TOWARDS A LOCAL-GLOBAL VISUAL FEATURE-BASED FRAMEWORK FOR RECOGNITION

BY ZHIPENG ZHAO

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, 2009

ABSTRACT OF THE DISSERTATION

Towards a Local-Global Visual Feature-Based Framework for Recognition

by ZHIPENG ZHAO

Dissertation Director: Ahmed Elgammal

General object and activity recognition is a fundamental problem in computer vision, which has been the subject of much research. Traditional approaches include model-based and appearance template-based methods. Recently, inspired by methods from the text retrieval literature, local visual feature-based models have shown a lot of success for recognition of objects or activities with large within-class geometric variability.

There are several challenges in this approach, namely feature selection and target modeling using these features. This thesis proposes a local-global visual feature-based framework for general object and activity recognition with novel methods for these problems:

1) Combinatorial and statistical methods for selecting informative parts to build statistical models for part-based object recognition. First a combinatorial optimization formulation is used for clustering on a weighted multipartite graph. Second, a statistical method for selecting discriminative parts from positive images is used to localize objects.

2) An entropy based vocabulary selection method for “bag-of-words” models for activity recognition.

3) Integrating both spatial and temporal information with appearance features for human activity recognition. This method models the human motions with the distribution of local motion features and their spatial-temporal arrangements.

The effectiveness of the proposed methods is demonstrated by several object recognition and activity recognition data sets, which include human facial expressions and hand gestures, etc.

This thesis also covers an interesting project regarding a framework of applying Discrete Fourier Transform to detect salient regions in images and video sequences. This framework generalizes the previous saliency detection methods and can be applied for saliency detection in the video sequences.

Acknowledgements

First and foremost, I would like to thank Professor Ahmed Elgammal, my thesis advisor, for his constant guidance during the past six years. Throughout my doctoral work, he encouraged me to develop my research skills. His energy, drive, and unequivocal encouragement to perform research that matters is an inspiration.

I would also like to thank the other members of my dissertation committee, Prof. Vladimir Pavlovic, Prof. Casimir Kulikowski and Dr. Vinay Shet for their advices and comments.

For all these years in the lab, I had countless interesting discussions with my lab mates, who gave me great helps in my research. Especially, I would like to thank Toufiq Parag, Ishani Chakraborty and Marwan Torki for their inputs and comments.

And I would like to use this opportunity to thank my family. My grandma, who brought me up and taught me to be a good person, with honor and integrity. I hope she is proud of me up there in heaven. My parents, who encouraged me to explore and discover the unknown at an early age. And my dear wife, Yun Ning, who provides comfort with love and allows me to focus on my research. And my lovely daughter, Veronica, who makes me happy along the way.

Last and not the lest, I would like to thank the Rutgers Ballroom Dancing Team. I had wonderful time with those undergraduate kids, during weekly practice and going out for competitions. It makes my graduate student life full of fun!

Dedication

To My Family and Those Who Have Helped Me along the Way

Table of Contents

List of Figures

List of Abbreviations

DoG	Difference of Gaussians
DFT	Discrete fourier transform
EM algorithm	Expectation-maximization algorithm
FFT	Fast Fourier Transform
HSM	Heterogeneous star model
KWIC	Key word in context
MEI	Motion energy image
MHI	Motion history image
PCA	Principal component analysis
pLSA	Probabilistic latent semantic analysis
SIFT	Scale invariant local feature
SVM	Support vector machine

Chapter 1
Introduction

1.1 Motivations

With recent advance of technology, huge amount of digital multimedia data, such as images and video clips, are produced by digital cameras, camcorders and cell phones. For example, in the financial district of London, many surveillance digital cameras are deployed on the streets to monitor suspicious activities. Every day and night, they keep feeding continuous video data into the security surveillance systems for analysis. Ordinary people are also important contributors for digital data. Thanks to the cheap prices, more and more people can afford consumer level cameras and camcorders. They can produce, post and share their photos and video clips on their own web sites, blogspace and other online media sharing services, such as image sharing service from Flickr (http://www.flickr.com), video clips sharing service from youtube (http://www.youtube.com), personal web space such as facebook (http://www.facebook.com) and myspace (http://www.myspace.com).

All these digital medial data generated everyday motivates people to explore possible business opportunities from these data. A common approach is to extract the information contained in these images and videos. However, it is very difficult and expensive to manually annotate the content for them. Currently, these data are simply too large to be handled by human experts. And even if we could do it, the expense will be very high. So automatic analysis for digital images and videos is the possible solution. Some commercial products have already been developed for analyzing these multimedia data. For example, in the field of video data analysis, On-Net Surveillance System, Inc (http://www.onssi.com/) has developed a complete suite of softwares which support video capture, content analysis and video intelligence. For image analysis, most of the famous search engines, such as Google (http://www.google.com), Yahoo Inc (http://www.yahoo.com) and Bing (http://www.bing.com) have support for image search, though most of these search are based on the text information surrounding the image to infer the content. Unlike the above, Like.com (http://www.like.com) has built one of the earliest visual-based search engines, where the visual content of the photos are used to retrieve similar items.

However, current state-of-art systems are still limited in their capabilities in handing complicated data. So how to automatically recognize the content of the media and later extract the information contained inside remains a big challenge for computer vision scientist around the world. And successfully solving this problem will lead to many applications far beyond what we previously mentioned. For examples:

1) Human computer interaction: Human gestures can be recognized by computers so that we will no longer solely reply on the traditional input devices such as keyboard and mouse to communicate with computers. Human computer interaction can be applied in applications such as signaling in high noise environment including airports and factories, sign language translation and gesture driven control for people with disability.

2) Manufacturing: The recognition of defects can be used in visual inspection for quality control during the manufacture of parts in the automotive industry or in the inspection of semiconductors. In a broader scope, the recognition of objects can help the visual control of robots during assembly of parts from pieces or during the calibration of robot control system.

3) Media analysis: Media analysis has many applications in the content-based indexing of sports video footage, personalized training in sports and image analysis for clinical study of patients.

4) Defense: The future “smart weapon” will have the recognition capability. Automatic target recognition system on these weapons can help navigate the cruise missiles or guide the air to surface missiles for targets.

1.2 The Goal and the Current Approaches

1.2.1 Goal

In order to extract the information from these multimedia data, a common approach is to recognize the objects in the images or the activities from the video sequences. Here the recognition is categorization in a more strict sense, because we want the computers to recognize all the instances in the same category, even for those the computers have not learnt before.

In the last two decades, extensive research has been conducted on general object or activity recognition in the field of computer vision. However, the state-of-art computer systems for analyzing and understanding these multi-media data are still quite limited and in many cases can only work in a constrained environment. The challenge comes from the following difficulties existing in this problem:

1) Variability within specificity: There might be large variability within the same category. For example, there are chairs with different size, style, color and texture within the same category. How to learn a generalized representation for all these different chairs in the same category from a finite number of training data is a big challenge.

2) Variations in scale, orientation and visibility: Even for the same instance of the target, the different scale caused by the distance between the object and the camera when the image is taken, the different pose which leads to the different orientations in the image, the different illumination conditions when the image is taken or the video is recorded, and whether part of the target is hidden all lead to very different intensity representation of the target in the image or video. Recognizing the target regardless those factors is very difficult.

3) Target of interest might have to be recognized in the context of multiple instances of the same or the different target and against the cluttered background: The introduction of other objects and the cluttered background makes the already difficult problem even worse. The cluttered background adds noises for the target recognition. The inclusion of other instances of objects might occlude the target and might have negative impact when inference of the target is based on the knowledge of spatial location and arrangement of the target.

Facing all these difficulties, the goal for research in the field of recognition is to find a general framework to recognize both object and the activity within the difficult context regardless of all the variations in the target class.

1.2.2 Current Approaches

Current approaches for object and activity recognition can be broadly divided into three categories: model based methods, appearance template search based methods and the local feature based methods.

In the model based methods, we learn the geometric models from the training data and infer the geometrically transformed target from the model. For object recognition in the image, we learn a geometric model for the object and later try to match the target in the test image with the geometric transformed projection of the model. For activity recognition, we can first build a three dimensional model then use it to locate and track the movement of limbs in the video sequences prior to the recognition of the activity.

However, for this method to work, we need to first obtain an accurate geometric model, which by itself is a difficult problem. And learning such a model could be an overshot for recognition. For example, for activity recognition, we might not need to track all limbs before we do the recognition. This motivates research on obtaining descriptor directly from the image or video without building the geometric model for recognition.

Appearance template search based method belongs to such methods. In this approach, appearance templates are directly learnt from the training data. For activity recognition, spatiotemporal descriptor, which could be motion energy image (MEI) and motion history image (MHI) [7] or is based on optical flow[14], is directly learnt from the data without tracking the limbs. For object recognition from the images, the appearance templates can work as classifier to search the image at different locations and scales for the best match of the target class.

However, such approaches are only successful in modeling the target with wide within-class appearance variation. These templates are often rigid and limited when the within-class geometric variation is large, such as in the case of recognizing a deformable target. An extreme example is that these approaches have difficulty in handling the occlusion problem.

Recently, local visual feature based methods for recognition have gained much attentions. In these approaches, the target is modeled as a collection of image patches or local motion features with distinctive appearance and spatial arrangements. The recognition is inferring the target class label based on the similarity in features' appearance and their arrangements. The spatial arrangements of the local visual features can be modeled as either fully connected model [17] or star model [18] or simply ignored such that all local visual features follow the "bag-of-words" model [94].

The local visual features based approach directly models the appearance and the spatial arrangements of local visual features, so it can avoid the difficult geometric modeling building process in model based method. Typically, it model the target object class on the statistical properties of the local visual features, so it is not rigid and can handle deformable target in some occlusion cases.

1.3 Our Approach and Contributions

Our approach is in line with the local visual feature based approach. We aim to build a model with both global and local, both appearance and spatiotemporal information from the local visual features for the target class. And this is a general framework, which is not be limited to specific object or action classes, and it can recognize object and action in the same fashion.

Typically, the whole process of local feature based recognition can be divided into four stages, as shown in Figure 1.1. The first stage is feature extraction and representation. Usually in this stage, low level vision feature detectors are applied to extract and represent the salient local visual features from the images or the videos. The next step is to select the informative local features that best characterize the target, because many of local features could come from the cluttered background and are irrelevant to the target. Then we try to model the target using the selected local features. In the fourth step, with the local features detected from the testing data, we will perform recognition using the model we learnt from the training data.

The work presented in this thesis address problems in all stages of the whole process.

1.3.1 Local Visual Features Selection and Its Application in Probabilistic Object Recognition Model

In this project, we try to learn a local visual feature based model for the target object in the image. The experimental setting is semi-supervised. So the training images are not segmented and we only know whether or not target objects exist in the images but not their locations. In most cases, the target objects coexist with cluttered backgrounds. So the initial salient local visual features, which are detected by low level vision detectors applied to the images, are large in number, redundant and often correspond to the clutters in the images. Finding actual object features coming from the target is essential for learning a representative object class model.

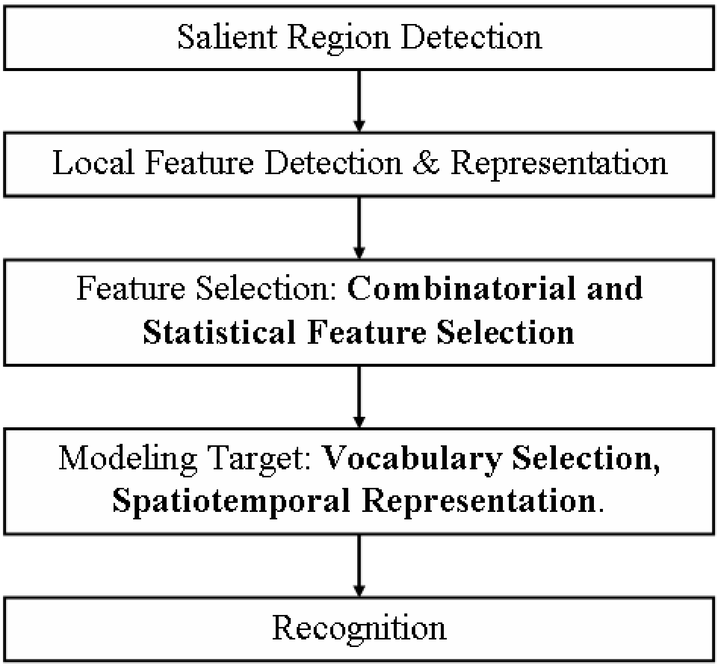

Salient Region Detection
Local Feature Detection & Representation
Feature Selection: **Combinatorial and Statistical Feature Selection**
Modeling Target: **Vocabulary Selection, Spatiotemporal Representation**.
Recognition

Figure 1.1: The workflow for a typical local visual feature based recognition framework

In our work, we introduce two complementary approaches for unsupervised selection of discriminative local features:

1) A combinatorial approach: In this approach, we want to find the best subsets of local visual features common to the positive examples but distant from the negative ones. This is a combinatorial problem because we want to find out the best subsets out of all possible combinations. We apply Akshay's [87] clustering method on a multipartite graph for this problem and obtain the optimal solution in $O(|E| + |V| * log|V|)$ time, given the defined score function for the subset.

2) A Statistical approach: In this approach, we want to find the local visual features that best discriminate between the positive and the negative classes. Inspired by the boosting method, we build classifiers upon local visual features and use their performance on the evaluation data sets as the criteria for choosing the characteristic local features.

3) Sequential combination of the above approaches: Since the above two approaches complement each other, we experiment with the sequential combination of these two methods and find it yields the best performance.

Because the above approaches do not use any property specific for computer vision problem, they can be applied for general feature selection in other applications as pre-process methods to remove noise from the original data and obtain the features from the target.

1.3.2 Vocabulary Selection for ”Bag-of-Words” Model

“Bag-of-words” model is a common modeling method for local visual features, which originated from text document representation. In this approach, words are modeled as independent from each other with the naive Bayesian assumption. And the document is represented as the joint distribution of words contained in the document. Typically, the joint distribution is approximated by the relative frequency of the words, which is the normalized histogram of the words.

Similarly, in computer vision, local features are quantized into visual words, typically via clustering on the local visual features and later assigning visual word labels to them. And the target is represented by these visual words in a “bag-of-visual-words” model similar to those in the text document representation.

However, not every visual word is equally important to represent the target. For a particular domain, some visual words are more characteristic than the others in describing the target. In our work, we introduce entropy based vocabulary selection methods for the “bag-of-words” model where the visual words are chosen based on the entropy of the clusters they come from.

In our work, we explore two methods for entropy based vocabulary selection:

1) Hard selection: In this method, we discard a certain percentage (defined as discard rate) of the least important visual words in the vocabulary. For different applications, we experiment with different discard rates for an optimal value.

2) Soft selection: In this method, we keep all the visual words in the vocabulary but assign different weights to them according to the their importance measured by the entropy.

Our approaches are different from the local visual features selection. The latter is selecting the local visual features coming directly from the data detected by the low-level vision feature detector. Our approaches are selecting clusters of local visual features based on their entropy. Since we assign visual word label to each of the cluster, this vocabulary selection is in a higher level in the recognition framework and has semantic meaning.

1.3.3 Integrating of Spatiotemporal Information into the “Bag-of-Words” Model

Similar to the original model in the text document representation, the traditional “bag-of-words” model used in computer vision also assumes the spatial and temporal independence among local visual features and ignores the spatial and temporal arrangements of local visual features. However, such arrangements could be very helpful for recognition because they provide the global context information for the local visual features.

In our work, we present two methods to capture the spatial and the temporal information in the representation for local visual features . Together with the appearance information contained in the local visual features, such rich representation has more discriminative power and can lead to better recognition performance:

1) Model the video sequence as the distribution of local visual features in a spatial-temporal pyramid structure. In this method, we recursively partition the visual features detected in a sequence along the x-y-t dimensions and use the concatenated weighted histogram from each subdivision as the representation for the sequence.

2) Model the key frames in the video sequence as the distribution of temporal integration of local visual features in a pyramid structure. In this method, we first select key frames based on the sum of the discriminative power of the visual words contained in them. Then for the key frame i, we recursively partition the visual features, which are detected in it and integrated from temporal adjacent frames, along the x-y dimension and use the concatenated weighted histogram from each subdivision as the representation for the sequence.

Because we have extended the “bag-of-words” model along two directions, namely the vocabulary selection and the integration of spatiotemporal information, we also test the sequential combination of both extensions. Experiments have shown the combination approach slightly improve the performance in most cases.

1.3.4 Salience Detection via Discrete Fourier Transform

Salience detection is an important early step in the recognition process. It indicates the regions where human usually pay attention to before the recognition takes place. Correct salience detection can lead to faster recognition in the image or from the video because the later stage of recognition, including feature detection, selection and the modeling, only need to be applied to the detected salient region.

In our work, we present a method for saliency detection from the frequency domain using Discrete Fourier Transform (DFT). This approach addresses the saliency detection as a redistribution of energy for the components with different frequency in the amplitude spectrum. After Discrete Fourier Transform of the original data, we apply logarithmic transform to the amplitude components such that the components at higher frequency can be comparable to those at lower frequency. After such transform, the amplitude components at higher frequency is no longer dominated by those from lower energy such that the reconstructed data from the transformed amplitude components with the corresponding phase components will indicate the salience regions.

This method explains other previously published methods and introduces its own logarithmic transform for the amplitude components. It can also be applied to the three dimensional data, e.g. video sequences, for salience detection.

Chapter 2

Related Work

In the last four decades, extensive research has been conducted on general object or activity recognition in the field of computer vision. One of the earliest computer vision experiments was carried out in M.I.T in the summer of 1965 [58]. It involved locating and recognizing individual block from a small database of blocks. It turned out that the recognition problem is much more difficult than people had previously expected, as the project was originally planned to be finished in a summer. Its difficulty lies in the following perspectives:

1) Large variability within the category.

2) Variations in scale, orientation and visibility for the same object.

3) Target might have to be recognized in context of multiple instances of the same or the different target and against cluttered background.

To address these difficulties, current approaches for recognition can be broadly divided into three categories:

1) Model based methods.

2) Appearance template search based methods.

3) Local visual feature based methods.

Since the research on recognition is a fundamental problem which has been studied for many years with huge amount of literature, I will only briefly review the related work in this chapter.

2.1 Model Based Method

Everyday experience tells that human usually recognizes a target by comparing it to the knowledge stored in the memory. One of the possible forms of the knowledge is the geometric model. Thus the model based approaches for object or activity recognition arise naturally. In these model based methods, we learn the geometric models from the training data and compare the target with the geometrically transformation from the model.

For object recognition in the image, we learn a geometric model for the object and later try to match the target in the test image by a geometrically transformed projection. The geometric model is typically obtained through integration of data points from several viewpoints of the training objects for information from all viewing angles. Then these data points are integrated in a coherent fashion to provide a two dimensional or three dimensional model for the target class.

Once geometric model is established, the recognition for object in the test image is carried out by matching the model and the object. This is usually performed in two steps. In the first stage, a correspondence is established between the model and the object. Such match tasks are usually solved by searching for all promising matches. Various attempts, which mainly using the geometric constrains, have been tested for reducing the search complexity. In the second stage, with the correspondence, a geometric transformation is derived such that the model can be projected onto the target in the image.

Grimson *et al.* [26, 27] proposed an object recognition system which used tree search to test all possible correspondences between the data and the model. Geometric constrains were applied to prune the search tree and avoid testing all combinations of possible correspondences. However, the number of combinations still increased rapidly with the complexity of the objects and the scenes, so it could only be used for recognition within limited conditions.

Ullman and Huttenlocher *et al.* [34, 35] suggested using a minimum amount of information with highly descriptive features in their alignment approach. Assuming pose consistency, they used a small number of pairs of model and image features to align the model with the image. Then the aligned model was compared directly with the image for verification by back projecting the object model to the image.

Other methods based on geometric hashing were suggested by Lamdan and Wolfson *et al.* [95, 43]. In these methods, an object was represented as geometric information about groups of model coordinates in a transformation invariant form and stored in hash table. At recognition phase, groups of target coordinates were used to index into the hash table and vote for possible correspondence with the model.

For activity recognition, typically a three dimensional model is built and used to locate and track the movement of limbs in the video sequences prior to the recognition of the activity. This requires a model of the body, whether a three dimensional model or a two dimensional view based model. Different models for human bodies, such as stick figures, two dimensional contours or volumetric model, are used for motion analysis of human body parts through video sequence. Usually, the the process for motion analysis can be divided into three steps [3]:

1) Feature extraction.

2) Finding feature correspondence.

3) High-level processing.

In the finding feature corresponding step, geometric model is used to establish correspondences between the images and the model data such that the tracking of features between the consecutive frames is automatically achieved.

Chen and Lee *et al.* [9] has used a stick figure to represent human body parts. This model includes 17 line segments and 14 joints. Both torso and hip parts were assumed to be rigid. Various kinetics and kinematics constraints were imposed for the analysis of the gait. Given a two dimensional projection, this method tried to recover three dimensional configuration by searching all possible combinations to locate the three dimensional coordinates of the joints and find their angles. So this method was computational expensive and required an accurate extraction of two dimensional stick figures.

Leung and Yang *et al.* [47] modeled the poses of human performing gymnastic movements with a two dimensional ribbon model. This model was made up with a body trunk, 5 U shaped ribbons, 7 joint points and several mid-points of the segments. The human outlines were extracted and used to interpret human motion. A complete outline of moving object was generated by edge detection and the side of the moving edge belongs to the moving object was determined by a spatiotemporal relaxation process. Then a description of the body parts and the appropriate body joints were obtained.

A collection of elliptical cylinders is one of the most commonly used volumetric models for human forms. Hogg *et al.* [30] has used such cylinder model to represent the human body parts with 14 elliptical cylinders. Each cylinder had three parameters: the length of the axis, the major and the minor axes of the ellipse cross section. Given the video sequence, a differentiating algorithm was applied to produce isolated regions of the moving object, which served as indications of the object's size, location and rough posture. Next, a Sobel filter [78] with a fixed threshold was used to extract the outline of the object. Then an exhaustive search was applied to find the corresponding posture, which was the best match for the image, to generate a three dimensional structural description for the walking person.

However, for these model based methods to work, we need to first obtain an accurate model, which by itself is a difficult problem. Firstly, errors present in the data acquisition step, such as digitization noise and system distortions can affect the modeling step, which in turn, could further complicate the matching step. Secondly, we need technique to register the data obtained from different view angles to generate the model in a coherent fashion. This may also introduce additional error. Thirdly, we need to incorporate all possible viewpoints to build a complete model. So the model construction phase might be difficult.

And the model based method requires geometric identification and location of the joints and body segments, which is difficult. On the other hand, learning such a model could be an overshot for recognition. For example, as pointed out by Polana and Nelson *et al.* [67], we might not need to track all limbs before we do activity recognition. In their work, they found that the movements of the torso was sufficient and they bound the walking object by a rectangle box, of which the centroid is used for tracking. So it is computationally more efficient to recognize the human activities by directly using the uninterpreted low-level visual features. This motivates research on obtaining descriptor directly from the image or video without building the geometric model prior to the recognition phase.

2.2 Appearance Template Search Based Method

Appearance template search based methods belongs to the school of methods without geometric models. In this approach, appearance templates are directly learnt from the low level visual features extracted from the training data without a geometric model. Then the appearance templates can be used to match the same low level visual features in the same representation from the testing data. The matching can be carried out using correlation or other classification methods such as support vector machine (SVM).

Another advantage of appearance template search based methods over the geometric model based methods is that the former contains appearance information, which is important for recognition while the later mostly depends on shape information, which is not sufficient. And acquiring the appearance model can be easier than acquiring the geometric models.

For object recognition from the images, the appearance template can work as a classifier to search the image at different locations and scales for the best match of the target class. H. Schneiderman and T. Kanade *et al.* [74] built a successful system to recognize faces from the images. Multiple face templates were built for different poses. And the templates were sliding across the image at different locations for the best match. Michael Oren *et al.* [65] used a similar approach to detect pedestrian by using wavelet templates. Normalized wavelet coefficients were used as templates and the detection window was shifted across all possible locations and scales. The matching value was the ratio of the coefficients in agreement or the comparison was done using support vector machine.

To speed up the process, Paul Viola *et al.* [89] adopted a variant of AdaBoosting method for feature selection and classification. In their approach, an attentional cascade of classifiers based on appearance templates, with their complexity from simpler to more sophisticate, was applied to the image. In this framework, simple, boosted classifiers could reject many of negative sub-widows, leaving the tasks of detecting all positive instances to more sophisticate classifiers. And series of such simple classifiers could achieve good detection performance which eliminated the need for further processing of negative sub-windows.

Eigen based representation is a common approach to represent target with different factors, such as poses, shapes and illumination conditions. In this approach, the eigen space, which is a lower dimensional feature subspace with image basis, is learnt from the training data. And the object, which is described as a linear combination of the image basis, can be represented as the projection into this feature space. M. Turk and A. Pentland *et al.* [84] used this approach to learn ”eigen faces”, which were the image basis in the eigen space, from the training data in a face recognition system. Then all the faces could be represented as a linear combination of these eigen faces. This approach could be further extended for learning both content and the style factors. W. Freeman *et al.* [80] have applied a bilinear model to factor out the style and the content information for recognizing letters with different styles and head poses from different persons. The work from Lee and Elgammal [46] has explored separating the style and content factors in a non linear manifold space.

Active shape and appearance model is another approach to statistically learn deformable objects through linear models of certain landmarks. In active shape model, Tootes *et al.* [11] learned a statistical model for the shape of the objects from the statistical distribution of the landmark points in the training data set. In the recognition phase, the model was iteratively deformed to fit the object in the test image. In active appearance model [10], statistical model for both object shape and appearance was learnt from the landmark distribution in the training data.

For activity recognition, spatiotemporal descriptor can be directly learnt from the data without tracking the limbs. Bobick *et al.* [7] used two temporal templates, Motion-Energy Images (MEI) and Motion-History Images (MHI) to represent activities. MEI indicated where there was motion and MHI indicated how the motion was happening. The template matching was done by using seven Hu moments [33] as features, which were reasonable discriminative in a translation and scale invariant setting.

Optical flow [31] is another commonly used low level motion feature. In the work of Polana and Nelson *et al.* [67], optical flow fields were computed between consecutive frames and partitioned into spatial grids in both X and Y directions. The motion magnitude in each cell was summed to form a high dimensional feature vectors for recognition. Efros et al [14] also proposed a spatiotemporal descriptor based on the global optical flow measurement.

However, appearance template search based approaches are only successful in modeling the target with large within-class appearance variation. These templates are often rigid and are limited when the within-class geometric variation is large, such as in the case of recognizing deformable target. An extreme example is that these approaches have difficulty in handling occlusion problems.

2.3 Local Visual Feature Based Method

Recently, the local visual feature based methods for recognition have seen many successful results [19, 50, 73, 96, 2, 8, 17, 18, 82]. In these approaches, the target is modeled as a collection of image patches or local motion features with distinctive appearance and spatial arrangements. The recognition is inferring the target class label based on the similarity of features' appearance and their arrangements. The spatial arrangements of the local visual features can be modeled either as fully connected model or star model [18] or simply ignored such that all local visual features follow the "bag-of-words" model [94].

Because these approaches directly model the appearance and the spatial arrangements of local visual features, they can avoid the difficult geometric model building process in model based methods. Typically, they model the target object class based on the statistical properties of the local visual features, so they are not rigid and can handle deformable target in some occlusion cases.

Historically, extensive research has been conducted in this direction. Here I will briefly mention a few milestones along the way. The paper by M. Fischler and R. Elschlager [19] was one of the earliest papers introducing the concept of pictorial structure, which modeled the target as a collection of local visual features. Schmid *et al.* [73] proposed a local feature based model with voting scheme. A series of papers from David Lowe [50, 51] described SIFT feature detector and representation, which is the current state-of-art local feature representation for objects in the image. Dorko *et al.* [13] introduced a part selection method, which aimed to remove local visual features detected from noisy background. For modeling of target by using local visual features, Jutta Willamowski *et al.* [94] suggested "bag-of-words" model, which originated from text document representation and Fergus *et al.* [17, 18] pioneered a probabilistic part based model.

Roughly speaking, there are four phases in the process of local feature based recognition: feature extraction and representation, feature selection, target modeling and target recognition. I will review each of them in the following subsections.

2.3.1 Feature Extraction and Representation

There are a large amount of literature regarding local visual feature detection and representation. Please refer to [49] for a comprehensive review.

Generally speaking, local visual features are meaningful, detectable parts of image or video sequence. Usually they are located in places where there exist sudden changes, such as edges, corners and where the information content is rich. For a good feature detection algorithm, it is desirable to have the following characteristics:

1) They can detect local visual features at highly informative regions such that most of the information from the image or video sequence are still retained in this sparse representation of image or video by using local visual features.

2) The number of visual features they can detect is sufficient to build robust statistical model for the target and at the same time, not too large for the propose of reducing the computational burden.

Moravec's corner detector [60] was one of the earliest corner detectors. It seeked the local maximum of the intensity changes by shifting a binary rectangle window over an image. However, the response of this detector was noisy and sensitive to edges. To reduce these shortcomings, Harris corner detector [29] was developed. This detector first computed the Harris matrix A, which was the second moment matrix and related to the derivatives of image intensity. Then the detector computed Harris matrix eigen values λ_1 and λ_2, which indicated the principal curvature of A. To reduce the computational complexity, the Harris corner metric $m_k = det(A) - kTr^2(A)$ replaced the eigenvalues for corner indication: i) if m_k is small, the pixel is in a uniform intensity region. ii) if $m_k < 0$, the pixel is on the edge. iii) if $m_k > 0$, the pixel is a corner. The detected feature point was invariant to rotation, but it failed to deal with scale changes.

David Low *et al.* [50, 51] pioneered a scale invariant local feature named the scale invariant local feature (SIFT). It included both detector and descriptor. The SIFT detector found the local maxima/minima of a series of difference of Gaussian (DoG) that occurred at multiple scales of the image. And the interest points were represented as the histograms of the gradient orientations in the scale of the points. Milolajczyk and Schmid *et al.* [57] developed the

Harris-Laplace detector by combining i) Harris corner detector. ii) the Laplace function for characteristic scale selection.

Another interesting approach for feature detection is based on salient region detection. Kadir and Brady [40] proposed the saliency region detector, which was based on the probability of density function of intensity values $p(I)$ computed over an elliptical region. The scale was selected to maximize the entropy density of the detected region.

Similar methods can be applied to video sequence for local motion feature detection, only with additional consideration for the changes in the temporal domains. These local visual features can be trajectories [98], flow vectors of corners [15, 79] or spatiotemporal interest points. Among them, spatiotemporal interest points can be obtained more reliably and thus be used widely in motion classification.

Schuldt *et al.* [76] generalized Harris corner detection to detect spatiotemporal features in the video sequences. The basic idea was to find gradients along x, y and t dimensions and the spatiotemporal corners were represented as local gradient vectors point in orthogonal directions spanning x, y and t dimensions. The second moment matrix was now 3×3 matrix and the response function was again based on the rank of this matrix. This feature detection algorithm was simple and elegant, however, the features it generated might be too sparse in some cases. To over this shortcomings, Dollar *et al.* [12] proposed a motion detector based on the application of separable linear filter. Two dimensional Gaussian kernel was applied along spatial dimensions and a quadrature pair of one dimensional gabor filter was applied along temporal dimension. Then the spatiotemporal was detected if the magnitude of the response function is larger than a threshold.

Typically, feature detection stage is followed by feature description stage. A good feature representation is desired to have the following characteristics:

1) The representation is robustic to the view point change and translation, scale and orientation transformation.

2) The representation is robustic to the change of illumination condition.

3) The representation is tolerant of object deformation and partial occlusion.

One of the earliest local descriptors used local derivatives [42]. Schmid and Mohr *et al.* [73] extended the local derivatives and used the local gray value invariants for image retrieval.

Freeman and Adelson *et al.* [20] proposed steerable filters, which were linear combination of a number of basis filters, for orientation and scale selection. The 2rd moments of detected image patches, which were computed based on the derivatives of x and y directions, were used by Van Cool *et al.*[24].

Local descriptor can also be used to represent both local and global shapes. After the edge in the image patch is detected, distance transform, which is the distance from all non-edge pixels to their nearest edge, can be used as a local shape descriptor. Shape context, introduced by Belongie [5], described the shape through distribution of the rest of the points in a polar coordinate system with a reference point as the origin.

The SIFT descriptor has been proved to be effective by many past research. The basic idea is to compute a histogram of gradient magnitudes and orientations in each cell partitioned from the neighborhood of interest points detected by SIFT feature detector. PCA-SIFT was proposed by Ke and Sukthankar *et al.* [41] to simplify the SIFT descriptor by applying principal component analysis(PCA) to normalized gradient patches. It can achieve fast matching and is invariant to image deformations.

2.3.2 Feature Selection

Because the initial number of the extracted features is large, and oftentimes the features are redundant or correspond to the clutters in the image, feature selection is important as it involves deciding which extracted features are most suitable for improving recognition rate. Finding features that come from the actual object can reduce the dimensionality of the problem and is essential for learning a representative object model to enhance the recognition performance.

Weber *et al.* [93] suggested the use of clustering algorithm to find the common object patches and to reject the background clutters from the positive training data. In this approach, large clusters were retained as they were likely to contain patches from the target object. A similar approach was used in [48]. However, there is no guarantee that a large cluster will contain only target object patches.

Dorko and Schmid *et al.* [13] extended this clustering based local feature selection approach in a supervised learning setting. In their work, classifiers, which were supporter vector machine (SVM) and Gaussian Mixture model, were built upon clusters of local features. Then the likelihood ratio and mutual information for these classifiers were used as criteria to choose the more informative cluster of local features.

Other feature selections methods combine feature selection with local feature representation. Viola and Jones *et al.* [89] selected rectangle features with an Adaboost trained classifier. Mahamud and Hebert *et al.* [54] selected discriminative object parts and developed an optimal distance measure for nearest neighbor search. Dashan Gao and Nuno Vasconcelos *et al.* [22] used discriminative saliency, which was defined as mutual information, for selecting the local visual features.

2.3.3 Target Modeling and Recognition

Since usually the modeling and the recognition phases are highly related, I will review both in one subsection.

A common approach for local visual features based modeling is to build a statistical model for the target. Both generative and discriminative models have been applied to local visual features. In the generative model, usually a joint probability distribution of the observation and the target are estimated while discriminative model directly estimates the conditional probability distribution of target given observation, which is used to predict target from the observation.

A series of work from Caltech vision groups follows the line of generative statistics modeling. Weber *et al.* [93] modeled objects as a flexible constellations of parts. A generative probabilistic model was defined through the joint probability density of parts and the hypothesis of the hidden data. This model explicitly accounted for shape variances, the randomness in the missing data due to detector error or occlusion and the image clutter. In the later work of [17], they simultaneously modeled all aspects of the object, including appearance, shape, relative scale and occlusion with a probability representation . In the learning stage, the model was estimated using expectation-maximization (EM) in a maximum-likelihood setting. In the recognition stage, this model was used in a Bayesian manner to classify images. In the work of [18], they proposed a heterogeneous star model(HSM) to simplify the training aspects of the constellation model. In this work, both learning and the recognition stages had a lower complexity such that this model could handle substantially increased number of the detected features. This enabled it to better model targets with significant intra-class variation in appearance. Xiaoxu Ma *et al.* [53] used a similar approach for vehicle classification. In their work, a repeatable and discriminative feature based on edge points and modified SIFT descriptors were used in a modified version of the constellation model. For human action classification, Niebles *et al* [62] used a probabilistic Latent Semantic Analysis (pLSA) model for a collection of spatial-temporal visual feature.

Numerous machine learning algorithms for discriminative models have been applied to the local visual feature framework. Boosting, which was proposed by Freund *et al.* [21], have been successfully used by Viola *et al.* [90] as the ingredient for a fast face detector and by Schneiderman *et al.* [75] to improve an already complex classifier. In the work of Opelt *et al.* [64], weak hypotheses were proposed based on different local visual features such as shape context and SIFT features. Boosting was used as the underlying learning technique. Recently, support vector machine (SVM) and kernel methods have begun to be used for appearance based object recognition. Pontil *et al.* [68] demonstrated the robustness of SVM to noise, bias in the registration and the moderate amount of partial occlusions. Schuldt *et al.* [76] used support vector machine (SVM) classification scheme for local space-time features in the application of human action recognition. Vidal-Naquet *et al.* [88] compared different combinations of features and classification schemes. They found out combining superior informative class-specific features with linear classification could obtain efficient object recognition than generic wavelet features with more complex Bayesian Network classification.

“bag-of-words” model is another commonly used model for local visual feature based approach. Inspired by a similar text retrieval approach, it models the objects by the distribution of words from a fixed visual code book. This code book is usually obtained by the vector quantization of local image visual features via clustering algorithm, e.g. k-means clustering. The “bag-of-words” model has been used successfully for object categorization [63, 94, 72, 51]. In the work of Schiele *et al.* [72], multidimensional receptive field histograms were used to approximate the probability density function of local appearance. David Lowe *et al.* [51] used a k-d tree with a best-bin-first modification to find the approximate nearest neighbors to the descriptor of the query. Nister *et al.* [63] implemented an hierarchical vocabulary tree to deal with the scalability of the problem. This tree allowed a larger and more discriminative vocabulary to be used efficiently. Lazebnik *et al.* [45] extended this line of work by using a hierarchical histogram to integrate the spatial information into the appearance information.

Recently, "bag-of-words" methods have also been applied in activity recognition, as demonstrated by [12, 76, 81]. In these methods, the local visual features are spatiotemporal motion features. Many work focused on a good local feature detection and representation [12, 76]. Typically they model the target with the histogram representation, ignoring the spatial temporal arrangements among the local visual features.

2.4 Saliency Detection

Saliency detection plays an important role when visual recognition must be performed in cluttered scenes. It has been a subject of research for a few decades. Broadly speaking, the saliency detection approaches can be divided into three major classes.

The first one treats the problem as detecting specific visual attributes such as edge and corners. These are usually edge and corner detectors and their detections have roots in the structure-from-motion literature. There are also approaches which use other low-level visual attributes such as contour [86]. A major limitation for these approaches is that they do not generalize well. For example, a corner detector will respond in a region that is strongly textured than a smooth region, even though the textured regions are not necessarily more salient than the smooth one.

Some of these limitations are addressed by more recent and generic formulation of saliency. One of the recent definitions for saliency is based on image complexity. Various complexity measures have been proposed in this vein. Lowe *et al.* [50] measured the complexity by computing the intensity variation in an image using the difference of Gaussian function. Sebe *et al.* [77] measured the absolute value of the coefficients of a wavelet decomposition of the image. And KaDir *et al.* [40] relied on the entropy of the distribution of local intensities. These definitions are flexible as they can detect any low-level attributes such as corners, contours and smoothed edges.

Recently, another approach for saliency detection from Fourier Frequency domain analysis has brought much attention. In the work of Xiaodi Hou *et al.* [32], the saliency regions were defined as spectral residual, which was the difference between the original signal and a smoothed one in the log amplitude spectrum. And the saliency map was obtained by transforming the spectral residual back to spatial domain. Chenlei Guo *et al.* [28] extended this line of work by transforming the amplitude components into one and only used phase spectrum of Fourier Transform for discovering the saliency region.

2.5 Summary

General object and activity recognition is one of the fundamental problems of computer vision and has been intensively studied for several decades, which leads to a large amount of literature. Traditional methods include geometric model based and appearance template search based methods. Recently, local visual feature based methods have gained their popularity. Because they are based on statistical properties of the local visual features, they are robust to large within-class geometric and appearance variance.

Typically, the process of local visual feature based methods for general object and activity can be divided into four stages, namely local feature detection and representation, local feature selection, target modeling with local features and the recognition via the model. For the modeling and recognition phases, both generative and discriminative approaches have been studied.

My work is in the vein of the local features based approach and addresses the problems in all phases of this framework. I explore the problem in the feature detection phase by introducing a method for saliency detection. I apply both statistical and combinatorial methods for local visual feature selection. For target modeling, I study a generative probabilistic model and different aspects of “bag-of-words” model for the vocabulary selection and integration of spatiotemporal information. So my work is in line with a promising approach to a fundamental problem in computer vision.

Chapter 3

Feature Selection

In this chapter, we introduce a framework that aims to select informative local visual features to build a probabilistic model for detection and categorization of target object, which is represented as a constellation of these features. The feature selection is a two stage method for choosing the local visual features which characterize the target object class and are capable of discriminating between the positive images containing the target object and the complementary negative ones. The first stage selection is done using a novel combinatorial optimization formulation on a weighted multipartite graph representing similarities between images patches across different instances of the target object. The following stage is a statistical method for selecting those images patches from the positive images which, when used individually, have the power of discriminating between the positive and negative images in the evaluation data. The individual methods have a performance competitive with the state of the art methods on a popular benchmark data set and their sequential combination consistently outperforms the individual methods and most of the other known methods while approaching the best known results.

3.1 Motivations

Object detection and class recognition is a classical fundamental problem in computer vision which has been the subject of much research. This problem has two critical components: representation of the images (image features) and recognition of the object class using this representation which requires learning models of objects that relate the object geometry to the image representation. Both the representation problem, which attempts to extract features capturing the essence of the object, and the subsequent classification problem are active areas of research and have been widely studied from various perspectives. The methods for recognition stage can be broadly divided into three categories: 3D model-based methods, appearance template search-based methods, and patch-based methods. 3D model-based methods [96] are successful when we can describe accurate geometric models for the object. Appearance based matching approaches are based on searching the image at different locations and different scales for the best match to an object "template" where the object template can be learned from training data and act as a local classifier [89, 74]. Such approaches are highly successful in modeling objects with wide within-class appearance variations such as in the case of face detection [89, 74] but they are limited when the within-class geometric variations are large, such as in detecting a motorbike.

In contrast, object recognition based on dense local "invariant" image features have shown a lot of success recently [19, 50, 73, 93, 2, 8, 17, 82, 18] for objects with large within-class variability in shape and appearance. In such approaches objects are modeled as a collection of patches or local features and the recognition is based on inferring object class based on similarity in patches' appearance and their spatial arrangement. Typically, such approaches find interest points using some operator such as [40] and then extract local image descriptors around such interest points. Several local image descriptors have been suggested and evaluated, such as Lowe's scale invariant features (SIFT) [50], entropy-based scale invariant features [40, 17] and other local features which exhibit affine invariance such as [4, 85, 71]. Other approaches that model objects using local features include graph-based approaches such as [16].

An important subtask in object recognition lies at the interface between feature extraction and their use for recognition. It involves deciding which extracted features are most suitable for improving recognition rate [93], because the initial set of features is large, and often features are redundant or correspond to the clutter in the image. Finding such actual object features reduces the dimensionality of the problem and is essential to learn a representative object model to enhance the recognition performance. This is precisely the focus of this paper: selecting the "best" features from the already extracted image features that are both exclusive and well represented in different images of the target object.

Unsupervised selection of discriminative patches is a fundamental problem for learning object models. Weber *et al.* [93] suggested the use of clustering to find common object patches and to reject background clutter from the positive training data. In such an approach large clusters are retained as they are likely to contain patches on the target object. A similar approach has been used in [48]. However, there is no guarantee that a large cluster will contain only object patches. Since the success of recognition is based on using many local features, such local features typically correspond to low level features rather than actual high level object parts. In this paper we introduce two complementary approaches to select discriminative object patches from a pool of patches extracted from the training images.

3.1.1 Contributions

We introduce two novel approaches for unsupervised selection of discriminative patches that explicitly takes into account the contrast between positive and negative examples in the training data. The first is a combinatorial optimization approach which optimally finds the best subsets of features common to the positive examples and distant from the negative examples. The second is a statistical approach which finds features that best discriminate the positive and negative examples. Experimental results show that each of the approaches enhances the recognition rate significantly. Since the two approaches are complementary in the way they select features, combining the two approaches in a sequential manner enhances the results even further. Finally, we use a probabilistic Bayesian approach for recognition where the object model does not need a reference patch [17]. Instead, object patches are related to a common reference frame.

The organization of this chapter is as follows. Section 3.2 formulates the problem of finding distinctive image patches from the positive images as a combinatorial optimization problem and a statistical problem, which are our main foci and are described in sections 3.3 and 3.4, respectively. Section 3.5 describes our recognition method and section 5.6 presents the results of applying the proposed methods on a benchmark dataset. Section 3.7 is the conclusion.

3.2 Problem Formulation and Framework

The problem we address can be stated as: Given a pool of local features (patches) extracted from a set of labelled training images containing positive and negative images of the target class, how can we choose (in an unsupervised way) the best features representing the object.

As feature extraction is not the primary focus of our investigation, we used the popular Kadir and Brady's feature extractor [40] to get the initial set of image patches for representing an image. Also we used a probabilistic method similar (in spirit) to [17, 61] for modeling the object class and for recognition. These choices allow us to focus on selecting the distinctive image patches from the positive class. The proposed selection algorithms are not tied by any means to the chosen feature extractor or recognition algorithm used in this paper and therefore can be used with any features and any recognition algorithm.

Undoubtedly, there can be many approaches for selecting a collection of image patches from images in the training data. Naively, it seems plausible to select patches from both the negative images and positive images, and classify a test image in the class to which it is closest. However, the space of negative images, devoid of any instance of the target image, is prohibitively large to allow any generalization on the negative class. So, one should rather train the classifier on the positive images using patches which are common to most of the positive images. This is based on the assumption that salience features of the target object will be present and captured from most of the positive images, and form a good representation for it. A potential side effect of focusing entirely on the positive images is the selection of undesirable patches corresponding to the background. A solution to this is by simultaneously considering the positive and negative images for selection the image patches representing both the saliencies of the target object while at the same time being exclusive/discriminative to the positive class. We present two approaches for realizing such a selection - a combinatorial approach and a statistical approach.

The combinatorial approach involves finding the subset of similar image patches shared in most of the positive images. To endow a discriminative power to the selected patches, we also consider their similarity to patches from the negative images. Thus, we wish to find such a subset of patches from the positive images where every patch is distant from the negative patches in the training data but highly similar to patches (in the selected subset) from other images in the positive images. Such selection is formulated as a combinatorial optimization problem on multipartite graph and is described in details in section 3.3.

Whereas the above described approach is a subset selection approach, the statistical approach analyzes an individual image patch from positive patches in an attempt to find patches which are both detectable and distinctive to the object class. This is achieved by determining if the patch has the power of discriminating between the positive and negative images in the evaluation data. Every patch from the positive training data is evaluated based on its performance in separating the positive and negative images in the evaluation data which was set aside from the training data a priori. If the image patch accurately predicts a significant number of evaluation images, it is selected. A detailed description of this procedure is provided in section 3.4.

The two approaches complement each other - apart from the obvious combinatorial and statistical nature of the formulation, the first does not involve any evaluation while evaluation is an integral part of the latter approach. Combinatorial selection mostly focuses on selecting patches which are over-represented in an ensemble of images of the target object, in contrast the statistical selection focuses on finding class-specific patches. From this perspective, combinatorial selection can be characterized as a method which has a low probability of losing a typical patch present in an image of the target object. On the other hand, the statistical selection is a method for eliminating, with high probability, patches which do not strongly belong to the target object. Due to their complementarity, one expects to gain by combining them. One way of combining them retaining advantages of both the methods is to initially use the combinatorial method for selecting the over-represented patches, and subsequently use the statistical method for filtering out the patches (from those selected at the first stage), which are not specific to the target object.

3.3 Combinatorial Selection of Characteristic Image Patches

We formulate the problem of finding the set of image patches that can help in discriminating between image with and without the target object as an combinatorial optimization problem on a multipartite graph. We first introduce some notations which will help in formalizing this problem. Suppose we are given a set $V^+ = \{V_1^+, V_2^+, \ldots, V_p^+\}$ of p images (positive class) containing the instances of the target object, and a set $V^- = \{V_1^-, V_2^-, \ldots, V_n^-\}$ of n images (negative class) which do not contain the target object. Recall that any arbitrary image is represented as a set of m salient image patches, so the image i^{th} from the positive class can be denoted as $V_i^+ = \{v_{i1}^+, v_{i2}^+, \ldots, v_{is}^+, \ldots, v_{im}^+\}$, where v_{is}^+ is the s^{th} image patch. Further, we also use V^+ to denote the set of all patches in V_1^+ through V_p^+, i.e. $V^+ = \cup_{\ell=1}^{p} V_\ell^+$; similarly, $V^- = \cup_{\ell=1}^{n} V_\ell^+$. The usage will become clear from the context.

We are interested in finding the subset of image patches from the set V^+ which are very similar to each other and, at the same time, distant from those in the set V^-. Furthermore, while finding image patches that characterize the target object, it is best to focus on similarities between image patches across different instances of the target object, rather than similarities between patches from the same image although they may be very similar. These two informal requirements can be conveniently expressed in a multipartite graph representation of the similarities between image patches from different images, as shown in Fig. 3.1. The right part of this figure shows an undirected edge weighted vertex weighted multipartite graph, $G = (V^+, E, W, N)$, with p partite sets V_1^+ through V_p^+ so that, as described earlier, $V^+ = \cup_{\ell=1}^{p} V_\ell^+$. The edges in the set $E \subseteq \cup_{i \neq j} V_i^+ \times V_j^+$, represent similarity between the image patches from different images while the weight w_{ab} on the edge connecting the vertices corresponding to the patches a and b represents the strength of their similarity. Each vertex in V^+ is also associated with a weight $N : V^+ \to \mathbb{R}^+$ which reflects its aggregated similarity to images patches in V^-. For any vertex $i \in V^+$, its vertex weight $N(i)$ is calculated as $N(i) = \sum_{s \in V^-} m_{is}^2$, where m_{is} is the similarity between image patch i and the image patch s from a negative image.

We consider the situation where the negative the images in training set do not contain any instance of the target object, and the positive images contain exactly one instance of the target object. Of course, it is possible to model more complex situations where the postive images contain multiple instances of the target object. However, we have focused on modeling the simpler situation. We now formulate the optimization problem for finding the subset of image patches which are characteristic of positive images and distant from patches in the negative images. In other words, we want to find a subset $H \subseteq V^+$ (so, $H = \cup_{\ell=1}^{p} H_\ell$, where $H_\ell \subseteq V_\ell^+$) of image patches from the positive images in which patches are very similar to each other and at the same time different from image patches in the negative images. To achieve this, any subset H is assigned score the $F(H)$ which measures the degree of similarity between the patches from different images in H and also their distinction from patches in V^-. This score is designed to be higher, as described later, for desirable subsets. The best subset, H^* is the

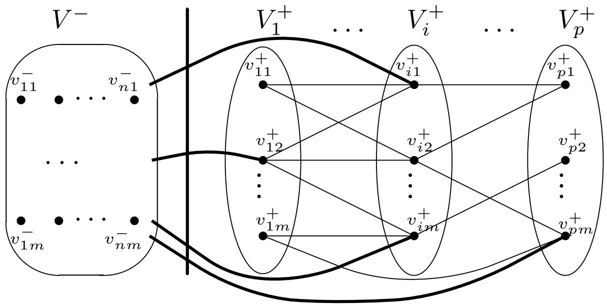


Figure 3.1: A multipartite graph representation for expressing similarity relationships between the image patches. Ellipse corresponding to V_i^+ represents the i^{th} instance of target image, and the m points inside this ellipse represent the image patches from this image. The patches from the images that do not contain the target object are represented inside the oval V^- without distinguishing between the images of those patches. The straight lines connecting the images patches across different instances of images represent the weighted similarity between them, while the thick curved lines represent the aggregated (weighted) similarity between an image patch from positive image to all image patches in the negative class. For visual clarity, weights are not shown on the edges.

globally optimal solution for the following criterion.

$$H^* = \arg \max_{H \subseteq V^+} F(H) \tag{3.1}$$

The score $F(H)$ is defined using a linkage function $\pi(i, H)$ which measures the degree of similarity of the patch i to patches from the other images in H.

$$F(H) = \min_{i \in H} \pi(i, H) \tag{3.2}$$

Thus, the score $F(H)$ for the subset H is linkage function value, $\pi(i, H)$, for the least similar patch in H. Then, the optimal solution H^* described in (3.1) corresponds to the subset of image patches where the similarity of the least similar patch is maximum.

The design of the linkage function is critical for a suitable problem formulation. It must be remarked that we only have the pairwise similarities between the image patches from different images and using this we must design the function $\pi(i, H)$. Also, recall that H is a multipartite subset, i.e. $H = \cup_{\ell=1}^{p} H_\ell$ where $H_\ell \subseteq V_\ell^+$ is a subset of patches from the image V_ℓ^+. If w_{ij} is the similarity value between the image patch from i from the image $I(i)$ and the image patch j from the image $I(j)$, then the linkage function is defined as:

$$\pi(i, H) = \sum_{\substack{\ell=1 \\ \ell \neq I(i)}}^{p} \left(\sum_{j \in H_\ell} w_{ij}^2 - \sum_{k \in V_\ell^+ \setminus H_\ell} w_{ik}^2 \right) - \beta N(i) \tag{3.3}$$

where $\beta \in \mathbb{R}^+$ is a constant factor for scaling $N(i)$, the weight associated with the vertex (i), defined as the aggregated similarity of i to the patches from the negative images. This scaling factor β serves to account for any imbalance between the number of positive and negative instances of the target object. The first term ($\sum_{j \in H_\ell} w_{ij}^2$) in the linkage function aggregates the similarity of the patch i from image $I(i)$ to patches from other images present in H. The second term ($\sum_{k \in V_\ell^+ \setminus H_\ell} w_{ik}^2$) estimates how the patch i is related to patches not included in H_ℓ. A large positive value of the linkage function $\pi(i, H)$ indicates that i is very similar to patches in H and different from the patches in the negative images or the patches from the positive images not included in H. According to this definition of linkage function, the optimal solution, H^* corresponds to a collection of image patches from different positive images each of which is highly similar to each other (as the least similar patch is highly similar to other patches) and very different from the patches in the negative images. So, such a formulation indeed serves our purpose of selecting characteristic and discriminative image patches.

This combinatorial optimization problem has been studied in [87] and it has been shown that an efficient algorithm exists for finding the global optimal solution H^* if the linkage function $\pi(i, H)$ is monotone increasing. The monotone increasing property requires that the value of the linkage function for the vertex i can only increase when the second argument H increases in a set theoretic sense, i.e. monotone increasing linkage function satisfies the condition: $\pi(i, H) \leq \pi(i, H \cup \{k\})$ for all $i \in H$ and for all $k \in V^+ \setminus H$. Indeed the linkage function defined in (3.3) satisfies this property. Observe that the third term $\beta N(i)$ is the vertex weight for i and is independent of H, so it does not affect the monotonicity property. Consider the effect of augmenting the subset H, by including $k \notin H$, on the linkage function value for the element i: when k is included in H, the value w_{ik} is deducted from the second term and added to the first term. So, $\pi(i, H \cup \{k\}) - \pi(i, H) = 2w_{ik}^2 \geq 0$, or $\pi(i, H) \leq \pi(i, H \cup \{k\})$.

Algorithm 3.3.1: ALGORITHM FOR FINDING $H^*()$

```
$t \leftarrow 1;\ \ H_t \leftarrow V^+;\ \ H^* \leftarrow V^+;$
$F(H^*) \leftarrow \min_{i \in V^+} \pi(i, V^+)$
while $(H_t \neq \emptyset)$
do
    $M_t \leftarrow \{\alpha \in H_t : \pi(\alpha, H_t) = \min_{j \in H_t} \pi(j, H_t)\};$
    $F(H_t) \leftarrow \min_{j \in H_t} \pi(j, H_t);$
    if $(H_t \setminus M_t) = \emptyset) \vee (\pi(i, H_t) = 0\ \ \forall i \in H_t)$
    then
        output $H^*$ as the optimal set and
            $F(H^*)$ as the optimal value.
    else
        $H_{t+1} \leftarrow H_t \setminus M_t;$
        $t \leftarrow t + 1;$
        if $(F(H_t) > F(H^*))$
          then
            $H^* = H_t;$
```

The algorithm for solving this combinatorial optimization problem is given [87], and is described in the pseudocode form in Algorithm 3.3.1 . This iterative algorithm begins by calculating $F(V^+)$ and finds the set M_1 containing the set of vertices from V^+ which have the minimum value of the linkage function i.e. $M_1 = \{\alpha \in V^+ : \pi(\alpha, V^+) = \min_{j \in V^+} \pi(j, V^+)\}$. The vertices in the set M_1 are removed from V^+ and the set H_2 is constructed as $H_2 = V^+ \setminus M_1$. At this point, the second iteration begins with the calculation of $F(H_2)$ and finds the set M_2. At the iteration t, the algorithm considers the set H_t as the input, calculates $F(H_{t-1})$, finds the subset M_t such that $F(H_{t-1}) = \pi(j, H_{t-1}),\ \forall j \in M_t$, and removes this subset from H_{t-1} to produce $H_t = H_{t-1} \setminus M_t$. Finally, the algorithm terminates at the iteration T, when $H_T = \emptyset$ or when $\pi(i, H_T) = 0\ \forall i \in H_T$. It outputs H^* as the subset H_j with the smallest j such that $F(H_j) \geq F(H_l)\ \forall l \in \{1, 2, \ldots, T\}$.

This problem formulation gives us one subset of similar image patches from the positive images and likely corresponds to some characteristic in the target object in those images. However, often an object has multiple salient characteristics, and these disjoint subset of patches corresponding to different characteristics of the target object can be found by removing the optimal solution H^* from the set V^+ and solving the optimization problem on the reduced set $V^+ \setminus H^*$. Thus, sequentially solving this optimization problem until we get optimal solutions with large values allows us to find the desired groups of image patches.

A complexity analysis of the method can be found in [87]. It runs in $O(|E| + |V| \log |V|)$ time, where E and V are the set of edges and vertices, respectively, in the graph.

3.4 Statistical Image Patch Selection

In the previous section we had focused on a combinatorial optimization formulation for finding subsets of patches characterizing the images from the positive class, and hopefully corresponding to salient regions in the target object. In this section, we formulate the same problem in a statistical framework by selecting, in isolation, those patches from the positive images which consistently appear in multiple instances of the positive images but only rarely appear in the negative images (barring some hypothetical and pathological cases). Intuitively, if an individual image patch from a positive image performs well in recognizing the images of the target object, a combination of a number of such image patches is likely to enhance the overall performance. This is because, barring a few pathological cases, the individual classifiers, although weak, can synergistically guide the combined classifier in producing statistically better results.

Our approach is different from the Boosting method [82]. Boosting is originally a way of combining classifiers and its use as feature selection is an overkill. In contrast, our statistical method does not boost the previous stage but filters out the over-represented and undesirable clusters of patches corresponding to background. In spirit, our approach is similar to [13]. We formalize this intuitive statistical idea in the following straightforward yet effective method for selecting the characteristic image patches, as complementary to the combinatorial selection method, which is the main contribution of this paper.

We select an image patch $v \in V^+$ from the positive images in the training data if it is able to discriminate between the positive and negative images in the evaluation data, $V_e = \{V_e^+, V_e^-\}$ with a certain accuracy. A complete description of this method requires description the classification method using a single image patch and the accuracy threshold. For classifying an image $\mathcal{V} \in V_e$ in the evaluation set, using a single image patch $v \in V^+$, we first calculate the distance, $D(\mathcal{V}, v) = \min_{\nu \in \mathcal{V}} d(\nu, v)$, between $\mathcal{V}$ and v defined as the Euclidean distance between v and the closest image patch from $\mathcal{V}$. For classifying the images in the evaluation data, we use a threshold, t on distance $D(\mathcal{V}, v)$; if $D(\mathcal{V}, v) < t$, the image $\mathcal{V}$ is predicted to contain the target object, otherwise not. Accordingly we can associate an error function, $\mathcal{E}r(\mathcal{V}, v, t)$ (defined below 6.19), which assumes a value 1 if and only if the classifier makes a mistake.

$$\mathcal{E}r(\mathcal{V}, v, t) = \begin{cases} 0, & \text{if } (D(\mathcal{V}, v) < t \wedge \mathcal{V} \in V_e^+) \vee \\ & \quad (D(\mathcal{V}, v) \geq t \wedge \mathcal{V} \in V_e^-) \\ 1, & \text{otherwise} \end{cases} \tag{3.4}$$

The performance depends on the parameter t, so we find an optimal circular region of radius t_v around v which minimizes the error rate of the classifier on the evaluation data. Finally, only those image patches from the positive images are selected which have recognition rate above a threshold, θ. A description of this algorithm, in the form of a pseudocode, is given in Algorithm 3.4.1. This algorithm takes the positive image patches V^+, patches from the evaluation data V_e, and the threshold θ as input and outputs $\widehat{H} \subseteq V^+$, the subset of selected image patches.

Algorithm 3.4.1: SELECT PATCHES, $\widehat{H}(V^+, V_e, \theta)$

```
$\widehat{H} \leftarrow \emptyset$;
for each $v \in V^+$
  do
    for each $\mathcal{V} \in V_e$
      do $D(\mathcal{V}, v) = \min_{\nu \in \mathcal{V}} d(\nu, v)$;
    $t_v \leftarrow \arg\min_{t \in \mathbb{R}^+} \sum_{\mathcal{V} \in V_e} \mathcal{E}r(\mathcal{V}, v, t)$
    $err \leftarrow \frac{1}{|V_e|} \sum_{\mathcal{V} \in V_e} \mathcal{E}r(\mathcal{V}, v, t_v)$
    if $(err < \theta)$
      then $\widehat{H} \leftarrow \widehat{H} \cup \{v\}$
```

3.5 Patches Based Probabilistic Model

Following the selection of characteristic image patches from the positive images, we used a probabilistic method for object class recognition. The selected image patches were used, simultaneously, to build a probabilistic model for the object class and the object reference frame. We assumed that a correctly classified object should also have a good approximated reference frame. In our work, we use centroid as the reference frame. Using the m observed image patches $v_k, \ (k = 1, \ldots, m)$, the problem of estimating the probability $P(O, C|V)$ of object class O and its centroid C given the image V can be formulated as (assuming independence between the patches and using Bayes' rule):

$$P(O, C|V) = \frac{P(V|O, C)P(O, C)}{P(V)} = P(O, C) \prod_{k=1}^{m} \frac{P(v_k|O, C)}{P(v_k)} \tag{3.5}$$

We wish to approximate the probability $P(v_k|O, C)$ as a mixture-of-Gaussians model using the observed patches from the training data. We simplify this by clustering all the patches selected from the training data into n clusters, $A_i, \ i = 1, \ldots, n$ and decompose $P(v_k|O, C)$ as

$$\begin{aligned} P(v_k|O, C) &= \sum_{i=1}^{n} P(v_k|A_i)P(A_i|O, C) \\ &= \frac{\sum_{i=1}^{n} P(v_k|A_i)P(O, C|A_i)P(A_i)}{P(O, C)} \end{aligned} \tag{3.6}$$

Substituting (3.6) in (5.5), we get

$$P(O, C|V) \propto \prod_{k=1}^{m} \frac{\sum_{i=1}^{n} P(v_k|A_i)P(O, C|A_i)P(A_i)}{P(v_k)} \tag{3.7}$$

While performing recognition, the term $P(v_k)$ can be ignored. Assuming that $P(C)$ and $P(O)$ are independent, we have

$$P(O, C|V) \propto \prod_{k=1}^{m} \sum_{i=1}^{n} P(v_k|A_i)P(O|A_i)P(C|A_i)P(A_i) \tag{3.8}$$

Since the clusters contain similar good features, we can assume that both the patch v_k and the centroid C from a cluster follow normal distribution. By calculating the sample mean and the sample covariance of these clusters, we can approximate the probability of v_k and C for each cluster $A_i, i = 1, \ldots, n$. We use μ_i^v and μ_i^c to denote the sample means for v_k and C, respectively, and Σ_i^v and Σ_i^c to denote the sample covariances for v_k and C, respectively. Then for cluster A_i we have $P(v_k|A_i) \sim \mathrm{N}(v_k|\mu_i^v, \Sigma_i^v)$ and $P(C|A_i) \sim \mathrm{N}(C|\mu_i^c, \Sigma_i^c)$. The rest of the terms in (3.8), can be approximated using the statistics from each of the cluster $A_i, i = 1, \ldots, n$. If the Cluster A_i has n_i points of which n_{ij} belong to the Class O_j, we can estimate the following: $P(A_i) = n_i / \sum_{i=1}^{n} n_i$ and $P(O_j|A_i) = n_{ij}/n_i$[1].

Now we can calculate equation (3.8). The result will give us an estimate for the probability of finding an object class centroid. If it is larger than a threshold, it will indicate the presence of an instance of the object class in the image. Equation 3.8 can be interpreted as a probabilistic voting where each patch gives a weighted vote for the object class and centroid given its similarity to each of the clusters. This formulation extends to handle scale variations by considering each pair of patches instead of each individual patch.

3.6 Experiment

3.6.1 Data Set

We applied the proposed image patch selection methods for recognizing images from the Caltech database (http://www.vision.caltech.edu/html-files/archive.html). This database contains four classes of objects: motorbikes, airplanes, faces, car rear end which have to be distinguished from image in the background data set, also available in the database. Each object class is represented by 450 different instances of the target object, which were randomly and evenly split into training and testing images. Of the 225 positive images set aside for selecting the characteristic image patches, 175 were used as the training images and the remaining 50 were spared to be used as evaluation data. In addition, the evaluation data also consisted of 50 negative images. The combinatorial and the statistical methods used the training and evaluation images slightly differently - while the combinatorial method selected images patches by simultaneously analyzing 175 positive (remaining 50 positive images from the evaluation data were not used in this method) and 50 negative images from the evaluation data, the statistical method selected patches from 175 positive images by judging their performance on 50 positive and 50 negative images in the evaluation data. The details of the data sets are summarized in table 3.1 and some samples from the data sets are shown in figure 3.2.

[1] It must be remarked that this model can be extended for modeling multiple object classes directly, however, since our problem consists of only one class, we have $P(O_j|A_i) = 1$.

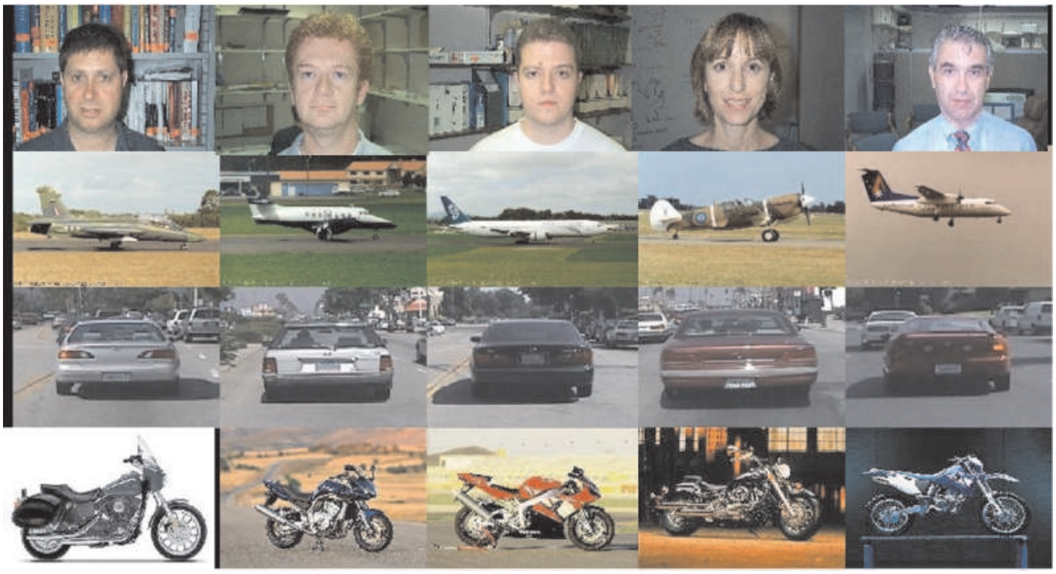


Figure 3.2: Sample images from the experiment data sets.

	Train	Evaluate		Test	
	Positive	Positive	Negative	Positive	Negative
Each class	175	50	50	225	225

Table 3.1: Details of the data sets used in our experiments.

3.6.2 Image Patch Detection and the Intensity Representation

We used region-based detector [40] for detecting informative image patches. This method finds regions that are salient over both location and scale. For each point on the image, an intensity histogram $P(I)$ is computed from a circular region of radius s. The entropy $H(s)$ of this histogram is then calculated and the local maxima of $H(s)$ are candidate scale for the region. The saliency of the region is measured by the entropy density, which is the entropy of the region over the area of the region. Then the region with the highest entropy density will provide the feature for learning and recognition. And each feature is defined by the location of the interest point and the scale for the maximum entropy density.

In our experiment, this region based local visual feature detector gives stable identification of features over different sizes and copes well with intra-class variability. The saliency measure is designed to be scaling invariant, the experiments have shown this is not the case because of aliasing and other effects. Please not that we turned the image to gray level image for representation to remove the bias in color.

Once we find the location and the scale for the image patch, we performed normalization for intensity and re-scaled the image patches to 11×11 pixels, and thus representing them as a 121 dimension intensity vectors. Then, principal component analysis (PCA), which is a common linear method for dimension reduction, was applied on these vectors to get a more compact 18 dimension intensity representation.

3.6.3 Experiment Setting

We extracted 100 image patches for each of the 175 training images, and 100 evaluation images. Following this, we applied the combinatorial and statistical methods individually and in a combination for removing the image patches from the background.

For the combinatorial image patch selection, we converted the Euclidean distance, $d(i, j)$ between the features from the patches i and j from different images to the similarity value $w_{ij} = d_{max} - d_{ij}$. The similarity values were thresholded using an empirically calculated value to convert the complete multipartite graph into a sparse graph containing 10% of the original edges. The same similarity threshold was used for considering similarity between patches from positive and negative images. We used $\beta = 3.0$ in the linkage function (3.3) to account for the imbalance in the number of positive images (175) and the negative images (50) used in the training data.

For statistical image patch selection, we built a simple classifier from each image patch in the training images and selected the one which led to a classifier with classification error rate less than 24%, an empirically calculated value.

We also used a sequential combination of the two methods. Figure 3.3 shows results from the three methods (statistical, combinatorial and their combination) for selecting image patches. The results show that both approaches are successful in removing a significant number of patches corresponding to background and the sequential combination of the methods performs the best.

After the image patch selection process, we computed the centroid for each object in the image. We used a 2D offset between the image patch and the object centroid as the spatial

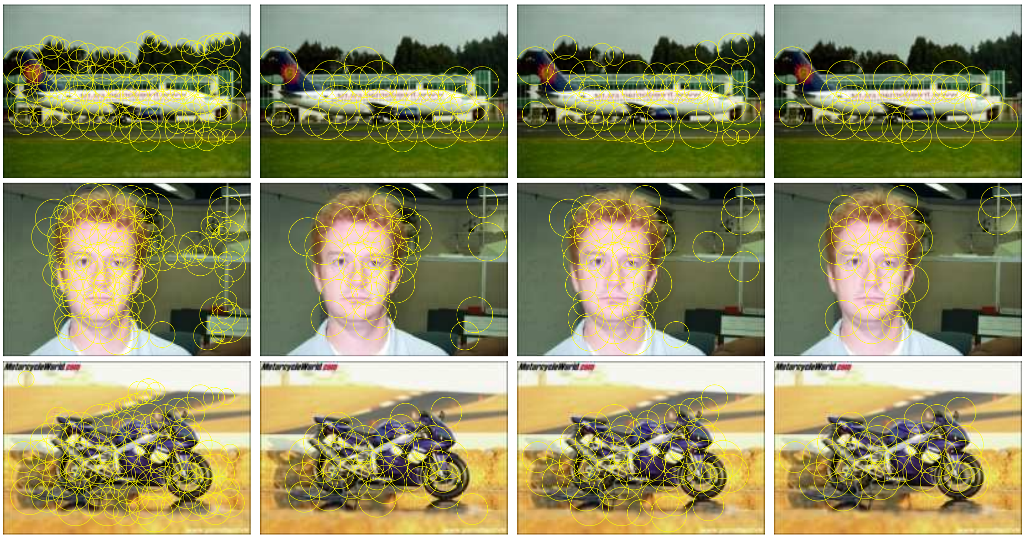


Figure 3.3: Image patch selection. The image patches are shown using a yellow circle on the images. The first column shows the image patches extracted by Kadir & Brady's feature detector. The second and third columns show image patches selected by combinatorial and the statistical methods, respectively. The patches selected by the sequential combination of the method are shown in column four.

feature for the image patch and concatenated it with the intensity feature vector as the feature representation for each image patch. We then used k-means algorithm for clustering them into 70 clusters (this number was empirically chosen) and calculated the statistics for them.

3.6.4 Experimental Result

In the testing phase, we used Kadir & Brady's feature detector for extracting the image patches. Then we calculated the probability of the centroid of a possible object in the image as an indicator of its presence.

Figure 3.4 shows the computationally estimated centroid for the object along with the image patches which contributed towards estimating this centroid. Observe that the estimated centroid was mainly voted by the image patches located on the object. It also shows some examples of misclassification. There are three major reasons for such misclassification. The first is the presence of multiple target objects in the image, as shown in the airplane example. In this scenario, there is no centroid which gets a strong probability estimation from the matched patches. The second is poor illumination conditions which seriously limits the number of initial image patches extracted from the object, as illustrated by the face example. Finally, as shown

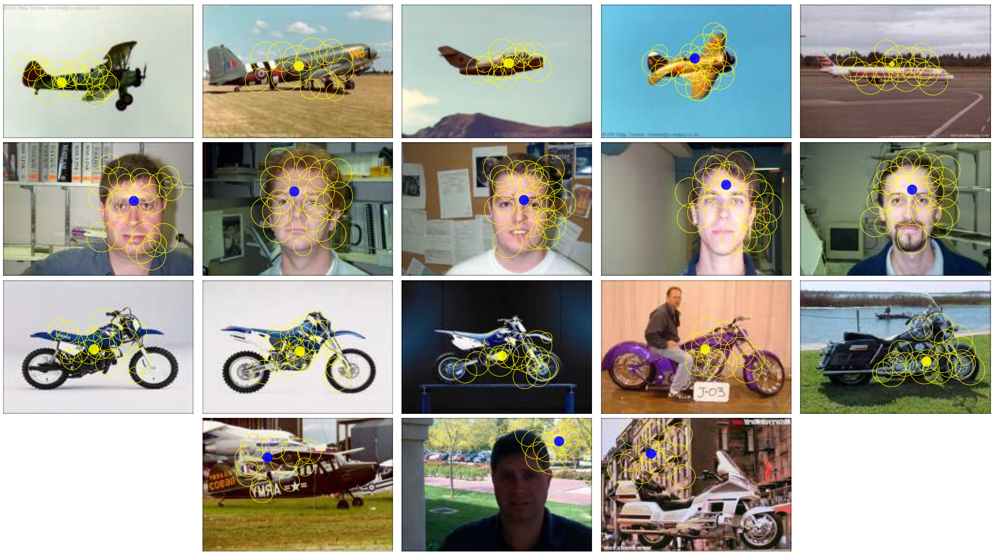


Figure 3.4: This figure demonstrates the estimation of object centroid in some typical testing image using the sequential combination of combinatorial and statistical approach. The estimated centroid is indicated by a dot with color contrast to the object. All the image patches contributed to this estimation are indicated by yellow circles. The bottom row of the images are some misclassification examples.

Dataset	No selection	combinatorial method	statistical method	combination	Fergus [17]	Opelt [64]
Airplane	54.2	88.9	94.4	95.8	90.2	88.9
Motorbike	67.8	92.9	94.9	95.8	92.5	92.2
Face	62.7	97.6	98.4	98.9	96.4	93.5
Car (rear)	65.6	97.8	96.7	99.3	90.3	n/a

Table 3.2: ROC equal error rates using different methods.

in the motorbike example, when the background is cluttered the initial patches are extracted from all over the image leading and, thereby, confusing the estimator.

We compared our result to the state of the art results from [17] and [64]. Table 3.2 gives the ROC equal error rates of our different approach and results from other recent methods. This shows our approaches yield comparable or better performance. The results are shown for no selection, combinatorial method only, statistical method only and the sequential combination of combinatorial and statistical methods. These results are also compared to other recent methods reporting equal error rate using this data set. We see that both the proposed methods perform well and their combination improves the recognition rates even further and yielding better results, quite often by a significant margin, than previous methods.

3.7 Summary

In this chapter, we have presented a combinatorial and a statistical method for selecting informative image patches for patch-based object detection and class recognition. The combinatorial visual feature selection method formulates problem as a combinatorial optimization problem on a weighted multipartite graph representing similarities between images patches across different instances of the target object. The statistical method selects those images patches from the positive images which, when used individually, have the power of discriminating between the positive and negative images in the evaluation data. Both of these methods when used alone and in combination, yield competitive recognition rates, and surpass the performance of many existing methods. Although these methods have been demonstrated in the context of image patch selection, they are general methods suitable for selecting a subset of features in other applications, which might be applied in other domain, e.g. for noise reduction in the preprocessing phase for data analysis.

Once we have the selected informative local visual features, we build a probabilistic model for object detection and recognition. This model is a generative model, which also incorporates into the representation the simple spatial information as two dimensional offset of each selected local features to the centroid of the object. Because we model the joint distribution of the object class and its location, we can detect and recognize the target object simultaneously.

Chapter 4

Vocabulary Selection

In this chapter, we present an entropy based vocabulary selection method which is used in the "bag-of-words" model for human action recognition. Inspired by the stop-words list concept in the text retrieval literature, this approach learns a list of insignificant and uncharacteristic visual words measured by their conditional entropy in the application domain and selects the more meaningful words for the representation of human action. The resulted model of "bag-of-meaningful-words" will be more compact and better representation for the target.

Our approach attacks the problem of visual word vocabulary selection in the middle level of "bag-of-words" framework and is different from other low level visual features selection methods. Instead of selecting each local visual feature detected by feature detector, this approach chooses clusters of local visual features based on their semantic meaning. Experiments have demonstrated improved performance over the baseline "bag-of-words" model and exceed or close to other known methods on the popular benchmark data sets.

4.1 Motivations

Recognizing human action from video sequences is a classical fundamental problem in computer vision with many applications including motion capture, human-computer interaction, environmental control and security surveillance. In this chapter, we focus on recognizing the actions of a person in a video sequence from the meaningful local visual word learnt in the application domain.

Our approach is motivated by the recent success of the "bag-of-words" model for general object recognition in computer vision [63, 94]. This representation, which is adapted from the text retrieval literature, models the objects by the distribution of words from a fixed visual code book. This code book is usually obtained by the vector quantization of local image visual features via clustering algorithm, e.g. k-means clustering. However, not every visual word in the vocabulary is equally important in characterizing the underlying actions. In the text retrieval literature, a search engine typically filters out a list of stop-words before it builds the representation for the article or web page. This stop-words list, which includes frequently occurring, insignificant and uncharacteristic words in the articles or web pages, is learnt from the training data. Similarly, after building the visual code book, we also need to learn a stop-words list to filter out meaningless visual words before we build a better representation for the human action in the video sequence.

In our approach, we first apply a spatiotemporal feature detector to the video sequences and obtain the local motions features. Then we generate a visual word code book by clustering the local motion features and assign a word label to each cluster. Next, we apply an entropy based method to measure the information contained in each cluster and generate a list of stop-words that do not characterize the underlying action in our application domain. Thus when we use the "bag-of-words" model for the action recognition, we can filter out the stop-words and represent the action only with the distribution of more meaningful visual words.

The contribution of our work lies in learning a visual stop-words list using an entropy-based method and generating a more discriminative representation for the action. This vocabulary selection process exists in the middle level of the "bag-of-words" framework and is different from other low level visual features selection. Experiments has shown that this "bag-of-meaningful-words" model leads to better performance than the popular "bag-of-words" approach and exceed or close to results from other published methods.

4.2 Related Work

Extensive research has been done in recognizing human activities. These approaches can be broadly categorized as model based, spatiotemporal template based and "bag-of-words" based. Model based approaches for activity recognition depend on locating and tracking body limbs in order to recognize the activity. That requires a model of the body, whether a 3D model or a 2D view-based model. We refer the reader to excellent surveys covering this topic, such as [3, 23, 59]. However, for the task of activity recognition, tracking the limbs is not necessary. That motivates research on obtaining spatiotemporal descriptors directly from the motion to recognize the activity without limb tracking. One of the earliest work on spatiotemporal descriptor was carried out by Polana and Nelson [66]. In Bobick and Davis's work [7], Motion-Energy-Image and Motion-History-Image are introduced as templates for different motion recognition. Efros *et al.* [14] also proposed a spatiotemporal descriptor based on global optical flow measurements. Spatiotemporal template approaches are holistic approaches where global descriptors are used with no local features extracted.

In contrast, "bag-of-words" based approaches detect local salient descriptors as visual words, which are then used to recognize the activity. The "bag-of-words" model has been used successfully for object categorization [63, 94]. Inspired by text categorization, it represents an object as a histogram of local features. Recently, "bag-of-words" methods have been used in activity recognition [12, 76, 81]. However, these approaches do not differentiate the importance of each word in the vocabulary. In the KWIC(Key word in context) indexing, H.P.Luhn first introduced the concept of stop-words list, which is a list of non-informing words to be ignored in the text retrieval [55]. Similarly, in the representation for human actions, there are also certain visual words which occur with similar frequency across different actions. The existence of such visual words gives us little information about the underlying actions. Including them in the representation for the action will only add noise for the recognition in the later stage.

An example of less meaningful visual words is illustrated in Figure 4.1, which shows three sequences of actions from the experimental data set. Though these sequences belong to different category, they share the same visual words, which are indicated by the colored cubic in the figure. Since such visual words appear across different classes, detecting them does not help much in the recognition of the underlying action classes. So they should be excluded from the vocabulary or be assigned less weights when we want to represent the underlying actions.

The approach we propose here tries to learn such visual stop-words list from the training data of the action video sequences. Since each visual word is a label assigned to the visual features cluster, we use the conditional entropy of the cluster to measure the importance of the word. By using the stop words list, we can only choose the meaningful words for the representation of the action.

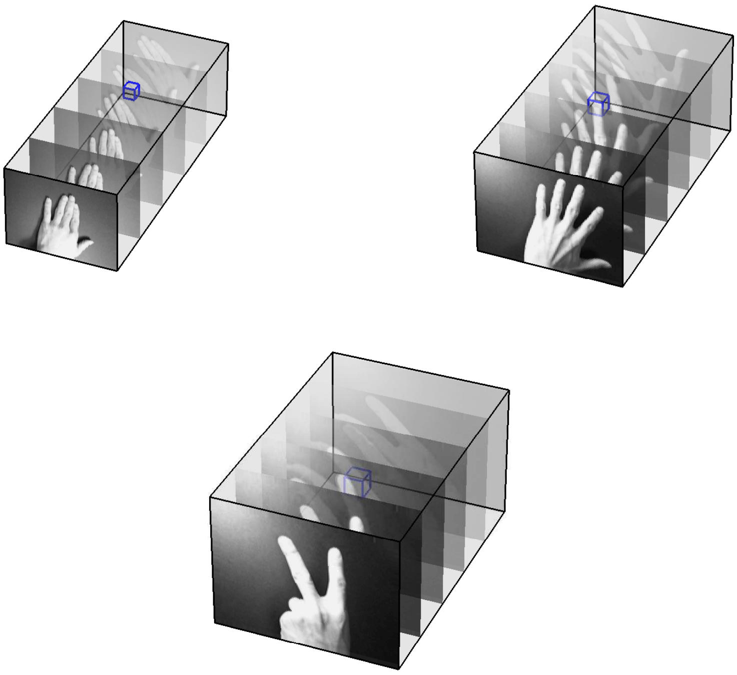


Figure 4.1: Less meaningful visual words. Three colored cuboid which belong to the same visual word appear across three different actions sequences. So the detection of such visual words does not help much in recognizing the underlying action in the sequence.

Other related directions and extensions for “bag-of-words” in the context of action recognition include [36, 92, 97, 99, 69]. In [36]’s work, the spatial orientation information were captured in the local features. In [92, 97], latent semantic model was applied to discover the activity types as topics in the hidden layer between the visual features and the video sequence. In [69], spatial-temporal correlograms were used to encode flexible long range temporal information into the spatial-temporal motion features.

4.3 Entropy Based Vocabulary Selection for “Bag-of-Word” Model

4.3.1 Feature Extraction

There are various methods for local motion feature detection and representation. Blank *et al.* [6] represent actions as space-time shapes and extract space-time features such as local space-time saliency, action dynamics, shape structure and orientation for action recognition.

Laptev and Lindeberg [44] propose an extended version of the interest points detection in the spatial domain [29] into space-time domain by requiring image values in space-time to have large variations in both dimensions. As noticed by [12] and observed by our experiments, the interest points detected using the generalized space-time interest point detector are too sparse to characterize and build model for complex actions. Therefore, we use the feature extractor from Dollar [12], which has been proven successful in [12, 62, 97, 69], for the detection and representation of the local motion features.

Like many interest point detectors, in [12], the space-time interest points are detected by applying separable linear filter to the video sequences. We are grateful that Dollar *et al.* have let us to use their code for location motion detection and representation. Here we will give a brief review of this method using the same notion as in [12].

With the assumption of a stationary camera or a preprocess to account for the camera motion, the response function has the following form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{4.1}$$

where $g(x, y; \sigma)$ is the $2D$ Gaussian smoothing kernel applying along the spatial dimension (x, y), with parameter σ corresponding to the spatial scale of the detector. h_{ev} and h_{od} are a quadrature pair of $1D$ Gabor filter applying along the temporal dimension. They are defined as $h_{ev}(t; \tau, \omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t; \tau, \omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2}$, with parameter τ corresponding to the temporal scale of the detector. In all cases, we chose $\omega = 4/\tau$, as did in [12]. To handle multiple scales, one must run the detector over a set of spatial and temporal scales. For simplicity, we run the detector using only one scale and rely on the code book to encode the few chances in scale that are observed in the data set.

It is noted in [12] that any region with spatially distinguishing characteristics undergoing a complex action can induce a strong response. To represent the motion feature, a cuboid of spatiotemporally windowed data surrounding the detected interest point (local maxima of response function) is extracted. In our experiments, it is set to be six times the scale of the detector to contain the volume contributing to the response function. We then compute the gradients of the intensities in the cuboid and flatten them into a vector. Finally, we project the vectors into a low dimensional space by PCA (Principal component analysis) and use the more compact representations as the motion features for the video sequences.

4.3.2 Build Visual Code Book

After the local motion feature detection and representation, we explored the detected data in the feature space by unsupervised clustering. By clustering, we can get a compact representation of the data, which is the visual code book and its related statistics.

The size of the code book is an important parameter for the representation. Because typically we will assume the resulted clusters follow Gaussian distribution such that the mean and variance will be sufficient statistics. If the cluster size is too small, we might put many unrelated data into one cluster such that the cluster does not follow normal distribution and the mean and variance are not sufficient for the description of the data in the cluster. If the cluster size is too large, the number of the data in one cluster might be too small such that the mean and the variance we compute from the samples are not good approximations for the true mean and variance in the cluster. So we need to find a balanced vocabulary size.

In our experiments, to build the code book, we perform k-means from a random subset of motion features from the training data. We have tested with different size for clustering and found out the typical vocabulary size for our experiments is K=250.

4.3.3 Entropy-Based Vocabulary Selection

Similarly to the representation for documents in the text retrieval literature, not all visual words in the code book are equally meaningful. We only need the informative words which characterize the underlying action for representation. The reasons are:

1) Some visual words occur with similar frequency across different actions. They are irrelevant to the underlying actions. Keeping them will only be nuisance for the recognition.

2) Keeping only the important visual words, we can have a more compact representation, which can speed up the recognition in the later stage.

The features extracted from Dollar[12] are low level visual features which do not contain higher level information about its relevance to the underlying actions in the video. Thus after they are clustered to build the code book, it is necessary to learn and generate a more meaningful vocabulary.

Each visual word represents the cluster from which it is assigned the word label. The information contained in the visual word is the information we can get from the cluster. In a particular application, if most of the local visual features contained in a given cluster come from one action, we know a visual word from this cluster will give us more information about this underlying action because this visual word can indicate this underlying action with more confidence.

A suitable measure for this information is entropy, which measures the uncertainty or the randomness of such a word. Given the set of action $A_1, \ldots, A_N$, we can compute $P(A_i|v_j)$, the conditional probability of each action given word v_j, from the training data. The conditional entropy given the visual word v_j can then be written as

$$E(Action|v_j) = \sum_{i=1}^{N} -P(A_i|v_j)logP(A_i|v_j) \tag{4.2}$$

The higher the entropy, the more uniformly distributed the actions are, given the visual word, therefore, the less information we can gain from knowing this word. The lower the entropy is, the more discriminative the visual word is. So we can use the reciprocal of the conditional entropy as the measure for the importance $I(v_j)$ of the visual word in the vocabulary.

$$I(v_j) = \frac{1}{E(Action|v_j)} \tag{4.3}$$

Thus we can select the vocabulary based on the importance of the visual words.

4.3.4 Two Methods for Vocabulary Selection

In the experiment, we used two methods for selecting the visual words. The first one is a “hard selection”, in which we discarded the bottom $p\%$ most important visual words. In this method, we removed the noise from the representation using the less meaningful visual words and we had a more compact representation. However, how to select the best value for p remains a parameter tuning problem. In our experiments, we have tested with different value of p and chosen the best value for each application.

The second method is a “soft selection”, in which we used the importance $I(v_j)$ as the weight assigned to the visual word v_j. In this method, we still keep all the words in the vocabulary but with different weights, we differentiate visual words with regard to their importance in the application domain, measured by their entropy. In this approach, we do not need to tune the parameter for the number of visual words in the vocabulary.

There are many related research for low level visual feature selection. In the work of Gy. Dorko and C. Schmid [13], likelihood ratio and mutual information have been used to select the scale-invariant parts for object class recognition. Zhao and Elgammal et al. [100] have proposed a two stage of statistical and combinatorial methods for image patches selection for recognition. However, in the “bag-of-words” model for object recognition, these work are for the local visual features and are in the low level of the framework. Our approach selects the vocabulary, which is the middle level of the framework and the selection has more semantic meaning for the underlying actions.

4.3.5 Recognition Algorithm

Once we extract and select the meaningful visual words, we represent a video sequence as “bag-of-meaningful-words” and use the histogram of these meaningful visual words to approximate the distribution as the feature for the underlying action. Because we use Chi-square distance as the metric for histogram representation in “bag-of-words” model and the Chi-square distance satisfies

$$c\chi^2(a, b) = \chi^2(ca, cb) \tag{4.4}$$

where c is a scalar, in “soft selection” method, we can directly embed the weight to the histogram representation.

So briefly, the whole action recognition algorithm is the following. In the training phase, we first build the code book from clustering the low level visual features. Then we select a more meaningful vocabulary based on the conditional entropy of the clusters. Next, for each training video sequence, we represent it as a histogram of “bag-of-meaningful-words” from our selected visual words and label the sequence with the underlying action. In the testing phase, for each testing video sequence, we represent it using the same meaningful vocabulary and apply the nearest neighbor algorithm for recognition.

The reason we choose nearest neighbor classification instead of more complicated algorithm are the following: Firstly, we aim to do a valid comparison with other published research using the same data sets. All the published results cited in this paper are obtained using nearest neighbor algorithm. Secondly, the focus of this paper is on how to better represent the data using “bag-of-meaningful-words” model instead of the simple “bag-of-words” one. By using the same simple classification method as other people did but get better results, it demonstrate the importance of this model, which is our major contribution.

By using more complicated discriminative methods, such as support vector machine (SVM) or boosting, the similar results might be achieved to some extend. For example, support vector machine only uses support vectors from the training data set for classification and boosting will assign less weight to weak classifiers with less discriminative power when the classification results from each weak classifier are combined. However, our approach addresses this issue as a representation problem, which is from a different perspective and at an earlier stage of the recognition process.

4.4 Experiments

4.4.1 Data Sets and Experiments Setting

We carried out our experiments on three data sets, namely facial expression data set from Dollar et al.[12], hand gesture data set from Wong et al. [97] and KTH human action data set from Schuldt et al. [76].

The facial expression data sets include two individuals, each expressing six different emotions under two lighting conditions. The expression are anger, disgust, fear, joy, sadness and surprise. Certain expressions are quite distinct, such as sadness and joy, while others might be quite similar, such as fear and surprise. Under each lighting condition, each individual is asked to repeat each of the six expression eight times. The subject always starts with neutral expression, expresses an emotion and back to neutral in about two seconds.

Table 4.1: Details of the data sets used in our experiments

Dataset	Facial Expression	Hand Gesture	KTH
No. of classes	6	9	6
No. of subjects	2	2	25
No. of trials per subject	8	10	1
No. of conditions	2	5	4
Total No. of Samples	192	900	593

The hand gesture data set involves two individuals performing nine hand gestures. The hand gestures have two components. The first is the shape of the hand, which includes close, open and v shape. The second is the orientation of the hand movement, which includes left, right and forward. So the combination is nine.

The KTH human action data are collected by [76]. There are 25 individuals performing the following six activities: walking, jogging, running, boxing, clapping and waving. Each individual has performed under four different conditions, the combination of indoor or outdoor and two different clothing. The clips have been sub-sampled (people are approximated eighty pixels in height) and contain compression artifacts. (This is the version of the data set available online). Similar to the facial expression data set, some of the different activities look quite similar, such as running and jogging.

In all data sets, each video sequence contains one activity. The video sequences were converted into gray level to avoid the bias in color. The details of the data sets are summarized in Table 4.1 and some sample images from the video sequences are shown in Figure 4.2.

For comparison in the experiments, we implemented a baseline approach using the “bag-of-words” representation. This baseline approach does not discard any meaningless visual words nor assigns different weights to the visual words.

We applied nearest neighbor classification method for the recognition of the underlying action. The experiments are carried out in leave-one-out cross-validation setting for 30 runs.

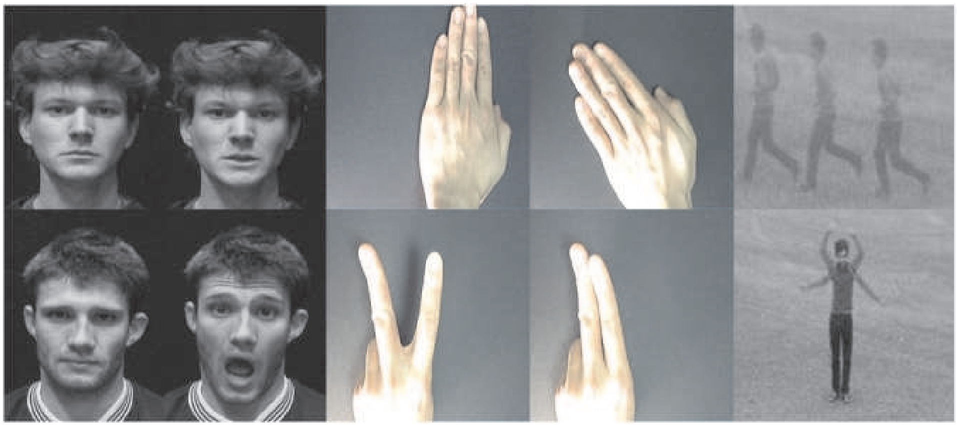


Figure 4.2: Sample images from the experiment data sets.

4.4.2 Code Book Size

The size of the code book, namely the initial vocabulary size, is one of the parameters we need to consider in the “Bag-of-the-words” model. As previously discussed, too large or too small will yield unsatisfied performance. It also depends on the specific application domain, as for different applications, different number of vocabulary is required to sufficiently describe the knowledge in that domain.

In our experiments, we explored the optimal value for the initial vocabulary size K empirically. We used the baseline approach without visual word discard since this was to decide the initial size of the vocabulary and no selection was involved yet. We adopted the leave-one-out cross validation setting for 30 times. Because the clustering results depend on the initial starting points, which are generated randomly, the recognition rates from different runs are different. We use the average recognition rate and its standard deviation as error bar to measure the performance, which is shown in Figure 4.3.

From the figure, we can see that for all these three applications, recognition rates first improve with the increase of vocabulary size. It is because with larger vocabulary size, the data are better described by these clusters. It is similar to larger vocabulary has better descriptive power in a language. Then we observed that the recognition rates do not change much after the vocabulary size reaches 250. This is because the structure of the language has been sufficiently explored by the current vocabulary set. Considering that with smaller size of the

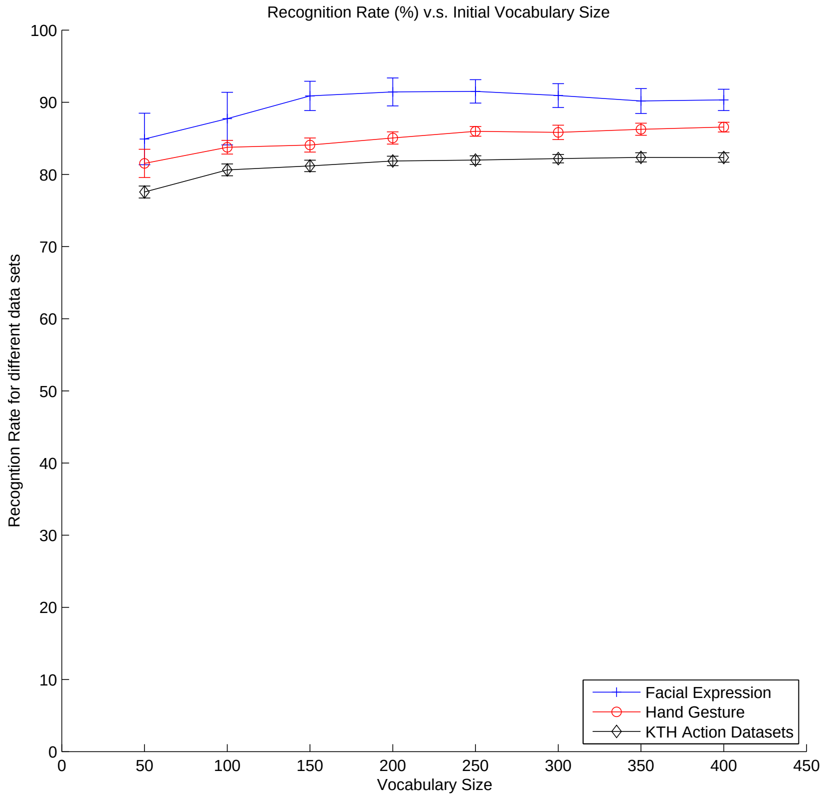


Figure 4.3: Average recognition rate for different initial vocabulary size using "Bag-of-the-words" model. Standard deviation is used as error bar for different data sets

vocabulary, we can achieve more compact representation, we empirically chosen K=250 as the initial vocabulary size.

4.4.3 Vocabulary Selection

In the experiments for vocabulary selection, we tested two methods of selection. We have experimented with "hard selection", which discarded the bottom $p\%$ important visual words. We also applied "soft selection", which weighed visual words differently according to their importance.

We first tested with different discard rates for visual words in the vocabulary. The results are shown in Figure 4.4. Since the size of stop words list is different for different application, we can see the discard rate position for the peak of the curve, which is the best recognition

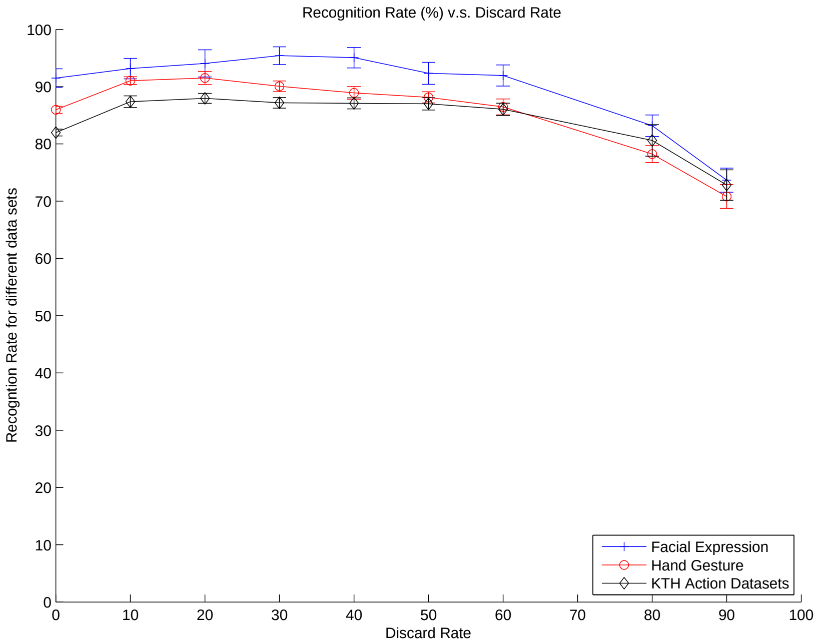


Figure 4.4: Recognition rate (%) v.s. discard rate (%) for different applications. The recognition rate peaks at different discard rate for different application. For hand gesture data set, the best recognition rate is at 20% discard rate. Recognition rate achieved best performance at 30% discard rate for facial expression data set and the optimal discard rate for KTH data set is at 20%. Standard deviation is used as error bar for different data sets

rate, is different for the three different data sets. However, they all perform better than the baseline "bag-of-words" model, which is at 0% discard rate without visual words selection. For each application, the recognition rate first increases with the increase of the discard rate. This is because with removal of the less meaningful words, we have a more characteristic representation for the action. Then with further increase of discard rate, we are likely to discard meaningful visual words, which leads to the decrease of the performance.

In the following discussion, we will use the recognition rate from the optimal discard rate for the performance of the "hard selection" methods.

We also experimented with "soft selection" method. Together, we compare our vocabulary methods with other published methods and list the performance of the baseline method, the best results from the "hard selection" method, the "soft selection" method with other published results on these data sets in Table 4.2.

Methods:	Base-line	Our method (hard selection)	Our Method (soft selection)	Wong [97]	Niebles [62]	Wang [92]
Facial Expressions	91.50	**95.41**	93.36	83.33	none	none
Hand Gestures	85.96	91.52	**91.83**	91.47	none	none
KTH Actions	81.98	87.96	87.47	83.92	81.50	**92.43**

Table 4.2: The average recognition rates (%) for facial expression, hand gesture and KTH human action data sets obtained from different algorithms.

The results in Table 4.2 demonstrate that our methods, both “hard selection” method and “soft selection” method, perform better than the baseline approach. It shows that the words selections for visual vocabulary are effective for the human action recognition applications and both proposed methods benefit from the “bag-of-meaningful-words” feature. The performance from both methods are similar or close to the published results from other methods, some of them using more complicated model, such as the latent semantic model from [97].

The comparison of “hard selection” method with the “soft selection” method on all three data sets using confusion matrices is given side by side in Figure 4.5. From these matrices, we can see some actions are easier to be confused with others, e.g. the jogging and running action from KTH human data sets. It also shows that in some cases, the “soft selection” method has better results than the “hard selection” method, and in other cases, it is the opposite. Thus the choice of selection method depends on the application. .

4.5 Summary

In this chapter, we have presented an entropy based vocabulary selection method for the “bag-of-words” model in action recognition. In this approach, we measure the importance of visual words by their conditional entropy learned from the clusters in the training data. Then for the representation of the action, we tried both “hard selection” method, which discards the less meaningful visual words, and the “soft selection” method, which weighs the visual words differently by their importance. Both methods have shown improved performance over the baseline “bag-of-words” model on a set of benchmark data sets and exceed or close to results from other published methods.

This entropy based vocabulary selection method has broader application than activity recognition. This is a general approach to measure the importance of words in the vocabulary and can be applied to other applications using the “bag-of-words” model, e.g. object recognition. Another possible application is to integrate our method into other extension of “bag-of-words” model, e.g. latent semantic model from [97] and the spatiotemporal model from [99]. I will present the results from this line of research in the next chapter.

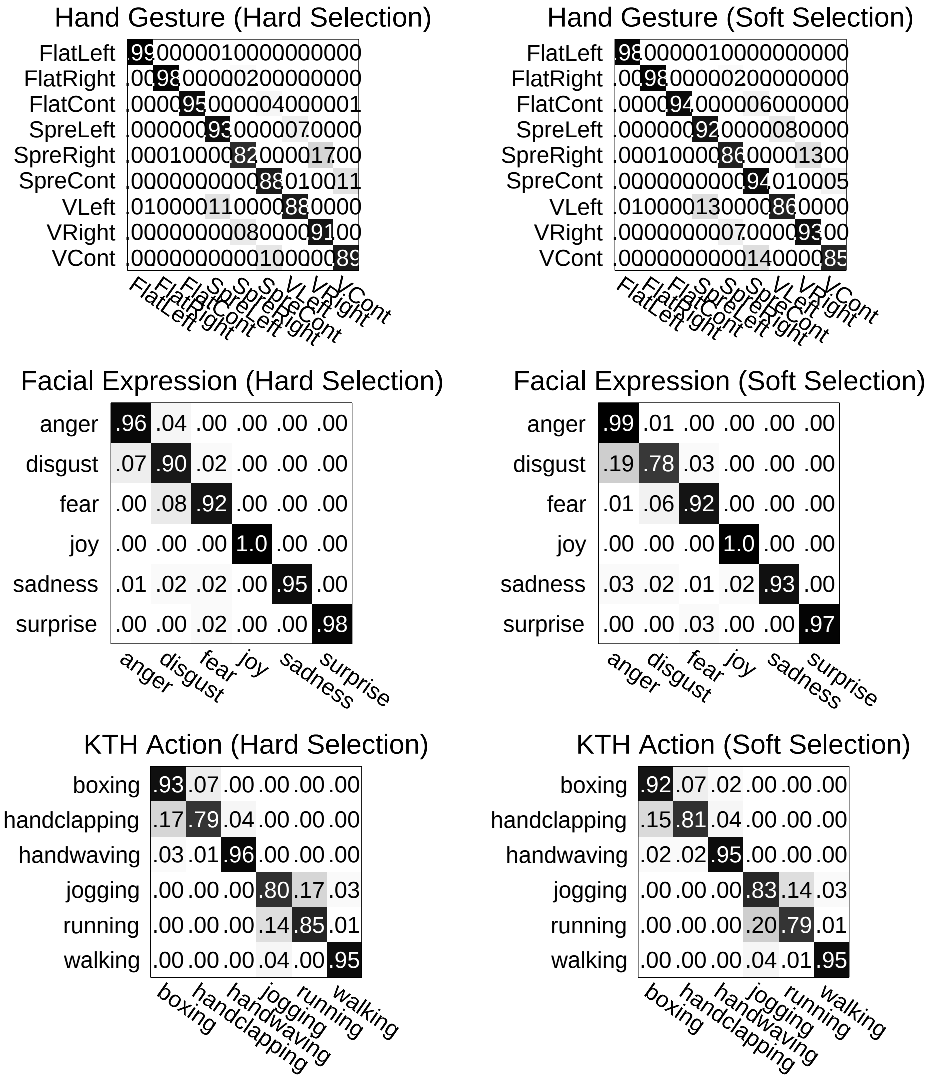


Figure 4.5: With leave-one-out cross-validation experimental setting, the confusion matrices on all three data sets from both "hard selection" method and "soft selection" method. The first row is for hand gesture data, the middle row is for facial expression data and the bottom row is for KTH human action data set. The results from the left column are from the "hard selection" method and the results from the right column are from the "soft selection" method.

Chapter 5

Spatiotemporal Representation

In the previous chapter, I have applied vocabulary selection method to build a "bag-of-meaningful-words" model for human activity recognition. In this chapter, I will present two approaches for human activity recognition by modeling the distribution of local visual motion features and their spatial temporal arrangements.

The first approach uses a spatiotemporal pyramid representation for recognizing facial expressions and hand gestures. This approach works by partitioning video sequence into increasingly fine subdivisions in the space and time domains and modeling the distribution of the local motion features inside each subdivision such that the set of motion features are mapped into spatial and temporal multi-resolution histograms. This spatiotemporal pyramid is built by weighting the histograms from the different layers of the subdivisions. The proposed approach is an extension of the orderless "bag-of-words" model by approximately capturing geometric and temporal arrangements of the local motion features. The experiments on facial expression and hand gesture data sets have demonstrated the significantly improved performance over state of art results on human activity recognition tasks by using our representation.

In the second approach, the local motion features used for the representation of a frame are the ones detected in this frame and others integrated from its temporal neighbors. The features' spatial arrangements are captured in a hierarchical spatial pyramid structure. By using frame by frame voting for the recognition, experiments have demonstrated improved performances over most of the other known methods on the popular benchmark data sets while approaching the best known results .

The above two approaches try to incorporate spatial and temporal information into the representation for local visual features, while the approaches introduced in previous chapter try to find more informative and relevant words in the vocabulary of "bag-of-words" model. Since these two directions are complementary, in this chapter, we also discuss sequential combinations of the methods from these two directions and apply them to human activity data sets, which generate comparable results.

5.1 Motivations

Recognizing human activities from image sequences is an appealing yet challenging problem in computer vision which has been intensively researched. In this chapter, we focus on recognizing the activities of a person in an image sequence from local motion features and their spatiotemporal arrangements.

Like the methods introduced in the previous chapter, our approaches presented in this chapter are also motivated by the recent success of "bag-of-words" model for general object recognition in computer vision[94, 63]. This representation, which is adapted from the text retrieval literature, models the object by the distribution of words from a fixed visual code book, which is usually obtained by vector quantization of local image visual features. However, this method discards the spatial and the temporal relations among the visual features, which could be helpful in the object recognition.

I have proposed two methods addressing this problem. These methods are inspired by the work of Lazebnik *et al.*[45], who have used a spatial pyramid to capture the global geometric information. This method partitions the image into increasingly fine sub-regions and computes the histogram of visual features from each sub-region. Using the concatenated weight histograms as the feature for the image, this image representation integrates both appearance and spatial information.

Following the same sprit, our first approach uses a hierarchical structure in the video sequence representation to integrate information from the spatial and temporal domains. The representation for this approach is illustrated in Figure 5.1. We first apply a spatiotemporal feature detector to the video sequence and obtain the local motion features. Then we generate a visual word code book by quantization of the local motion features and assign word label to each of them. Next we divide the data volume spatially and temporally into finer subdivisions and compute the histograms of the visual words in each cell. Finally, we concatenate the

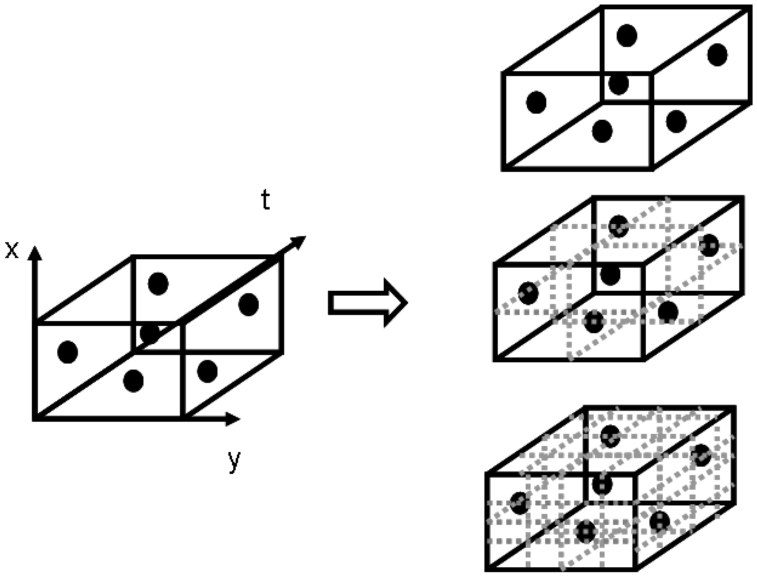


Figure 5.1: The representation for the action in a video sequence is built from the motion features detected in it. Then a spatiotemporal pyramid (e.g. $L = 2$) is applied to model the spatial temporal arrangements among the features.

histograms from all cells and use it as the feature for the whole video sequence.

The second approach we proposed uses a hierarchical representation for the frames of the video sequence to integrate information from the spatial and the temporal domains. We also first apply a spatiotemporal feature detector to the video sequence and obtain the local motion features. Then we generate a visual word code book by quantization of the local motion features and assign word label to each of them. Next for each frame, we integrate the visual words from its nearby frames, divide the frame spatially into finer subdivisions and compute in each cell the histograms of the visual words detected in this frame and its temporal neighbors. Finally, we concatenate the histograms from all cells and use it as the feature for this frame. The representation for a frame i is illustrated in Figure 5.2.

The contribution of both approaches lies in that besides the appearance information contained in the local motion features, our representation also captures both the spatial and the temporal relations among the features, which leads to better performance than the popular “bag-of-words” approach.

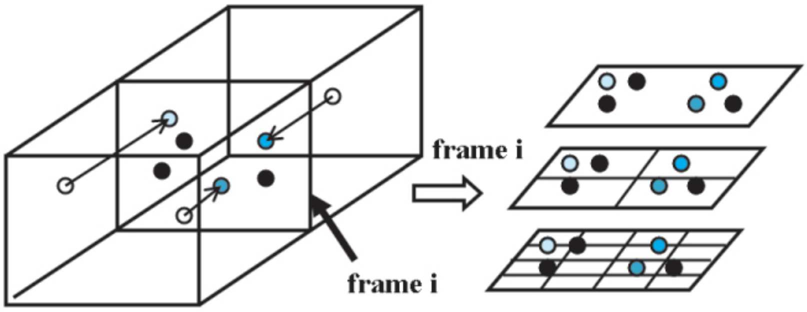


Figure 5.2: The representation for a frame i is built from the motion features detected in it and integrated from the nearby frames. The closer a features is to frame i, the higher weight it is assigned, represented by a darker circle. Then a spatial pyramid (e.g. $L = 2$) is applied to model the spatial arrangements among the features.

The organization of this chapter is as follows. The related work are summaried in Section 5.2. In Section 5.3 we introduce the spatiotemporal pyramid representation framework and its application for recognizing human actions in the video sequences. In Section 5.4, we introduce our second approach which uses local motion features to build the spatiotemporal representation for frames in the video sequences. In Section 5.5, we combine the visual word vocabulary selection methods introduced in previous chapter with the spatiotemporal information integration approaches presented in this chapter. Section 5.6 shows the results of applying the proposed methods on facial expression, hand gesture and human action data sets. Section 5.7 is the summary.

5.2 Related Work

Extensive research has been done in recognizing human activities. The approaches can be broadly categorized as model based, spatiotemporal template based and local visual features based methods. Please refer to Chapter 2 and the Related Work section in Chapter 4 for detailed discussion.

“Bag-of-words” based approach belongs to the school of local visual feature based methods. In this approach, local visual features are detected and represented as visual words, which are used to recognize human activities. “bag-of-words” has been used successfully for object categorization[94, 63]. Originally used in text categorization, it represents the object as histogram of local features. Recently, “bag-of-words” methods have been used in activity recognition[76, 12, 81]. However, these approaches lack the relations between the features in the spatial and the temporal domains which are helpful for recognition. There are many recent research on extending “bag-of-words” to add the spatial relation in the context of object categorization [70, 1, 56, 25, 45]. In particular, pyramid match kernel [25, 45] used the weighted multi-resolution histogram intersection as a kernel function for classification with sets of image features.

The approaches we propose here try to simultaneously model the spatial and temporal relations of the local motion features. In our first approach, we build a hierarchical structure along both spatial and temporal dimensions for representation. This approach is inspired by [25, 45]. However, our approach is a representation which embeds the spatial and temporal information while [25, 45] proposed a matching kernel. Our approach uses Chi-square distance as a distance function while [25, 45] used weighted histogram intersection to satisfy as a kernel function.

In our second approach, the temporal information are captured by integrating the local motion features from the temporally nearby frames and the spatial information are captured by using a hierarchical spatial pyramid in the representation.

Other related works for “bag-of-words” in the context of activity recognition include [36, 92, 97]. In [36]’s work, spatial orientation information were captured in the local features. In [92, 97], latent semantic model was applied to discover the activity types as topics in the hidden layer between the visual features and the video sequence. Different from them, our approach uses the spatiotemporal pyramid as a representation and simultaneously integrates the spatiotemporal relation among visual features with their appearance information.

5.3 Spatial Temporal Pyramid Representation

5.3.1 Pyramid Representation for Sets of Points

The goal of our approach is to find a representation for a set of points in a spatiotemporal space. This representation should capture the distribution of a set of points in the space in a way that it is suitable for measuring the similarity between the sets. Inspired by [25, 45], we propose a pyramid representation to capture the spatiotemporal distribution for sets of points.

Let X and Y be two sets of points in a d-dimensional space, $X = \{x_i | x_i \in R^d\}$, $Y = \{y_j | y_j \in R^d\}$. We recursively partition the space into subdivisions. Intuitively, we measure the distance between X and Y as the sum of the distances between the distributions of the points from each data set in each of the subdivisions. The distributions of the points are approximated by the number of points in the subdivision whose distances are measured by Chi-square distance.

We start by constructing a sequence of increasingly finer binary partitions at resolution 0,..., L, such that the partition at level l has 2^l cells along each dimension with a total of $D = 2^{dl}$ cells for this level. We denote the number of points from X in the ith cell at level l as $H_X^l(i)$. The distance between X and Y at this level, represented as H_X^l, H_Y^l respectively, is the sum of the distances for each corresponding cell,i.e.:

$$dist(H_X^l, H_Y^l) = \sum_{i=1}^{D} \chi^2(H_X^l(i), H_Y^l(i)) \tag{5.1}$$

where $\chi^2(\cdot, \cdot)$ is the Chi-square distance. This is similar to representing the sets of points X and Y at level l by concatenating $H_X^l(i)$ and $H_Y^l(i)$ into histograms respectively and measuring their Chi-square distance. Therefore, we can use these concatenated histograms as the representations for H_X^l and H_Y^l respectively.

In this pyramid structure, different level captures different scale of variance. The representation for a set of points X should include all $H_X^l, l = 0, .., L$ and the distance between X and Y, represented as H_X and H_Y respectively, should include the distances from all levels:

$$dist(H_X, H_Y) = \sum_{l=0}^{L} dist(H_X^l, H_Y^l) \tag{5.2}$$

This is again similar to representing X and Y by concatenating their histogram representations from all levels into a long histogram respectively and measuring their distance. Therefore, we can use these concatenated histograms as the representations for H_X and H_Y respectively.

5.3.2 Weighed Pyramid Representation for Sets of Points

Since different information are captured at various levels of the pyramid, different weights should be assigned for each level of them. At finer resolution, the correspondence between two sets is captured more accurately. Therefore, we penalize the similarity information gained at a coarser level and give more weights to the similarity measured by the histogram distance at a finer resolution. The weight we assign at level l is: $weight(l) = 1/2^{L-l}$ for $l = 0, .., L$. The weighted distance between X and Y is:

$$dist(H_X, H_Y) = \sum_{i=0}^{L} \frac{1}{2^{L-l}} dist(H_X^l, H_Y^l) \tag{5.3}$$

Since Chi-square distance satisfies

$$c\chi^2(a, b) = \chi^2(ca, cb) \tag{5.4}$$

where c is a scalar, we can directly embed the weight to the histogram representation. Putting everything together, our representation for a set of points X is the concatenated weighted histogram from all levels of the pyramid.

Since the distance between these representations is the sum of the Chi-square distances between each element, which by themselves are metric, it is easy to prove that:

1) $dist(H_X, H_X) = 0$

2) $dist(H_X, H_Y) = dist(H_Y, H_X)$

3) $dist(H_X, H_Z) \leq dist(H_X, H_Y) + dist(H_Y, H_Z)$

Therefore, the distance defined on our representation is a metric.

Our representation is suitable for the set of points in a spatiotemporal space because it is an approximation of the distribution of the points in the spatiotemporal space. Since it is histogram based representation, Chi-square distance is suitable for measuring the similarity between the sets of points.

In Lazebnik et al.'s work[45], pyramid match kernels are proposed to use a pyramid structure to find approximate correspondence at different levels between two sets. Our work differs from them in that:

1) Our goal is to find a suitable representation to integrate the spatial and temporal relation for a set of points. It embeds the weights in the representation to reflect the importance of different pyramid layers. The work in [45] is seeking a suitable kernel function for two sets of points.

2) Because our representation is a concatenated histogram, we measure the distance by Chi-square distance. The pyramid match kernels use histogram intersection as the distance function to satisfy the Mercer's condition.

3) Our representation captures the distribution of the points in both spatial and temporal space, so it can be applied for human action recognition in the spatiotemporal domain. The pyramid matching kernels are only used for natural scene categorization in 2-D images.

5.3.3 Pyramid Representation for Human Action

Motivated by the "bag-of-words" approach while still considering the spatial and temporal arrangements of the features, we model human activity as a set of local motion features points located in the three dimensional spatiotemporal space. Then the spatiotemporal pyramid representation can be used for the set of feature points. We divide the video sequence spatially and temporally into increasingly finer subdivisions and compute the distribution of the feature points in each cell for all levels. The final representation for the activity is the concatenated weighted histogram from all levels.

We also want to consider the appearance information of the local motion features and model them as words. We apply k-means clustering in the visual feature space to quantize all local motion features into K discrete types and assign word label to each of them. In each subdivision, we use the histogram of the visual words instead of the number of points as the approximation for the point distribution. This representation contains both appearance information, as the histogram in each cell, and the spatiotemporal information, which comes from the spatiotemporal pyramid structure.

This representation is a straightforward extension of the popular "bag-of-words" method. In each subdivision, all the local motion features are modeled as "bag-of-words". When $L = 0$, it reduces to the standard "bag-of-words" representation. For better computation efficiency, we normalize the vector by the total weights of all elements.

The complexity of this representation is linear with the size of motion words vocabulary. For L level and K motion words, the dimensionality of the resulting representation is $K\sum_{l=0}^{L} 8^l = K\frac{1}{7}(8^{L+1}-1)$. In our experiments we observe that the performance does not improve much when $L > 2$. In previous chapter, we have observed that the vocabulary size $K = 250$ has optimal performance. Therefore, we use the setting of $K = 250$ and $L = 2$, which leads to a 18250-dimension vector for the activity representation.

5.3.4 Feature Extraction

I have discussed our feature extraction method used in the experiments in previous chapter. To make this section self contained, I will reiterate it briefly.

There are various methods for motion feature detection and representation, such as presented in [76] and [12]. As noticed by [12] and observed from our experiments, the interest points detected by generalized space-time interest points detector from [76] are too sparse to build model for many complex activities. Therefore, we utilized the one from Dollar[12], which has been proven successful in [12, 62, 97].

Like many interest point detectors, in [12], the space-time interest points are detected by applying separable linear filter to the video sequences. With the assumption of a stationary camera or a preprocess to account for the camera motion, the response function has the following form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{5.5}$$

where $g(x, y; \sigma)$ is the $2D$ Gaussian smoothing kernel applying along the spatial dimension (x, y), with parameter σ corresponding to the spatial scale of the detector. h_{ev} and h_{od} are a quadrature pair of $1D$ Gabor filter applying along the temporal dimension. They are defined as $h_{ev}(t; \tau, \omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t; \tau, \omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2}$, with parameter τ corresponding to the temporal scale of the detector. In all cases, we chose $\omega = 4/\tau$, as did in [12].

To represent the motion feature, a cuboid of spatiotemporally windowed data surrounding the detected interest point (local maxima of response function) is extracted. In our experiments, it is set to be six times the scale of the detector to contain the volume contributing to the response function. We then compute the gradients of the intensities in the cuboid and flatten them into a vector. Finally, we project the vectors into a low dimensional space by principle component analysis (PCA) and use the more compact representations as the motion features for the video sequences.

To build the code book, we perform k-means from a random subset of motion features from the training data. As experimented and reported in previous chapter, the typical vocabulary size for our experiments is K=250.

5.3.5 Recognition Algorithm

Since the video sequences' representations contain rich information in the spatiotemporal and the appearance domains, they can serve as classifiers for the underlying activity types. Here we employ nearest neighbor classification algorithm. For test video sequence, we label it with the same label from the most similar sequence in the training data sets.

5.4 Spatiotemporal Representation for the Frame

5.4.1 Feature Extraction

We use the feature extractor from Dollar[12] for the local motion features' detection and representation, which has been proven successful in [12, 62, 97]. In this method, the motion features are detected by applying separable linear filter to the video sequences. They are represented by the intensity gradients of a cuboid of spatiotemporally windowed data surrounding the detected interest point. To build the code book, we perform k-means from a random subset of motion features from the training data. The typical vocabulary size for our experiments is K=250.

5.4.2 Key Frames Selection by Their Discriminative Power

Intuitively, not all frames from a video sequence are equally important. We only need a few informative frames that characterize the activity for recognition. The reasons are:

1) Some video frames are irrelevant to the underlying activity, e.g. the frames with no action in them. They could be nuisance for the recognition.

2) We can greatly speed up the recognition process if we only use the informative key frames without losing important information..

The feature exactor from Dollar[12] can detect the local informative motion features for each frame. They are encoded by the visual words $v_1, \ldots, v_K$, obtained from the clustering, where K is the vocabulary size. We can measure the discriminative power of each visual word. Entropy is a suitable measure for the discriminative power of a given visual word since it measures the uncertainty or the randomness of such a word. Given the set of activities $A_1, \ldots, A_N$, we can compute $P(A_i|v_j)$, the conditional probability of each activity given visual word v_j, from the training data. The conditional entropy given the visual word v_j can then be computed as

$$E(Activity|v_j) = \sum_{i=1}^{N} -P(A_i|v_j)logP(A_i|v_j) \tag{5.6}$$

The higher the entropy, the more uniformly distributed the activities are given the visual word, therefore, the less discriminative at the visual word. The lower the entropy is, the more discriminative the visual word is. We can use the conditional entropy of the visual words to measure the discriminative power of a given frame F. To do that, we use a function $g(\cdot)$ which is defined as:

$$g(F) = \sum_{j=1}^{K_F} \frac{1}{E(Activity|v_j)} \tag{5.7}$$

where K_F is the number of the visual words in frame F. The higher the score of $g(F)$, the more discriminative the frame F is.

We selected the top $p\%$ most discriminative frames for recognition. p is set to 25 in the experiment, which is an empirical chosen number. These top discriminative frames are called the key frames.

5.4.3 Temporal Integration of the Motion Features

Since a frame is correlated to its temporal neighbors, we build its representation from the motion features detected in it and its neighbor frames, weighed by the features' temporal distance to this frame. Intuitively, the further the distance is, the less weight it should be assigned to. Therefore, for a frame i, the weights assigned to the motion features from frame j are:

$$Weight(i,j) = e^{-\frac{dist(i,j)}{\sigma^2}} \tag{5.8}$$

where $dist(i,j)$ is the first norm distance between frame i and j and σ is the bandwidth for a smooth weight, which is empirically set to be 5 in our experiments. Thus the temporal relations of the features to frame i are captured by the different weights. The weights are 1 for the motion features detected at frame i and are close to 0 for those from the distant frames. Therefore, only the motion features from nearby frames contribute significantly to the integration.

5.4.4 Spatial Representation for the Frame

With all the temporally weighted motion features for the frame, our next goal is to find a representation to model the spatial relations of these features in a way that it is suitable for measuring the similarity between the frames.

Let X and Y be two sets of motion features from two frames respectively. Inspired by [45], we represent the frame in a spatial pyramid. For each level $l, l = 0, \ldots, L$, we divide the frame along x and y dimensions into $2^{2\times l}$ subdivisions. Intuitively, we measure the distance between X and Y as the sum of the distances between the corresponding cells of all levels from X and Y. Each cell can be described as the histogram of the weighted motion features in it and the distances between them are measured by Chi-square distance. So the distance between X and Y is formulated as:

$$dist(X,Y) = \sum_{l=0}^{L} \sum_{i=1}^{2^{2\times l}} \chi^2(H_X^l(i), H_Y^l(i)) \tag{5.9}$$

where $H_X^l(i)$ is the histogram of the weighted motion features from the ith cell in level l from X and $\chi^2(\cdot,\cdot)$ is the Chi-square distance. This is similar to representing X and Y by concatenating their histogram representations from all cells in all levels into a long histogram respectively and measuring their distance. Therefore, we can use these concatenated histograms as the representations for the frames.

Since different information are captured at various levels of the pyramid, different weights should be assigned to each of them. At finer resolution, the correspondence between two sets are captured more accurately. Therefore, we penalize the similarity information gained at a coarser level and give more weights to the similarity measured by the histogram distance at a finer resolution. The weight we assign at level l is: $weight(l) = 1/2^{L-l}$ for $l = 0, \ldots, L$. The weighted distance between X and Y is:

$$dist(X, Y) = \sum_{l=0}^{L} \frac{1}{2^{L-l}} \times \sum_{i=1}^{2^{2\times l}} \chi^2(H_X^l(i), H_Y^l(i)) \tag{5.10}$$

Because Chi-square distance satisfies

$$c\chi^2(a, b) = \chi^2(ca, cb) \tag{5.11}$$

where c is a scalar, we can directly embed the weight to the histogram representation. Putting everything together, our representation for a frame is the concatenated weighted histogram from all cells in all levels of the pyramid. In our representation, the temporal relations are modeled as the different weights assigned to the motion features and the spatial relations are captured in the spatial pyramid structure. With the motion features as the visual words, our representation simultaneously integrate appearance, spatial and temporal information.

Our representation for the frame is a straightforward extension of the popular “bag-of-word”. In each subdivision, all the local motion features are modeled as “bag-of-words”. When $L = 0$ and $\sigma = 0$, it reduces to the standard “bag-of-word” representation. In our experiments we observe that the performance does not improve much when $L > 1$. Therefore, we use the setting of $K = 250$ and $L = 1$, which leads to a 1250-dimension vector for the frame representation. For better computation efficiency, we normalize the vector by the total weight of all elements.

5.4.5 Recognition Algorithm

Since the frames’ representations contain rich information in the spatiotemporal and the appearance domains, they can serve as classifiers for the underlying activity types. For each frame from the test frame, we label it with the closest frame in the training data sets and employ a majority voting throughout the sequence.

5.5 Combination of Vocabulary Selection and Integrating Spatiotemporal Information Method

We have extended the “bag-of-words” approach for general recognition framework in two directions. In the previous chapter, we introduce visual word selection methods, which aims to select the most important words in a specific application domain to build more informative code book. In the previous sections of this chapter, we conduct research on integrating spatiotemporal information among these visual words into the representation of the underlying activities in the video sequences. Since these two directions are complement to each other, naturally, we experiment with the sequential combinations of them.

Since such combinational approach is built upon previous approach, we adopt the previous tuned up parameters as the setting for this approach. For example, the initial vocabulary size is set to k=250. And σ is also set to 5 as the smoothing weight for the temporal distance in the spatiotemporal representation for the frame.

I will briefly review these approaches. For visual word selection, we have experimented with two approaches. Both methods use the conditional entropy of visual words to measure their importance. The first one is “hard selection” method, which discards an empirically selected percentage of initial less important visual words for a more compact representation. Different application has different discard rate. The second one is a “soft selection” method, which does not discard any visual words but assigns different weights to different visual words measured by their conditional entropy. These weights later will be embedded into the representation for the “bag-of-words” model.

In the spatiotemporal integration approaches, the first one is to represent the whole activity as visual words distribution approximated by histogram in three dimensional spatial temporal space. The second approach is to represent the informatively selected key frames as the distribution of temporally integrated visual words in a spatial pyramid.

We combine these methods, which results four combination approaches. We will discuss the performances from these approaches in details in Section 5.6.

Dataset	Facial Expression	Hand Gesture	KTH
No. of classes	6	9	6
No. of subjects	2	2	25
No. of trials per subject	8	10	1
No. of conditions	2	5	4
Total No. of Samples	192	900	593

Table 5.1: Details of the data sets used in our experiments.

5.6 Experiments

5.6.1 Data Sets and Experimental Setting

We carried out our experiments in three data sets, namely facial expressions data set from Dollar et al.[12], hand gestures data set from Wong et al.[97] and KTH human action data set from Schuldt et al.[76]. In all data sets, each video sequence contains one activity. These data sets are public data sets and have been widely used for activities recognition. For detailed description of these data sets, please refer to the experiment section of previous chapter.

In our experiments, the video sequences were converted into gray level to avoid bias in color. The details of the data sets are summarized in Table 5.1 and some sample images from the video sequences are shown in Figure 5.3. In the experiments, we implemented a baseline approach using the “bag-of-words” representation for comparison. The recognition rates were obtained using leave-one-out cross-validation unless noted otherwise.

5.6.2 Pyramid Representation Approach for Human Action

Facial Expression

With the same experiment setting as in [12], we trained on one subject under one of the two lighting conditions and tested on: (1) the same subject under the same illumination, (2) the same subject under different illumination, (3) a different subject under the same illumination, and (4) a different subject under different illumination. Since Dollar’s implementation[12] used a “bag of words” approach, we used it as the baseline algorithm. We compared the confusion matrices in the first two scenarios from the two approaches in Figure 5.4. In the same subject under

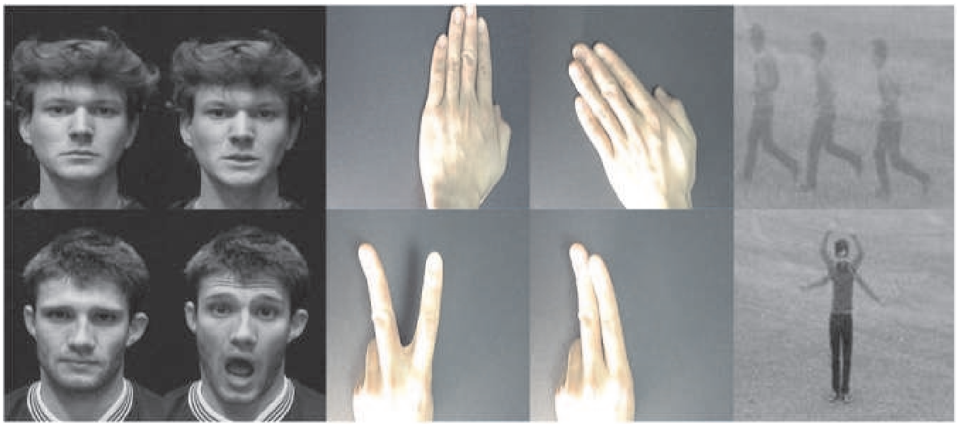


Figure 5.3: Sample images from the experiment data sets.

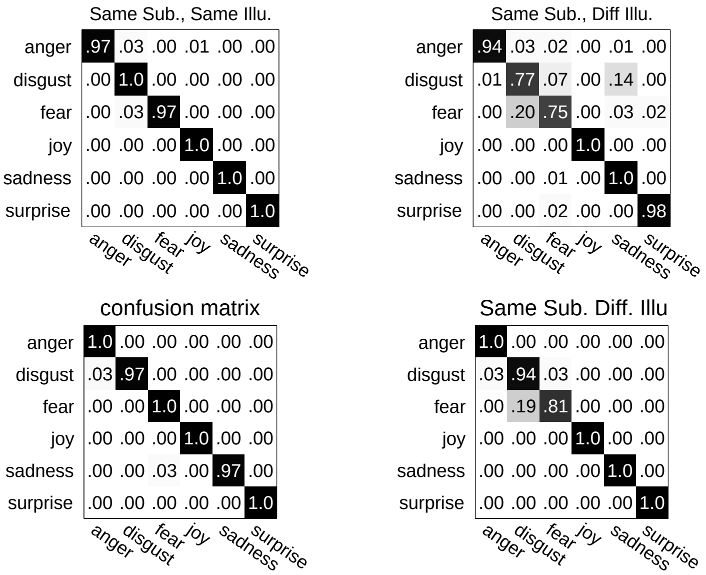
Methods	Same Sub. Same Illu.	Same Sub. Diff. Illu.	Diff Sub. Same Illu.	Diff Sub. Diff Illu.
Baseline	98.83	90.46	58.67	47.71
1 level Pyramid	$\mathbf{99.33}$	$\mathbf{95.29}$	78.33	69.29
2 level Pyramid	98.17	94.75	$\mathbf{78.92}$	$\mathbf{73.67}$

Table 5.2: The facial expression recognition rates($\%$)in different scenarios from the baseline "bag-of-words" algorithm and the spatiotemporal pyramids with different number of levels.

the same illumination scenario, the recognition task was easy. The baseline algorithm already achieved very hight recognition rate, so our approach only slightly improved the results. In the same subject under different illumination scenario, our approach has shown great improvements for all facial expression types.

We also show in Table 5.2 the recognition rates in all four different scenarios from the baseline "bag-of-words" algorithm and spatiotemporal pyramid representation with different number of layers. From the baseline algorithm, the average recognition rate across different scenarios is 73.92%. Both recognition rates from our pyramid representation with different number of levels have shown significantly improved performance. And the best result is 86.38% from the 2-level configuration.

We also tested on the facial expressions data set with the same experiment setting as in [97], which was the leave-one-out cross-validation. The average recognition rates along with



Same Sub., Same Illu.

	anger	disgust	fear	joy	sadness	surprise
anger	.97	.03	.00	.01	.00	.00
disgust	.00	1.0	.00	.00	.00	.00
fear	.00	.03	.97	.00	.00	.00
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.00	.00	1.0	.00
surprise	.00	.00	.00	.00	.00	1.0

Same Sub., Diff Illu.

	anger	disgust	fear	joy	sadness	surprise
anger	.94	.03	.02	.00	.01	.00
disgust	.01	.77	.07	.00	.14	.00
fear	.00	.20	.75	.00	.03	.02
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.01	.00	1.0	.00
surprise	.00	.00	.02	.00	.00	.98

confusion matrix

	anger	disgust	fear	joy	sadness	surprise
anger	1.0	.00	.00	.00	.00	.00
disgust	.03	.97	.00	.00	.00	.00
fear	.00	.00	1.0	.00	.00	.00
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.03	.00	.97	.00
surprise	.00	.00	.00	.00	.00	1.0

Same Sub. Diff. Illu

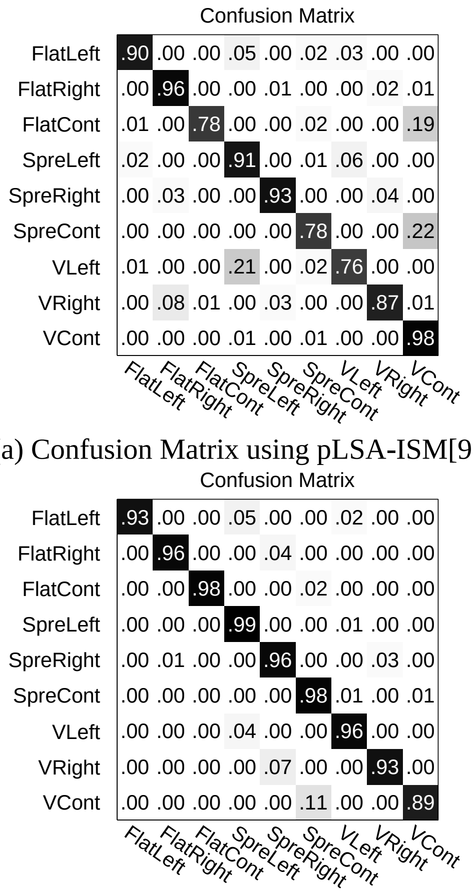
	anger	disgust	fear	joy	sadness	surprise
anger	1.0	.00	.00	.00	.00	.00
disgust	.03	.94	.03	.00	.00	.00
fear	.00	.19	.81	.00	.00	.00
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.00	.00	1.0	.00
surprise	.00	.00	.00	.00	.00	1.0

Figure 5.4: Comparison of recognition rates in the first two scenarios. The top row shows the confusion matrices from Dollar's implementation[12] and the bottom row shows the confusion matrices from our 1 level spatiotemporal pyramid representation.

Methods	Accuracy (%)	Std. Deviation
Base line	91.50	1.09
Pyramid (1 level)	**95.24**	1.35
Pyramid (2 level)	94.29	1.56
pLSA [97]	50.00	none
pLSA-ISM [97]	83.33	none

Table 5.3: The facial expression recognition rates along with their standard deviations obtained from different algorithms with leave-one-out cross-validation experiment setting.

their standard deviations from the different algorithms for the six types of facial expressions are listed in Table 5.3. This has shown the pyramid representation can improve on the "bag-of-words" baseline model and even achieve better performance than the complicated probabilistic latent semantic models[97]. Their standard deviations are not very large, which indicate that our results are not very sensitive to the randomness caused by clustering algorithm in our approaches.



Confusion Matrix

	FlatLeft	FlatRight	FlatCont	SpreLeft	SpreRight	SpreCont	VLeft	VRight	VCont
FlatLeft	.90	.00	.00	.05	.00	.02	.03	.00	.00
FlatRight	.00	.96	.00	.00	.01	.00	.00	.02	.01
FlatCont	.01	.00	.78	.00	.00	.02	.00	.00	.19
SpreLeft	.02	.00	.00	.91	.00	.01	.06	.00	.00
SpreRight	.00	.03	.00	.00	.93	.00	.00	.04	.00
SpreCont	.00	.00	.00	.00	.00	.78	.00	.00	.22
VLeft	.01	.00	.00	.21	.00	.02	.76	.00	.00
VRight	.00	.08	.01	.00	.03	.00	.00	.87	.01
VCont	.00	.00	.00	.01	.00	.01	.00	.00	.98

(a) Confusion Matrix using pLSA-ISM[97]

Confusion Matrix

	FlatLeft	FlatRight	FlatCont	SpreLeft	SpreRight	SpreCont	VLeft	VRight	VCont
FlatLeft	.93	.00	.00	.05	.00	.00	.02	.00	.00
FlatRight	.00	.96	.00	.00	.04	.00	.00	.00	.00
FlatCont	.00	.00	.98	.00	.00	.02	.00	.00	.00
SpreLeft	.00	.00	.00	.99	.00	.00	.01	.00	.00
SpreRight	.00	.01	.00	.00	.96	.00	.00	.03	.00
SpreCont	.00	.00	.00	.00	.00	.98	.01	.00	.01
VLeft	.00	.00	.00	.04	.00	.00	.96	.00	.00
VRight	.00	.00	.00	.00	.07	.00	.00	.93	.00
VCont	.00	.00	.00	.00	.00	.11	.00	.00	.89

(b) Confusion Matrix using 2 level spatial-temporal pyramid

Figure 5.5: The confusion matrices for recognizing the nine types of hand gestures. The top confusion matrix is obtained by using pLSA-ISM[97] and the bottom confusion matrix is from our work by using 2 level spatiotemporal pyramid representation.

Hand Gesture

We used the leave-one-out cross-validation experiment setting, which is the same as in [97], for hand gesture recognition. In this experiment, the video from one objects under one capturing condition was used in testing and the remaining was used in training.

The confusion matrices for recognizing the nine types of hand gestures are shown in Figure 5.5. The top confusion matrix is obtained by using pLSA-ISM[97] and the bottom confusion matrix is obtained from our approach by using 2 level spatiotemporal pyramid representation. It shows that by using the pyramid representation, the recognition rate has improved on every categories except the last one.

Methods	Accuracy (%)	Std. Deviation
Base line	85.96	0.68
Pyramid (1 level)	95.57	0.65
Pyramid (2 level)	**96.75**	0.61
pLSA [97]	76.94	none
pLSA-ISM [97]	91.94	none

Table 5.4: The hand gesture recognition rates obtained from different algorithms.

The average recognition rates along with their standard deviations from different algorithms for all hand gestures are shown in Table 5.4. This has shown the pyramid representation can achieve much better recognition rate than the "bag-of-words" approach and even exceed the results from complicated probabilistic latent semantic models[97]. The standard deviations from our methods are not very large, which indicate that our results are not very sensitive to the randomness caused by clustering algorithm in our approaches.

From experiments on both data sets, we do not observe any significant increase in performance beyond 2-level pyramid configuration. This is because when $l = 2$, the 64 subdivisions of the whole video sequence already roughly capture the sets of points' locations in the spatiotemporal domain while maintain tolerance for the locations variance in each cell. With more levels, the number of features points falling into each cell will be decreased, so the histograms might not be a good approximation for the feature distribution.

5.6.3 Frame Spatiotemporal Representation Approach

We set the parameters for the experiments empirically. The bandwidth for a smooth temporal weight, σ, is set to 5. The vocabulary size K, which is the cluster number in the k-means clustering algorithm, is set to 250. In our experiments, we observe that the performance does not improve much when the level of the spatial pyramid $L > 1$. Therefore, we use the setting of $K = 250$ and $L = 1$ or $L = 2$, which leads to a 1250-dimension or 5250-dimension vector for the key frame representation.

Methods	same sub. same illu.	same sub. diff. illu.	diff sub. same illu.	diff sub. diff illu.
Baseline	98.83	90.46	58.67	47.71
Our method (L=1)	$\mathbf{100.00}$	$\mathbf{96.25}$	74.38	71.71
Our method (L=2)	98.37	93.13	$\mathbf{74.83}$	$\mathbf{73.25}$

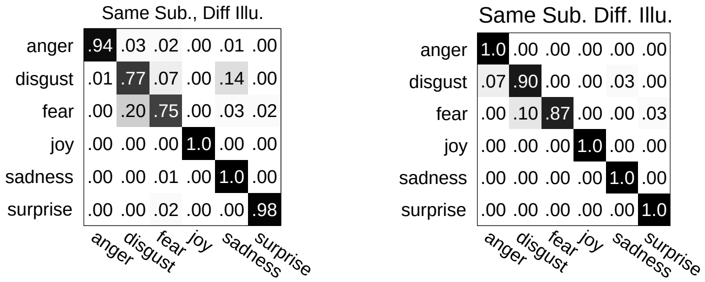
Table 5.5: The facial expression recognition rates($\%$) in different scenarios from the baseline "bag-of-words" algorithm and from our spatiotemporal representations with different spatial levels.

Experimental Results

With the same experimental setting on facial expression data set as in [12], we trained on one subject under one of the two lighting conditions and tested on: (1) the same subject under the same illumination, (2) the same subject under different illumination, (3) a different subject under the same illumination, and (4) a different subject under different illumination. The recognition rates in each scenario from Dollar's implementations[12], which is the baseline "bag-of-words" approach, and from our approaches are shown in Table 5.5. We can see that in the first scenario, the recognition task is easy. The baseline algorithm already achieved very high recognition rate, therefore our approaches only slightly improved the results. For the rest of the cases, our approaches with both configurations have demonstrated significant improvements.

As an example, the confusion matrices of the six facial expressions in the second scenario from the baseline implementation and our spatiotemporal representation with L=1 are reported in Figure 5.6. It shows improvements on recognition rates in every category of the facial expressions.

From the experiment on facial expression data set, we do not observe any significant increase in performance beyond 1-level spatial pyramid configuration. This is because when $l = 1$, the 4 subdivisions of the key frame already roughly capture the sets of feature points' locations in the spatial domain while maintain tolerance for the locations variance in each cell. With more levels, the number of features points falling into each cell will be decreased, so the histograms might not be a good approximation for the feature's distribution. So we will use



Same Sub., Diff Illu.

	anger	disgust	fear	joy	sadness	surprise
anger	.94	.03	.02	.00	.01	.00
disgust	.01	.77	.07	.00	.14	.00
fear	.00	.20	.75	.00	.03	.02
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.01	.00	1.0	.00
surprise	.00	.00	.02	.00	.00	.98

Same Sub. Diff. Illu.

	anger	disgust	fear	joy	sadness	surprise
anger	1.0	.00	.00	.00	.00	.00
disgust	.07	.90	.00	.00	.03	.00
fear	.00	.10	.87	.00	.00	.03
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.00	.00	1.0	.00
surprise	.00	.00	.00	.00	.00	1.0

Figure 5.6: Comparison of the confusion matrices in the second scenario on the facial expression data set. The left confusion matrix is from Dollar's implementation[12] and the right confusion matrix is from our spatiotemporal representation with L=1.

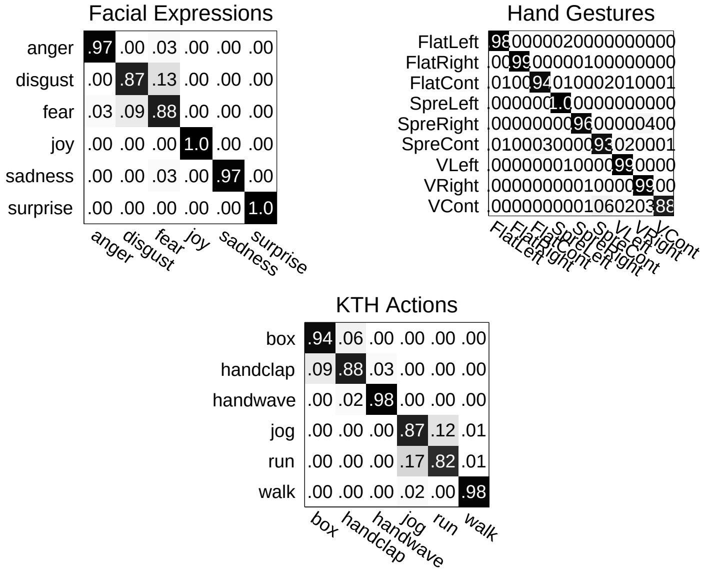
Methods:	Base-line	Our method	Wong [97]	Niebles [62]	Wang [92]
Facial Expressions	91.50	$\mathbf{94.83}$	83.33	none	none
Hand Gestures	85.96	$\mathbf{95.83}$	91.47	none	none
KTH Actions	81.98	91.17	83.92	81.50	$\mathbf{92.43}$

Table 5.6: The average recognition rates ($\%$) for facial expression, hand gesture and KTH human action data sets obtained from different algorithms. Our method is frame spatiotemporal representation approach

$l = 1$ for the rest of the experiments.

With leave-one-out cross-validation experimental setting, we tested the baseline and our proposed methods on all data sets. The confusion matrices from our method with L=1 are shown in Figure 5.7.The average recognition rates for all data sets, compared with other published results, are reported in Table 5.6. This demonstrates that our approach improves the "bag-of-words" baseline model and outperforms most of the other known methods while approaching the best known result.

To show the sensitivity of recognition rates of our methods on different data sets, the standard deviation along with the recognition rate is shown in Table 5.7



Facial Expressions

	anger	disgust	fear	joy	sadness	surprise
anger	.97	.00	.03	.00	.00	.00
disgust	.00	.87	.13	.00	.00	.00
fear	.03	.09	.88	.00	.00	.00
joy	.00	.00	.00	1.0	.00	.00
sadness	.00	.00	.03	.00	.97	.00
surprise	.00	.00	.00	.00	.00	1.0

Hand Gestures

	FlatLeft	FlatRight	FlatCont	SpreLeft	SpreRight	SpreCont	VLeft	VRight	VCont
FlatLeft	.98	.00	.00	.02	.00	.00	.00	.00	.00
FlatRight	.00	.99	.00	.00	.01	.00	.00	.00	.00
FlatCont	.01	.00	.94	.01	.00	.02	.01	.00	.01
SpreLeft	.00	.00	.00	1.0	.00	.00	.00	.00	.00
SpreRight	.00	.00	.00	.00	.96	.00	.00	.04	.00
SpreCont	.01	.00	.03	.00	.00	.93	.02	.00	.01
VLeft	.00	.00	.00	.01	.00	.00	.99	.00	.00
VRight	.00	.00	.00	.00	.01	.00	.00	.99	.00
VCont	.00	.00	.00	.00	.01	.06	.02	.03	.88

KTH Actions

	box	handclap	handwave	jog	run	walk
box	.94	.06	.00	.00	.00	.00
handclap	.09	.88	.03	.00	.00	.00
handwave	.00	.02	.98	.00	.00	.00
jog	.00	.00	.00	.87	.12	.01
run	.00	.00	.00	.17	.82	.01
walk	.00	.00	.00	.02	.00	.98

Figure 5.7: With leave-one-out cross-validation experimental setting, the confusion matrices on all three data sets from our spatiotemporal representation with L=1.

Data set	Recognition Rate	Standard Deviation
Facial Expressions	94.83	0.83
Hand Gestures	95.83	1.21
KTH Actions	91.17	1.54

Table 5.7: The average recognition rates (%) and standard deviation for facial expression, hand gesture and KTH human action data sets obtained from our method. Our method is frame spatiotemporal representation approach

5.6.4 Combinations of Previous Approaches for Human Action

This section I list the experimental results from the four combinations of previous approaches in the direction of "bag-of-meaningful-words" model and spatiotemporal model for the underlying activities in the video sequences. The recognition rates for all data sets are shown in Figure 5.8.

For comparison, I also list results from the original approaches. Because our approach for spatiotemporal representation for the whole video sequence can not handle the global motion case, there is no result for the KTH data sets, which contains global movement of human figure in the sequence. Thus, we also did not experiment with the combination of soft selection and

Methods:	Facial Expression	Hand Gesture	KTH Actions
Hard Selection	95.41	91.52	87.96
Soft Selection	93.36	91.83	87.47
Representation for sequence	95.24	96.75	none
Representation for key frame	94.83	95.83	91.17
Soft selection + Rep. for sequence	95.00	**97.25**	none
Soft selection + Rep. for key frame	**95.70**	96.33	91.16
Hard selection + Rep. for sequence	93.33	95.53	none
Hard selection + Rep. for key frame	95.30	95.70	**91.50**

Table 5.8: The average recognition rates (%) for facial expression, hand gesture and KTH human action data sets obtained from different combinations of “bag-of-meaningful-words” model and spatiotemporal models.

hard selection with such approach for KTH data set.

The results have shown that in most cases, the sequential combinations of approaches from previous this chapter and previous chapter produce comparable and even slightly better recognition rates. These make sense because the combinations have the benefits from both methods.

5.7 Summary

In this chapter, we addressed the human action recognition problem by incorporating spatial temporal information into the “bag-of-words” model. We present two approaches. In the first approach, we used a spatiotemporal pyramid representation for human activity recognition. This approach simultaneously integrates the spatiotemporal relation among local motion features with their appearance information and embeds these rich information in the pyramid representation for the video sequence. However, in this approach, we are only working on video sequences which contain activity starting and ending in a neutral position without global motion. This makes it easy to partition the sequence in the spatial and temporal domains. In the future, we intend to investigate methods to detect periodicity of the activity and compensate for global motion such that our approach can be applied in more general scenario.

In the second approach, we have presented a spatiotemporal key frame representation for human activity recognition. First, our approach selects the key frames of the video sequences based on their discriminative power. Next, our approach simultaneously integrates the spatiotemporal relations among local motion features with their appearance information and embeds these rich information in the representation for the selected key frames.

Our work differs from the pyramid match kernels[25, 45] in that: 1) Our goal is to find a suitable representation to integrate the spatiotemporal relations among motion features. The work in [25, 45] is seeking a suitable kernel function for two sets of image features. 2) Because our representation is a concatenated histogram, we measure the distance by Chi-square distance. The pyramid match kernels use histogram intersection as the distance function to satisfy the Mercer's condition.

Experiments have been conducted for the sequential combinations of methods from this chapter and previous chapter. The results show comparable and slight better recognition rates on all experimental data sets, which can be explained as the combination of methods have benefits from both schools of approaches.

Chapter 6

Saliency Detection

In the research of digital signal processing, a common approach for analyzing the digital signal in time domain is to transform it into frequency domain via Fourier transform. From the theory of Fourier transform, we know that any signal can be represented as an infinite weighted sum of an infinite number of sinusoids at different frequency. By using the Fourier transform, we change the basis in the time domain to a set of sinusoids in the frequency domain, which gives us new perspectives to study the signal as the sum of sinusoids with different frequency.

Fourier transform (FT) has been applied in computer vision in a similar way. Because the majority of the research subjects in computer vision are images and video sequence, which can be seen as the two dimensional x-y signal and three dimensional x-y-t signal respectively, we can use the same signal processing technique and apply 2-D Fast Fourier Transform (FFT) and 3-D FFT to them. After the transformation, we have new representation for the objects in frequency domain, which give us new insights from the frequency and energy perspectives.

In this chapter, we will use a Fourier Transform based method to detect saliency region in two dimensional images and three dimensional video sequences. Saliency detection is very important processing stage in the general object recognition framework, which can help fast local visual feature detection and representation.

6.1 Motivation

When human does object recognition, typically, the first step is object detection, which aims at extracting an object from its background before recognition[32]. Many traditional methods model this process as the detection of specific categories of objects [17], [74]. These approaches are based on the training of the particular features of the target. However, this is different from what human vision system does. When human detects object, he does not have the prior information about what he expects to see. And human can achieve this efficiently. Usually at a glance, people can detect what he is interested in, which is much faster than most of traditional methods based on complicated training models.

So how does the human visual system work to efficiently detect a general interesting location, where the object is likely to exist? It is believed to be a two step procedure [83, 52]. The first step is a fast and simple pre-attentive process and the second step is a slow and complex attention process. It is during the first step that low level vision features such as orientation, edges and corners ”pop-up” to human eyes as salient regions. These regions contain general features which catch human’s attention and not specific for a particular object category. And the detection of these regions should be very efficient and fast.

Saliency detection has many potential applications. As the first step in object recognition, saliency detection can help us choose the locations where we will apply more complex algorithms for recognition. This approach can increase the speed for the processing of the images and videos.

Saliency detection can also be applied to image and video compression. After we detect the salient regions, we can compress them with less information loss and the other regions, which are likely to be the background, with more information loss. Since human pays more attention to the salient region, this approach will lead to better perceptual results with the same information loss. Saliency detection can also help in image segmentation. Saliency detection is to find the regions which catch human’s attention. The result is inline with human’s perception of segmentation for the interesting regions against the uninterested background.

To find the salient regions in a given image, many models have been proposed in the field of machine vision. In [39, 37, 38], Itti and Koch proposed a saliency model to stimulate the visual search process of human based on [83] and built a system called Neuromorphic Vision C++ Toolkit. Recently, Walther [91] further extended this model to create Saliency tool box and applied it for object recognition task. However, these models demand high computational cost and tuning for various parameters.

Recently, Xiaodi Hou [32] proposed an approach to detect the saliency region from the frequency domain based on Fourier Transform. This method first computes the spectral residual, which is the difference between the original image and its smoothed version in the log amplitude spectrum. Together with the phase spectrum, the spectral residual is transformed back to spatial domain to construct the saliency map, which marks the salient region. This method does not rely on parameters and can detect salient regions rapidly. Chenlei Guo [28] extends this method by only keeping the phase spectrum, which contains the location information for the inverse transform. This method also achieved promising results.

In this chapter, we will first discuss a general framework for the saliency detection in the frequency domain. This approach models the saliency detection as an enhancement method for the amplitude of the high frequency components in the Fourier spectrum of the digital signal. In this approach, the salient region, where many varying sinusoidal components exist, will be enhanced by re-distribution of the energy in the frequency domain. This framework explains the methods from [32] and Chenlei Guo [28] as special cases and introduces a new method which achieves similar results as those from [32] and [28]. Secondly, we will extend this framework to apply to the three dimensional data and detect salient region from video sequences by using Fourier Transform along the temporal dimension.

In section 6.2, we will introduce the general framework and apply it for image and video data in section 6.3. Experiments in section 6.4 will demonstrate this framework and the discussion and conclusion are given in section 6.5.

6.2 A General Framework for Saliency Detection

6.2.1 Background in Discrete Fourier Transform

Typically, a video clips is a series of consecutive frames. Let $f_t(x, y)$ be the 2D image of size $H \times W$ at frame t, the video clips of N frames can be stacked along time axis and constructed as a 3D spatial-temporal image of size $H \times W \times N$:

$$f(x, y, t) = f_t(x, y) \tag{6.1}$$

for $x = 0, 1, 2W - 1, y = 0, 1, 2H - 1, t = 0, 1, 2, , N - 1$. In this representation, we can process 3D video clips in the same way as 2D images.

The formal definition of Fourier transform of a three dimensional signal $f(x, y, t)$ is

$$F(u,v,w) = \int\int\int_{-\infty}^{+\infty} f(x,y,t)e^{-i2\pi(ux+vy+wt)}dxdydz \tag{6.2}$$

In computer vision research, the images and video are discrete digital signal. So three-dimensional discrete Fourier Transform is used to transfer a 3D spatial-temporal image of size $H \times W \times N$:

$$F(u,v,w) = \sum_{x=0}^{W-1}\sum_{y=0}^{H-1}\sum_{t=0}^{N-1} f(x,y,t)e^{-i2\pi(ux/W+vy/H+wt/N)} \tag{6.3}$$

After the Fourier transform is applied, image or video sequence, which is typically a real function in the spatial space, is transformed into frequency domain, typically as a complex function:

$$F(u,v,w) = R(u,v,w) + j * I(u,v,w) \tag{6.4}$$

where $R(u,v,w)$ and $I(u,v,w)$ are the real and imaginary components of $F(u,w,w)$, i.e.

$$R(u,v,w) = \frac{1}{W*H*N}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}\sum_{t=0}^{N-1} f(x,y,t) * cos[2\pi(\frac{ux}{W} + \frac{vy}{H} + \frac{wt}{N})] \tag{6.5}$$

$$I(u,v,w) = \frac{1}{W*H*N}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}\sum_{t=0}^{N-1} f(x,y,t) * sin[2\pi(\frac{ux}{W} + \frac{vy}{H} + \frac{wt}{N})] \tag{6.6}$$

The Fourier transform can also be represented in polar form, as pair of amplitude component $Amplitude(u,v,w)$ and phase components $Phase(u,v,w)$:

$$Amplitude(u,v,w) = R^2(u,v,w) + I^2(u,v,w) \tag{6.7}$$

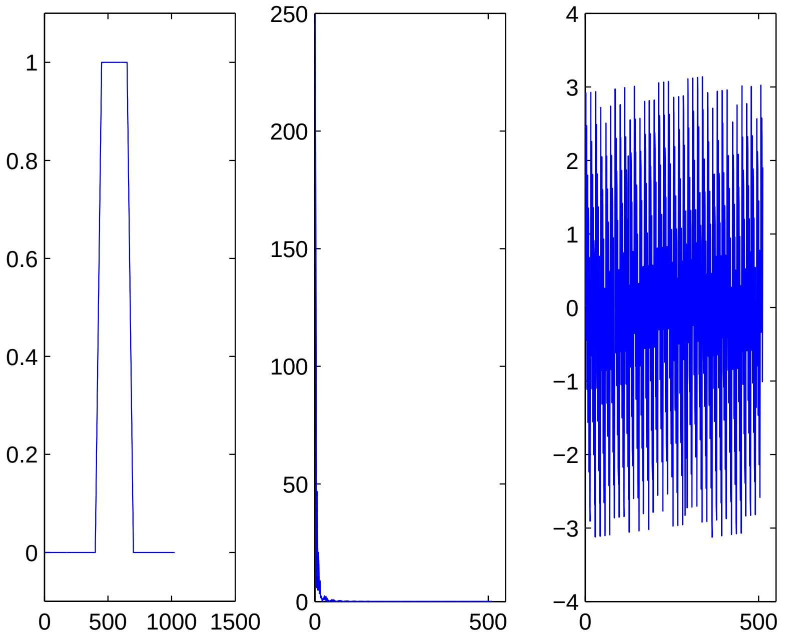


Figure 6.1: The left one is the original signal. The middle one is the amplitude spectrum and the right one is the phase spectrum of the original signal by Fourier Transform.

$$Phase(u,v,w) = tan^{-1}(\frac{I(u,v,w)}{R(u,v,w)}) \tag{6.8}$$

In the Fourier frequency spectrum, the phase components contain the location information and the amplitude components contain the spatial structure information.

6.2.2 Saliency Detection for One-Dimensional Signal

Let's first discuss a one dimensional positive impulse signal as an example and give its waveform, amplitude and phase spectrum from Fourier Transform in figure 6.1.

From the theory of Fourier Transform, we know that a signal in spatial domain can be transformed into frequency domain and decomposed into amplitude spectrum and phase spectrum. Usually, the amplitude at lower frequency is much larger than those from the higher frequency. Since the energy is indicated by the amplitude, we can also say the energy is concentrated more at the lower frequency portion of the spectrum.

The saliency region of the signal is where many irregularities exist. These irregularities are composed by many sinusoidal components with different frequency in the spectral domain. To make it even more obvious, we need to enhance the high frequency components by scaling their amplitude such that their scales are more comparable to those in low frequency components. From the energy perspective, this is to redistribute the energy so that the energy from high frequency components is no longer dominated by those from the low frequency components.

From statistics, we know Logarithmic transformation can be applied for skewed data to reduce the variability. To enhance the high frequency component, we can also apply Logarithmic transformation to the amplitude spectrum of the signal to reduce its variability.

Formally, we formulate our approach of saliency detection as enhancing the high frequency components by applying logarithmic transformation to the amplitude spectrum of the signal and reconstructing the saliency map by Inverse Fourier Transform from these enhanced components. Let $S(t)$ denotes the signal S at time t, $FFT(S)$ and $FFT^{-1}(s)$ denote Fourier and Inverse Fourier Transform respectively and $R(s)$ and $P(s)$ for the amplitude and phase spectrum respectively. Our approach is:

1. Fourier transform of signal S:

$$s(f) = FFT(S(t)) \tag{6.9}$$

2. Take the amplitude and phase spectrum respectively:

$$Amplitude(f) = R(s(f)) = R(FFT(S(t))) \tag{6.10}$$

$$Phase(f) = P(s(f)) = P(FFT(S(t))) \tag{6.11}$$

3. Apply logarithmic transformation to $Amplitude(f)$ for enhancement:

$$Enhance(f) = log(Amplitude(f)) \tag{6.12}$$

4. Construct saliency map by Inverse Fourier Transform from the enhanced amplitude spectrum and the corresponding phase spectrum:

$$\begin{aligned} SaliecyMap(t) &= FFT^{-1}(Enhance(f)exp(iPhase(f))) \quad (6.13) \\ &= FFT^{-1}(log(Amplitude(f))exp(iPhase(f))) \quad (6.14) \end{aligned}$$

The results of applying our method to the signal in Figure 1 are shown in the second row of Figure 2. The salient region are at around $x = 0.45$ and $x = 0.6$ as indicated by the two peaks at those locations in the saliency map.

6.2.3 The Relations with Other Methods

Both Xiaodi Hou [32] and Chenlei Guo[28] methods are similar to this framework. They just use different transformation to enhance the high frequency components. In the work of Xiaodi Hou [32], the log spectral residual is defined as

$$Residual(f) = log(Amplitude(f)) - h(f) * log(Amplitude(f)) \quad (6.15)$$

In which the second term is smoothing $log(Amplitude(f))$ in its local neighborhood with a linear average filter $h(f)$. Usually the second term is close to $log(Amplitude(f))$, so we can assume a frequency f' close to f whose log amplitude spectral can approximate the second term. Thus the $Residual(f)$ can be rewritten as

$$\begin{aligned} s(x) &= log(Amplitude(f)) - log(Amplitude(f')) \\ &= log(\frac{Amplitude(f)}{Amplitude(f')}) \quad (6.16) \end{aligned}$$

The construction of the saliency map from [9] can be rewritten as:

$$\begin{aligned} s(x) &= F^{-1}(e^{Residual(f)+i*Phase(f)})^2 \\ &= F^{-1}(e^{Residual(f)} * e^{i*Phase(f)})^2 \\ &= F^{-1}(\frac{Amplitude(f)}{Amplitude(f')} * e^{i*Phase(f)})^2 \quad (6.17) \end{aligned}$$

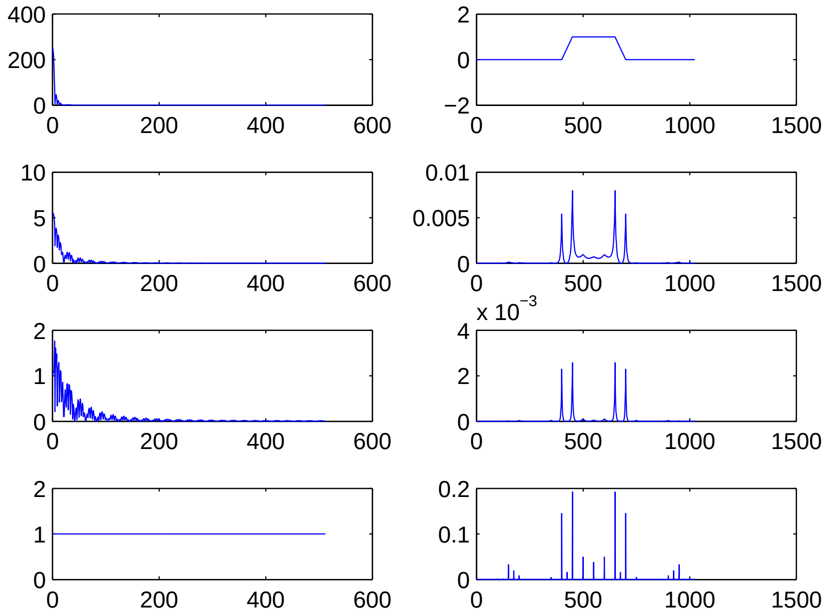


Figure 6.2: The saliency detection for a one dimensional signal. The first row is the amplitude spectrum and the original signal. The second row results from using our proposed transform. The third row results from using transform from [32] and the forth row results from using transform from [28]. For the bottom three rows, the left column is the transformed amplitude spectrum and the right column is the constructed saliency map.

So instead of applying logarithmic transformation to the amplitude spectrum, this method uses the ratio of amplitude component and its average from its neighborhood. In this way, the amplitude components are normalized locally and the values of the components at both low frequency and high frequency are comparable. So this is a way to enhance the components at high frequency. The result of applying this method to the signal in Figure 6.1 are shown in the third row in Figure 6.2 for comparison.

In the work of [28], the information from amplitude information is ignored and the amplitudes at all frequencies are set to one. The construction of saliency region is:

$$s(x) = F^{-1}(e^{i*Phase(f)})^2 \tag{6.18}$$

This is also an enhancement for high frequency components because now the energy is uniformly distributed and the impact for saliency map construction from high frequency components is no longer dominated by those from the low frequency. The result of applying this method to the signal in Figure 6.1 is shown in the forth row in Figure 6.2 for comparison.

6.3 Saliency Detection for Images and Video Sequences

6.3.1 Saliency Detection for Images

The image can be seen as a two dimensional digital signal and the salient regions in the image are the places where many varying sinusoidal components locate. As in the same way for one-dimensional signals, we can detect the salient region in images by enhancing the high frequency components.

Since the saliency, such as edges, corners etc, can come from both x and y dimensions, we first apply two-dimensional Fourier Transform to get the amplitude and phase spectrum. We then apply logarithmic transformation to the components from amplitude spectrum for enhancement. In the end, we construct the saliency map from these enhanced amplitude components and their corresponding phase components. The comparison with the results from other transform from [32] and [28] is in section 6.4.

6.3.2 Saliency Detection for Video Sequences

Saliency detection can also be applied to video sequences for motion detection. Motions are at the places where there are many changes in the temporal domain. If we model the frames of a video sequence as a group of signals changing along the temporal dimension, the motion can also be detected as the saliency region when we enhance the high frequency components in the amplitude spectrum along the temporal dimension.

To detect the saliency, we first apply Fourier Transform along the temporal dimension to the three dimensional video sequence data. Then we apply logarithmic transform to the amplitude spectrum to enhance the high frequency components. In the end, we construct the saliency map by Inverse Fourier Transform from the enhanced amplitude and phase spectrum. Some motion detection samples are shown in section 6.4.

6.4 Experiments

We have carried out experiments on both still images and video clips. Because saliency detection plays an important role in identifying the regions for object or motion detection, we use the object or motion detection to measure the results as in [32] and [28].

Since saliency map is an explicit representation for the possible object or motion locations, we use a simple threshold segmentation method for detection. Give the saliency map $S(x)$ of an image, the object map is obtained:

$$O_s(x) = \begin{cases} 0, & \text{if } S(x) > Threshold \\ 1, & \text{otherwise} \end{cases} \tag{6.19}$$

Empirically, we set $threshold = cE(S(x))$, where $E(S(x))$ is the average intensity of the saliency map and c is a constant we can change. The selection of the threshold is a tradeoff problem between false alarm and the neglect of the object.

6.4.1 Still Image

In our experiments for still images, we adopted the same setting from [32] for comparison. In this experiment, 62 images of natural scene are downsized to 320×240 to test the performance of our proposed method and the methods from [32] and [28]. The ground truth is the object regions hand labeled by human.

Samples for the saliency maps constructed from various methods are shown in Figure 6.3. It is shown that the saliency maps are similar across different methods and are compatible with the object regions labeled by human.

6.4.2 Video Sequences

We used a video sequence capturing passing vehicles on a street in a rainy day for experiment. This clip has 181 frames, each of which is 120×160 pixels. This clip captures a streetlight pole, which is periodically partially occluded by rain. This might lead to false motion detection from other methods that detect motion by comparing the frame by frame pixel difference. Our method suppresses this low frequency periodicity by enhancing high frequency components for

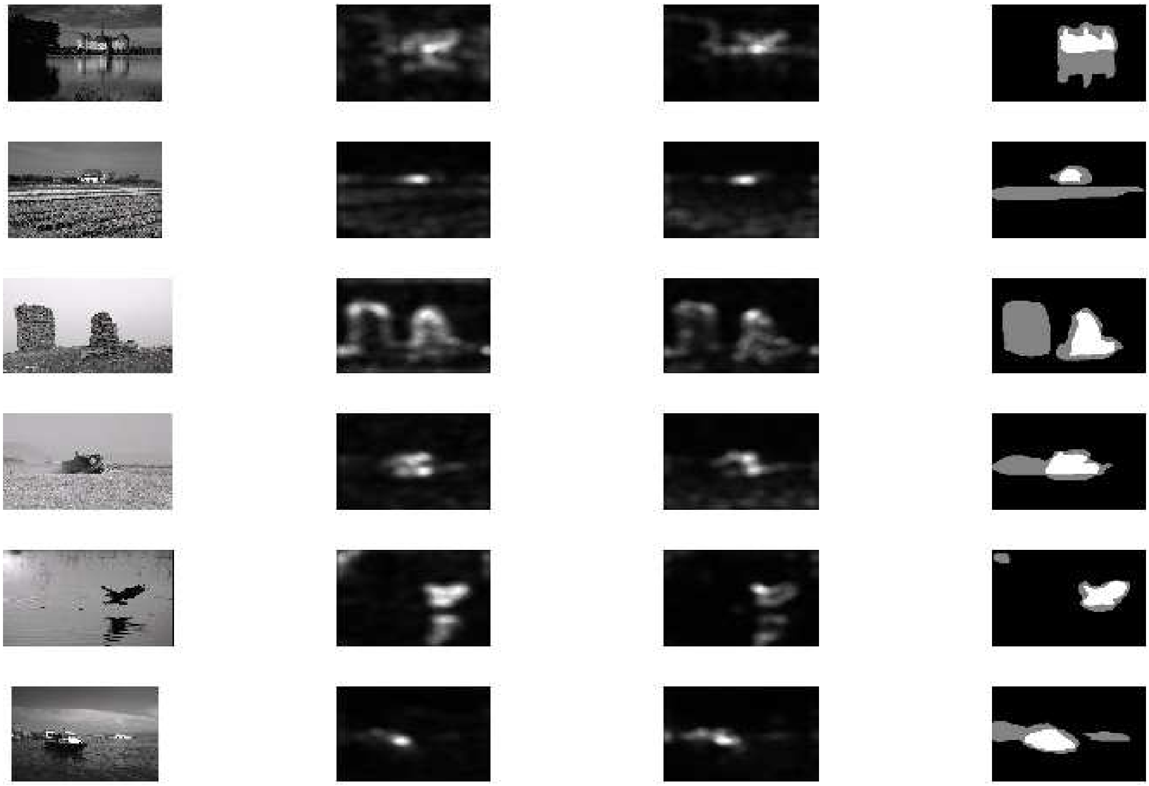


Figure 6.3: Saliency map from different methods. The first column is the original images. The second column is the saliency map constructed from our method. The third column is the saliency map from [32], the fourth column is the saliency map from [28] and the fifth column is the object map hand labeled by human.

more irregularity, such as the passing cars. Four snapshots of the video sequence with passing cars are shown in Figure 6.4, together with the corresponding saliency maps and the object maps.

6.5 Summary

In this chapter, we propose a framework for general object saliency detection. Our approach addresses the saliency detection problem in frequency domain as an enhancement problem for high frequency components. Our contributions are:

1. Model the saliency detection problem in a more general enhancement framework, which explains other saliency detection methods from [9] and [10] and introduces a new method which has similar results.

2. Extend the saliency detection framework for video sequences via Fourier Transform along the temporal dimension.

Saliency detection is an important step in the local visual feature framework for object and activity recognition. Successfully solving this problem can help fast detection of local visual features.

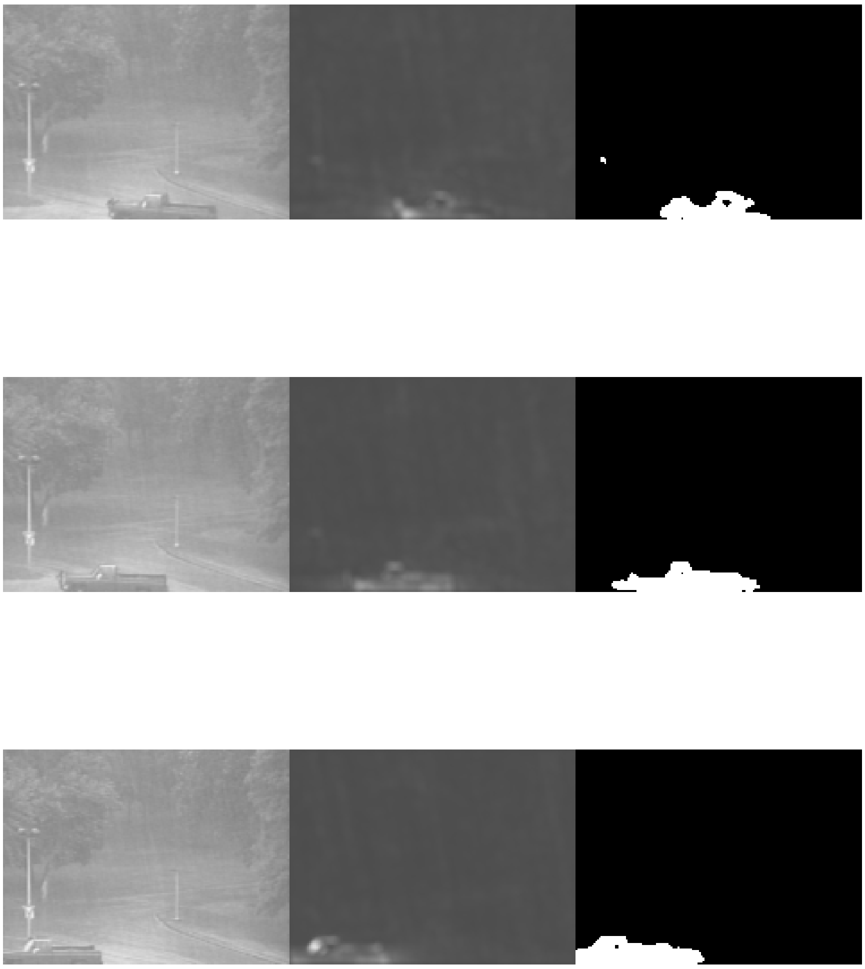


Figure 6.4: Saliency map and the object map from a video sequence. These are four snapshots of the video sequence. The left frames are from the original clip. The middle frames are the corresponding saliency maps and the right frames are the object maps built from saliency maps by threshold segmentation

Chapter 7

Conclusions

Recognition is one of the essential problems in computer vision with many applications. Recent advances of the computer power of modern computers also push the progress of research on recognition. Traditional methods such as model-based methods, appearance template based methods have been intensively studied and gained success in many applications. Recently, local visual features based methods, which usually infer the recognition results from statistical models of local visual features, have gained much attention from computer vision researchers. This statistical based methods can model target class without accurate geometric model, which is difficult to obtain, and handle more complicated scenarios, such as deformable target, partial occlusion. Because of these advantages, local visual features based methods can work on many problems which can not be successfully solved by traditional methods.

This thesis follows the local visual feature framework and addresses some important problems in this framework. Namely, they are:

1) How to select good local visual features in the images and build a statistical model that model the target class and location simultaneously.

2) In the popular “bag-of-words” model, how to select the vocabulary for this model such that the resulting model is more compact and contains less irrelevant words for the representation.

3) Also for the “bag-of-words” model, which is based on the naive Bayesian assumption of independence among local visual features, how to improve this model by integrating spatial and temporal information into the representation for activity recognition in a three-dimensional space.

For the first problem, we propose a general object recognition system featuring a two stage method for selecting local image features, which characterize the target object class. The first stage uses a combinatorial optimization formulation for clustering a weighted multipartite graph. The following stage is a statistical method for selecting discriminative local visual features from the positive images. This recognition system integrates spatial information of local feature into recognition by estimating the joint probability of the target class and its relative location. This work improves the recognition rate over the other known methods on a benchmark data set.

To choose a more compact and relevant vocabulary for the “bag-of-words” model, we need metrics to measure the importance of visual words in the representation of an application. Conditional entropy comes naturally because it indicates the purity of the distribution of activity given a cluster of local visual features, which typically corresponds to a visual word. So in chapter 4 of this thesis, we introduce conditional entropy based vocabulary selection methods to build “bag-of-meaningful-words”. Both hard selection, which simply discards some irrelevant words in the vocabulary, and soft selection, which assigns different conditional entropy based weights to visual words in the vocabulary have been proposed. Experiments on various representative human action data sets, which include facial expression, hand gesture and general human activities in KTH data sets, have shown improved performance over the traditional “bag-of-words” model.

To further integrate spatial and temporal information into the “bag-of-words” model for human activity recognition, we model the human motion with the distribution of local motion feature and their spatial-temporal arrangements. Instead of applying histogram to approximate the distribution in the traditional local visual feature space, we also consider both the spatial and the temporal space and use the histogram in a three-dimensional space to approximate the joint distribution.

Two methods have been proposed. The first approach uses a spatiotemporal pyramid representation for human activity. This approach works by partitioning video sequence into increasingly fine subdivisions in the space and time domains and modeling the distribution of the local motion features inside each subdivision such that the set of motion features are mapped into spatial and temporal multi-resolution histograms. This spatiotemporal pyramid is built by weighting the histograms from the different layers of the subdivisions. The proposed approach is an extension of the orderless “bag-of-words” model by approximately capturing geometric and temporal arrangements of the local motion features.

However, this approach assumes the target human actions do not have global motion or the locations of local motion features in the three-dimensional spatial temporal space are normalized, for example by estimating the global motion, to cancel its impact such that the spatial temporal location of local motion features are comparable. So we introduce the second approach, in which the local motion features used for the representation of a frame are the ones detected in this frame and others integrated from its temporal neighbors. The features' spatial arrangements are captured in a hierarchical spatial pyramid structure. Since for this approach, only local visual features from neighboring frames are considered, the global motion among them will be small and can be assumed to be zero when modeling. By using frame by frame voting for the recognition, experiments have demonstrated improved performances over most of the other known methods on the popular benchmark data sets while approaching the best known results. One of the drawbacks from frame by frame voting scheme is that it is computational expensive. Again we used entropy as the measure to select the informative frames for voting. Since the voting from frames are independent from each other, we believe we can use parallel computing to help solve this problem.

We also explored analyzing computer vision problem from the frequency domain. The problem we are addressing is saliency detection, which is to detect the places where human usually pay attention to at the first glance. Successfully solving this problem can help fast detection of local visual features. The salient regions typically contain more information than other places and will pop up to the human eyes. From frequency analysis perspective, these regions are places where many varying sinusoidal components exist. However, typically, these high frequency sinusoidal components are dominated by low frequency components. If these places are enhanced by re-distribution of the energy in the frequency domain and compared with the original images, the large difference indicates the possible location for saliency. In this thesis, we explain the previous two methods for frequency analysis of saliency detection as special cases of a school of a more general approach and present our own transformation for re-distribution of energy in frequency domain.

Therefore, some very important problems in different phases of the global local visual features framework for recognition have been addressed in this thesis in various ways. They are salience detection in the local visual detection and representation phase, local visual feature selection in the feature selection phase, entropy based words selection in vocabulary for “bag-of-more-meaningful” words, and integration of spatial temporal information into the representation in the target modeling and recognition phases. So my thesis is in line with a promising approach to recognition, a fundamental problem in computer vision.

References

[1] A. Agarwal and B. Triggs, “Hyperfeatures: Multilevel local coding for visual recognition,” in *ECCV06*, 2006, pp. I: 30–43.

[2] S. Agarwal and D. Roth, “Learning a sparse representation for object detection.,” in *ECCV*, 2002, pp. 113–130.

[3] J. K. Aggarwal and Q. Cai, “Human motion analysis: A review,” *Computer Vision and Image Understanding*, vol. 73, pp. 428–440, 1999.

[4] A. Baumberg, “Reliable feature matching across widely separated views,” pp. 774–781.

[5] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[6] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *ICCV ’05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, (Washington, DC, USA), IEEE Computer Society, 2005, pp. 1395–1402.

[7] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[8] E. Borenstein and S. Ullman, “Class-specific, top-down segmentation.,” in *ECCV*, 2002, pp. 109–124.

[9] Z. Chen and H. Lee, “Knowledge-guided visual perception of 3-d human gait from a single image sequence,” vol. 22, pp. 336–342, 1992.

[10] T. F. Cootes, C. Taylor, and M. M. Pt, “Statistical models of appearance for computer vision,” 2004.

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, 1995.

[12] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *ICCCN ’05: Proceedings of the 14th International Conference on Computer Communications and Networks*, 2005, pp. 65–72.

[13] G. Dorkó and C. Schmid, “Selection of scale-invariant parts for object class recognition.,” in *ICCV*, 2003, pp. 634–640.

[14] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in *ICCV ’03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003, p. 726.

[15] C. Fanti, L. Zelnik-Manor, and P. Perona, “Hybrid models for human motion recognition,” in *CVPR ’05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1*, (Washington, DC, USA), IEEE Computer Society, 2005, pp. 1166–1173.

[16] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition.,” *IJCV*, vol. 61, no. 1, pp. 55–79, 2005.

[17] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning.,” in *CVPR (2)*, 2003, pp. 264–271.

[18] R. Fergus, P. Perona, and A. Zisserman, “A sparse object category model for efficient learning and exhausitive recognition.,” in *CVPR*, 2005.

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

[20] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, 1991.

[21] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *EuroCOLT ’95: Proceedings of the Second European Conference on Computational Learning Theory*, (London, UK), Springer-Verlag, 1995, pp. 23–37.

[22] D. Gao and N. Vasconcelos, “Discriminant saliency for visual recognition from cluttered scenes,” in *NIPS*, 2004.

[23] D. M. Gavrila, “The visual analysis of human movement: A survey,” *Computer Vision and Image Understanding: CVIU*, vol. 73, no. 1, pp. 82–98, 1999.

[24] L. J. V. Gool, T. Moons, and D. Ungureanu, “Affine/ photometric invariants for planar intensity patterns,” in *ECCV ’96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, (London, UK), Springer-Verlag, 1996, pp. 642–651.

[25] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in *ICCV ’05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2005, pp. 1458–1465.

[26] W. Grimson and T. Lozano Perez, “Model-based recognition and localization from sparse range or tactile data,” vol. 3, no. 3, pp. 3–35, 1984.

[27] W. Grimson and T. Lozano Perez, “Localizing overlapping parts by searching the interpretation tree,” vol. 9, no. 4, pp. 469–482, July 1987.

[28] C. Guo, Q. Ma, and L. Zhang, “Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform,” 2008, pp. 1–8.

[29] C. Harris and M. Stephens, “A combined corner and edge detector,” 1988, pp. 147–152.

[30] D. Hogg, “Model-based vision: A program to see a walking person,” vol. 1, no. 1, pp. 5–20, February 1983.

[31] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” pp. 389–407, 1992.

[32] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” 2007, pp. 1–8.

[33] M. K. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. IT-8, pp. 179–187, February 1962.

[34] D. Huttenlocher and S. Ullman, “Object recognition using alignment,” 1987, pp. 102–111.

[35] D. Huttenlocher and S. Ullman, “Recognizing solid objects by alignment,” 1988, pp. 1114–1124.

[36] N. Ikizler and P. Duygulu, “Human action recognition using distribution of oriented rectangular patches,” in *HUMO07*, 2007, pp. 271–284.

[37] L. Itti and C. Koch, “A saliency-based search mechanism for overt and covert shifts of visual attention,” *Vision Research*, vol. 40, pp. 1489–1506, 2000.

[38] L. Itti and C. Koch, “Computational modelling of visual attention,” *Nat Rev Neurosci*, vol. 2, no. 3, pp. 194–203, March 2001.

[39] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

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

[41] Y. Ke and R. Sukthankar, “Pca-sift: a more distinctive representation for local image descriptors,” in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2004, pp. 506–513.

[42] J. J. Koenderink and A. J. van Doom, “Representation of local geometry in the visual system,” *Biol. Cybern.*, vol. 55, no. 6, pp. 367–375, 1987.

[43] Y. Lamdan, J. Schwartz, and H. Wolfson, “On recognition of 3-d objects from 2-d images,” 1988, pp. 1407–1413.

[44] I. Laptev, “On space-time interest points,” *Int. J. Comput. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[45] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *CVPR ’06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[46] C. Lee and A. Elgammal, “Dynamic shape outlier detection for human locomotion,” vol. 113, no. 3, pp. 332–344, March 2009.

[47] M. Leung and Y. Yang, “First sight: A human body outline labeling system,” *T-PAMI*, vol. 17, pp. 359–377, 1995.

[48] T. K. Leung and J. Malik, “Recognizing surfaces using three-dimensional textons,” in *ICCV (2)*, 1999, pp. 1010–1017.

[49] J. Li and N. M. Allinson, “A comprehensive review of current local features for computer vision,” *Neurocomput.*, vol. 71, no. 10-12, pp. 1771–1787, 2008.

[50] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. of the International Conference on Computer Vision ICCV, Corfu*, 1999, pp. 1150–1157.

[51] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[52] W. J. M, “Guided search 2.0. a revised model of visual search,” *Psychonomic bulletin and review*, vol. 1, pp. 202–238, 1994.

[53] X. Ma and W. E. L. Grimson, “Edge-based rich representation for vehicle classification,” in *ICCV ’05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, (Washington, DC, USA), IEEE Computer Society, 2005, pp. 1185–1192.

[54] S. Mahamud and M. Hebert, “The optimal distance measure for object detection,” 2003, pp. I: 248–255.

[55] S. H. Manning, C. D., *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[56] M. Marszaek and C. Schmid, “Spatial weighting for bag-of-features,” in *CVPR ’06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2118–2125.

[57] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[58] M. Minsky, “Project mac robotics,” 1965.

[59] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Comput. Vis. Image Underst.*, vol. 104, no. 2, pp. 90–126, 2006.

[60] H. Moravec, “Towards automatic visual obstacle avoidance,” 1977, p. 584.

[61] E. Murphy-Chutorian and J. Triesch, “Shared features for scalable appearance-based object recognition.,” in *WACV/MOTION*, 2005, pp. 16–21.

[62] J. Niebles, H. Wang, H. Wang, and F. Li, “Unsupervised learning of human action categories using spatial-temporal words,” in *BMVC06*, 2006, p. III:1249.

[63] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *CVPR ’06*, 2006, pp. 2161–2168.

[64] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, “Weak hypotheses and boosting for generic object detection and recognition.,” in *ECCV (2)*, 2004, pp. 71–84.

[65] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, “Pedestrian detection using wavelet templates,” 1997, pp. 193–199.

[66] R. Polana and R. Nelson, “Detecting activities,” in *DARPA93*, 1993, pp. 569–574.

[67] R. Polana, R. Nelson, and A. Nelson, “Low level recognition of human motion (or how to get your man without finding his body parts),” in *In Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, Press, 1994, pp. 77–82.

[68] M. Pontil and A. Verri, “Support vector machines for 3d object recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 637–646, 1998.

[69] S. Savarese, A. DelPozo, J. Niebles, and F. Li, “Spatial-temporal correlatons for unsupervised action classification,” 2008, pp. 1–8.

[70] S. Savarese, J. Winn, and A. Criminisi, “Discriminative object class models of appearance and shape by correlatons,” in *CVPR ’06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2033–2040.

[71] F. Schaffalitzky and A. Zisserman, “Multi-view matching for unordered image sets, or ”how do i organize my holiday snaps?”.,” in *ECCV (1)*, 2002, pp. 414–431.

[72] B. Schiele and J. L. Crowley, “Object recognition using multidimensional receptive field histograms,” in *ECCV ’96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, (London, UK), Springer-Verlag, 1996, pp. 610–619.

[73] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval.,” *IEEE PAMI*, vol. 19, no. 5, pp. 530–535, 1997.

[74] H. Schneiderman and T. Kanade, “A statistical method for 3d object detection applied to faces and cars,” 2000, pp. 45–51.

[75] H. Schneiderman and T. Kanade, “Object detection using the statistics of parts,” *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[76] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in *ICPR04*, 2004, pp. III: 32–36.

[77] N. Sebe and M. S. Lew, “Comparing salient point detectors,” *Pattern Recogn. Lett.*, vol. 24, no. 1-3, pp. 89–96, 2003.

[78] F. G. Sobel, I., “A 3x3 isotropic gradient operator for image processing,” 1968.

[79] Y. Song, L. Goncalves, and P. Perona, “Unsupervised learning of human motion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 814–827, 2003.

[80] J. B. Tenenbaum and W. T. Freeman, “Separating style and content with bilinear models,” *Neural Comput.*, vol. 12, no. 6, pp. 1247–1283, 2000.

[81] C. Thurau, “Behavior histograms for action recognition and human detection,” in *HUMO07*, 2007, pp. 299–312.

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

[83] A. M. Treisman and G. Gelade, “A feature-integration theory of attention,” *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, January 1980.

[84] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience, vol3 no 1 pp 77-86, 1994*.

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

[86] S. Ullman and A. Shashua, “Structural saliency: The detection of globally salient structures using a locally connected network,” in *MIT AI Memo*, 1988.

[87] A. Vashist, C. Kulikowski, and I. Muchnik, “Ortholog clustering on a multipartite graph,” in *Proceedings of Algorithms in Bioinformatics (WABI), LNCS*, volume 3629, 2005, pp. 328–340.

[88] M. Vidal-Naquet and S. Ullman, “Object recognition with informative features and linear classification,” in *ICCV ’03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, (Washington, DC, USA), IEEE Computer Society, 2003, p. 281.

[89] P. Viola and M. Jones, “Robust real-time object detection,” *International Journal of Computer Vision 2002*.

[90] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features.,” in *CVPR*, 2001.

[91] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch, “Attentional selection for object recognition - a gentle way,” in *in Proc. of 2nd Workshop on Biologically Motivated Computer Vision (BMCV’02*, Springer, 2002, pp. 472–479.

[92] Y. Wang, P. Sabzmeydani, and G. Mori, “Semi-latent dirichlet allocation: A hierarchical model for human action recognition,” in *HUMO07*, 2007, pp. 240–254.

[93] M. Weber, M. Welling, and P. Perona, “Unsupervised learning of models for recognition,” in *ECCV (1)*, 2000, pp. 18–32.

[94] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan, “Categorization nine visual classes using local appearance descriptors.,” in *IWLAVS*, 2004.

[95] H. Wolfson and Y. Lamdan, “Geometric hashing: A general and efficient model-based recognition scheme,” 1988, pp. 238–249.

[96] H. J. Wolfson and I. Rigoutsos, “Geometric hashing: An overview,” *IEEE Computational Science & Engineering*, vol. 4, no. 4, pp. 10–21, /1997.

[97] S. Wong, T. Kim, and R. Cipolla, “Learning motion categories using both semantic and structural information,” in *CVPR07*, 2007, pp. 1–6.

[98] A. Yilmaz and M. Shah, “Recognizing human actions in videos acquired by uncalibrated moving cameras,” in *ICCV ’05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, (Washington, DC, USA), IEEE Computer Society, 2005, pp. 150–157.

[99] Z. Zhao and A. Elgammal, “Information theoretic key frame selection for action recognition,” in *BMVC08*, 2008.

[100] Z. Zhao, A. Vashist, A. M. Elgammal, I. B. Muchnik, and C. A. Kulikowski, “Combinatorial and statistical methods for part selection for object recognition,” *Int. J. Comput. Math.*, vol. 84, no. 9, pp. 1285–1297, 2007.

Vita

Zhipeng Zhao

Education

Ph.D. Computer Science, Rutgers University, New Jersey 2009

M.S. Statistics, Rutgers University, New Jersey 2007

M.S. Computer Science, Old Dominion University, Virginia 2000

B.S. Computer Science, Tsinghua University, Beijing, P.R. China 1997

Publications

- Zhipeng Zhao and Ahmed Elgammal. “Entropy-Based Vocabulary Selection for Action Recognition”. Submitted to the Asian Conference on Computer Vision (ACCV09), 2009.
- Zhipeng Zhao and Ahmed Elgammal. “Human Activity Recognition from Frames Spatiotemporal Representation”. In the International Conference on Pattern Recognition (ICPR08), 2008.
- Zhipeng Zhao and Ahmed Elgammal. “Information Theoretic Key Frames Selection for Action Recognition”. In British Machine Vision Conference (BMVC08), 2008.
- Zhipeng Zhao and Ahmed Elgammal. “Spatiotemporal Pyramid Representation for Recognition of Facial Expression and Hand Gestures”, In International Conference on Automatic Face and Gesture Recognition (FG08), 2008.
- Zhipeng Zhao, Akshay Vashist, Ahmed Elgammal, Ilya Muchnik and Casimir Kullikowski. “Combinatorial and Statistical Methods for Part Selection for Object Recognition”. International Journal of Computer Mathematics, volume 84 Issue 9, 2007.
- Zhipeng Zhao and Ahmed Elgammal. “A Statistical Selected Part-Based Probabilistic Model for Object Recognition”. In International Workshop on Intelligent Computing in Pattern Analysis/Synthesis, (IWICPAS’06), Xi’an, China, 2006. LNCS 4153 pp95-104
- Akshay Vashist, Zhipeng Zhao, Ahmed Elgammal, Ilya Muchinik and Casimir Kullikowski. “Discriminative Part Selection using Combinatorial and Statistical Models for Part-Based Object Recognition”. In Beyond Patches Workshop in conjunction with CVPR06 2006.