LEARNING THE MANIFOLDS OF LOCAL FEATURES AND THEIR SPATIAL ARRANGEMENTS

by MARWAN TORKI

A Dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

written under the direction of

Ahmed Elgammal

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2011

ABSTRACT OF THE DISSERTATION

Learning the Manifolds of Local Features and Their Spatial Arrangements

by Marwan Torki

Dissertation Director: Ahmed Elgammal

Local features play an important role for many computer vision problems; they are highly discriminative and possess invariant properties. However, the spatial configuration of local features plays an essential role in recognition. Spatial neighborhoods capture local geometry and collectively provide shape information about a given object. In this dissertation we studied explicit and implicit ways to exploit the joint feature-spatial arrangement in images for recognition problems. We introduce a framework to learn an embedded representation of images that captures the similarity between features and the spatial arrangement information. The framework was successfully applied in object recognition and localization context. The framework was also applied for feature matching across multiple images. We also showed the viability of the framework in regression from local features for viewpoint estimation. We also studied implicit ways to exploit the feature-spatial manifold structure in the data without explicit embedding and within a transductive learning paradigm for object localization. We learned the labels of the local features from an object class in a manner that provides spatial and feature smoothing over the labels. To achieve that we adapted the Global and Local Consistency Solution for Label Propagation to our implicit manifold model to infer the labels of local features. We showed excellent accuracy rates with very low false positive rates on the learned features labels in the test images.

Acknowledgements

I would like to thank my family for their love and support all along. It was a dream of us all and we can finally feel it. I want to express my deepest gratitude to my advisor, Prof. Ahmed Elgammal, who was guiding me along this path for full five years. His advices and encouragements were more than valuable and his support was unlimited. I would like to thank my committee Prof. Casimir Kulikowski, Prof. Vladimir Pavlovic and Dr. Sanjiv Kumar of Google research for their very useful comments to improve the quality of my dissertation

Dedication

To Parents and Family

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

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 dissertation we present a framework for visual recognition that emphasizes the role of local features, geometry and 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 [43, 78, 108, 135, 2, 15, 40, 124, 39]. 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 interest points using some operator such as corners [55] and then extract local image descriptors around such interest points. Several local image descriptors have been suggested and evaluated [86], such as Lowe's scale invariant features (SIFT) [78], Geometric Blur [11], and many others. 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 [137, 112, 86].

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 is a fundamental trade-off in part-structure approaches in general: The more discriminative and/or invariant a feature is, the less frequent 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 [137, 112] 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. [81, 66, 50, 114]. On the other end of the trade-off, 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. [9, 125].

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 [91], learning activity and pose manifolds for activity recognition and tracking [36, 128], 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 [91], or by establishing a correspondence framework between features or landmarks, e.g. [28].

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 nonlinear, have been based on wholistic image representations where images are represented as vectors [91, 120, 129, 36]. 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 theoretically 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. [81, 66, 114] 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) [28] 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 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 [50], string

kernels [33], 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.

1.2 Contributions

In this section we highlight the key contributions of the dissertation.

1.2.1 Fusing Feature Similarities and Spatial Arrangements of Local Features in a Common Embedding Space

The first contribution in this dissertation is to learn a low-dimensional representation from a bunch of local features from different images. The learned embedding representation preserves both the spatial arrangements of local features within an image and the feature similarities between features from different images. To achieve such representation we propose an objective function solution of which can be computed in a closed form using eigenvector decomposition. This new low-dimensional representation fuses both the feature similarities and spatial arrangements of local features from different images in a common embedding space, which enhances the task of learning a similarity measure between images. Details on the embedding will be presented in chapter 3.

1.2.2 Defining Similarity Measure between Images

The dimensionality reduction provides a global embedding for the feature points as the whole embedding is affected by all the feature points and thus the distances in the embedding space are affected by all the points embedded. This makes the task of learning an image to image kernel in the new embedding space smoother than computing a kernel that relies only on the features from two images. Here comes another contribution of the dissertation, we provide a distance measure in the embedding space obtained by dimensionality reduction on the feature points. This measure reflects the feature and spatial similarities between feature points as intended and moreover it provides smooth distance measure between images which will correctly capture smooth image manifolds. Fig. 1.1shows a view manifold example which is correctly captured using our similarity measure.Details on the image to image distance will be presented

in chapter 4.

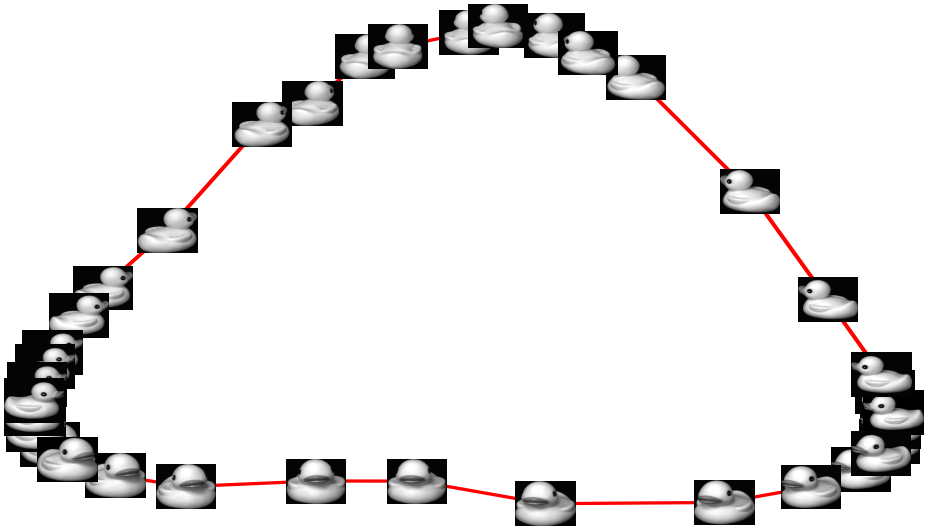


Figure 1.1: Examples of view manifold learned from local features for toy example

1.2.3 Solving The Out-of-Sampling Problem Embedding New Features

In manifold learning, embedding large number of features can be prohibitive in certain situations. Usually this problem is tackled in two steps. First, data is sampled and an embedding for sample points is computed to form an initial embedding. Second, the remaining points are out-of-sampled using some approximation technique [117]. Out-of-sampling is also needed when it is required to embed test data for recognition purposes. However, for either of the two cases current approximation methods do not consider the case where the out-of-sampled features are structured in groups like our case where every set of features belongs to a single image are structured via the spatial structure within that set. Our contribution is to solve the out-of-sample problem where the features that belong to the same image are embedded in a way that respects the spatial structure within an image and, in same time, reflects the feature similarity between the features of the new embedded image and the already embedded features in the initial embedding. We provide a closed form solution in chapter 3.

1.2.4 Learning Image Manifolds from Local Features

As a direct consequent of image-to-image distance, image manifolds can be learned. The contribution here is to utilize image manifolds in object recognition tasks like unsupervised object categorization, object classification and object localization. Our approach outperforms many state-of-the-art methods in corresponding problems. Details and results are given in chapters 4 and 5.

1.2.5 Regression Framework from Local Features

Many computer vision problems are regression problems, for example viewpoint and pose estimation, age estimation, facial expression intensity, etc. However, there is no previous work on regression problems using collections of local features where there is no correspondence available. For example enforcing the spatial structure on a learned view manifold makes it easier to capture the underlying manifold in a smooth way and enables for learning continuous regularized regression functions. Our contribution is to provide kernel-based regression framework from local features. Details will be presented in chapter 6.

1.2.6 Scalable Multi-Set Feature Matching

Feature matching is a very important and fundamental problem in computer vision. Many state-of-the-art matching techniques try to achieve spatially consistent feature matching using quadratic assignment. These methods deals with the spatial consistency by adding higher order terms between pairs of features which grow the size of the problem in quadratic order of the original number of features to be matched. The embedding framework that we propose in this dissertation has three merits. First, the embedding preserves both spatial arrangements of local features within an image and the feature similarities between features from different images. Second it allows for multiple images to matched together without the need of solving a quadratic assignment for every pair of images. Third it involves one eigenvector decomposition problem whose size is linear in the number of features from all images which makes our solution scalable compared to quadratic assignment methods. Our contribution is to formulate the feature matching problem as a graph embedding problem which involves one eigenvector

problem to embed all features at once. Also another contribution in our solution is to handle multiple images that need to be matched together while quadratic assignment methods can not. Details will be presented in chapter 7.

1.2.7 Implicit Feature-Spatial Manifold

The feature spatial manifold embedding that we mentioned in previous subsection 1.2.1 is an explicit way to obtain the low-dimensional representation of the feature points. The dimensionality reduction requires solving an eigenvector decomposition problem. However, in certain problems we are given some label information, which can be viewed as a supervised embedding space for the feature points. Using the label information would alleviate the need of solving the dimensionality reduction problem. Learning the Feature-spatial manifold reduces to learning a graph structure that reflects inter image spatial arrangements and intra image feature similarities. Learning the graph structure of the spatial visual manifold is the final contribution in this dissertation. We show the usefulness of spatial visual manifold in the object class localization problem. Details will be presented in chapter 8.

Chapter 2

Background

2.1 Local Features in Object Recognition

During the last two decades lots of research in computer vision community had focused on local features and their usage in many different problems such as stereo vision, image registration, mosaicing, structure from motion, motion segmentation, tracking, instance recognition, object recognition, object detection, etc. Invariant properties for the local features to image transformations, distinctiveness, and robustness to occlusion of the local features make them more plausible to be used in the wide range of problems in computer vision community. In fact, the top cited computer vision paper for the past ten years is ”Distinctive Image Features from Scale-Invariant Keypoints ” by Lowe [79, 78], where the SIFT descriptor for keypoints in images was presented. This gives us an insight on how important it would be to utilize local features to develop state-of-the-art methods in many object recognition systems and other computer vision problems in general.

2.1.1 Feature Detectors

The local features are point locations in images associated with vectorized descriptor. Lots of researches had been conducted to show how to compute candidate interest point locations for useful local features in an image. These interest points include Harris corners [55], Harris-affine regions [85], Hessian-affine regions [85], maximum stable extremal region (MSER) [83], salient regions[58], and more. These feature/region detectors try to find the candidate patches in the image that makes the local feature informative, invariant to geometric (affine, rotation or scale) transformations and repeatable to facilitate recognition tasks. Another method to detect the feature points was introduced in [11] by sampling the edges according to edge strength

scores.

2.1.2 Feature Descriptors

Another important part is the description of the local features. The descriptor should be invariant to viewing angle, illumination, compression, blurring, zooming, etc. The descriptor is summarizing the information in the patch around the location of the local feature. Examples are Scale Invariant Feature Transform (SIFT) [79], Geometric Blur (GB) [11], Gradient Location Orientation Histogram(GLOH) [86], Histogram of oriented gradient (HOG) [32], Shape Context (SC) [9], etc. Many of the mentioned descriptors are histograms representing local edge orientation distribution, for example SIFT [79] is represented by a 3D histogram of gradient locations and orientations. Also Shape context [9] is similar to the SIFT descriptor, but is based on edges. Shape context is a 2D histogram of edge point locations and orientations.

2.1.3 Performance of Detectors and Descriptors

Several evaluation studies on local features have been published. An evaluation on region detectors was presented in [87]. This evaluation was based on the repeatability of the features and on matching image pairs under different viewing conditions. The conclusion as indicated by [87] is that the performance of all presented detectors declines slowly, with similar rates, as the change of viewpoint increases. There does not exist one detector that outperforms the other detectors for all scene types and all types of transformations.

Another study on evaluating descriptors has been presented in [86], again the evaluations was based on matching tasks under different viewing conditions. The conclusion as indicated by [86] was that in most of the tests, GLOH obtains the best results, closely followed by SIFT. This shows the robustness and the distinctive character of the region-based SIFT descriptor. Shape context also shows a high performance. However, for textured scenes or when edges are not reliable, its score is lower.

Another evaluation study [89] combined the evaluation on detectors and descriptors together. In [89] the evaluation is done on 3D objects, and it was found that the best overall choice is using an affine-rectified detector [85] combined with a SIFT [79] or shape context

descriptor [9].

2.1.4 Bag of Visual Words Models

Inspired by bag of words model in text categorization [13], many works,e.g. [31, 66, 144], addressed the object recognition problem with no dependence on the spatial configuration. A simple approach has been followed to utilize local features in object recognition tasks, called the bag of words model, which can be summarized as:

1. Detect local features using a feature detector and compute local feature descriptors.
2. Compute a dictionary of visual words from training images by clustering the feature descriptors using K-means or other clustering technique.
3. Compute a histogram representation for each training image based on the frequencies of the visual words.
4. Learn a classifier from training images.
5. Assign every feature in the test image to its nearest visual word.
6. Compute a histogram representation for the test image.
7. Classify the resulting histogram to decide the category of the test image.

Advantages of the bag of visual words model lies in its simplicity, it implicitly inherits the discriminative nature of the local features in the dictionary building step, also the model is able to summarize every image in a single vector, which ease the categorization tasks.

An interesting work by Boiman et al. [14] showed that a system that is based on nearest neighbor search between query features and all the features belongs to one class can outperform a visual word system. This actually means that, the feature space quantization into visual words caused some loss in representing the query features.

It is not difficult task to compute an image-to-image kernel if images are in the same metric space. Once an image-to-image kernel is computed, Kernel SVM [107] can be learned to classify the different categories. The bag of words model tries to build signatures of the same

dimensionality for all images involved in an object recognition system. In [144] example distance metrics are used like the Earth Mover Distance(EMD) [104] or the Chi squares distance. Later these distances are transformed into SVM kernels using Gaussian kernel.

Another important kernel is the pyramid matching kernel (PMK) in [50]. This method can handle collections of local features in images without the need of building a dictionary. Instead a multi resolution pyramid is used to bin the features and histogram intersection kernel is computed.

2.2 Encoding Shape based on Local Features

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 objects only based on the statistics of the features (parts) as an unordered set, e.g. bag-of-words approaches [31, 66, 144]. However, the performance of object recognition systems was shown to be improved by utilizing the shape or the arrangements of the local features [114].

2.2.1 Grouping of Local Features within Spatial Neighborhood

Several bag-of-words extensions are offered to encode the spatial relations within the learned dictionaries. The work of [112] extended the bag-of-words vocabulary to include doublets that encode spatially local co-occurring regions. Doublets are defined as pairs of visual words that co-occur within a local spatial neighborhood.

Similar ideas for encoding the spatial relations of visual words are studied by [143] and Spatial Keyton Histogram is introduced. Another important direction is to do feature selection within the higher order features that encode spatial relationships of a learned bag of words model. The exhaustive nature of higher order features need to be handled to avoid exponential growth of the cardinality of the learned features. Towards this end the work of [77] proposed to use feature selection framework on the learned visual words and then the higher order features are learned using only the selected ones. The feature pool is updated using the added higher order features and the process continues. This approach is not only handling co-occurrence of pairs of visual words but also it handles the case of co-occurrence of tuples of size N visual

words and, in same time, it does not need to generate all the higher order tuples before feature selection.

All the mentioned approaches start with a visual word dictionary that is learned on the descriptors space, the next step is to do the pairing between the learned visual words. However, a recent approach [90] build a dictionary on the concatenated paired descriptors of locally close features and thus they can learn a Local Pairwise Codebook (LPC).

2.2.2 Part Based Models

Pairwise distances and relative locations between parts have also been used to encode the spatial structure, e.g. [1]. Felzenszwalb and Huttenlochers Pictorial structure [38] uses spring like constraints between pairs of parts to encode the global object structure. However, [38] restricts the connections to a tree, which makes learning and inference more tractable.

Sacrificing the ease of inference on pictorial structures comes the fully connected constellation model, where the assignment of features to parts becomes intractable for moderate numbers of parts P. The trade-off between the number of features and the number of parts is crucial in the constellation model and would prevent from having many features in images. The constellation model by [17, 135, 40] consists of a number of parts whose relative positions are encoded to constrains the part locations given a central coordinate system and pairwise covariances. Fig: 2.1 shows the constellation model and pictorial structure (Tree) for five parts model.

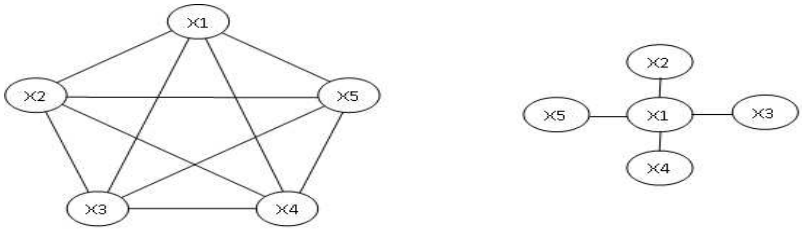
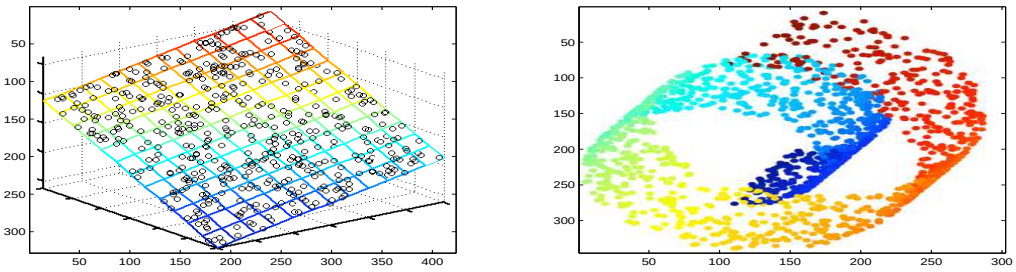


Figure 2.1: Different part models. Left:Constellation model. Right:Pictorial Structure(Tree).

2.3 Manifold Learning for Object Recognition

Manifold learning is a very powerful tool for data analysis. The question to be answered via manifold learning techniques is how to reveal a low-dimensional structure in a high dimensional data. There are two kinds of low-dimensional structures can be found in the data namely linear and nonlinear structures. As can be seen in fig 2.2 the linear structure means the data lies on a low-dimensional subspace. Where the nonlinear structure means the data lies on a low-dimensional manifold. The low-dimensional representation is intended to maintain pairwise relationships between data points. In other words ,nearby points remain nearby and distant points remain distant. The problem of dimensionality reduction can be defined as given *M*,



(a) Linear Structure (b) Non-Linear Structure

Figure 2.2: Left: Linear structure where the data lies on a low dimensional subspace.Right: Non-Linear structure where the data lies on a low dimensional manifold.

points $x_1, x_2, \ldots, x_M \in \Re^D$, these points need to be embedded into $y_1, y_2, \ldots, y_M \in \Re^d$, where $d << D$. Under certain geometric constraints that preserves the topology of the data.

2.3.1 Linear Methods

The linear methods are suitable when the input data lie on a low-dimensional subspace. The outputs returned by these methods are related to the input patterns by a simple linear transformation. Example methods are principal component analysis (PCA) [57] and multi dimensional scaling (MDS) [30]. For example, the objective in PCA is to obtain a low-dimensional representation while maximizing the variance. This is achieved by finding a set of orthonormal bases $\{e_j\}_{j=1}^{d}$, which are the top d eigenvectors of the covariance matrix $C = \frac{1}{M}\sum_i x_i x_i^T$. The resulting embedding $y_{ij} = x_i e_j$.

On the other hand the MDS tries to preserve the inner product between the input points.

This can be achieved by finding the spectral decomposition of the gram matrix $G = X^T X$ Where X is $M \times D$ matrix for all the input points. Finding the top d eigenvectors of this Gram matrix $M \times M$ by $\{v_j\}_{j=1}^{d}$ and their corresponding eigenvalues by $\{\lambda_j\}_{j=1}^{d}$, the resulting embedding of MDS are given by $y_{ij} = \sqrt{\lambda_j} v_{ji}$.

2.3.2 Nonlinear Methods

The nonlinear methods, also called graph based methods, are suitable when the input data lie on a low-dimensional manifold. Linear methods tend to fail and the points will be projected on each other. The nonlinear methods start with graph construction step, where the graph approximates the geodesics between the data points. The graph nodes are the data points and the edges are the pairwise weights that are based on neighborhood structure. Spectral decomposition is then performed on the graph and the lower dimensional representations of the data points can be computed directly from the corresponding eigenvectors. Since we are more interested in nonlinear methods we summarize two of the widely used methods namely Laplacian Eigen-Maps [93] and Isometric feature mapping (ISOMAP) [119]. Other methods includes local linear embedding (LLE) [103], Maximum variance unfolding (MVU) [136], etc.

ISOMAP embedding is a clear example where the goal is to to preserve geodesic distances as measured along manifold. Fig 2.3 shows that the geodesic distance on the manifold is not equivalent to the Euclidean distance for same points and that is why linear methods like MDS would fail to unfold the underlying manifold.

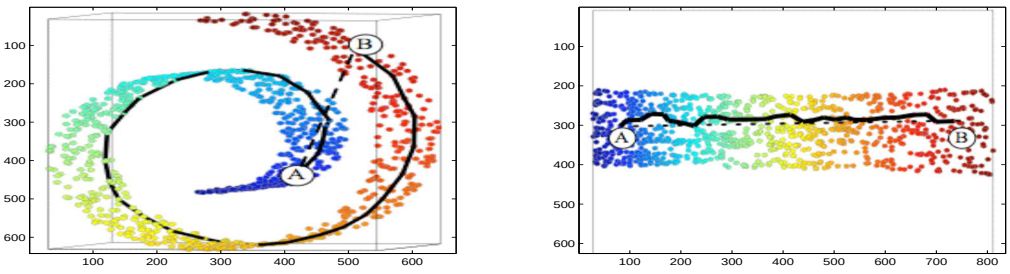


Figure 2.3: Geodesic distance on the manifold between the points A and B is not equivalent to the Euclidean distance.

In ISOMAP, the first step is to build adjacency graph based on k nearest neighbors. The second step is to compute the pairwise distances between all nodes along shortest paths through the graph. This can be done using Djikstras algorithm. the third step is to apply MDS on the

distance matrix Δ and finally produce the d significant eigenvectors of the Gram matrix.

Of particular interest to this dissertation is the Laplacian eigenmaps, where a weighted adjacency matrix (graph) on the original features is defined as $\mathbf{W}_{ij} = exp^{-\|x_i - x_j\|^2/2\sigma^2}$. The following objective function need to be minimized

$$\Phi(Y) = \sum_{i,j} \|y_i - y_j\|^2 \mathbf{W}_{ij}, \tag{2.1}$$

Therefore, the minimization problem reduces to finding

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}^T\mathbf{D}\mathbf{Y}=I} tr(\mathbf{Y}^T\mathbf{L}\mathbf{Y}), \tag{2.2}$$

where $\mathbf{L}$ is the Laplacian of the matrix $\mathbf{W}$, i.e., $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{D}$ is the diagonal matrix defined as $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. As mentioned in [93], the constraint $\mathbf{Y}^T\mathbf{D}\mathbf{Y} = I$ removes an arbitrary scaling factor in the embedding. The solution is provided by the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem $\mathbf{L}y = \lambda\mathbf{D}y$. The embedded M points are stacked in the d vectors.

It was shown in [93] that the laplacian eigen embedding based on Nearest Neighbor graphs preserves local optimality criterion. The local optimality criterion is the key for unfolding the manifolds in high dimensional spaces to be presented in lower dimensional spaces.

2.3.3 Unified View of Dimensionality Reduction Methods

The dimensionality reduction methods that we mentioned in the above subsections can be viewed as instances for a unified dimensionality reduction framework. One of the important studies to relate different methods into a common framework is the study by Ham et al. [54] where a kernel interpretation of KPCA, ISOMAP, LLE, and Laplacian Eigenmap was proposed and it was shown that these methods share a common KPCA formulation with different kernel matrices. The construction of a kernel matrix is equivalent to mapping the data to points in a Hilbert space so that the resulting kernel is positive definite.

Also in [10] a common formulation for the MDS, ISOMAP, LLE, spectral clustering, and Laplacian Eigenmap was proposed with an out-of-sample extension. It was shown that a common algorithm can be used to build a unified framework in which these algorithms are seen as learning eigenfunctions of a kernel.

Another study [142] is more general as it supports both supervised and unsupervised methods in dimensionality reduction in a common framework which is called graph embedding framework. LLE, laplacian eigenmaps,ISOMAP, PCA, KPCA, LDA,LPP are considered as instances of the graph embedding framework.

2.3.4 Large Sale Dimensionality Reduction

The work in [117] examined the problem of extracting a low-dimensional manifold structure given very large sized data points (millions) of high dimensional data. The computational challenges of nonlinear dimensionality reduction via ISOMAP and Laplacian Eigenmaps, using a graph containing millions of points make the problem intractable. The study proposed two approximate spectral decomposition techniques for large dense matrices (Nystrom and Column-sampling), providing a theoretical and empirical comparison between these techniques. The large scale method for dimensionality reduction was examined on Laplacian eigenmaps and ISOMAP and successfully applied on datasets of sizes up to 65 millions face images for classification and clustering tasks.

2.3.5 Applications for Manifold Learning in Object recognition

In computer vision problems the problem of dimensionality reduction should be addressed properly. Usually the data (images) comes in high dimensional vector form. But, suppose there is an underlying lower dimensional structure in the data that controls the relation between images of similar objects from the same viewpoint, or images of same object from different viewpoints or illumination conditions. Manifold learning methods will reveal the underlying manifolds which can lead to better inference and recognition. Here comes the seminal work of Murase and Nayar [91] where it was shown how linear dimensionality reduction using PCA [57] 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 [120] and multi-linear tensor analysis [129]. A way to handle the nonrigid objects is to use landmarks as done in Active Shape Models and Active Appearance Models [28, 27]. The deformation are modeled through linear models of certain

landmarks through a correspondence frame. Thus the ordered sets of landmarks acts as vectorized representation of the images.

The introduction of nonlinear dimensionality reduction techniques such as Local Linear Embedding (LLE) [103], Isometric Feature Mapping (Isomap) [119], and others [119, 103, 8, 16, 65, 136, 88], made it possible to represent complex manifolds in low-dimensional embedding spaces in ways that preserve the manifold topology. Along the same direction manifold learning approaches have been used successfully in many problems such as human body pose estimation and tracking [36, 37, 128, 67].

Chapter 3

Feature-Spatial Embedding Framework

In this chapter we propose a framework to embed bunch of local features that are extracted from different images. The challenge is to encode different sources of similarities among the local features. Within an image, the spatial proximities between the local features plays an important role for describing the shape of an object. In different images, the appearance similarity plays more important role in recognition tasks. Fusing both similarities helps in defining a new representation of the local features that takes into consideration the spatial arrangements of local features within an image and maintains the appearance similarity of local features within different images.

3.1 Problem Statement

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[1]. For example, the feature descriptor can be a SIFT [79], GB [11], 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

[1]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.

$Y^k = \left\{ y_1^k, \cdots, y_{N_k}^k \right\}$ 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 chpater we will give specific examples on some kernels we can 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, diffent object recognition tasks can be performed like object classification, regression and category discovery, etc... .

3.2 Objective Function

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

$$\Phi(Y) = \sum_k \sum_{i,j} \|y_i^k - y_j^k\|^2 \mathbf{S}_{ij}^k + \sum_{p,q} \sum_{i,j} \|y_i^p - y_j^q\|^2 \mathbf{U}_{ij}^{pq}, \tag{3.1}$$

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:

$$\Phi(Y) = \sum_{p,q} \sum_{i,j} \|y_i^p - y_j^q\|^2 \mathbf{A}_{ij}^{pq}, \tag{3.2}$$

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

$$\mathbf{A}_{ij}^{pq} = \begin{cases} \mathbf{S}_{ij}^k & p = q = k \\ \mathbf{U}_{ij}^{pq} & p \neq q \end{cases} \tag{3.3}$$

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. 3.2 reduces to the problem of Laplacian embedding [8] of the point set defined by the weight matrix $\mathbf{A}$. Therefore the objective function reduces to

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}^T \mathbf{D} \mathbf{Y} = I} tr(\mathbf{Y}^T \mathbf{L} \mathbf{Y}), \tag{3.4}$$

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 [8]. 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.

3.3 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 also can use an affine invariant kernel based on subspace invariance [134]. 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 [134] that an affine representation can be achieved by QR decomposition of the projection matrix of $\mathbf{X}$, *i.e.*

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

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 $\mathbf{Q}'$ and thus the produced kernel is affine invariant with regard to the pointset.

3.4 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, which we used in chapter 7 and [122] is a soft correspondence kernel that enforces the exclusion principle based on the Scott and Longuet-Higgins algorithm [109].

Given the feature affinity $\mathbf{G}$ between features in sets p and q, we need to solve for a permutation matrix $\mathbf{C}$ that permutes the rows of $\mathbf{G}$ in order to maximize its trace, i.e.,

$$\psi(\mathbf{C}) = tr(\mathbf{C}^T\mathbf{G})$$

The permutation matrix constraint can be relaxed into an orthonormal matrix constraint on the matrix $\mathbf{C}$. Therefore, the goal is to find an optimal orthonormal matrix $\mathbf{C}^*$ such that

$$\mathbf{C}^* = \arg \max_{s.t. \mathbf{C}^T\mathbf{C}=\mathbf{I}} tr(\mathbf{C}^T\mathbf{G}) \tag{3.5}$$

It was shown in [109] that the optimal solution for 3.5 is

$$\mathbf{C}^* = \mathbf{U}\mathbf{E}\mathbf{V}^T$$

where the SVD decomposition of $\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{E}$ is obtained by replacing the singular values on the diagonal of $\mathbf{\Sigma}$ by ones. The orthonormal matrix $\mathbf{C}^*$ are used as the feature weights $\mathbf{U}^{pq} = \mathbf{U}^{qp^T}$ after setting the negative values to 0.

3.5 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 [10] 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 [140][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 3.1. 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 3.1[3]. *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 3.1, and adding the new constraint, we reach the following optimization problem in $\mathbf{Y}$

[2]We are not using the approach in [140] 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

$$\begin{aligned} &\min \quad tr(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) \\ &s.t. \quad y_i^k = \hat{y}_i^k, i = 1, \cdots, N_k, k = 1, \cdots, K \end{aligned} \tag{3.6}$$

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

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}^\tau & \mathbf{U}^{\nu^T} \\ \mathbf{U}^\nu & \mathbf{S}^\nu \end{pmatrix} \tag{3.7}$$

where $\mathbf{A}^\tau$ is defined in Eq 3.3 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 3.4 is not needed anymore since the equality constraints avoid the degenerate solution.

Unlike the problem in Eq 3.4, which is quadratic programming with quadratic constraints that can be solved by as an eigenvalue problem, the problem in Eq 3.6 is a quadratic programming with linear equality constraints. It was shown in [140] 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.

Let $\mathbf{L}$ be the Laplacian of the matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}^\tau & \mathbf{U}^{\nu^T} \\ \mathbf{U}^\nu & \mathbf{S}^\nu \end{pmatrix}$, which can be rewritten as

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}^\tau & \mathbf{L}^{\nu\tau^T} \\ \mathbf{L}^{\nu\tau} & \mathbf{L}^\nu \end{pmatrix} \tag{3.8}$$

Where τ denotes the training data. The objective function 3.6 can be written as

$$\begin{aligned} &\min \quad tr([\mathbf{Y}^\tau\mathbf{Y}^\nu]^T \begin{pmatrix} \mathbf{L}^\tau & \mathbf{L}^{\nu\tau^T} \\ \mathbf{L}^{\nu\tau} & \mathbf{L}^\nu \end{pmatrix} [\mathbf{Y}^\tau\mathbf{Y}^\nu]) \\ &s.t. \quad y_i^\tau = \hat{y}_i^\tau, i = 1, \cdots, N^\tau \end{aligned} \tag{3.9}$$

The objective function 3.9 can be expanded as

$$\phi(\mathbf{Y}) = \min tr(\mathbf{Y}^{\tau^T}\mathbf{L}^\tau\mathbf{Y}^\tau + \mathbf{Y}^{\tau^T}\mathbf{L}^{\tau\nu}\mathbf{Y}^\nu + \mathbf{Y}^{\nu^T}\mathbf{L}^{\nu\tau}\mathbf{Y}^\tau + \mathbf{Y}^{\nu^T}\mathbf{L}^\nu\mathbf{Y}^\nu) \tag{3.10}$$

The first term is constant since $y_i^\tau = \hat{y}_i^\tau, i = 1, \cdots, N^\tau$. Afterwards we can differentiate $\phi(\mathbf{Y})$ w.r.t $\mathbf{Y}^\nu$ and equate the derivative $\frac{\partial\phi(\mathbf{Y})}{\partial\mathbf{Y}^\nu}$ to zero, then we have

$$2 \times \mathbf{L}^\nu\mathbf{Y}^\nu = -(\mathbf{L}^{\nu\tau} + \mathbf{L}^{\tau\nu^T})\mathbf{Y}^\tau \tag{3.11}$$

Since $\mathbf{L}^{\nu\tau} = \mathbf{L}^{\tau\nu T}$ and given the definition of laplacian $\mathbf{L}$ of $\mathbf{A} = \mathbf{D} - \mathbf{A}$ this implies that $\mathbf{L}^{\nu\tau} = -\mathbf{U}^{\nu}$. This will result in

$$\mathbf{L}^{\nu}\mathbf{Y}^{\nu} = \mathbf{U}^{\nu})\mathbf{Y}^{\tau} \tag{3.12}$$

and hence

$$\mathbf{Y}^{\nu} = (\mathbf{L}^{\nu})^{-1}\mathbf{U}^{\nu}\mathbf{Y}^{\tau} \tag{3.13}$$

3.5.1 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 objective function in Sec 3.2 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 3.4.

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 3.6

Chapter 4

Image Embedding from Local Features

The question that we address in this chapter is how can we learn image manifolds from collections of local features from different images in a smooth way that captures the feature similarity and spatial arrangement variability between images. We benefit from the feature-spatial embedding framework introduced in chapter 3 to build a representation that preserves both the local appearance similarity as well as the spatial structure of the features. We further embedded features from a new image by using the solution we introduced in chapter 3 for the out-of-sample. By solving these two embedding problems and defining a proper similarity measure in the feature embedding space, we can reach an image manifold embedding space.

4.1 From Feature Embedding to Image Manifold Embedding

The embedding achieved in chapter 3 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 distance to measure the distance between two sets of features from two images, define as

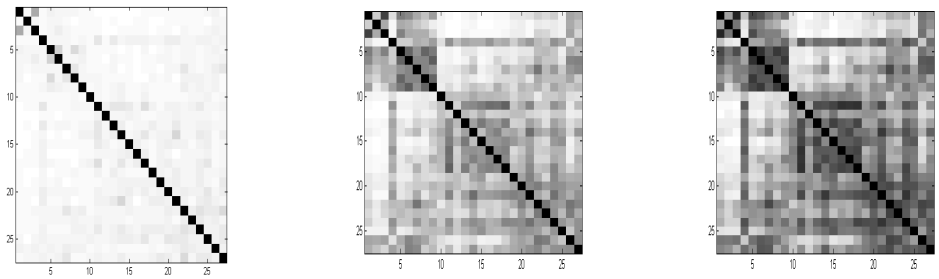
$$H(X^p, X^q) = \max\{\max_j^{l\%} \min_i \|y_i^p - y_j^q\|, \max_i^{l\%} \min_j \|y_i^p - y_j^q\|\} \tag{4.1}$$

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 [30], LLE [103], Laplacian Eigenmaps [8], 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.

Although the percentile-based Hausdorff measure is more robust than the original Hausdorff kernel, the resulting $\mathbf{H}$ is not a positive definite distance matrix, i.e., the $\mathbf{H}$ does have negative eigenvalues, hence the images cannot be assumed to lie in a metric space.



(a) $\mathbf{H}$,l=1 (b) $\mathbf{H}$,l=.5 (c) $\mathbf{H}^+$

Figure 4.1: Different dissimilarity matrices. The $\mathbf{H}$, $l = 1$ distance matrix is far from giving any meaningful clusters in the matrix. For both $\mathbf{H}^+$ and $\mathbf{H}$, $l = .5$ the clusters can be seen on the diagonal (every 9 rows are in the same cluster). However the blocks are more strong on the $\mathbf{H}^+$ distance matrix.

We use $\mathbf{H}$ to compute a positive definite version $\mathbf{H}^+$ that uses the eigenvectors corresponding to the positive eigenvalues. We can obtain an Euclidean distance matrix that represents the percentile-based Hausdorff measure of similarity and hence the images are presented in a metric space which describes the similarity between images in a more sensible way. This way that we follow is called spectrum transformation for a non-metric proximity matrix. The spectrum transformation on the dissimilarity matrix $\mathbf{H}$ works as denoising step [139].

Figure 4.1 shows three dissimilarity matrices. The original Hausdorff distance matrix, i.e. $l = 1$, which is a metric distance, the median Hausdorff kernel, i.e. $l = .5$ and the positive definite version of the median Hausdorff matrix. The third matrix is used to obtain the image embedding in figure 4.4.

4.2 Image Manifold Examples

4.2.1 Visualizing View Manifold

COIL data set [91] has been widely used in holistic recognition approaches where images are represented by vectors [91]. This is a relatively easy data set where object view manifold can be embedded using PCA using the whole image as a vector representation [91]. It has also been used extensively in manifold learning literature, 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 4.2 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.

4.2.2 Shape Classes

We used the "Shape" dataset [114]. 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. To achieve the image embedding we used MDS on the Hausdorff measure. Fig. 4.3 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.

4.2.3 TUD/ETHZ Objects

We use the same dataset that has been used in [60], it has three categories $\{$Motorbikes, Giraffes, Car-side view$\}$ with different sizes 115, 87 and 100 respectively. This dataset is very challenging due to the heavy clutter in the scenes and the multi-instances nature of some images

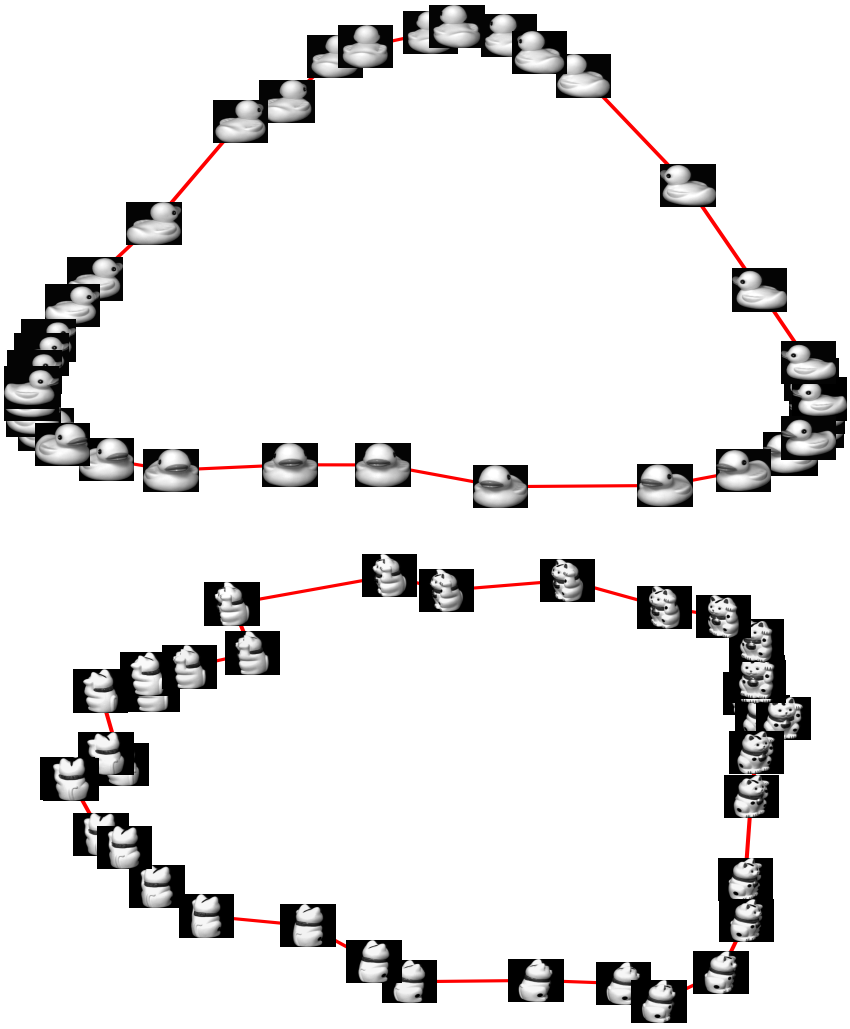


Figure 4.2: Examples of view manifolds learned from local features

Figure 4.3: Manifold Embedding for 60 samples from Shape dataset using 60 GB local features per image

in the Motorbikes and Giraffes classes. We select 27 random samples (9 samples per class) to form an initial feature embedding.

4.2.4 Caltech Subsets

We used different subsets of Caltech-101: Caltech-4-I (faces, airplanes, motorbikes, leopard) as used in [112, 131, 51], Caltech-4-II (faces, airplanes, motorbikes, cars-rear) as used in [41, 56] and Caltech-6 (faces, airplanes, motorbikes, cars-rear, ketch, watches) as used in [41, 56]. 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. To visualize the obtained manifold, we show in Fig. 4.5 the embedded image manifold (first two dimensions) obtained after the initial feature embedding (12 images per class, 60 features per image) for Caltech-4-I. 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,

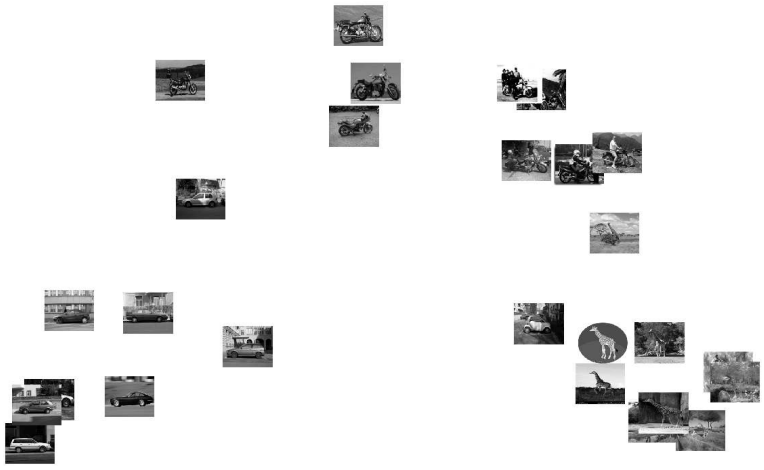


Figure 4.4: Embedding 9 samples from three classes Motorbikes and Car-Side view(TUD) and Giraffes(ETHZ) based on the common feature embedding framework. The clustering is very clear, only one sample is mis-clustered in this example

as can be seen in Fig. 4.5-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. Using the whole data set we can achieve a more comprehensive embedding of all images. This is shown in Fig. 4.6 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 Hausdorff measure (Eq 4.1) 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.

Figure 4.5: 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.

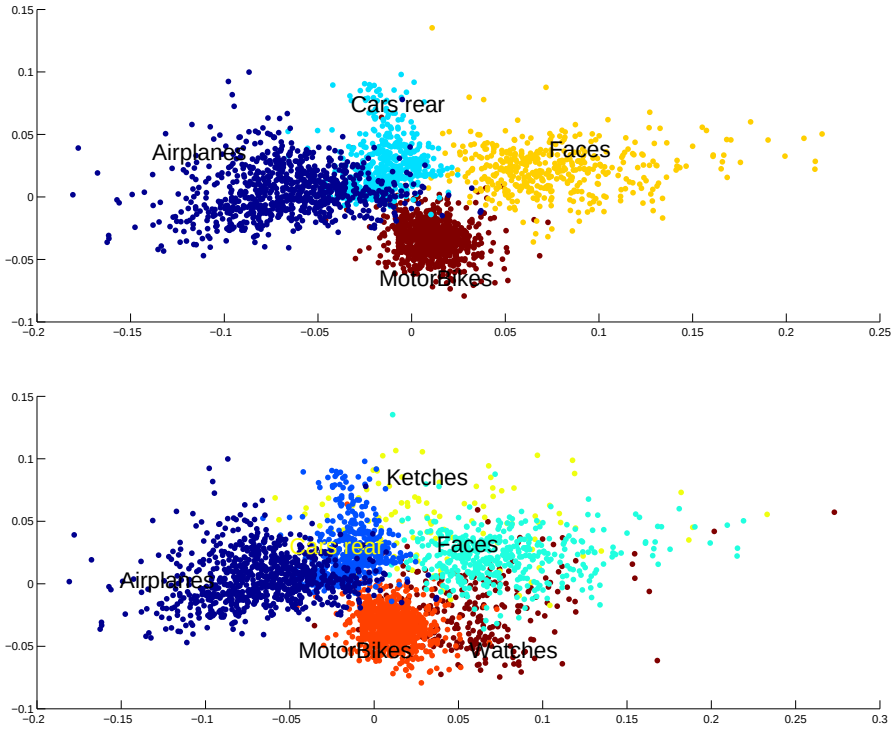


Figure 4.6: Manifold Embedding for all images in Caltech-4-II, Caltech-6. Only first two dimensions are shown.

Chapter 5

Applications: Object Recognition

In chapters 3, 4 we propose a novel representation to learn a common embedding for local features from different object categories. The learned representation takes into account the feature similarities across different image instances and the feature spatial arrangement within each image instance. Such a representation can be used in recognition problems. In this chapter we show different applications including object classification, localization and unsupervised category discovery. We show that we outperform state-of-the-art methods in different object recognition tasks.

5.1 Introduction

In the previous chapters we proposed four components will enable us to solve important problems in object recognition. The four components are a feature embedding space, an image similarity measure induced by this space, an out-of-sample solution, and an image manifold embedding space. In this chapter we show several results obtained for recognition problems including object classification, object localization and unsupervised category discovery. We show comparisons to state-of-art-methods with clear improvements.

5.2 Results: Object Classification

In all experiments we used the Geometric Blur features (GB) [11]. It was shown in [114] 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.

	training/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 5.1: Shape dataset: Average accuracy for different classifier setting based on the proposed representation

5.2.1 Shape Dataset

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. 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 5.1 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 [114] 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 [79], GB [11], and KAS [42] features. Table 5.2 shows the reported accuracy in [114] for comparison. All reported results are based on 2:1 ratio for training/testing split. Unlike [114] 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.

Accuracy %			
Feature used	SIFT	GB	KAS
Our approach	-	91.27	-
bag of words (reported by [114])	75	69	65
Localized bag of words ([114])	88	86	85

Table 5.2: Shape dataset: Comparison with reported results

5.2.2 Caltech 101

The recognition accuracy of the proposed approach was evaluated using subsets of the Caltech-101 dataset [75]. 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 [112, 131, 51], Caltech-4-II (faces, airplanes, motorbikes, cars-rear) as used in [41, 56], Caltech-6 (faces, airplanes, motorbikes, cars-rear, ketch, watches) as used in [41, 56]. 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 Hausdorff measure (Eq. 4.1). Table 5.3 shows the recognition accuracy using different number of training data and three different classifiers: 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. 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 5.1). This is also expected since, unlike “Shape” dataset, Caltech101 dataset contains lots of clutter besides the objects.

		# training 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 5.3: 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.

5.3 Results: Object Localization

The goal of this experiment is to evaluate the robustness of the proposed approach to clutter in the context of object localization. Many approaches that encode feature geometry are based on a bounding box, e.g. [114, 50]. 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.) Table 5.4 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 [60]), and Bounding Box Miss Ratio (BBMR).

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 5.4: Object localization results - Caltech101-4

Categories	Our Approach $\mathbf{H}^{+}$	Our Approach $\mathbf{H}$	Baseline	[60]	[68]	[51]	Baseline [68]
Caltech-4	**99.54**$\pm$0.31	98.83	96.43	98.55	98.03	86	87.37
Caltech-5	**98.59**$\pm$0.47	94.32	96.28	97.30	96.92	NA	83.78
Caltech-6	**97.48**$\pm$0.57	93.57	94.03	95.42	96.15	NA	83.53

Table 5.5: Caltech-4,5 and 6: Average clustering accuracy, best results are shown in bold.

5.4 Results: Unsupervised Object Categorization

5.4.1 Equal Cardinality -Caltech

In this experiment we follow the setup by [51, 60, 68] 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. In all experiments we used GB features [11]. The input of our algorithm is a set of M unlabeled images with the number of object categories C. 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.

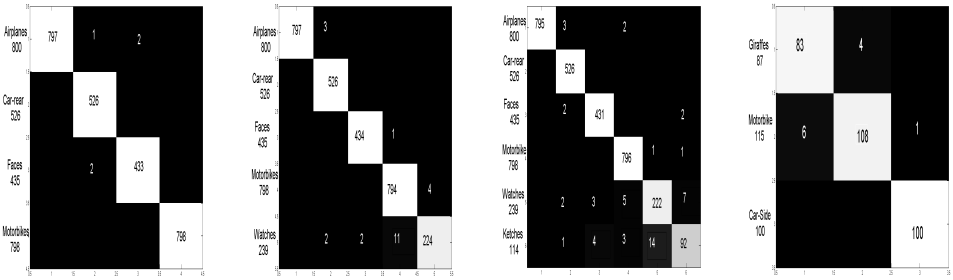
We randomly select $12 \times C$ random samples to form an initial embedding that is used to generate initially the common feature embedding of all features. We select 120 features per image for initial embedding and we out-of-sample 420 features (at the most) per image. This results in a common feature embedding that has $100C \times 420$ features. We chose dimensionality of the common feature embedding = 120. Table 5.5 shows comparative evaluation, the state of the art results in [60, 68]. 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 [60, 68] can be outperformed by using a

common feature embedding space that encodes the spatial proximity and appearance similarity in same time, which is done without an explicit matching step.

An interesting part of the results is that the baseline is giving very nice results when compared to the baseline reported in [68] about 10% difference (last column). The baseline in [68] is also using similarity based on the appearance only. This means adding the positive definiteness to the $\mathbf{H}_{descriptor}$ has a great impact on the unsupervised category discovery problem.

5.4.2 Different Cardinality -Caltech

In the previous setup, only 100 images per class were randomly chosen. However, the whole collections of Caltech-4,5 and 6 are more challenging due to the large sub clusters problem. The larger the class the higher the probability to find sub clusters, these sub clusters sizes are very much comparable to the sizes of the small classes. For example the motorbikes set is having 798 images, while the ketches are just 114 images. Thus the sub clusters in the motorbikes are very reasonable candidates for a clustering algorithm like NCUT[1].



(a) Caltech-4 (b) Caltech-5 (c) Caltech-6 (d) TUD/ETHZ-3

Figure 5.1: Confusion Matrices for different setups using the whole data at once

We use same selection for the parameters as in the previous settings to compute the initial embedding. For NCUT we use 8-NN for Caltech-4 , 16-NN for Caltech-5 and 24-NN for Caltech-6. In figure 5.1 we show the confusion matrices for the three cases. We achieve accuracy 99.80$\%$ for Caltech-4 , this means only 5 samples of the whole 2559 images are

[1]Normalized cut prefers balanced clustering.

mis-clustered. For Caltech-5 we achieve 99.18% accuracy, for Caltech-6 we acheive 98.28% accuarcy. We used the $\mathbf{H}^+$ to define the distance matrix of all the images.

5.4.3 Different Cardinality TUD/ETHZ

We use the same dataset that has been used in [60], it has three categories {Motorbikes, Giraffes, Car-side view} with different sizes 115, 87 and 100 respectively. This dataset is very challenging due to the heavy clutter in the scenes and the multi-instances nature of some images in the Motorbikes and Giraffes classes. We select $9 \times C$ random samples to form an initial feature embedding. We select 160^2 features per image for initial embedding and we out-of-sample 560 features (at the most) per image. We chose dimensionality of the common feature embedding = 100.

Again our results are better than the reported results in [60]. The accuracy in this experiment is 96.36% while in [60] the average accuracy was 95.47%, this means only 11 samples of the 302 samples were mis-clustered. For NCUT we use 8-NN weighted graph of the $\mathbf{H}^+$.

[2] We increased the number of features per image since the image resolution in this dataset is higher than in Caltech subsets.

Chapter 6

Regression From Local Features

In this chapter we propose a framework for learning a regression function form a set of local features in an image. The regression is learned from an embedded representation that reflects the local features and their spatial arrangement as well as enforces supervised manifold constraints on the data. We applied the approach for viewpoint estimation on a Multiview car dataset, a head pose dataset and arm posture dataset. The experimental results show that this approach has superior results to the state-of-the-art approaches in very challenging datasets .

6.1 Introduction

Many problems in computer vision can be formulated as regression problems where the goal is to learn a continuous real-valued function from visual inputs. For example, viewpoint estimation of an object, head pose estimation, age estimation from faces, estimating illumination direction, articulated object joint angles, limb position, etc. In many of these applications, the regression is learned from a vectorized representation of the input. For example, in head pose estimation, researchers typically learn regression from vectorized representation of the raw image intensity, e.g., [133, 6, 46, 92, 53].

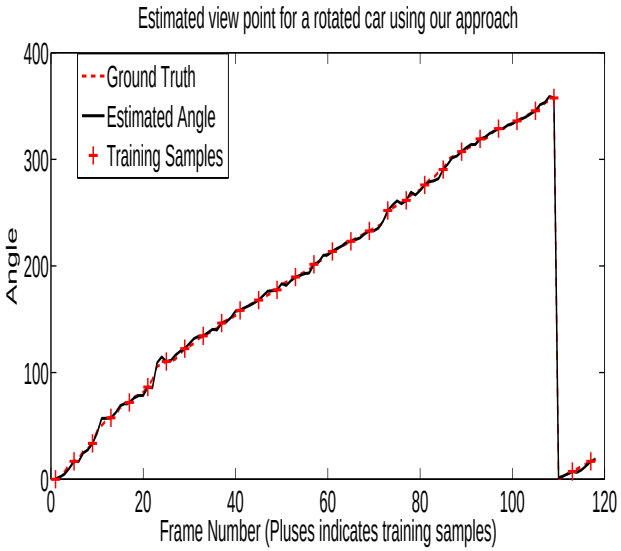
In the last decade, there have been a tremendous interest in recognition from highly discriminative local features such as SIFT [79], Geometric Blur [11], etc. Most research on generic object recognition from local features have focused on recognizing object from a single viewpoint or from limited viewpoints, e.g., frontal cars, side view cars, rear cars, *etc*. Very recently, there have been some interest on object classification from multi-view setting [23, 63, 106, 105, 76, 115]. There have been also some promising results on pose recovery (3D viewpoint estimation) from local features for generic object class [106, 105, 76, 115]. The problem of object classification from multi-view setting and pose recovery are coined together.

Pose (viewpoint) recovery is a fundamental problem that has been long studied for rigid objects with no within class variability [45]. A very challenging task is to solve for the pose for a generic object class, e.g. , recovering the pose of a chair instance that was never seen before in training. Most of recent work on viewpoint estimation from local features are based on formulating the problem as a classification problem [105, 106, 76, 116, 121, 115] where the viewpoint is discretized into a few number of bins, 4, 8, or 16 and a classifier is used to decide about the viewpoint. Obviously, the accuracy of such classifiers is limited by how coarse the viewpoint is discretized. Such treatment does not allow for continuous estimation of the viewpoint and can not interpolate between the learned views.

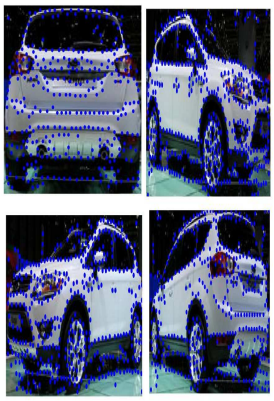
Viewpoint estimation is fundamentally a continuous regression problem, where the goal is to learn a regression function from the input. Similar are other problems such as posture estimation. *The question we address in this chapter is how to learn a regression function from local features: their descriptors and their spatial arrangement.*

Local features are designed to have some geometric invariant properties. For example, SIFT [79] is view invariant. From two close viewpoints, we expect to see the same local features. Such local features can be useful in viewpoint estimation only if we consider apart views. If our goal is to accurately recover the viewpoint, local features' descriptors only are not enough. It is obvious that the spatial arrangement of feature will play a more important role in this case. Recent work have addressed this though encoding the spatial information through a pyramid spatial subdivision [95], or through enforcing geometric constraints at test time [76]. Relative distances between parts have also been used [105, 105]

In this chapter we introduce an approach for learning a regression function from local features. The approach is inspired by the feature embedding approach introduced in chapter 3 where we have shown that an embedded representation that encodes both the features' descriptor and their spatial arrangement can be achieved. In this chapter we show how such an embedding can be used to achieve regression from local features that takes into consideration the feature descriptor and the spatial feature arrangement. The regression is achieved by defining a proper kernel in the embedding space. We show how a supervised manifold constraints can be enforced in the embedding. For example, for viewpoint estimation, we can enforce that the viewpoint to lie on a one dimensional manifold. In the resulting embedding space, image



(a) Estimated Angle and Ground Truth



(b) Sample Views

Figure 6.1: Regression on a single car: (Left) Absolute Error computed using our approach is plotted with the ground truth, they are very close to each other. (Right) sample views of the car with features detected on it.

similarity can be measured in a way that reflect smooth changes in the functions to be learned, e.g. the smooth changes in viewpoint. Therefore, we can learn a regression function from local features that can accurately estimate viewpoint from a small number of training example and a small number of features. The experimental results show that this approach has superior results to the state-of-the-art approaches in very challenging datasets (e.g. in a challenging multi-view car data set we have 67% improvement over [95]).

Figure 6.1 shows an example of our results in estimating the viewpoint of a car from local features. In this example we used 30 instances for learning, around 12° apart, with 200 local features, with no correspondences established. The regression function can estimate the viewpoint with less than two degrees error.

6.2 Kernel-based Regression from Local Features:

6.2.1 Kernel Regression Framework

The training data is a set of input images, each represented with a set of features. Let us denote the input images (sets of features) by $X^1, X^2, \cdots X^K$, where each image is represented by $X^k = \{(x_i^k \in \mathbb{R}^2, f_i^k \in \mathbb{R}^{\mathcal{F}})\}, i = 1, \cdots, N_k$. Here x_i^k denotes the feature spatial location and

f_i^k is the feature descriptor and $\mathcal{F}$ denotes the feature descriptor dimension. For example, the feature descriptor can be a SIFT, HOG, 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., $N = \sum_{k=1}^{K} N_k$.

Each input image is associated with a real-value, $v^k \in \mathbb{R}$, for example, v^k can be the angle representing the viewpoint, or the head pose of the k-th image. Therefore, the input is pairs in the form (X^k, v^k). For simplicity, here we show how regression can be done to real numbers, extension to real-valued vectors is straight forward. Extension to real-valued vectors is necessary for problems like articulated posture estimation where joint angles are estimated.

The goal is to learn a regularized mapping function $g : 2^{\mathbb{R}^2 \times \mathbb{R}^{\mathcal{F}}} \rightarrow \mathbb{R}$. Notice that unlike traditional regression, the input to such a function here is a set of features from an image with any number of features. This function should minimize a regularized risk criteria, which can be defined as

$$\sum_k \|g(X^k) - v^k\| + \lambda \Phi[g] \tag{6.1}$$

where the first term measured the error in the approximation, the second term is a smoothness function on g for regularization, and λ is a regularization parameter. From the representer theorem [61] we know that such a regularized regression function admits a representation in the form of linear combination of kernels around the training data points (or a subset of them). Therefore, we seek a regression in the form

$$v = \hat{g}(X) = \sum_j b_j K(X, X^j) \tag{6.2}$$

Therefore, it is suffice to define a suitable positive definite kernel $K(\cdot, \cdot)$ that measures the similarity between images. Once such kernel is defined we can solve the coefficients b_j by solving a system of linear equations [100].

6.2.2 Enforcing Manifold Locality Constraint

To achieve a smooth image similarity kernel from local features, we learn an embedded representation of the features and their spatial arrangement, as was described in chapter 3. 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 seek 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 approach as described in chapter 3 satisfies two constrains: Inter-image feature affinity and Intra-image spatial structure. Besides these two constraints, we need to add a third constraint that enforces manifold locality, we denote that by *Supervised Manifold Locality Constraint*. The idea is to enforce existing manifold structure in the data, features from images neighboring each other on the manifold should be embedded close to each other. For example, if images are labeled with viewpoints, such label can be used to define a neighborhood for each image. Since we are using the labels to define the neighborhood, this is a supervised enforcement of data manifold constraint. Enforcing manifold constraints have been shown to highly improve regression results in many applications [6, 102, 133, 53]. However all these applications used vectorized representations of the raw intensity.

We can enforce the manifold constraint in a supervised way from the labels v^k. This can be achieved by amending the objective function in chapter 3 Eq. 3.1 by supervised weights between images as

$$\Phi(Y) = \sum_k \sum_{i,j} \|y_i^k - y_j^k\|^2 \mathbf{S}_{ij}^k + \lambda \sum_{p,q} \sum_{i,j} \|y_i^p - y_j^q\|^2 w(p,q) \mathbf{U}_{ij}^{pq}, \tag{6.3}$$

where $w(p,q)$ denotes a weight function that measure the supervised affinity between images X^p and X^q as implied by their labels v^p and v^q. There are many way to define such weights. If we set all the weights to one, we reduce to an unsupervised embedding as in chapter 3 Eq. 3.1. The weights can be set to reflect labels distances, i.e., $w(p,q) = \mathcal{G}(v^p - v^q)$. For example a Gaussian function can be used or alternatively, the weights can be set to reflect neighborhood structure by using a uniform window kernel. Therefore the matrix $\mathbf{A}$ can be redefined as

$$\mathbf{A}_{ij}^{pq} = \begin{cases} \mathbf{S}_{ij}^k & p = q = k \\ \mathcal{G}(v^p - v^q) \cdot \mathbf{U}_{ij}^{pq} & p \neq q \end{cases} \tag{6.4}$$

6.2.3 Feature Embedding based Regression

Since each image is represented in the embedding space by a set of Euclidean coordinates in that space, the similarity in the embedding space can be measured by a suitable set kernel that

measure the distance between two sets of embedded features representing two images. There are a variety of similarity measures that can be used. For robustness, we use a percentile-based Hausdorff distance to measure the distance between two sets of features from two images in the embedding space, defined as

$$H_l(X^p, X^q) = \max\{\max_j^{l\%} \min_i \|y_i^p - y_j^q\|, \max_i^{l\%} \min_j \|y_i^p - y_j^q\|\} \tag{6.5}$$

where l is the percentile used. 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 not a metric and therefore does not guarantee a positive semi-definite kernel. Therefore we use this measure to compute a positive definite matrix $\mathbf{H}^+$ by computing the eigenvectors corresponding to the positive eigenvalues of the original $\mathbf{H}_{pq} = H_l(X^p, X^q)$. The regression problem now can be solved by using kernels based on matrix $\mathbf{H}^+$ in the embedding space, e.g., Radial Basis Function (RBF) kernels are used. Therefore, we can solve for the regression parameter in Eq. 6.2.

Given the learned regression function, it can be applied to any new image. However, the features in that new image has to be mapped first to the embedding space. Therefore, the regressor for a new test image X will be in the form

$$v = \hat{g}(X) = \sum_j b_j K(\mathcal{O}(X), Y^j) \tag{6.6}$$

where $\mathcal{O}(X)$ is a function that maps the features in a test image X into a set of coordinates in the embedding space, i.e.,

$$\mathcal{O}(X) : \{(x_i, f_i)\} \longrightarrow \{y_i\}$$

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 $\mathbf{L}^\nu$, $\mathbf{U}^\nu$

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

where $\mathbf{Y}^\tau$ is an $N \times d$ matrix stacking of the embedding coordinate of the training features.

6.2.4 Image Manifold-based regression:

The regression can be also learned from an image manifold embedding space, which can be obtained using the similarity kernel defined on the feature embedding space. This is a second embedding where each image is represented by a single point in a Euclidean space. However the problem with this approach is that for any test image two out of sample problems have to be solved: First, out of sample on the features should be used to map them to the feature embedding space. Second, the embedded set of features has to be used to achieve the image coordinate in the image embedding space using a second out of sample. The advantage of learning a regressor from the image embedding space is that enforcing manifold constraints on the images can be easier in that space. However, a two stage embedding and two out of sample problems disencourages this approach.

6.3 Experiments

6.3.1 Regression on a single car example

We use a single car sequence (first car) from the dataset introduced by [95] to demonstrate the different setups for our approach and to show the effect of the different parameters. The sequence contains 118 views of a rotating car. We changed the following parameters: 1) The number of training samples to learn the feature embedding, which are also used as RBF centers: 15, **30**, and 40. 2) The dimensionality of the embedding space: 20, 40, 80, **100**, 160 and 200. 3) Manifold supervision neighborhood size: **30**°, 45°, 60° and ∞, where ∞ means unsupervised embedding. We change one parameter at a time while we fix all other parameters with a default value (shown in bold above) . In all experiments we fix the RBF scale to 0.05 of the median Hausdorff distance in the data. We measure the mean and standard deviation of the absolute error (MAE, std(AE)), between the estimated and the ground truth viewpoints. Table 6.1 shows the obtained results for various settings. Fig 6.1 shows the estimated and ground truth angles for the default base case: 30 training samples, 100 dimensions, 30° neighborhood. The MAE in this case is 1.94°. From the table we can see that, in general, the accuracy in the regression does not change much with the change in the parameters. We can see that when the number of training samples increased from 15 to 30 the mean absolute error dropped to half of its value,

Train	Supervised	Dim	MAE°	std(AE)
30	Yes/30°	20	2.34	1.99
30	Yes/30°	40	2.06	1.65
30	Yes/30°	80	2.04	1.64
30	Yes/30°	100	1.94	1.63
30	Yes/30°	160	1.93	1.63
30	Yes/30°	200	1.95	1.59
15	Yes/30°	100	5.47	4.21
30	Yes/30°	100	1.94	1.63
40	Yes/30°	100	1.84	1.66
30	Yes/45°	100	1.94	1.5
30	Yes/60°	100	2.09	1.66
30	No/∞	100	2.16	1.83

Table 6.1: Regression on a single car

increasing the training size after that does not change the accuracy much. Also we can see that the dimensionality of the embedding space is insignificantly affecting MAE. Notice that the there is an error in the ground truth itself of the same order as the error in the estimation. So, this experiments basically shows that we can achieve accurate regression on a single object from local features from a small number of sparse training samples. In the next experiment we show results on the whole dataset.

6.3.2 Multi-View Car Dataset

In this experiment we use 'Multi-View Car Dataset' that was introduced recently in [95] which captures 20 rotating cares in an auto show. The dataset is very challenging as the cars are accompanied with much clutter even within the detected bounding boxes. It has large class variation in appearance, shape, and texture of the cars in this dataset. We use this data set since it provides finely discretizes viewpoint ground truth, the discretization varies in each car sequence. Such ground truth facilitates evaluation of the accuracy of our regression approach. Other datasets like PASCAL VOC 2006 gives only 4 viewpoint class labels {'Front', 'Back', 'Left', 'Right'} and the dataset that was used in [105, 106, 116] only has 8 viewpoints classes. Moreover, it is really hard to find a dataset with ground truth that covers the whole range of viewpoint with realistic challenging conditions. However, there are some drawbacks and

challenges in this dataset: 1) The high within-class variation makes it hard for a regressor or classifier to generalize. 2) Ground truth accuracy problems: The viewpoint is calculated using the time of capturing assuming a constant velocity, which affects on the ground truth. There are some frames of the same car that are having same time of capturing but there is slight change in the pose and in few frames the cars are partially occluded by passing people. 3) Some cars are highly symmetric from a side view, that makes classifiers subject to 180° reflection error in some views. Such reflection error exist in other datasets as well and reported in the results of [95, 105, 116]. 4) Some cars are very odd, and even visually it is very hard to discriminate between whether the car front or rear is facing the camera.

The dataset has been used for viewpoint classification in [95] where the viewpoint was discretized into 16 bins. In [95] their goal was to classify the car pose using a bag-of-words technique that is based on a spatial pyramid of histograms. They build 16 SVM classifiers for the 16 bins to cover the 360 range of rotation (i.e., bin size is 22.5 °). We use the results of [95] as a baseline since it incorporates both the features and their spatial arrangement through the spatial pyramid structure. *The approach proposed in [95] resulted in 41.69% viewpoint classification accuracy from bounding box input. In contrast, given a similar 16 bin setting, our approach results in 70% accuracy using the same bounding box as inputs, that is over 67% improvement over the state of the art result.*

In our regression experiment, we use the same split of training and test sets as [95]. The dataset contains 20 sequences for 20 rotating cars. The total number of images is 2137, the first ten cars are used for training (1103 images) and the last ten cars for testing (1034 images). We used only 135 images (sampled randomly from 4 sequences of train data) to learn an initial feature embedding. Each image is represented using 50 geometric blur local feature descriptor [11]. The initial feature embedding is then expanded using out-of-sampling to include all the training images with maximum of 350 features per images (the number of features extracted per image varies).

We learn our regression model using Radial Basis Functions (RBF) as described in section 6.2. We examine the effect of "supervision", i.e., enforcing the view manifold constraint on the initial embedding by defining a neighborhood for each image not to exceed 45° difference. For quantitative evaluation, we use the Mean Absolute Error (MAE) between the estimated and

Method	MAE 90% percentile	MAE 95% percentile	MAE	AE<22.5	AE<45
Results [95] (Baseline)	–	–	46.48	41.69%	71.2%
Unsupervised(RBF)	27.17	32.65	39.2	50.09%	73.6%
Unsupervised(RBF)Leave One Out	22.57	27.12	35.87	63.73%	76.84%
Supervised (RBF)	19.4	26.7	33.98	70.31%	80.75%
supervised (RBF)Leave One Out	23.13	26.85	34.9	55.83%	76.65%
Unsupervised(SVR)	29.52	34.44	40.60	41.19%	70.12%
Supervised (SVR)	25.23	30.63	36.07	57.9%	78.6%

Table 6.2: Regression on Multi-View car dataset, baseline and different variants of our approach

ground truth viewpoint. In addition we also used the MAE of 90% percentile of the absoulte errors and the 95% percentile of the absoulte errors. These are used because, typically, a very small percentage of the test data produces very large error (180°) due to reflection, which biases the MAE. While MAE is a good measure for validating regression accuracy, it is not suitable for comparison with classification-based viewpoint estimation approaches which uses discrete bins, such as [95, 105, 116]. Therefore, we also used the estimated viewpoint to compute the error in discritized viewpoint classifier. For example, to achieve an equivalent of a 16 bin viewpoint classifier, we compute the percentage of test samples that satisfies $AE \leq 22.5$, where the absolute error $AE = |Est.Angle - GroundTruth|$. With this measure we can compare to 16 bin classifier used in [95]. To achieve an equivalent of an 8 bin viewpoint classifier, we also compute the percentage of test samples that satisfies $AE \leq 45$.

For comparative evaluation, we evaluate different supervised and unsupervised setting within our framework as described in chapter 3, in addition we used the results from [95] as a baseline. We also evaluated a support vector regressor (SVR) based on our framework. For each setting we evaluated the 10/10 split as described above and also a leave one out split (learn on 19 cars and test on 1).

We show our results in table 6.2, we might find the following observations:
– The MAE is ranging from 33.9° to 40.60° which seems to be a large error. However, this might be misleading because if we compare reported results like in [116] in which they learn classifiers for 8 bins, the reported average accuracy on the diagonal of the confusion matrix is 66%. In this case this means only 66% of the testing set is recovered within the bins and any error adds at least 45 $^\circ$. In the last two columns of table 6.2 show that around 65% of

testing samples are giving $AE < 22.5^\circ$ and around 80% or more are giving $AE < 45^\circ$. The source of the higher MAE is then coming from few instances with large reflection errors (around 180 degrees), this also clear in the percentiles MAEs. Comparing our results to the reported confusion matrices in [95, 116][1] we can find that our approach has a lower reflection effect in the estimated angles. In figure 6.3 a, only few test samples lies in the last bin of the histogram.

– As we can see the supervised setting is giving the best results for this dataset. This confirms that enforcing the neighborhood constraint on the manifold is in fact boosting the regression results.

– Also we can observe that using the leave one out settings for regression is not improving beyond few degrees over the split settings. This means that our approach is generalizing well so that it does not gain much by including as many training samples.

Figure 6.2: Regression on a Multi-view car dataset: Top left corner shows how the arrows reflect he estimated angle. The ground truth is shown along with the estimated angle. Yellow arrows for ground truth and Magenta for our results, features are shown as blue dots(Best viewed in color)

[1]The confusion matrix in [95] was shown without the actual numbers in it, but after we contacted the authors of [95] they sent us the actual values in their confusion matrix.

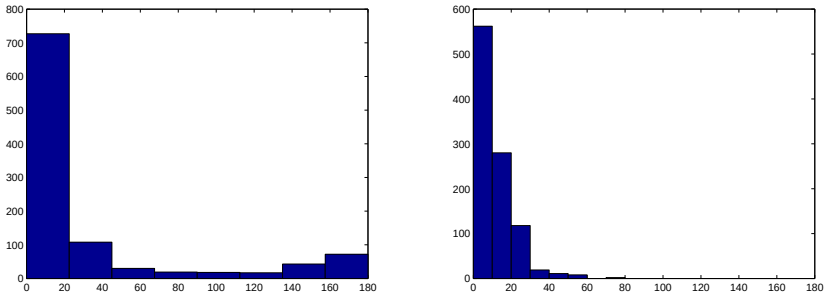


Figure 6.3: Histogram of absolute error: Left: for Multi view car datset. Right: for face dataset.

6.3.3 Face Pose Estimation in Uncontrolled Environment

In this experiment we used 'Face Pose' dataset that was introduced recently in [3, 4]. It has been used in inferring the face pose of freely downloaded faces from the web. The pose ranges from -90° to 90 °, the ground truth is manually labeled for 11900 images, 10900 of them were used for training and 1000 for the testing. The images that were used in [3] experiments are 60x60 bounding boxes that were normalized using a Euclidean warp. The dataset is a real world challenging set which exhibits much variation in controlling factors like illumination, scale, expression and pose as well as partial occlusion and background clutter. However, we want to mention the drawbacks of the dataset. First the distribution of the pose degrees is very biased and only few examples are beyond the range [-50°,50°] which affects the regression we learn. Second as mentioned in [3] the manual labeling is not so accurate since four subjects were asked to label every image and the pose is then averaged. The correlation of the manually labeled poses between different subjects was $\approx .75$ [3].

In our regression experiment, we use the same training set and same test set as [3], and we compare our results in terms of the MAE and Pearson Correlation Coefficient (PCC) as they provided in [3]. We used 250 images (sampled randomly from train data) to learn the initial feature embedding of dimensionality 50 for each feature. Each image is represented using 24 geometric blur local feature descriptors. The initial feature embedding is then expanded using out of sample to include all the features from training images with maximum of 72 features per images (the number of features extracted per image is not equal). The dimension of images is the reason for fewer extracted features per image when compared to the cars dataset.

We learn our regression model as we did in the cars dataset. We examine the effect of supervision on the initial embedding by defining a neighborhood for each image not to exceed 15 ° difference. The histogram of absolute error in figure 6.3 show that in around 86% of the case the estimated error is less than 20 °.

We achieved an MAE error of 10.92° and 11.15° for the unsupervised and supervised cases respectively and PCC of .81 and .79 respectively. In [3] the reported results is MAE=13.21 and PCC=.76. We have better MAE for both the supervised and unsupervised settings. This shows that from sparse local features we can achieve better results in regression in this example. The most noticeable point is that the unsupervised is behaving better than the supervised setting. Although this might seem strange, but the distribution of poses of the training samples and the testing samples is very biased towards the region [-50°,50°] and actually in the 10900 training samples there is not a single image with pose in the interval [-80°,-90°], Under this condition enforcing the neighborhood in the region that have few samples in the training will result in a poor generalization. We show in figure 6.3 the histogram of absolute error it shows high accuracy of the estimating the face pose using our framework.

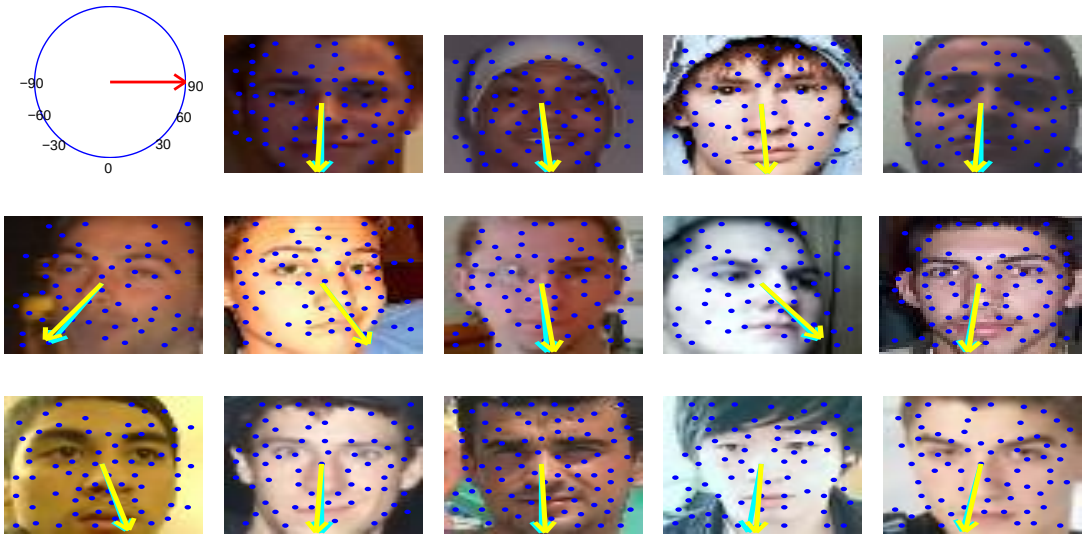


Figure 6.4: Regression on a Face Pose estimation dataset: Top left corner shows how the arrows reflect he estimated angle. The ground truth is shown along with the estimated angle. Green arrows for ground truth and Yellow for our results, features are shown as blue dots(Best viewed in color)

6.3.4 Arm Posture Estimation

As we mentioned earlier, our approach is general and can be used in different regression problems, not only viewpoint estimation. We show here articulated body posture estimation for a subject who moves his arms freely. We used the sequence from [102]. The local features are affected very much by the clutter. The ratio of features on the hands to the extracted features is about 10%, all the features in each frame are used in the regression. The sequence contains 200 frames, 25 equally spaced are chosen for training (12.5% of the sequence). Initial Embedding: 150 features from 20 training frames, dimensionality 250. We then compute out of sample embedding for all 25 training frames, each with 450 features. Then we learn the regressor parameters for the hands and elbows joints positions from the 25 training frames. The regressor was used to estimate the position of the hands and elbows joints in the rest of the frames. We evaluated the estimation using 75 frames marked with ground truth and the error is 18 pixel in average per estimated parameter (image size is 640x480). Sample results are shown in the figure.

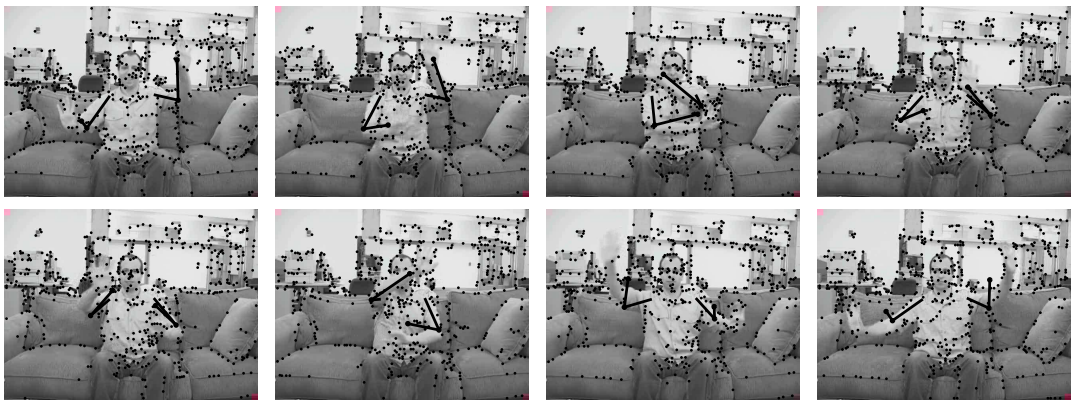


Figure 6.5: Regression example for articulated body posture estimation: shown are frames 20,40,60,80,100,120,140,160

Chapter 7

Multi-Set Feature-Spatial Matching

We introduce a novel framework for multi-set feature matching among multiple sets in a way that take into consideration both the feature descriptor and the features spatial arrangement. As introduced in chapter 3 we can learn an embedded representation that combines both the descriptor similarity and the spatial arrangement in a unified Euclidean embedding space. The solution can be obtained directly by solving one Eigenvalue problem which is linear in the number of features. Therefore, the framework is very efficient and can scale up to handle a large number of features. The matching step is taking place after the feature-spatial embedding which ensures that the resulting feature embedding preserves within image spatial structure and in same time it preserves the feature similarity between different images. Experimental evaluation is done using different sets showing outstanding results compared to the state of the art; up to 100% accuracy is achieved in the case of the well known Hotel sequence.

7.1 Introduction

Finding correspondences between features in different images plays an important role in many computer vision tasks. Several robust and optimal approaches have been developed for finding consistent matches for rigid objects by exploiting a prior geometric constraint [126]. 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 [79]). 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 [31]. This is also motivated by

avoiding the high complexity of solving for spatially consistent matches.

The problem we address in this chapter is how to find 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. [73, 29, 20, 125]). Typically 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 [20]. Even though some efficient solutions (e.g. linear complexity in the problem description length) have been proposed for such a problem [29] 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[1].

Besides this scalability limitation, most of the state of the art algorithms for matching can only match two sets of points. They do not generalize to match multiple sets of features.

In this chapter, we introduce a framework for feature matching among multiple sets of features in one shot, where both the feature similarity in the descriptor space, as well as the local spatial geometry are enforced. *This formulation brings three achievements to the problem:*
1) Graph Matching through Embedding: We formulate the problem of consistent matching as an embedding problem 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 an objective function enforcing both the feature similarity and the spatial arrangement. Such embedding space acts as a new

[1]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.

unified feature space (encoding both the descriptor and spatial constraints) where the matching can be easily solved. The framework is illustrated in Fig 7.1.

2) Matching Multiple sets in one shot: The proposed framework directly generalizes to matching multiple sets of features in one shot through solving one Eigenvalue problem. Consistent matching of multiple sets of features is a fundamental problem, for which very few solutions have been proposed.

3) Scalability: An interesting point in 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 the proposed 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 Eigenvalue problem *which size is linear in the number of features in all images*.

Extensive evaluation on several standard datasets shows that the proposed approach gives better or comparable results to the state of the art algorithms [73, 29, 19, 125] that uses quadratic assignment. In fact, we achieve 100% correct matching on a standard benchmark with our multiset setting. The experiment results also show that the proposed approach can find consistent matching under wide range of variability including: 3D-motion, viewpoint change, rotation, zooming, blurring, articulation and nonrigid deformation.

7.2 Related Work

7.2.1 Matching Under Geometric Constraints

Geometric matching techniques such as RANSAC [44], interpretation trees [52], Hough transform [7], or alignment [126] can be used to efficiently explore consistent correspondence hypotheses when the mapping between image features is assumed to have some parametric form (e.g., a planar affine transformation), or obey some parametric constraints (e.g., epipolar ones). These methods work well for rigid transformations. However, these methods cannot be easily extended to the case of non-rigid transformations where the number of transformation parameters often scales with the cardinality of the data set [25].

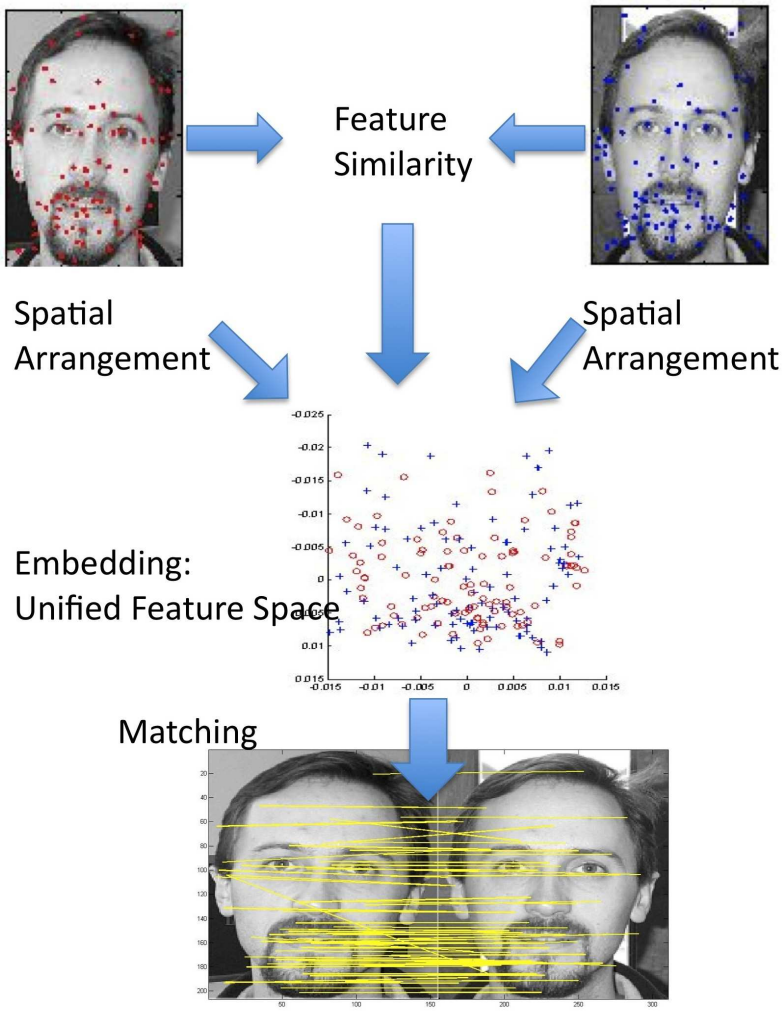


Figure 7.1: Motivating Example on two faces

7.2.2 Shape Vs. Appearance Based Matching Approaches

Depending on the application, matching algorithms are either using the appearance or the shape (arrangement) of the feature points to decide matches. Appearance-based matching, e.g. [34, 79, 86], requires a good descriptor that is invariant under different viewing condition. In such case, the matching is done in the descriptor space. Comparative studies like [86, 89] recommended SIFT [79] based descriptors for the task of feature matching. On the other hand, shape matching algorithms are desired for recognition tasks, e.g., [9, 49, 111, 132]. Such algorithms use the spatial location of the feature points or descriptors derived from these locations.

7.2.3 Spectral Correspondences as Graph Matching

Spectral methods [109, 111, 132, 62, 127, 73, 29, 34, 141, 118] has been widely used for the problem of feature matching. All mentioned approaches solve a graph matching problem to compute correspondences. The definition of the graph matching problem varies.

The feature matching problem can be casted as a **BiPartite Graph Matching** [34, 109] in which a node compatibility matrix is built using either the spatial locations of feature points [109] or the descriptor information [34].The goal is to find a permutation matrix that maximizes $tr(P^T C)$ where C is the node compatibility matrix .

The problem also can be casted as a **Graph Isomorphism** problem [111, 132, 62, 127]. 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. This formulation uses the spatial locations of feature points to construct weighted or Unweighted graphs to be matched and the goal is to find a permutation matrix that will bring one graph to the other. Spectral methods for Graph Isomorphism differ in the way of building the weighted/unweighted graph and in the way they compute the solution. Some of them use the adjacency matrix [111, 132, 62, 127] but in [118] they used the Laplacian of the adjacency matrix. Typically the weighted matrix that represent the graph use Euclidean based kernels because it is both rotation and translation invariant.Alternatively affine invariant kernels might be better to build the weight matrix. Using Affine invariant kernel would be more robust towards image transformations [141]. However, the affine invariant kernel used in [141] is

not robust to noise and can break easily. Also some approaches use a set of the eigenvectors to compute the correspondences [62, 141, 132] instead of using all eigenvectors.

The graph matching can also be casted as **Quadratic Assignment Problem** in which both node compatibility and edge compatibility are used together. Unfortunately the quadratic assignment problem is NP-Hard [5] and thus most of the techniques that used the quadratic assignment formulation will end up with approximations to the solution. A spectral relaxation of the quadratic assignment problem is done in both [73, 29] by considering only the spectrum of the edge compatibility matrix which of quadratic size of the original graph sizes.

Graph Matching and Problem Size

As we discussed above the spectral correspondences depends on the definition of the graph matching problem. The complexity of the matching problem will vary according to the way the graph matching problem is defined.

Bipartite graph matching (Linear Assignment): Given two graphs the matching is solved via combinatorial graph matching algorithm such as the Hungarian algorithm [97] which is polynomial time $O(n^3)$ where n is the number of nodes in a graph. Instead, spectral decomposition of the cost matrix can yield an approximate relaxed solution, e.g. [34, 109] to the permutation matrix P. The size of the problem is linear in the number of nodes of each graph.

Graph Isomorphism: The problem size remains linear in most of graph isomorphism formulation for spectral methods and it reduces to compute the eigenvalue decomposition of each graph. An alternative approach for solving graph isomorphism constructs an associate graph of the two graphs and uses replicator equations to reach equilibrium state of the graph nodes [99]. The number of nodes of the associate graph is of quadratic size of the number of nodes of original graphs(i.e. $N = n^2$) and they solve for a quadratic programming problem iteratively.

Quadratic Assignment: The quadratic assignment is considered as the state of the art solution for the graph matching problem [20], such formulation enforces edgewise consistency on the matching. Since the size of the problem is quadratic because of the edge compatibility matrix , the solutions introduced for the quadratic assignment problem includes different approximations,such as spectral methods [73, 29]. Graduated assignment [49] which consists

of a series of first-order approximations to the quadratic assignment objective function. Dual decomposition which solves for linear subproblems and small quadratic problems instead of solving large quadratic problem [125]. In [11] a gradient descent approximation was done to get rid of the integer quadratic programming overhead. Another approximations using Relaxation labeling and probabilistic methods define a probability distribution over mappings, and optimize using discrete relaxation algorithms [138, 24]. In [18]the problem itself is approximated by identifying approximate models for the original problem and finding the exact solution for these models.

7.2.4 Learning Graph Matching:

In [20] 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 [29], which is an important finding. In [74] using smoothing based optimization they learned the edge compatibility matrix of quadratic size $N = n^2$ instead of the node compatibility matrix as [20] and they showed that it leads to better matching results.

7.2.5 Matching Multiple Sets

There is very few papers that addressed solving for multiset correspondences in a fundamental way. In image sequences the problem can be addressed by forward tracking a set of features [110] also this appears in structure from motion applications and photo tourism [113]. In [26] a deterministic annealing-like approach was introduced to find correspondences between multiple point sets and was used to obtain a shape average, which is updated through the iterations of the deterministic annealing optimization. These approaches are different from our framework in several aspects. First they do not consider the feature descriptor as [110, 26] information and thus there have been used in the context of tracking or building an average shape from examples. They don't generalize to be used in object recognition scope. Also the matching is computed in a pairwise manner [26] or incrementally as in [113, 110] and these approaches can't be computed in one shot.

7.3 Feature Matching

The embedding achieved through minimizing the objective function Eq 3.2 represents a Euclidean "Feature" space encoding both the descriptors' similarity and the local spatial structures. Solving for matching will be a straight forward task in such space. Embedding all the input points in such a way will result in a consistent set of matches, which means the pairs of matches will obey some common transformation between the two point sets. Therefore there is no need to explicitly add pairwise consistency constraints as done in quadratic matching approaches [11, 29, 73, 125]. The objective function in Eq 3.2 is general. We can easily see that algorithms that use only spatial constraints are a special case by replacing the off-diagonal blocks in the affinity matrix $\mathbf{A}$ by a unity block. On the other hand, matching algorithms that use the feature similarity constraints only is a special case by replacing the diagonal blocks in the affinity matrix $\mathbf{A}$ by an identity block. Notice that the size of the matrix $\mathbf{A}$ is linear in the number of input points, i.e., for the case of matching two sets, $\mathbf{A}$ is an $(N_1 + N_2) \times (N_1 + N_2)$ matrix. In contrast, other approaches that enforces pairwise consistency [11, 29, 73, 125] use a consistency matrix that is quadratic in size $N_1 N_2 \times N_1 N_2$. Such quadratic order hinders the scalability of such matching techniques. Figure 7.2 summarizes our framework for the case of two sets only. It shows the generality of the framework, also it shows the interaction between different components in our approach.

7.3.1 Matching Settings

We present three settings in which our framework can be used depending on the application.

Pairwise Matching (PW): Given two sets of features, the matching reduces to solving a bipartite graph matching problems between two sets of embedding coordinates. We give details about how to obtain the matching in Sec 7.3.2.

Multiset Pairwise Matching (MP): If we have multiple sets of features and we would like to find pairwise matching between each pair of sets, then embedding all the features in all the sets will give a global unified feature space. Pairwise matches between any two sets can also be solved as a bipartite graph matching where the weights are defined in the embedding coordinates. In this case, the global solution is expected to enhance the pairwise solution. This

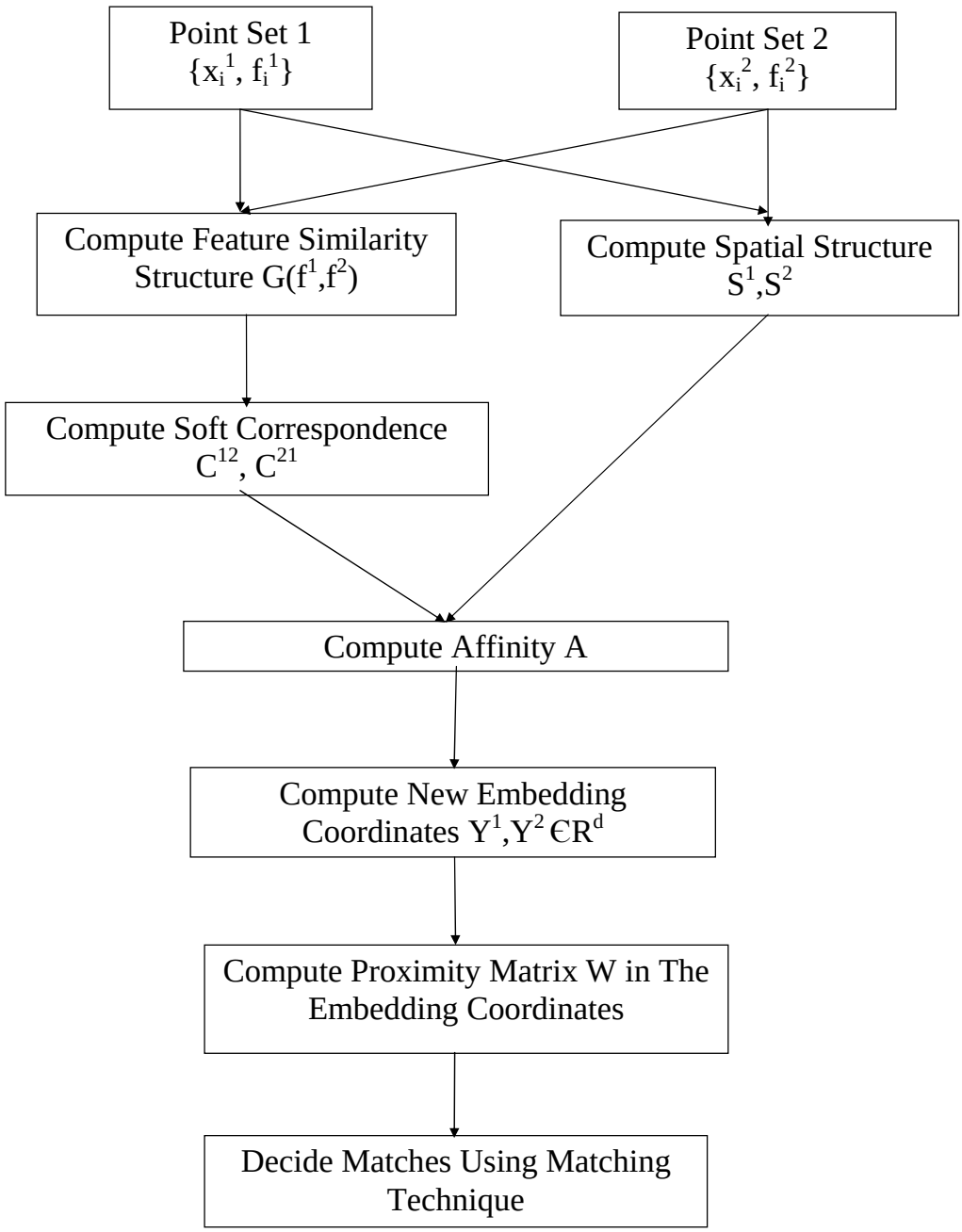


Figure 7.2: Illustration of our framework entities and interaction between them

is shown in the experiment in Sec 7.4.2. We give details about how to obtain the matching in Sec 7.3.2.

Multiset Clustering (MC): If we have multiple sets of feature points the unified embedding should bring correspondent features from different sets to be close to each other. In that sense, clustering can be used to in the embedding space to obtain matching features. In this paper we applied k-means clustering in the embedding coordinate to find the feature groups. Other clustering techniques can be used. The problem can also be formulated as a Multi-partite graph matching in the embedding space.

In Sec. 7.4.2 we show the results obtained by applying these three settings on the well known 'Hotel' sequence.

7.3.2 Matching Criterion

The embedding coordinates achieved by solving the objective function 3.1 guarantees that the Euclidean distances between the embedded points reflect both the spatial and descriptor constraints. Therefore, the matching problem reduces to solving a bipartite matching problem in the embedding space. This can be solved by many approaches such as the Hungarian algorithm [97] and others. However, in particular we used the Scott and Longuet-Higgins (SLH) algorithm [109] as matching criterion in the embedding space. The conditions required for the Scott and Longuet-Higgins matching are satisfied by the embedding since all the points are lying on the same plane and there are no large rotation. We compute an $N_1 \times N_2$ Ecuildian distance based weight matrix $\mathbf{W}$ in the embedding space using a Gaussian kernel and then we compute an orthonormal matrix $\mathbf{P}^*$ in a way similar to Eq. 3.5. We decide a match if the element $\mathbf{P}^*_{ij}$ is maximum in its row and its column. In addition we add the condition that the second largest element in its row and its column is far by threshold ratio as done in [34].

The main reason we chose the SLH algorithm over the Hungarian algorithm as a matching criterion is its ability to reject false matches. The Hungarian algorithm finds a matching for each feature even though that match might not be good, which is not a desired characteristic.

7.4 Results

In this section we show both quantitative and qualitative results on different data set. Despite that our focus is on non-rigid matching, we also show results on rigid matches for quantitative and comparative evaluation [2].

7.4.1 Non-Rigid Matching

Fig. 7.3 shows some matching results on nonrigid motions. We used sequences from the KTH dataset [3]. Fig. 7.3-top shows the results of our pairwise matching (**PW** setting) using SIFT features on four frames of a walking motion, i.e., 6 pairs. Our approach boosted the matches obtained to double of the original SIFT matches. Fig. 7.3-bottom shows the result of the multi-set setting (**MC**) applied on 13 frames of a half cycle of hand waving. Due to the low resolution in the sequence, a small number of features are detected (around 25 features per frame). Enforcing the global matching with the spatial constraints boosted the number of matches to from 44 to 73 and correct matches can be found on the moving parts for all the 13 frames.

Fig. 7.4 shows sample matches on motorbike and airplane images from Caltech101 [75]. In each case we used eight images and used the Multiset Pairwaise (**MP**) to match all pairs. In these experiments we used affine kernels and Geometric Blur [11] features.

7.4.2 Comparative Evaluation: 3D Motion (Wide Baseline Matching)

Goal: This experiment aims at evaluating our proposed framework compared to the state of the art reported results including linear and quadratic assignment based approaches [29, 20, 125, 132, 72, 35] .

Data: We use the CMU 'Hotel' sequence with the same manual labeling of 30 landmark points employed in [20]. This dataset has been used in [20, 125] to compare the performance of graph matching methods. The sequence contains 101 frames that shows a 3D motion of the 'Hotel' object. The experiment is done using the same setting as [20, 125]: 15 frames are sampled (every 7 frames), that gives 105 pairs of images to match.

---

[2] To the best of our knowledge there is no available non-rigid dataset with ground-truth matches.

[3] http://www.nada.kth.se/cvap/actions/

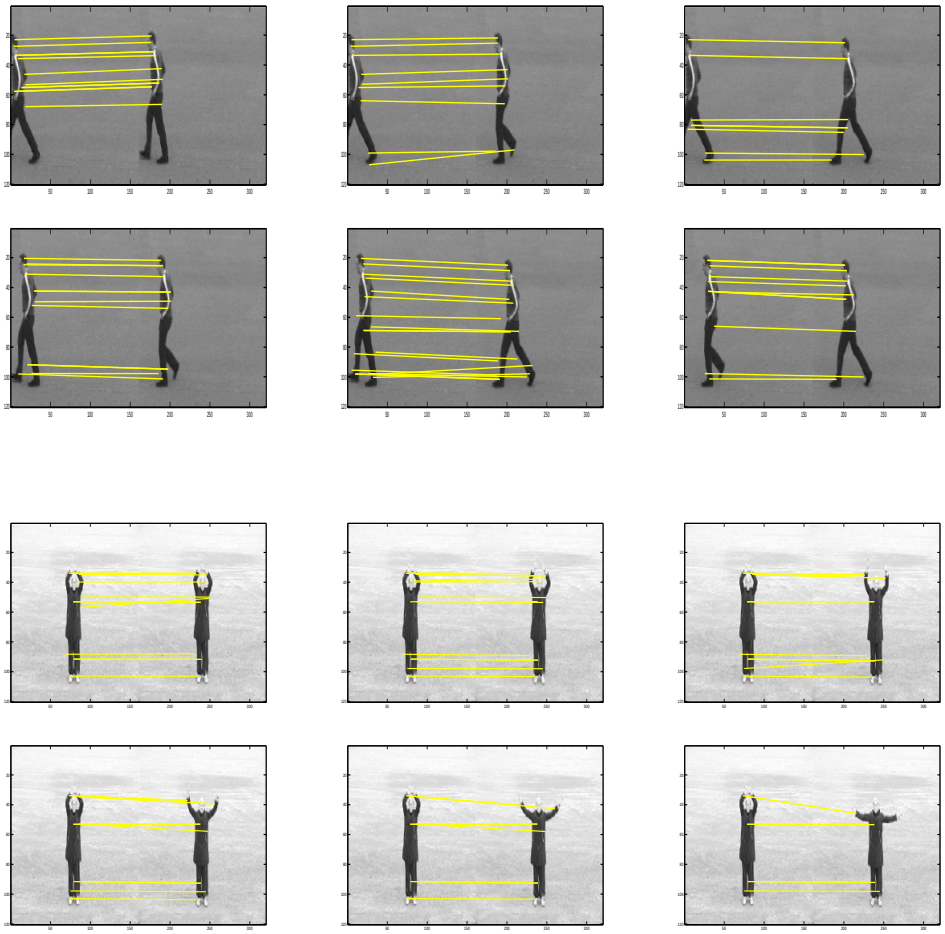


Figure 7.3: Top: Results on non rigid walking sequence (matched pairwise). Bottom: Sample results on hand waving sequence matched on a 13 frames in one shot (multiset). Shown is the first image matches with the consecutive odd frames in the 13 frames

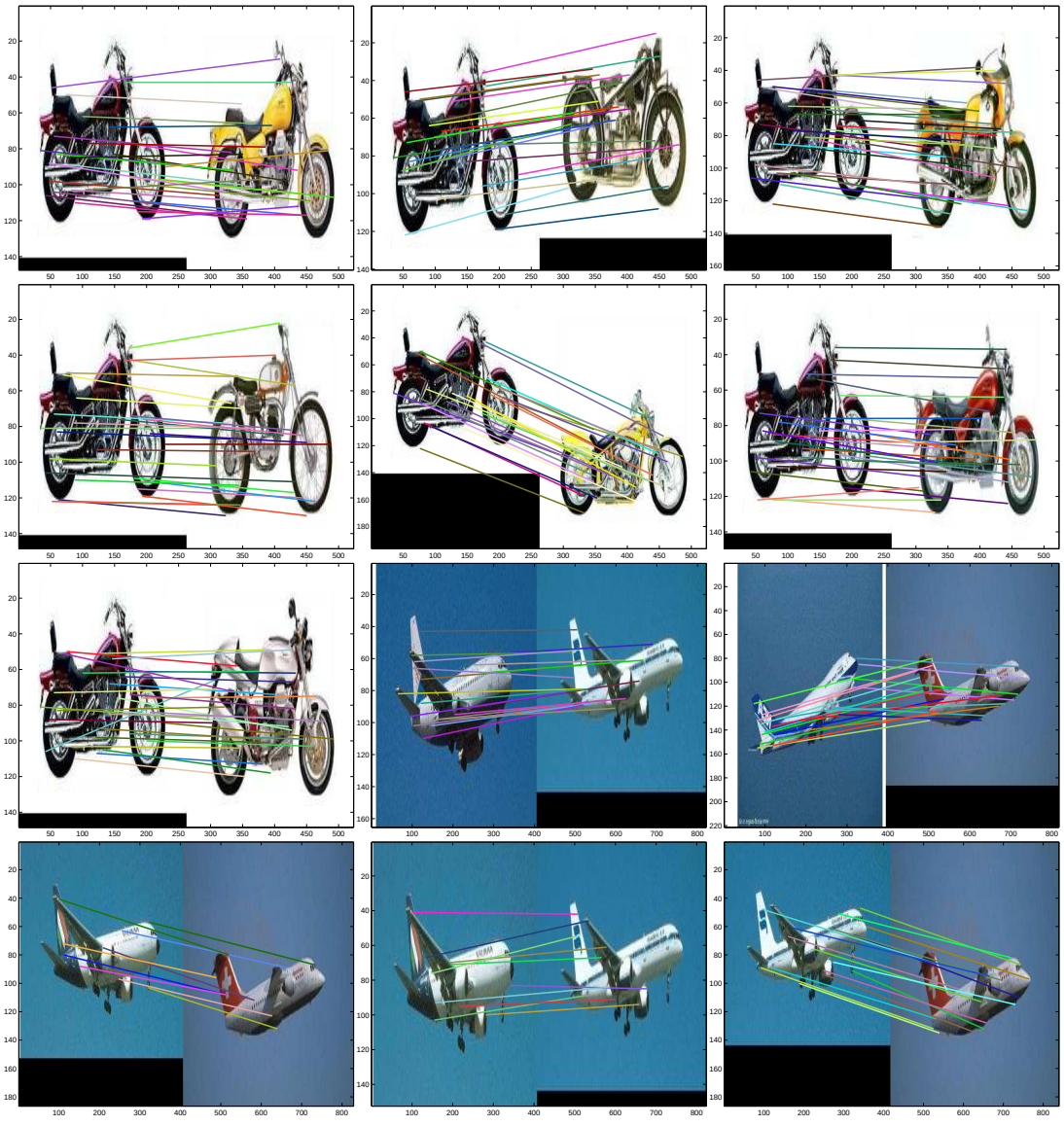


Figure 7.4: Sample results on Caltech 101 images. Best seen in color.

Competitive Approaches: In all cases we used the Shape context [9] as the feature descriptor (except for KPCA). We compared the following: 1)The KPCA matching [132] is an example of an algorithm that only uses the spatial structure. 2) Descriptor-only linear assignment: we used the Hungarian algorithm applied to the shape context descriptor. In this case only feature similarity is used. We used the histogram distances as our metric as it was introduced in [9]. 3) Our approaches: The three settings described in Sec 7.3.1: Pairwise (**PW**), Multiset pairwise (**MPW**) and Multiset with clustering (**MC**). We used a Euclidean double exponential kernel to encode the spatial structure, and Gaussian kernel on the *same* shape context descriptor for descriptor similarity. 4) Dual Decomposition approach proposed in [125]. This is a quadratic assignment approach that uses an iterative solution. 5) Results reported in [125], which are state of the art algorithms using quadratic optimizations. That includes [29] a spectral relaxation of the graduated assignment, [72, 35] and max-product belief propagation on a quadratic pseudoboolean optimization [125]. 6) Results reported in [20] after learning on another sequence (CMU 'House' sequence) using both quadratic and linear assignment with learning.

Evaluation: Evaluation is based on the mismatch ratio and the complexity of the problem. Table 7.1 shows that our basic **PW** outperforms all approaches that use linear complexity and outperforms some of the state of the art quadratic algorithms, e.g., [29, 72]. Using our multiset **MPW** and **MC** we reach 95.56% and 100% accuracy, which is not reached by any of the competing algorithms. It is very important to notice that the size of our affinity matrix A in the case of the multiset of 15 frames is just 450×450 and for the case of the pairwise matching is 60×60, where the size for one edge compatibility matrix for any of the quadratic assignment approaches is 900×900. Table 7.1 shows the complexity of the problem and the mismatch ratio. Fig 7.5 shows the matches obtained from all the 15 frames using our multiset approach.

7.4.3 Robustness: INRIA datasets

Data: In this experiment we use the INRIA datasets, which has been used by [86] for comparing descriptors. This dataset contains seven subsets that covers several effects such as viewpoint change, zooming, rotation, blurring and lighting change. Each of the seven datasets has a ground truth *Homography* matrix computed between the first image in each set and the other images in same dataset. Overall there are 36 matching problems given their ground truth.

Algorithm	Error Rate	Problem complexity
KPCA [132]	35.5%	Linear
Linear Assign. W/SC [97]	11.81%	Linear
Our Approach PW	**9.24%**	Linear
Our Approach MPW	**4.44%**	Linear
Our Approach MC	**0.0%**	Linear
SMAC [29]	15.97%	Quadratic
Fusion [72]	13.05%	Quadratic
COMPOSE [35]	4.51%	Quadratic
Belief Propagation [125]	0.06%	Quadratic
Dual Decomposition [125]	0.19%	Quadratic
Learning(LA) [20]	12-17%	Linear
Learning(GA) [20]	10-14%	Quadratic

Table 7.1: State of the art results on the ‘Hotel’ Sequence

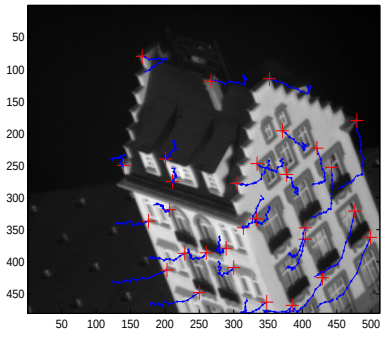


Figure 7.5: Matches obtained in 15 frames of the ‘Hotel’ sequence using one-shot multiset matching

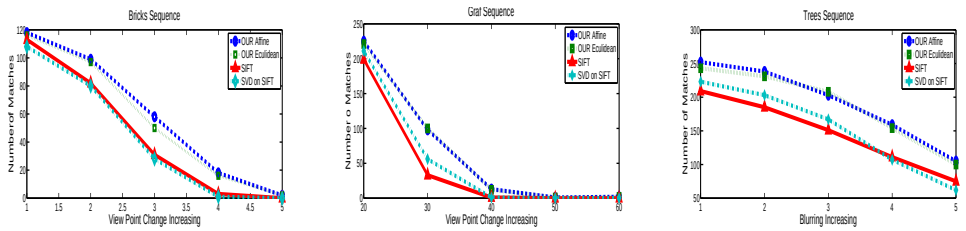


Figure 7.6: Number of matches affected by Different effects. left,middle) Increasing view point Change(Bricks and Graf), right) Increasing Blurring (Trees)

Goal: We use the INRIA data set to evaluate the robustness of the pairwise matching version of our framework to the various imaging effect in a dataset with ground truth. We also evaluate the behavior of the matching under strong affine transformation using both the Euclidean and the affine invariant kernels. This set demonstrates the scalability of our approach to handle a very large number of feature points (from 130 to 1250 SIFT features per image). That shows the value of our approach compared to the quadratic assignment approaches, which typically can only handle a number of features limited to around 100. We use the ground truth *Homography* matrices just for evaluating the resulting matches, since our approach does not assuming any geometric transformation prior.

Competitive Approaches: in this experiment we compare 1) The basic SIFT matches [79] as a baseline. 2) SVD-SIFT [34]: This approach uses SVD decomposition on a Gaussian proximity matrix in the SIFT descriptor space. 3) Our Pairwise matching approach with both a Euclidean Gaussian spatial kernel and an affine invariant kernel. In all cases we are using the same set of SIFT descriptors.

Results: Table 7.2 shows that for all the datasets, our approach with either kernels gives the highest number of correct matches. The last column gives the number of features in the first image for each dataset. This result shows that enforcing the spatial consistency improves the descriptor matches. Fig. 7.6 shows the number of matches as a function of the viewpoint change or the blurring. The results show that the Euclidean kernel gives comparable results to the affine invariant kernel even under a very large viewpoint change. We selected the scale for the spatial kernel as a constant-multiple of the maximum distance between feature points in each image. In general, we found that selecting a scale large enough for the Euclidean kernels would give results comparable to affine invariant kernels, this is consistent with what was stated in [109].

Dataset(Effect)	SIFT Matching [79]	SVD on SIFT Matching [34]	Our Approach	Our Affine Approach	1^{st}Image Feature Count
Graf (ViewPoint)	47	54	66	**67**	464
Boat (Zoom&Rotation)	99	87	**108**	**108**	467
Bark (Zoom&Rotation)	49	47	**55**	**55**	392
Bricks (ViewPoint)	46	44	58	**59**	310
Trees (Blurring)	146	153	186	**191**	642
Cars (Lighting)	60	17	65	**70**	134
Bikes (Blurring)	227	229	**239**	237	400

Table 7.2: Average number of correct matches for each dataset from INRIA datasets

Chapter 8

Implicit Feature Spatial Manifold Learning through spatial consistent label propagation

In this chapter we propose a novel approach to integrate feature similarity and spatial consistency of local features to achieve the goal of localizing an object of interest in an image. The goal is to achieve coherent and accurate labeling of feature points in a simple and effective way. We adapt the Global and Local Consistency Solution to our method of label propagation to infer the labels of local features in a test image from known labels. This is done in a transductive manner to provide spatial and feature smoothing over the learned labels. We show the value of our novel approach by a diverse set of experiments with successful improvements over previous methods and baselines classifiers.

8.1 Introduction

Object localization is a fundamental problem in computer vision. The detection and accurate localization of a given object under general settings with high class variation, different viewing conditions, presence of occlusion and clutter is a challenge. Local features descriptors, such as SIFT [79] and other similar descriptors, have been shown to be useful for object localization and recognition as they are highly discriminative and possess invariant properties. The spatial configuration of the local features is also important to decide the presence or absence of an object since it captures shape information which markedly reduces the rate of false positives. A good localization algorithm should find good object candidates with low false alarms.

Many researchers have addressed the localization problem as finding candidate patches that have high probability/score of lying on the object and at the same time rejecting patches that are likely to be false alarms [82, 12, 96, 84, 71, 47, 32, 70]. Most of these approaches use multiple cues and do not depend on local features alone. In [82] an aspect graph encodes the

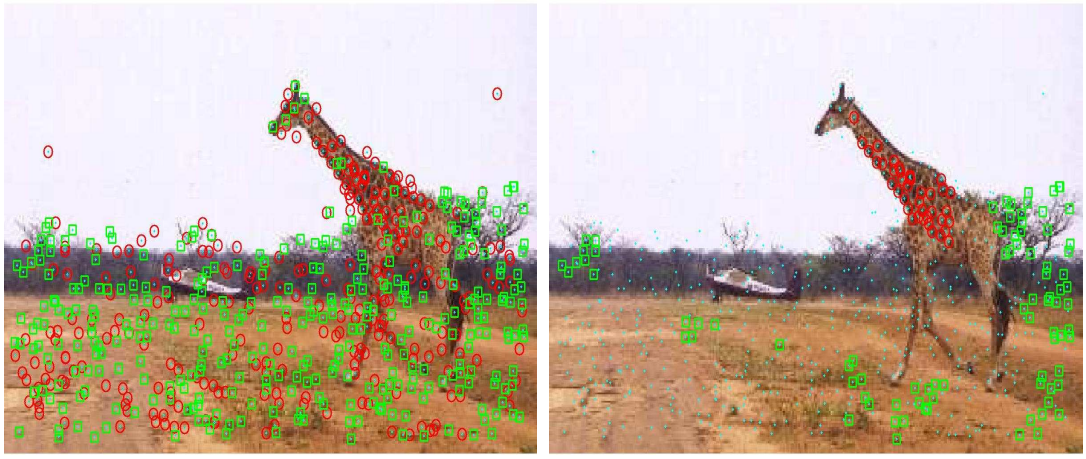


Figure 8.1: The left image shows the SVM classification of the local features and the right image shows the result of our localization approach. Red and green points are foreground and background, respectively

shape of the object and *shape masks* are learned to reduce the hypothesis space. In [96, 70] *segmentation* cues are augmented with local features to find accurate localization. In [84] a *hard matching* is established. Other approaches use different types of context cues [48].

Although it is reasonable to consider more cues beyond local feature descriptors and their locations to solve localization, it is also desirable to enhance localization without adding more cues. Enhancing the usage of local features is complementary to other state of the art achievements in localization. In this chapter we only use local features defined by a feature descriptor and location in the image. We do not use any more cues.

Similar to our approach are [101, 60, 68] in which the local features are pruned heavily to find the good features to be used in sophisticated localization algorithms. Similar to [101], our approach can be understood as a way of pruning local features so that only candidate features for the object class and background class are considered for further higher level processing to accurately find the object of interest.

In this chapter we pose the object localization problem as a transductive learning problem on a graph structure. *Graph-based methods* for both transductive and semi-supervised learning are widely used in many applications where the structure of unlabeled data can be combined with the structure of labeled data to learn better labeling [21]. This approach works well if

there is a valid manifold assumption on the underlying data and hence the intrinsic manifold describes neighborhood relationships over the labels. A characteristic problem is that a feature may lie in more than one feature-space and hence lie in more than one manifold. For example, in the object localization problem using local features, local features can lie in two different spaces, namely, the feature descriptor space and the spatial x-y feature location on the image coordinates.

A successful approach of object class localization using local features must handle the feature descriptor and feature location spaces accordingly. Under class variation (like many real objects) there might exist multiple manifold structures in the descriptor space. Simply, the manifold can be broken into several clusters where every cluster has its own manifold structure. This is what visual code book methods try to capture. The idea of exploiting the manifold structure in the feature descriptor and spatial domain was recently addressed in [123]. Unlike [123] where they explicitly embed the feature manifold and perform inductive learning in that embedded space, we exploit the manifold structure in the data implicitly without embedding and within a transductive learning paradigm.

The spatial arrangement of local features is useful in many aspects. Spatial neighborhoods gives us local geometry and collectively provides shape information about a given object. Spatial neighborhoods also inherently provide smoothness over labels since we expect to see the same labels in close proximity to each other. This is used in MRFs for segmentation [130] where the points are typically defined on a grid.

The contribution of this chapter is that we pose the object class localization problem as classifying the features of a test image using transduction on a graph composed of the training features as well as the test features. Every training feature has a label and using transduction we can infer the labels of the test features. We propose a new technique to capture similarity among data points which share two structures: the spatial structure, which refers to the spatial arrangement of local features within an image, and visual structure, which refers to the feature similarities between local features in the whole data set. We call our approach Spatial-Visual Label Propagation (SVLP) and can be used to detect objects and/or their parts in images. In addition our approach is independent of the actual local feature descriptor being used (e.g. Geometric Blur (GB) [11], SIFT [79], etc.).

8.2 Problem Definition

We denote the i^{th} feature in the k^{th} image by $f_i^k = (v_i^k, x_i^k,)$, where $v_i^k \in \mathbb{R}^{Desc}$ is the feature descriptor and $x_i^k \in \mathbb{R}^2$ is the feature coordinate in the image. The feature descriptor can be an image patch or local descriptor such as SIFT, Geometric Blur, etc. The labeled training data consisting of K sets of feature points, $X^1, X^2, \cdots X^K$ in K images where $X^k = \{(f_i^k, y_i^k)\}$. Here $y_i^k \in \mathbb{R}^C$ denotes the class label and C is the number of classes (e.g. foreground/background or object parts as classes). For the binary case $C = 2$ and for the k^{th} image we have $y_i^k = [1, 0]$ if the feature f_i^k belongs to the object class and $y_i^k = [0, 1]$ if otherwise.

During testing, an unlabeled test image is given with its associated set of features $\{f_i = (x_i, v_i)\}$ and corresponding labels $Y = \{y_i\}$ which are unknown. The goal is to label these features in the test image. Once the labels are discovered we can localize the object (or part) of interest by its local feature labels. The labeling should reflect what we learned from the training data about the features and their local spatial arrangement as well as coherent regions in the test image.

A fundamental assumption in label propagation is label consistency: points in close proximity to each other are likely to have similar labels [145]. This is often called the manifold assumption. The key difference in our problem is that the consistency or manifold assumption in our case has two folds: spatial consistency: close by features on the same image should have the same label, feature consistency: similar features across the different images should have the same label. The question is how to construct a graph that reflects spatial and feature similarity and allows label propagation in a way that preserves both similarities. Simple concatenation of the feature descriptor and its location in the image cannot be considered since this will give rise to the issue of how to do deal with a test image without knowledge about the location(s) of object(s) of interest.

The SVLP approach captures the local spatial arrangement between the feature points by computing a local kernel based on the spatial arrangement of the local features in one image (intra-image). SVLP also captures the similarities between the features in the descriptor space across the different images (inter-image). Thus augmenting these two types of similarities

in one graph is important to find a meaningful, accurate and coherent labeling. The intra-image spatial structure in the test image is also important to find the coherent labeling. Finally SVLP aims at finding long range (global) relations between the features by propagating local information through diffusion between the spatial and visual appearance information. We use the SVLP as a transductive solution to induce the desired labeling of the feature points in the test image.

8.3 Background on Label Propagation Algorithms

As explained in [21], label propagation relies on the idea of building a graph whose nodes are data points and edges represent similarities between points. Known labels are used to propagate information through the graph in order to label all nodes [146, 145]. Finding a way to propagate labels from labeled data to unlabeled data has many applications, for example, interactive image segmentation [130], image annotation [59], visual code book generation [22].

Graph Construction: Given a point set $X = \{x_1, \cdots x_l, x_{l+1} \cdots, x_n\}$ and a label set $L = \{1, \cdots c\}$ the first l points have labels $\{y_1, \cdots, y_l\} \in L$ and the remaining points are unlabeled. The goal is to predict the labels of the unlabeled points.

The graph built by label propagation methods represents the geometry of the data induced by both labeled and unlabeled data. It defines a weight matrix W : W_{ij} is non zero iff x_i and x_j are "neighbors". One choice of the W matrix is a k-nearest neighbor matrix: $W_{ij} = 1$ iff x_i is among the k-nearest neighbors of x_j or vice versa (and 0 otherwise). In our approach we used k-nearest neighbors with $k = 20$ to create a sparse graph to ease computational load.

Propagation By Iteration: Given the similarity graph of $n = l + u$ nodes (l labeled nodes and u unlabeled nodes), the labeled nodes $1, 2, \cdots, l$ will propagate their labels to their neighbors. The process is repeated until convergence of the labels is achieved. The iterative formula can take different forms depending on fixing or changing the labels of the labeled points.

Harmonic Function Solution: One example of a label propagation algorithms is the harmonic function solution by [146]. In this algorithm the known labels remain unchanged. The algorithm consists of computing an affinity matrix W using a gaussian kernel. The intra-image local structures are not utilized to compute the unknown labels.

Global and Local Consistency (GLC): Another solution for label propagation is the GLC method [145]. This method provides more interactions between the labeled and unlabeled data. The known labels are not fixed and so they can also change through the iterations. The algorithm is summarized in the following steps.

(1) Compute an affinity matrix W using Gaussian kernel with bandwidth σ.

(2) Compute the normalized affinity $S = D^{-1/2}WD^{-1/2}$, in which $D = \sum_j W_{ij}$.

(3) Initialize $\hat{Y}^{(0)} = (y_1, \cdots, y_l, 0, 0, \cdots, 0)$.

(4) Choose parameter $\alpha \in [0, 1)$.

(5) Iterate $\hat{Y}^{(t+1)} = \alpha S\hat{Y}^{(t)} + (1-\alpha)\hat{Y}^{(0)}$, until convergence.

(6) Label point x_i by the converged upon $\hat{y}_i^{(\infty)}$. The convergence of the sequence is proved regardless of the initial labeling Y. During iterations each point receives two contributions coming from its neighbors through S and its initial value $\hat{Y}^{(0)}$ respectively.

$$\hat{Y}^{(\infty)} = (1-\alpha)(I - \alpha S)^{-1}\,\hat{Y}^{(0)} \tag{8.1}$$

Now computing $\hat{Y}^{(\infty)}$ can be done without iterations. This shows that the iteration result does not depend on the initial value of the iteration. Also we can notice that $(I - \alpha S)^{-1}$ is in fact a graph or a diffusion kernel. We can also define $S = \begin{pmatrix} S_{ll} & S_{lu} \\ S_{ul} & S_{uu} \end{pmatrix}$.

8.4 Approach

8.4.1 Motivating Example

Two-Image Example: We illustrate the interaction between labeled and unlabeled features on a simple example where the features in the first image are all labeled and the features in the second image are all unlabeled.

The Harmonic Function Solution utilizes the visual structure across the two sets of feature points and also the spatial structure of the feature points in the test image. We note here that the the spatial structure of the first image is not utilized in any way to compute the labels of the feature points of the second image.

The GLC Solution in equation 8.1 utilizes the full S matrix, this actually means the whole

graph structure in the S will be utilized to induce the labels of the unlabeled features in the second image. More precisely the intra-image spatial structure that was ignored in the harmonic solution (represented by S_{ll} in GLC) will be diffused to the unlabeled feature points in the second image. We write down the expansion of equation 8.1 for the unlabeled features only as

$$\begin{aligned}\hat{Y}_u^{(\infty)} &= \left(\alpha S_{(ul)}^1 + \alpha^2 S_{(ul)}^2 + \cdots\right) Y_l \\ &+ \left(I_u + \alpha S_{(uu)}^1 + \alpha^2 S_{(uu)}^2 + \cdots\right) Y_u \end{aligned} \tag{8.2}$$

We note that the $\hat{Y}_u^{(\infty)}$ is getting its label from two terms. The first term depends on the labels of the ground truth labels Y_l and it also depends on $S_{(ul)}^p$. The terms $S_{(ul)}^p$ are the normalized similarities between labeled (training) and unlabeled (testing) features. The superscript p represents the order of the block matrices S which can be replaced by a summation of components consisting of S_{uu}, S_{ul} and S_{ll} which will be shown in equation 8.3. The second term in equation 8.2 depends on the unknown labels Y_u (which can be given some initial values using some external classifier, it also can be initialized as zeros) and it also depends on $S_{(uu)}^p$. The terms $S_{(uu)}^p$ are the normalized similarities between the unlabeled (testing) features.

The first order blocks $S_{(uu)}^1$ and $S_{(ul)}^1$ do not encode the spatial structure of the training image S_{ll}. On the other hand, the higher order blocks $S_{(uu)}^p$ and $S_{(ul)}^p$ do encode the spatial structure of the training image S_{ll}. This can be noticed if we further expand the terms $S_{(ul)}^2, S_{(uu)}^2, S_{(ul)}^3$ and $S_{(uu)}^3$ in terms of the original S blocks

$$\begin{aligned} S_{(ul)}^2 &= S_{ul}S_{ll} + S_{uu}S_{ul} \\ S_{(uu)}^2 &= S_{ul}S_{lu} + S_{uu}S_{uu} \\ S_{(ul)}^3 &= S_{ul}S_{ll}S_{ll} + S_{ul}S_{lu}S_{ul} + S_{uu}S_{ul}S_{ll} + S_{uu}S_{uu}S_{ul} \\ S_{(uu)}^3 &= S_{ul}S_{ll}S_{lu} + S_{ul}S_{lu}S_{uu} + S_{uu}S_{ul}S_{lu} + S_{uu}S_{uu}S_{uu} \end{aligned} \tag{8.3}$$

The higher orders blocks ($S_{(uu)}^p$,$S_{(ul)}^p$) already have the term S_{ll}. This shows that the unknown labels $\hat{Y}_u^{(\infty)}$ are not only affected by the similarity across the labeled and unlabeled data points, but in fact it is affected also by the similarity in the training points. In other words the spatial structure of the training points is reflected on the propagated labels.

Conclusion: The two-image example above leads us to a number of conclusions. First, the

diffusion kernel $(I - \alpha S)^{-1}$ that is used in the GLC solution is capturing the long-term relationships (*i.e.* between pre-convergence and post-convergence labels) in the **whole graph** constructed from the two sets of feature points, labeled and unlabeled (coming from the single training image and single testing image). On the other hand the diffusion kernel used in the harmonic function solution is capturing only the long term relationships in the unlabeled data (in our case, the test image).

Second, although it seems less intuitive to change the labels of the training set, we find that it is fundamental to change the labels in the training features so that we can benefit from the spatial structure in the training image. We understand that changing the labels for labeled data is sound when the labeled data has some overlap between the classes. In our addressed problem of object class localization from local features this is also sound, because the features that are close to the boundary of an object will have much confusion between its original label and the labels of surrounding features. This will lead to find some features that might change its label depending on its neighborhood structure.

Third, the two images example gives us an intuition of how to design the terms in the weight matrix W when we construct the graph, this will be reflected on the normalized weight matrix (S). We see that we need to define some spatial structure for the features from each image in the training set. We see that we need to define some structure that represents the visual appearance similarity between the image in the training set and the image in the test set. In our problem where the local features are defined by two different vectors (descriptor and spatial location), it is easy to see that the spatial structure can be inferred to assure coherent labeling in the spatial space. Also the visual structure can be inferred from the feature descriptor similarity in the descriptor space so that the features that have high similarity in descriptor space can be labeled similarly.

8.4.2 Constructing W for SVLP

We first define the W matrix as a block matrix where the block W_{uu} is computed as a Gaussian kernel $K_x(.,.)$ on the spatial structure of the local features spatial arrangements on the test image. The blocks W_{ul} and W_{lu} are computed as Gaussian kernels $K_v(.,.)$ on the visual appearance structure of the local features between test and training images. The block W_{ll}

should be designed to reflect both the intra-image spatial structure within each image of training images as well as the inter-image visual appearance structure between features in different training images. W_{ll} is defined as a block matrix where the blocks on the diagonal represent the spatial structure within each of the training images and the off-diagonal blocks represent the visual structure between different images in the training set.

Equation 8.4 shows an example W matrix that has K training images and one test image. W_k^S is the spatial structure for image k, where W_{ij}^V is an visual structure kernel between features in image i and features in image j

$$W = \left(\begin{array}{c|c} W_{ll} = \begin{pmatrix} W_1^S & W_{12}^V & \cdots & W_{1K}^V \\ W_{21}^V & W_2^S & \cdots & W_{2K}^V \\ \vdots & \ddots & & \vdots \\ W_{K1}^V & \cdots & \cdots & W_K^S \end{pmatrix} & W_{lu}^V \\ \hline W_{ul}^V & W_{uu}^S \end{array}\right) \tag{8.4}$$

8.4.3 Objective Function for SVLP

We write down our objective function as the sum of three terms. The first term is the smoothness constraint on the intra-image spatial structures, The second term is the smoothness constraint on the inter-image visual structures. The third term is the fitting constraint, which means there should not be too much change from the initial label assignment. This avoids oscillations in the label values during the iterations.

In our formulation the first two terms mean that nearby points defined by the graph structure should not change their labels very often to allow the neighborhood structure to control the labeling process.

$$\begin{aligned} \Psi(\hat{Y}) &= \sum_p \sum_{i,j} W_p^S(i,j) \| \frac{\hat{Y}_p(i)}{\sqrt{D_{ii}}} - \frac{\hat{Y}_p(j)}{\sqrt{D_{jj}}} \|^2 \\ &+ \sum_{p,q,p\neq q} \sum_{i,j} W_{pq}^V(i,j) \| \frac{\hat{Y}_p(i)}{\sqrt{D_{ii}}} - \frac{\hat{Y}_q(j)}{\sqrt{D_{jj}}} \|^2 \\ &+ \mu \sum_i \|\hat{Y}(i) - Y(i)\|^2 \end{aligned} \tag{8.5}$$

Where D is the diagonal matrix $D = \sum_j W_{ij}$. W is defined in 8.4. p and q are the image indices. Once W is constructed, equation 8.5 can be rewritten as

$$\Psi(\hat{Y}) = \sum_{i,j} W(i,j) \| \frac{\hat{Y}(i)}{\sqrt{D_{ii}}} - \frac{\hat{Y}(j)}{\sqrt{D_{jj}}} \|^2 + \mu \sum_i \|\hat{Y}(i) - Y(i)\|^2 \quad (8.6)$$

Equation 8.6 reduces directly to the same cost function as [145] and the minimization can be computed in closed form as equation 8.1.

8.4.4 Algorithm

We summarize our algorithm in the following steps

- Training (Constructing W_{ll}). Given K training images with labeled local features.
  1. For $k = 1 : K$
     Construct the blocks W_k^S as $W_k^S(i,j) = exp(\|x_i^k - x_j^k\|^2/(2\sigma_x^2))$.
  2. For a certain p and $q = 1 : K$ where $p \neq q$
     Construct the blocks W_{pq}^V as $W_{pq}^V(i,j) = exp(\|v_i^p - v_j^q\|^2/(2\sigma_v^2))$.
  3. Construct W_{ll} as in equation 8.4
- Testing (Construct full W and do the transduction) Given a test image with unlabeled local features
  1. Construct the block W_{uu}^S as $W_{uu}^S(i,j) = exp(\|x_i^u - x_j^u\|^2/(2\sigma_x^2))$.
  2. For $k = 1 : K$
     Construct the blocks W_{uk}^V as $W_{uk}^V(i,j) = exp(\|v_i^u - v_j^k\|^2/(2\sigma_v^2))$.
  3. Construct $W_{ul}^V = [W_{u1}^V | W_{u2}^V | \cdots W_{uK}^V]$.
  4. Construct $W_{lu}^V = (W_{ul}^V)^T$.
  5. Compute $S = D^{-1/2} W D^{-1/2}$.
  6. Iterate $\hat{Y}^{(t+1)} = \alpha S \hat{Y}^{(t)} + (1-\alpha)\hat{Y}^{(0)}$ until convergence, where α is a parameter in the range $(0, 1)$.
  7. Let $\hat{Y}^*$ denote the limit of the sequence $\{\hat{Y}^{(t)}\}$. Label each point $\hat{y}_i$ as a label $\hat{y}_i = arg\max_{j \leqslant c} \hat{Y}_{ij}^*$.

8.5 Experiments

In the experiments reported in this chapter we used Geometric Blur (GB) [11] and SIFT [79] as the local feature descriptors. Towards the end of this section we briefly compare between these two descriptors. The datasets used in the experiments are: Caltech-101 [75], TUD Motorbikes and Cows [80], ETHZ Shape Classes-Giraffes [101], GRAZ02-Bikes [94].

8.5.1 Caltech-101

In this experiment we performed object class localization via feature labeling for all the classes in the Caltech-101 [75], each class separately. The Caltech-101 dataset is widely used by the community in categorization tasks. Here we carried out the localization for all the 101 classes to show that we can apply our method for object class localization across very different kinds of objects ranging from animals, man-made, indoor objects, etc. One main reason behind using this data set is that the ground truth is given via a contour surrounding the object of interest, which facilitates the quantitative validation of our localization approach.

Every training image has at most 300 (the number of local features actually vary significantly depending on the class) local features. These local features are described by GB [11] descriptors and their spatial location in their images. The detected local features within the contours are labeled as object class and the features outside the contours are marked as background class. We ran our algorithm 5 times on all classes for each of three different training settings (sizes 10, 20 and 30). By using our SVLP method the labels of the test image feature points are inferred and thus this leads to localization of the object of interest. Similar to many other researchers in object class localization from local features [60, 68, 101], we report the q percentile of features that scored the highest in the object class or background class.

In figure 8.2 the performance of our localization method is compared to two baselines which are the 1-NN classifier based on the feature descriptor alone and the SVM classifier based on the feature descriptor alone. We applied SVLP given different numbers of training samples per class and we fed the SVM estimated solution to our algorithm as an initial Y_u. The figure shows that the q percentile SVLP significantly improves over the baselines, even with a very large portion of features included in the accuracy measure, i.e. 80%. We note here that

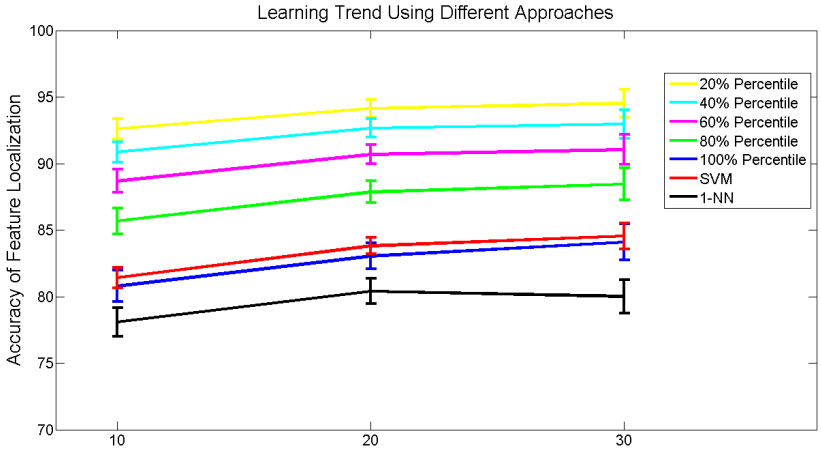


Figure 8.2: Learning Trend: changing the training size per class improves the results.

	F	M	A	W	K
SVLP	0.1028	0.0384	0.1365	0.0213	0.0769
SVM	0.2359	0.0487	0.2030	0.0902	0.1372
1-NN	0.3229	0.1667	0.2721	0.0732	0.2197
[60]	.30	0.11	0.21	.08	.19
[68]	.15	0.07	0.177	.03	0.08
[123]	.31	.003	.02	-	-

Table 8.1: False Positive Rates (FPR) for different methods. F: Faces, M: Motorbikes, A: Airplanes, W: Watches and K: Ketch. Results by [123] are at 20% percentile and is not comparable directly with other entries.

most other localization approaches use only the best 20% of the local features in measuring the accuracy. This improvement is very meaningful as the SVLP is always finding a spatially coherent feature labeling. As q decreases the localized features on the object of interest become more and more confident localized features. We also made another observation that the closer a feature lies to the core of the object, the stronger the confidence it receives using our approach.

For comparative evaluations we mainly consider the approaches [60, 68]. The reasons behind this selection are the following. Firstly, similar to [60, 68], our goal is to localize features into foreground/background classes. Due to this, we use the same evaluation measure (FPR) as [60, 68]. Using FPR is a more sensible choice over bounding box overlap ratio when evaluating sparse local feature localization. Secondly, since the localization in [60, 68] is performed after clustering the images with very high accuracy (around 98%). These approaches localize the features that belong to the object in every individual cluster independently and

Figure 8.3: Sample Results on ETHZ-Giraffes, TUD-Cows, TUD-Motorbikes and Caltech-101. Every row represents the percentile at which the localization is inferred. The top row shows the top 80% percentile of the features are localized, second row 20%. Red are foreground localized features. Green are background localized features. Detected features are shown in cyan. Best viewed in color with zooming.

hence the object is known to be in the image with high probability (around $.98$). In other words the unsupervised part ($i.e.$ clustering) of their approaches does not increase the hardship of their feature ranking problem. Thirdly, we use only $10 - 30$ training images which is much more challenging than the 100 images per class in [60, 68]. The much larger number of training images they select balances the unsupervised ranking they perform on their features. Lastly, we favored the setup in [60, 68] over ours because localization results by our approach are based on the object contour while their approaches are based on bounding boxes. In addition, we reported accuracy in 60% of the scoring features which adds weaker features than the 50% they use in their evaluation. This addition of weaker features degrades the (FPR) in our case.

The best 5 classes that improved over the SVM baseline using 20 training images were $\{$crocodile, crocodile-head, pagoda, hedgehog, cougar-body$\}$ and the least 5 improving classes were $\{$ewer, car-side, watch, dollar-bill, inline-skate$\}$. We notice that the biggest improvement takes place when the object of interest is a living object class which appears in very cluttered background. The least improvement is for well localized objects in their datasets.

	Classname	$q = 80\%$	$q = 60\%$	$q = 40\%$	$q = 20\%$
1	car-side	.9896	.9963	.9982	.9986
2	dollar-bill	.9788	.9917	.9967	.9986
3	windsor-chair	.9691	.9869	.9968	.9992
4	nautilus	.9686	.9849	.9910	.9972
5	faces-easy	.9644	.9850	.9946	.9984
97	sea-horse	.7837	.8246	.8625	.8876
98	ant	.7779	.8107	.8353	.8489
99	flamingo-head	.7756	.7968	.8176	.8414
100	star-fish	.7740	.7965	.8156	.8320
101	lamp	.7670	.7933	.8178	.8404

Table 8.2: Accuracy for best 5 and worst 5 classes on Caltech-101. These results were taken after training using 20 sample images. q here represents the percentile of highest scoring features taken.

8.5.2 Generalization to Subsets of LabelMe

Caltech-101 is designed for single object categorization tasks. To evaluate the generalization of our proposed approach on different datasets which might have different distributions, we used training example from Caltech-101 and tested on images from the LabelMe datasets [64] with multiple object instances. We used subsets of LabelMe datasets that have been used by [68]. In this experiment we trained from four Caltech 101 classes namely $\{$Motorbikes, Cars-rear, Faces, Airplanes$\}$. Since the object scales are very different in Caltech-101 and LabelMe, we adapted a pyramid of scales on the test images. We show some results of the localized features in Figure 8.4.

8.5.3 TUD / ETHZ Datasets

We experimented on three other datasets to analyze the performance of our approach compared with an SVM and 1-NN baselines. The first dataset is TUD-Motorbikes which is part of the PASCAL collection [98] which is known to contain hard images for the object localization problem. The hardship lies in the fact that the images have different resolutions, scales, background, heavy clutter and multiple instances per image. The second dataset is TUD-Cows which is a simple dataset with varying skin textures on the body of the cows in the images. The third dataset is ETHZ-Giraffes which contains images of giraffes under different deformation conditions (i.e. the giraffes' necks vary in shape from fully extended to leaning downwards).

Figure 8.4: Generalization to some example from LableMe dataset. Features with top 25% confidence are shown. Red for foreground localized features. Green for background localized features. Detected features shown in cyan. Best viewed in color with zooming.

	Classname	ETHZ-Giraffes	TUD-Cows	TUD-Motorbikes
Accuracy	SVM	.5980	.8550	.5776
	KNN	.5878	.8259	.5655
	$q = 80\%$	.7322	.9339	.6703
	$q = 60\%$	.7649	.9714	.7026
	$q = 40\%$	.7977	.9874	7327
	$q = 20\%$	.8251	.9933	.7601
FPR	SVM	.4036	.3119	.4826
	KNN	.3972	.2217	.4914
	$q = 80\%$	.2049	.1357	.3763
	$q = 60\%$	.1670	.1072	.3414
	$q = 40\%$	.1294	.0812	.3093
	$q = 20\%$	.1061	.0536	.2835

Table 8.3: This table shows a comprehensive comparison of the presented approaches using different percentiles q as well as two baseline classifiers: SVM and K Nearest Neighbors (K=1, results did not vary significantly with variations of K)

The images in this dataset are also challenging as they exist in multiple scales, resolution, multiple instances per image and contain extensive clutter in the form of vegetation.

For TUD-Cows and ETHZ-Giraffes we set the number of training images to 20. For TUD-Motorbikes we used 30 training images. The number of training images are approximately $21 - 26\%$ of the size of the respective datasets. The much larger portion of the dataset can then be used for testing. In all three datasets we used 300 SIFT descriptors. The reason why we used the SIFT descriptor in these datasets is because GB failed on images containing vegetation in the form of bushes, trees, grass, etc. The reason behind this is that GB is not multi-scale and due to the large variance in the local structures of vegetation it is not able to generalize over the background class. SIFT on the other hand captures multiple scales of the local structures in the images and hence is able to discriminate between object and background classes with higher accuracy.

8.5.4 Object Parts Localization

For qualitative evaluation of our approach on part localization we carried out object part localization for some classes. Namely $\{$Caltech-Motorbikes, TUD-cows [69]$\}$. The parts of the objects are manually annotated via bounding boxes in the train images. We used TUD-cows to test how our part localization works in the case of non-rigid objects with articulation.

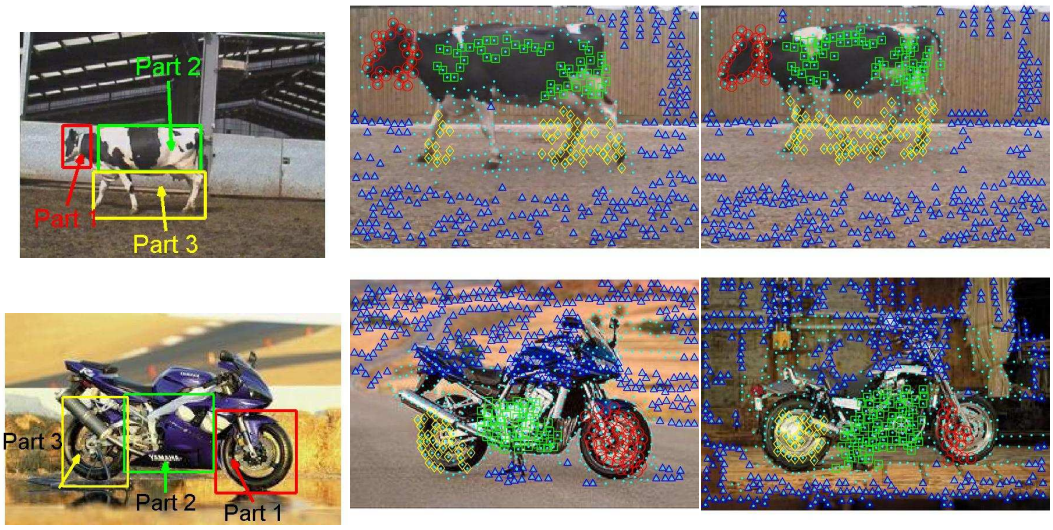


Figure 8.5: Object part localization. Left: bounding boxes defining the parts used during training. Middle and Right: some part localization results on TUD-cows and Caltech-Motrobikes. Features with top 60% confidence are labeled. Red for part 1 localized features. Green for part 2 localized features. Yellow for part 3 localized features. Blue for background localized features. Detected features shown in cyan. Better Viewed in color and zooming.

We used 20 images for training, each has 300 GB features. As shown in figure 8.5 we defined three parts motorbike using bound boxes by gathering the front wheel and some part of the attached handle, the second part is the engine area and the third part is the rear wheel and some part of the seat. We defined three parts on the Cow object using bound boxes as the head, body and legs. In both cases the remaining features are considered as back ground class. Notice that in the motorbike example, the front and back wheels have similar appearance and in the cow example the head and body have similar texture. Successfully localizing the parts in these example shows that the approach is in fact learning about the feature spatial arrangement. We can see (Figure 8.5) that the part labels are retrieved efficiently, here we use 60% percentile to show the localized features for each class.

8.5.5 Multiple Base-Learners

To avoid the problem that may arise due to over-sizing matrix W beyond computational bounds when dealing with large sets of images and features, we present a slightly adapted approach to the above. This approach uses multiple base-learners to train using smaller overlapping subsets of the training set. Each base-learner is identical to the presented approach and runs using the

Figure 8.6: Sample results from the challenging GRAZ02-Bikes dataset using 7 multiple base learners. The top row shows the 80% percentile and the bottom shows the 20% percentile. What may seem like a false positive bike detected in the background of the left image is actually a bike wheel. Same color legend as figure 8.5 Best viewed in color, with zooming.

algorithm outlined in 8.4.4. Each base-learner uses the same number of features as before and in this case we used SIFT as our descriptor. At testing, the local features in the test image with the most votes from these base-learners are selected as the most confident foreground / background features.

We ran this multiple base-learner approach on the more challenging GRAZ02-Bikes dataset [94]. The setup consisted of 7 learners. Each one of them was trained on 18 images from the training set of 110 (which was partitioned from the 300 images). As you can see the small training subsets overlap with each other to cover the full training set. Figure 8.6 shows sample results of this approach.

Chapter 9

Conclusions

The work we presented in this dissertation has abridged the gab between the usage of local features with their spatial arrangement in object recognition and manifold learning in data analysis. Current object recognition systems depend heavily on local features due to its discriminative nature, which ease the recognition tasks. However, different kinds of manifolds, e.g. view manifold and object class manifold, are already present in the data and revealing the underlying manifold structure in the data is expected to boost the recognition rates. That was confirmed through diverse set of problems that were addressed in the body of the dissertation.

In this dissertation 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 called the embedding "feature-spatial embedding" which provides an explicit low-dimensional representation of collection of local features from different images. We also introduced a relevant solution for the out-of-sample problem, which is essential to be able to embed large data sets. We defined a distance measure between images using the feature-spatial embedding framework. Given these three 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. Results showed that the framework can achieve superior results in recognition and localization.

Furthermore, we proposed a kernel regression framework based on manifolds of local features and their spatial arrangement. To the best of our knowledge this is the first work to address regression problems from local features without either establishing full correspondences between features from different images or using a holistic representation of the images. We tested the regression framework on different problems such as viewpoint estimation, face pose estimation and arm posture estimation. The results showed that the state-of-the-art methods

can be outperformed with our kernel regression framework for viewpoint estimation problems.

The feature embedding framework allowed us to solve the problem of feature matching, which is a very fundamental computer vision problem. Feature matching with spatial consistency usually involves a higher order quadratic assignment problem. In our framework we preserve spatial consistency of the features without the need of quadratic assignment. Also, our framework allowed us to match multiple sets by solving a single graph embedding problem. The results shows that both rigid and non-rigid cases can be solved using same framework.

At the very end, we also proposed to learn an implicit spatial-visual manifold without the need of computing an explicit low-dimensional embedding for the feature points. We achieved that by utilizing the label information that comes with the local features. We tested that implicit spatial-visual manifold with a transductive learning framework for object and part localization, which resulted in high accuracy and low false positive localization rates.

References

[1] S. Agarwal, A. Awan, , and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2004. 13

[2] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002. 1

[3] J. Aghajanian and S. Prince. Face pose estimation in uncontrolled environments. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. 53, 54

[4] J. Aghajanian, J. Warrell, S. J. Prince, P. Li, J. L. Rohn, and B. Baum. Patch-based within-object classification. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009. 53

[5] K. Anstreicher. Recent advances in the solution of quadratic assignment problems. *Math. Program.*, 2003. 61

[6] V. Balasubramanian, J. Ye, and S. Panchanathan. Biased manifold embedding: A framework for person-independent head pose estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 42, 46

[7] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 1981. 58

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003. 18, 21, 22, 28

[9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2002. 2, 10, 11, 60, 69

[10] 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 *Proceedings of Advances in Neural Information Processing (NIPS)*, 2004. 16, 24

[11] A. C. Berg. *Shape Matching and Object Recognition*. PhD thesis, University of California, Berkeley, 2005. 1, 9, 10, 19, 35, 36, 39, 42, 50, 62, 63, 66, 75, 83

[12] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 73

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003. 11

[14] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 11

[15] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002. 1

[16] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In *The ninth international workshop on AI and statistics*, 2003. 18

[17] M. C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1998. 13

[18] T. Caelli and T. Caetano. Graphical models for graph matching: Approximate models and optimal algorithms. *Pattern Recognition Letters*, 2005. 62

[19] T. Caetano, L. Cheng, Q. Le, and A. Smola. Learning graph matching. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007. 58

[20] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2009. 57, 61, 62, 66, 69, 70

[21] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006. 74, 77

[22] H. Cheng, Z. Liu, and J. Yang. Sparsity induced similarity measure for label propagation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009. 77

[23] H. Chiu, L. Kaelbling, and T. Lozano Perez. Virtual training for multi-view object class recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 42

[24] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1995. 62

[25] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Compter Vision and Image Understanding (CVIU)*, 2003. 58

[26] H. Chui, A. Rangarajan, J. Zhang, and C. M. Leonard. Unsupervised learning of an atlas from unlabeled point-sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2004. 62

[27] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1998. 17

[28] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *Compter Vision and Image Understanding (CVIU)*, 1995. 2, 4, 17

[29] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. *Proceedings of Advances in Neural Information Processing (NIPS)*, 2006. 57, 58, 60, 61, 62, 63, 66, 69, 70

[30] T. Cox and M. Cox. *Multidimentional scaling*. Chapman & Hall, 1994. 14, 28

[31] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004. 11, 12, 56

[32] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 10, 73

[33] M. Daliri, E. Delponte, A. Verri, and V. Torre. Shape categorization using string kernels. In *SSPR06*, 2006. 5

[34] E. Delponte, F. Isgrò, F. Odone, and A. Verri. Svd-matching using sift features. *Graph. Models*, 2006. 60, 61, 65, 71, 72

[35] J. Duchi, D. Tarlow, G. Elidan, and D. Koller. Using combinatorial optimization within max-product belief propagation. *Proceedings of Advances in Neural Information Processing (NIPS)*, 2007. 66, 69, 70

[36] A. M. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 2, 4, 18

[37] A. M. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 18

[38] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 2005. 13

[39] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google"s image search. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005. 1

[40] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 1, 13

[41] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 31, 37

[42] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2008. 36

[43] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transaction on Computer*, 1973. 1

[44] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Readings in computer vision: issues, problems, principles, and paradigms*, 1987. 58

[45] D. Forsyth and J. Ponce. *Computer Vision, a modern approach.* Prentice Hall, 2002. 43

[46] Y. Fu and T. Huang. Graph embedded analysis for head pose estimation. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, 2006. 42

[47] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009. 73

[48] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet. Multi-class object localization by combining local contextual interactions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 74

[49] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1996. 60, 61

[50] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005. 2, 4, 12, 38

[51] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 31, 37, 39

[52] W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1987. 58

[53] G. Guo, Y. Fu, C. Dyer, and T. Huang. Head pose estimation: Classification or regression? In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2008. 42, 46

[54] J. Ham, D. D. Lee, S. Mika, and B. Schlkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003. 16

[55] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988. 1, 9

[56] A. D. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005. 31, 37

[57] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986. 14, 17

[58] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004. 9

[59] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 77

[60] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 29, 38, 39, 41, 74, 83, 84, 85

[61] G. S. Kimeldorf and G. Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 1970. 45

[62] S. Kosinov and T. Caelli. Inexact multisubgraph matching using graph eigenspace and clustering models. *SSPR/SPR*, 2002. 60, 61

[63] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 42

[64] LabelMe. The open annotation tool. http://labelme.csail.mit.edu/. 86

[65] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Proceedings of Advances in Neural Information Processing (NIPS)*, 2003. 18

[66] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition (CVPR), 2006. 2, 4, 11, 12

[67] C.-S. Lee and A. M. Elgammal. Coupled visual and kinematic manifold models for tracking. *International Journal of Computer Vision (IJCV)*, 2010. 18

[68] Y. J. Lee and K. Grauman. Shape discovery from unlabeled image collections. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 39, 40, 74, 83, 84, 85, 86

[69] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004. 88

[70] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision (IJCV)*, 2008. 73, 74

[71] B. Leibe, K. Mikolajczyk, and B. Schiele. Segmentation based multi-cue integration for object detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2006. 73

[72] V. Lempitsky, C. Rother, and A. Blake. Logcut: Efficient graph cut optimization for markov random fields. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007. 66, 69, 70

[73] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005. 57, 58, 60, 61, 63

[74] M. Leordeanu and M. Hebert. Smoothing-based optimization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 62

[75] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Compter Vision and Image Understanding (CVIU)*, 106, 2007. 37, 66, 83

[76] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 42, 43

[77] D. Liu, G. Hua, P. A. Viola, and T. Chen. Integrated feature selection and higher-order spatial feature extraction for object categorization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 12

[78] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999. 1, 9

[79] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004. 9, 10, 19, 36, 42, 43, 56, 60, 71, 72, 73, 75, 83

[80] B. C. M. Fritz, B. Leibe and B. Schiele. Integrating representative and discriminant models for object category detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005. 83

[81] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 2, 4

[82] M. Marszalek and C. Schmid. Accurate object localization with shape masks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 73

[83] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Computing*, 2004. 9

[84] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 73, 74

[85] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)*, 2004. 9, 10

[86] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2005. 1, 10, 60, 69

[87] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 2005. 10

[88] P. Mordohai and G. Medioni. Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting. In *Proceedings of the international joint conference on Artificial intelligence*, 2005. 18

[89] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision (IJCV)*, 2007. 10, 60

[90] N. Morioka and S. Satoh. Building compact local pairwise codebook with joint feature space clustering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 13

[91] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision (IJCV)*, 1995. 2, 4, 17, 29

[92] E. Murphy Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2009. 42

[93] P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003. 15, 16

[94] A. Opelt and A. Pinz. Object localization with boosting and weak supervision for generic object recognition. In *Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA)*, 2005. 83, 90

[95] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 43, 44, 48, 49, 50, 51, 52

[96] C. Pantofaru, G. Dorko, C. Schmid, and M. Hebert. Combining regions and patches for object class localization. In *CVPR Workshops*, 2006. 73, 74

[97] C. Papadimitriou and K. Stieglitz. *Combinatorial Optimization Algorithms and Complexity*. Prentice Hall, 1982. 61, 65, 70

[98] PASCAL. The pascal object recognition database collection. http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html. 86

[99] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 1998. 61

[100] T. Poggio and F. Girosi. Network for approximation and learning. *Proceedings of the IEEE*, 1990. 45

[101] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007. 74, 83

[102] A. Rahimi, B. Recht, and T. Darrell. Learning appearance manifolds from video. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 46, 55

[103] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Sciene*, 2000. 15, 18, 28

[104] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 1998. 12

[105] S. Savarese and F. Li. 3d generic object categorization, localization and pose estimation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007. 42, 43, 49, 50, 51

[106] S. Savarese and F. Li. View synthesis for recognizing unseen poses of object classes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 42, 43, 49

[107] B. Schlkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002. 11

[108] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1997. 1

[109] G. Scott and H. Longuett-Higgins. An algorithm for associating the features of two images. *The Royal Society of London*, 1991. 23, 60, 61, 65, 71

[110] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2005. 62

[111] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 1992. 60

[112] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their localization in images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005. 1, 2, 12, 31, 37

[113] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *SIGGRAPH*, 2006. 62

[114] M. Stark and B. Schiele. How good are local features for classes of geometric objects. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007. 2, 4, 12, 29, 35, 36, 37, 38

[115] H. Su, M. Sun, L. Fei Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009. 42, 43

[116] M. Sun, H. Su, S. Savarese, and L. Fei Fei. A multi-view probabilistic model for 3d object classes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 43, 49, 50, 51, 52

[117] A. Talwalkar, S. Kumar, and H. A. Rowley. Large-scale manifold learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 6, 17

[118] J. Tang, D. Liang, N. Wang, and Y. zheng Fan. A laplacian spectral method for stereo correspondence. *PR Lett.*, 2007. 60

[119] J. B. Tenenbaum. Mapping a manifold of perceptual observations. In *Proceedings of Advances in Neural Information Processing (NIPS)*, 1998. 15, 18

[120] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 2000. 4, 17

[121] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 43

[122] M. Torki and A. Elgammal. One-shot multi-set non-rigid feature-spatial matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 23

[123] M. Torki and A. Elgammal. Putting local features on a manifold. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 75, 84

[124] A. B. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 1

[125] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 2, 57, 58, 62, 63, 66, 69, 70

[126] S. Ullman. Aligning pictorial descriptions: An approach to object recognition. *Cognition*, 1989. 56, 58

[127] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1988. 60

[128] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 2, 18

[129] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002. 4, 17

[130] F. Wang, X. Wang, and T. Li. Efficient label propagation for interactive image segmentation. *Proceedings of the international joint conference on Machine Learning Applications*, 2007. 75, 77

[131] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 31, 37

[132] H. Wang and E. R. Hancock. Correspondence matching using kernel principal components analysis and label consistency constraints. *Pattern Recognition*, 2006. 60, 61, 66, 69, 70

[133] X. Wang, X. Huang, J. Gao, and R. Yang. Illumination and person-insensitive head pose estimation using distance metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 42, 46

[134] Z. Wang and H. Xiao. Dimension-free afne shape matching through subspace invariance. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 22

[135] M. Weber, M. Welling, and P. Perona. Unsupervised Learning of Models for Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2000. 1, 13

[136] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 15, 18

[137] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *IWLAVS*, 2004. 1, 2

[138] R. Wilson and E. Hancock. Structural matching by discrete relaxation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1997. 62

[139] G. Wu, E. Y. Chang, and Z. Zhang. Learning with non-metric proximity matrices. In *ACM MULTIMEDIA*, 2005. 28

[140] S. Xiang, F. Nie, Y. Song, C. Zhang, and C. Zhang. Embedding new data points for manifold learning via coordinate propagation. *Knowl. Inf. Syst.*, 2009. 24, 25

[141] Z. Xue and E. K. Teoh. A novel eigenvector approach to pose and correspondence estimation. *Systems, Man, and Cybernetics IEEE International Conference*, 2000. 60, 61

[142] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2007. 17

[143] L. Yang, P. Meer, and D. J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 12

[144] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision (IJCV)*, 2007. 11, 12

[145] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *Proceedings of Advances in Neural Information Processing (NIPS)*, 2004. 76, 77, 78, 82

[146] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003. 77