2002.

[3] A. Baumberg. Reliable feature matching across widely separated views. In *CVPR*, pages 774–781, 2004.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 2002.

[6] 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.

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

[8] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, pages 109–124, 2002.

[9] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In *Proc. of the Ninth International Workshop on AI and Statistics*, 2003.

[10] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *TPAMI*, 2009.

[11] 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.

[12] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. *NIPS*, 2006.

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

[14] M. Daliri, E. Delponte, A. Verri, and V. Torre. Shape categorization using string kernels. In *SSPR06*, pages 297–305, 2006.

[15] E. Delponte, F. Isgrò, F. Odone, and A. Verri. Svd-matching using sift features. *Graph. Models*, 2006.

[16] O. Duchenne, F. Bach, I. S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *CVPR*, 2009.

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

[18] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *CVPR*, volume 1, pages 478–485, 2004.

[19] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[20] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR (2)*, pages 264–271, 2003.

[21] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhausitive recognition. In *CVPR*, 2005.

[22] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *TPAMI*, 30(1):36–51, 2008.

[23] M. Fischler and R. Elschlager. The representation and matching of pictorial structures, 1973. IEEE Transaction on Computer c-22(1): 67-92.

[24] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *TPAMI*, 1996.

[25] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, volume 2, pages 1458–1465 Vol. 2, Oct. 2005.

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

[27] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of The Fourth Alvey Vision Conference*, 1988.

[28] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[29] T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 2001.

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

[31] N. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. In *NIPS*, 2003.

[32] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. pages II: 2169–2178, 2006.

[33] C.-S. Lee and A. Elgammal. Coupled visual and kinematics manifold models for human motion analysis. *IJCV*, July 2009.

[34] Y. J. Lee and K. Grauman. Shape discovery from unlabeled image collections. In *CVPR*, 2009.

[35] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *In ECCV workshop on statistical learning in computer vision*, pages 17–32, 2004.

[36] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *ICCV*, 2005.

[37] 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.

[38] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[39] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[40] M. Marszaek and C. Schmid. Spatial weighting for bag-of-features. In *CVPR*, pages II: 2118–2125, 2006.

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

[42] P. Mordohai and G. Medioni. Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting. In *Proc. of AJCAI*, 2005.

[43] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *IJCV*, 2007.

[44] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 14:5–24, 1995.

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

[46] A. Opelt, A. Pinz, , and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006.

[47] C. Papadimitriou and K. Stieglitz. *Combinatorial Optimization Algorithms and Complexity*. Prentice Hall, 1982.

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

[49] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. *ICCV*, 2007.

[50] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV (1)*, pages 414–431, 2002.

[51] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *TPAMI*, 19(5):530–535, 1997.

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

[53] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *IVC*, 1992.

[54] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *ICCV*, 2005.

[55] M. Stark and B. Schiele. How good are local features for classes of geometric objects. In *ICCV*, pages 1–8, Oct. 2007.

[56] J. Tenenbaum. Mapping a manifold of perceptual observations. In *NIPS*, volume 10, pages 682–688, 1998.

[57] J. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12:1247–1283, 2000.

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

[59] M. Torki and A. Elgammal. Putting local features on a manifold. In *CVPR*, 2010.

[60] A. B. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. In *CVPR*, 2004.

[61] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. *ECCV*, 2008.

[62] T. Tuytelaars and L. J. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, 2000.

[63] S. Ullman. Aligning pictorial descriptions: An approach to object recognition. *Cognition*, 1989.

[64] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *TPAMI*, 1988.

[65] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with gaussian process dynamical models. In *CVPR*, pages 238–245, 2006.

[66] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensebles: Tensorfaces. In *Proc. of ECCV, Copenhagen, Danmark*, pages 447–460, 2002.

[67] H. Wang and E. R. Hancock. Correspondence matching using kernel principal components analysis and label consistency constraints. *PR*, 2006.

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

[69] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.

[70] K. W. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *CVPR*, volume 2, pages 988–995, 2004.

[71] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *IWLAVS*, 2004.

[72] 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.