2010

Posture detection by kernel PCA-based manifold learning

Peng Cheng
University of Wollongong

Posture Detection by Kernel PCA-based Manifold Learning

A thesis submitted in fulfillment of the

requirements for the award of the degree

Master of Computer Science

from

UNIVERSITY OF WOLLONGONG

by

Peng Cheng

School of Computer Science and Software Engineering

September 2010

Dedicated to

Yuan-ying Cheng and Er-liang Zhang

Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Peng Cheng
September 29, 2010

Abstract

A Posture detection system aims to identify and localize any specific types of postures in images and video sequences. Unlike human or pedestrian detection where only one class of objects is required to be detected, posture detection is designed to detect multiple classes of postures. It remains a challenging problem because human bodies are complex and articulated with very diversified appearances. Posture detection often relies on a good generalization of the variations from large quantity of training examples that cover different situations. In this thesis, we devise a new posture detection framework that combines the histogram of gradient (HOG)-based feature with a novel manifold-based open-set classifier designed to achieve a better generalization. In this framework, each posture class is represented by a complex manifold that lies in the high-dimensional visual input space. The manifold is learned using Kernel PCA. Classification of a new observation is achieved by comparing it to each trained posture manifold. In addition, a new greedy Kernel PCA approximation algorithm is proposed to speed up the learning of the posture manifolds. The approximation algorithm seeks to remove the redundant training samples in the kernel space while best retaining the accuracy of kernel mapping, resulting in a new kernel PCA model that provides almost

identical learning and classification ability to the original kernel PCA with significantly lower computational cost. Both the detection framework and approximation algorithm were tested on 2D and 3D artificial datasets and real human and posture datasets. The results have shown that the approximation algorithm is effective and the proposed framework can provide accurate and efficient detection of different postures with a relatively small training set.

Acknowledgments

Great thanks to my supervisors, Associate Professor Wanqing Li and Professor Philip Ogunbona, for their guidance and demonstration of the required attitudes and principles of a researcher, and their constant encouragement and motivation. Also great thanks to everyone in the Advanced Multimedia Research Lab: Ce Zhan, Nguyan Duc Thanh, Alister Coordiner, Shenbo Guo, Jun Hu, Zhenqiang Jiang and Zhou Sun for sharing their ideas and offering constructive discussions. This thesis would have not not been completed without their help.

Contents

List of Tables

List of Figures

0.1 Glossary

Adaboost : Adaptive boosting, a boosting algorithm for ensemble learning.

auROC : Area under Receiver's Operating Characteristic curve, a scalar equal to the integration of an ROC curve from 0 to 1, used to measure the performance of classifiers.

DET : Decision Error Trade-off curve, a curve showing the missing rate versus the false positive per window (FPPW).

DR : Detection Rate, a scalar denoting the percentage of successful detection per window.

DT : Distance Transform, a 2D pixel map showing the distance from pixels to a given contour.

DoG : Derivative of Gaussian.

FPPW : False Positive per Window, a scalar denoting the percentage of false detection per window.

GMM : Gaussian Mixture Model, a statistic model for probabilistic estimation of a multivariate distribution.

HMM : Hidden Markov Model, a random process model primarily used in modeling temporary events.

HOG : Histogram of Gradient, a visual feature mainly used in human detection and body representation.

ICA : Independent Component Analysis: a toolset consisting of several matrix decomposition and factorization techniques that aim to isolate statistically mutually independent bases from a set of data.

ISM : Implicit Shape Model, an object detection framework by synthesizing local detection results through spatial voting.

Isomap : Isometric mapping, a nonlinear manifold embedding technique to reconstruct a mapping space that preserves the graph-distance.

KPCA :Kernel Principal Component Analysis, a generalized nonlinear manifold learning framework that performs the principal component analysis in a kernelized feature space.

KNN :K-Nearest Neighbors, a simple example-based classifier that labels each unknown datum to the majority of its K-Nearest Neighbors.

LBP : Local Binary Pattern, a visual descriptor that is invariant to illumination.

LLE : Locally Linear Embedding: a nonlinear manifold embedding technique to reconstruct a mapping space that minimizes the change of distances between adjacent data.

LPP : Locally Preserving Projection, a linear dimensionality reduction technique that aims to find a linear mapping to a lower dimension space that minimizes the same objective function with Laplacian eigenmap.

MDS : Multidimensional Scaling, a toolset that aims to reconstruct a equivalent dataset in an explicit Euclidean space from a similarity or dissimilarity matrix.

MHI : Motion History Image, a 2D greyscale image characterising motion information of a binary video.

MRF : Markov Random Field, a 2D statistic graphic model used to model spatially correlated random variables.

NMF : Non-negative Matrix Factorization, a matrix decomposition and factorization technique that aims to isolate non-negative bases from a dataset.

PCA : Principal Component Analysis.

RANSAC : Random Sample Consensus, a fast model estimation meta-algorithm.

RBF : Radial Basis Function, a bivariate function in the form of $f(||x - y||^2)$.

RMI : Recurrent Motion Image, a 2D greyscale image characterising recurrent motion information of a binary video.

ROC : Receiver's Operating Characteristic curve, a curve showing the detection rate versus the false positive detection.

ROI : Region of Interest.

SIFT : Scale Invariant feature transform.

SVM : Support Vector Machine.

0.2 Notations

$E(.)$: expectation of a variable or a set of random vectors.

$cov(.)$: covariance matrix of a variable or a set of random vectors.

$p(X)$: marginal probability or likelihood of a random variavle X.

$p(X|Y)$: conditional probability or likelihood of X given Y.

X : the column matrix of a training dataset.

x_i : the i^th vector in the training dataset.

c_i : the class of the i^th example.

$\phi(.)$: the implicit non-linear mapping that maps [.] into an infinite-dimensional feature space.

Φ : the implicit column matrix of $\phi(x_i)$.

$k(.,.)$: the positive semidefinitive bivariate kernel function that defines $\phi(.)$ by its inner-product.

K : the inner-product matrix of $\phi(x_i)$ where $K_{ij} = k(x_i, x_j)$.

H : the constant matrix for centralizing data in the feature space. $H = I_n - 1_n$, I_n is an $n \times n$ identity matrix and 1_n denotes a $n \times n$ matrix in which each element takes the value of $1/n$.

$\hat{\Phi}$: the implicit column matrix of $\phi(x_i)$ centered at zero mean.

$\hat{\phi}(x_i)$: the i^th column vector of $\hat{\Phi}$.

$\hat{K}$: the inner-product matrix of $\hat{\Phi}$, $\hat{K} = HKH$.

λ_i : the i^th eigenvalue.

D : the diagonal matrix with each diagonal element $D_{ii} = \lambda_i$.

P : the implicit column matrix of eigenvectors of $cov(\hat{\Phi})$ (also known as principal components).

A : the column matrix of eigenvectors of $\hat{K}$.

z : an arbitrary datum sample in the test dataset to be classified.

c : the ground truth of z.

$y(.)$: the function that maps z into the subspace of principal components in the feature space.

H_A : the abbreviation for HA, it is the most important matrix in defining the KPCA model and the projection $y(.)$.

$w(.)$: polynomial part of $y(.)$ defined by $w(.) = (H_A)^T k(X, .)$, where $k(X, .) = [k(x_1, .), k(x_2, .) \dots k(x_n, .)]^T$

b : the constant part of $y(.)$ defined by $b = (H_A)^T K 1_n$

U : the multidimensional scaling result of K.

$\tilde{X}$: the column matrix of a subset of the training dataset selected by a kernel approximation algorithm.

$\tilde{P}$: shortened P for faster KPCA mapping $y(.)$.

$\tilde{H_A}$: shortened H_A for faster KPCA mapping $y(.)$.

$W_{\tilde{X}}$: the w-mapping of the subset $\tilde{X}$ into the KPCA mapping space (the column space of the original principal components P).

Q **and** R : QR decomposition of $D_n^{\frac{1}{2}} W_{\tilde{X}}$.

V : the set of $D_n^{\frac{1}{2}} W_X$ orthonormalized with $W_{\tilde{X}}$.

Chapter 1

Introduction

1.1 Motivation

Posture detection aims to identify and locate specific types of postures from images or video sequences. It is one of the fundamental problems in computer vision and is an important step in many applications like security surveillance, human-machine interaction and semantic image/video retrieval. Unlike human detection that can be regarded as a two-class classification problem, the posture detection is often required to detect human body and recognize specific posture types simultaneously. The key challenge is that human body is a highly articulated and deformable object and its appearance can vary substantially due to various factors such as clothing, viewpoints, illuminations and shadows. As a result, learning the discriminative appearance features without any prior knowledge is very challenging. This thesis is concerned with the problem of posture detection under a realistic assumption that training samples are only available for the postures to be detected and there are no negative samples (e.g. samples for uninterested postures and non-human scenes).

Many attempts have been made for more effective and efficient learning of the

appearance of humans and postures. Most work mainly has focused on the following two issues.

1. Extraction of features that can discriminate postures and a cluttered background and are invariant to irrelevant variations.

2. Effective representation of the postures that can be obtained from a limited number of examples.

In this thesis, we seek to find a new posture representation. Specifically, the manifold representation is proposed. This representation aims to enclose the change of appearance caused by continuous movements and, hence, the variations of postures by smooth lower-dimensional manifolds. By using kernel principal component analysis (KPCA) as a manifold learning tool, the learning of multiple postures does not need any negative examples. Based on the manifold representation, a new open-set classifier has been proposed for posture detection. The classifier works by measuring the distance between a test example and the closest manifold. Experiments on several posture databases have verified the performance of the proposed detection method.

A major drawback of KPCA based manifold representation is that its computational cost is proportional to the number of examples that define the manifold, and is often impractical to be used in real-time detection. In this thesis we have proposed an approximation of KPCA that can significantly reduce the number of required examples while best preserving the configuration of posture manifolds. Experimental results have shown that the proposed approximation outperforms other kernel approximation approaches in both speed and accuracy, and is proved to be effective in constructing a faster detector.

1.2 Contributions

The key contributions of this thesis are:

1. A new KPCA approximation method which significantly accelerates the KPCA calculation and therefore broadens the applications of KPCA.

2. A new posture detection method that employs KPCA for learning and identifying posture manifolds without negative sample.

1.3 Publication List

The following publications have resulted from the work reported in this thesis:

- Peng Cheng, Wanqing Li and Philip Ogunbona, Greedy approximation of kernel PCA by minimizing the mapping error. In *Proc. of Digital Image Computing: Techniques and Applications.* pages 303–308, 2009

- Peng Cheng, Wanqing Li and Philip Ogunbona, Kernel PCA of HOG features for posture detection. In *Proc. of Image and Vision Computing New Zealand.* pages 415–420, 2009

1.4 Outline of the thesis

The rest of the thesis is organized as follows. Chapter two provides a survey of the literature published in the recent ten years on human and posture detection. Based on the strategies adopted, the existing methods are first categorized into *Detection by*

patterns and *Detection by local descriptors*. They are then reviewed critically. In addition, the criteria and benchmark datasets used to measure and compare the detection performance are described. The major challenges related to posture detection are then analyzed.

Chapter three presents an open-set classification framework which is built upon the manifold learning using Kernel PCA (KPCA). In particular, it introduces an improved KPCA learning algorithm that avoids the problem of numerical instability that may exist for high dimensional data. The open-set classification is achieved by measuring the reconstruction error. To overcome the problem of high computational cost associated with conventional KPCA, we propose in this chapter a new approximation algorithm that aims to find a reduced KPCA to approximate the kernel mapping. Experimental results are given on both real and simulated data.

Chapter four presents a new approach for detecting human postures from single images. This approach follows the detection by pattern framework. In the training stage, KPCA is employed to learn the manifold span of a set of examples represented in the feature space by the histogram of gradient (HOG) feature. The HOG is able to represent a posture effectively. In the detection stage, the open-set classifier presented in the previous chapter is iteratively applied on the HOG of every detection window of the image to identify and locate the postures. Experimental results are given on some popular datasets.

Chapter five concludes the thesis. It summarizes the advantages and disadvantages of the proposed algorithms and presents some possible future work.

Chapter 2

Literature Review

This chapter provides a survey of the literature published in the recent ten years on human/posture detection . Specifically, we categorize the existing methods into two approaches *Detection by patterns* and *Detection by local descriptors*, and each approach has been comprehensively reviewed. Also, in this chapter the various criteria used to evaluate a posture detector are presented and challenges in posture detection are analyzed.

2.1 Overview

Vision-based human motion analysis has been an active research topic for the past ten years in computer vision due to its scientific challenges and numerous applications including security surveillance, human-machine interaction, unmanned systems and information retrieval. Its ultimate objective is to develop a system that can understand the human motion from visual information and further predict the intention of humans.

Given a video sequence, a human motion analysis system often consists of the following four components:

1. **Posture detection**: it aims to identify and locate specific types of postures from images or videos; the output includes one or several labelled bounding boxes that describe the posture classes, positions and scales of the detected postures. The detector should ignore the presence of any postures in the scene that are not of interest to the application.

2. **Posture estimation**: it aims to infer the human kinematic model from a localized posture patch or silhouette, each patch or silhouette should be well-bounded and contains only one human, the output is a 2D or 3D kinematic model.

3. **Posture tracking**: given a human model that is initialized with the posture configuration in the first frame of a video, it tracks the movement of body and limbs by using visual information and temporal correspondence amongst the following frames.

4. **Action/Gait analysis**: given a set of identified human postures or consecutive kinematic models that represent an action, it identifies the action or the identity of the subject performing the action.

Since a human motion can be defined by a sequence of static postures, extraction and analysis of postures are usually, explicitly or implicitly, the first and most important component in such a system. Its performance is often critical to the overall performance of the system.

Definition of posture varies according to the application context. Briefly, posture refers to the configuration of human body, consisting of information about relative positions and directions of limbs (the arms and legs), head and torso. This definition

Figure 2.1: Top view of a walking person, its shape can only be recognized if the viewpoint is known.

is often used in *posture estimation* or *posture tracking* where the objective is to extract and track the human kinematic model. However, it has also been shown that in both neural perception and many practical motion analysis systems the recognition of human motion does not require an accurate kinematic model. Instead, human motion can be represented as a sequence of finite *salient postures*. Motion analysis system based on this representation has been shown to be effective [45].

This thesis is concerned with the detection of salient postures or simply postures which are defined by examples. This problem is different from the commonly-known human detection problem, which has been studied extensively in previous literature. The aim of the human detection is to detect any human body appearing in an image regardless of its posture. In posture detection, a defined set of postures need to be recognized and postures outside this set are not of interest. Furthermore, we assume that only positive examples, i.e. examples of the postures of interest are available.

Compared to the detection of other types of objects, the uniqueness of posture detection is that a posture has a very flexible visual pattern due to the articulated and diverse nature of body appearance. Therefore, it is believed that human perception of human body may involve empirical learning of appearance and understanding of kinematic structure under different environment and visual context. Figure 2.1 shows

an extreme condition, where the strange shape of human body is not familiarized by most people (therefore empirical learning of human shape from this angle is impossible). However it is still easy for a human to identify this shape based on the knowledge of the kinematic structure and the aspect of the camera.

Intuitively, posture detection can be carried out using a two step framework: human detection followed by posture estimation. However, in practice most existing human detection algorithms are designed only for upright standing figures. If we are only interested in specific postures (e.g. fouls in sporting matches or malicious acts in public spaces), the human detection will be unlikely to yield a good result. In this case, detection of the specified postures will be more appropriate. In addition, posture estimation often requires searching a huge feature space to find proper kinematic configurations for all human bodies in an image and this is impractical for detection.

Despite its significance in human motion analysis, posture detection has rarely been studied as an independent problem in previous literature. Due to this reason, this chapter will mainly focus on the publications on human detection whilst others have also approached posture tracking or estimation, but in general they focus primarily on detection [86] [25] [36]. In terms of the aims, human detection may be considered as a special case of the problem of posture detection. Theoretically, some human detection methods may be extended for posture detection by replacing the two-class classifier with its multi-class version even though the extension may not be effective and efficient.

According to our taxonomy, the existing methods are categorized into two approaches:

1. **Detection by patterns**: works by searching over the whole image space. At every location, an image window at certain scale is classified into one of the desired or learned postures or non-postures by comparing the image window to examples in a defined set of postures. Usually the comparison is performed in a representative or discriminative features space.

2. **Detection by local descriptors**: works by first detecting interest points or regions denoting different body parts, then finding the postures by inference from these points using the structural information of the human body.

In this chapter, over 60 publications are surveyed. These publications represent the state-of-the-art of their respective era in achieving better performances compared to previous systems or gaining better robustness when applied to new working conditions and assumptions.

The remaining part of this chapter will be organized as follows: sections 2.2 and 2.4 provide the literature survey of the approaches denoted as *detection by patterns* and *detection by local descriptors* respectively. Section 2.5 describes the commonly used methods to evaluate performances and efficiencies of the surveyed approaches. Section 2.6 discusses potential challenges of the problem of posture detection and possible solutions to overcome them. The last section concludes this chapter.

2.2 Detection by patterns

Given an image or a frame of a video sequence with an unknown number of postures of interest, a typical detection by patterns framework consists of five steps: first, regions

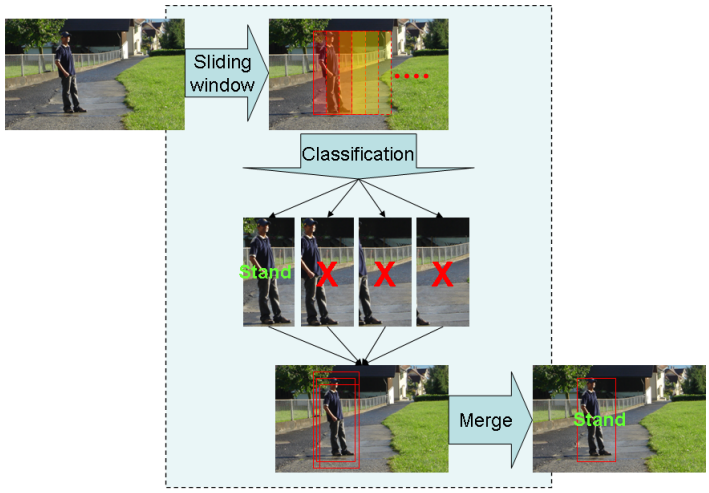


Figure 2.2: The framework of detection by patterns

of interest (ROIs) are extracted from the image often using simple pixel-level image processing techniques. The ROIs indicate the areas in which postures are likely to appear. This step is not a necessary step, but it does reduce the search space. Then, a sliding window iteratively scans over all possible positions in the ROIs. Very often the entire image area is scanned without a need for the first step. At each position, an image patch is extracted. In the third step, for each patch, features are extracted. In the forth step, each patch is classified into one of the trained postures or into an unknown or negative class. Depending on how the image is scanned, it would result in many overlapping bounding boxes specifying the positions of a possible posture. In the last step, adjacent bounding boxes are merged together to form one posture as the final decision. This framework is illustrated in Figure 2.2.

2.2.1 Extraction of ROIs

ROI extraction aims to find the areas in which the postures of interest are likely to appear so as to limit the search space. This process is often required to be fast and, therefore, pixel-level information and simple image processing techniques are usually employed. For an image, this process is usually omitted since there is no robust way to define the ROIs. For a video, motion information is often used to define the ROIs by assuming that human body is usually moving in contrast to the static or homogeneously moving background. The most simple method to extract the ROIs is background subtraction when the background is static or slowly changing. It extracts foreground regions from background by finding all pixels that are not consistent with the background.

Background subtraction usually involves only comparison between pixels and can be performed quickly. In the simplest form, the background can be represented as a single image and pixels in a video frame are considered as foreground if their brightness or color deviate significantly from the corresponding background pixels [9]. The idea was later extended to adaptive background subtraction [70] to handle unstable and slowly changing environment (e.g. illumination of outdoor scene may change slowly). In this method, an adaptive background model, such as the one based on Gaussian Mixture Models (GMM), is constructed to model the variation of each background pixel independently, [56][4][84][85][87][86][51][39]. Shadows may be removed based on the assumption that the shadow pixels have the same hue as the background but are of lower brightness [86] than the non-shadow pixels in the background.

Background subtraction is fast and stable but cannot handle a moving background.

Such a problem can be overcome by motion based segmentation, a technique that estimates the movement of brightness pattern of pixels (including both foreground and background pixels) between two or more consecutive frames of a video based on their adjacency and brightness. Two typical motion segmentation techniques are used in the surveyed works: optical flow determination and affine background estimation.

The optical flow is a 2D vector field that represents the relative movements of brightness pattern of pixels in two consecutive frames. Optical flow provides cues to segment and identify background regions and moving foreground objects. Determining optical flow from the brightness patterns of various pixels is an ambiguous problem because two different motions can lead to identical displacement of the pattern (a phenomenon known as aperture effect). Therefore traditional optical flow algorithms usually perform poorly on estimating the true motion of a complex articulated object, such as humans. As a result, the assumption of 'blob flow' is often imposed. It assumes that the optical flow in a blob (that may correspond to a fixed body part of human body or an entire body) can be represented by a homogeneous affine transform (consisting of translation, scaling and rotation). This idea was adopted in [12].

The Affine background estimation [24, 81, 24] works under the assumption that the background is homogeneously 'shifting' (e.g. background of a video captured by a sweeping surveillance camera or a vehicle-based camera) and can be approximated by a static image undergoing affine transform, and usually covers a large portion of the image. It aims to estimate the affine transform by maximizing the matching score between 2 consecutive frames. After the estimation, a motion compensated background subtraction can be used to find the ROIs. In [18], the affine transform is determined

by reducing the difference of the transformed picture of previous frame and the next frame at the four corners. In [81], the Affine transform is determined by the following method. First, calculate the optical flow field by the diamond search flow estimation. Then, exclude the flow field on regions where brightness is homogeneous (considering any aperture effect, the estimation on these regions are most likely to be erroneous). Finally, the optical flow field is approximated by a global affine transform; a random sample consensus (RANSAC) robust estimator is applied to find the affine transform that can best approximate the flow field. The main idea of the RANSAC is to randomly select a subset of the image pixels to generate a model hypothesis and to check how well this model fits to the remaining image pixels. This is done iteratively a fixed number of times or until some accuracy condition is met.

2.2.2 Feature extraction

For a reliable detection of postures, the selection of proper visual features is critical. The chosen features have to be sufficiently discriminative in reference to the postures and invariant to appearance variations caused by the various factors as discussed above. Note that the property of invariance is no less important than the discriminative property because of the special articular nature of the human body. A very detailed feature may have a high discriminative power but low invariance, which makes it difficult to use since an universal classifier cannot be generalized from limited number of examples. To deal with the problem of partial occlusions, the features should also be local-sensitive: where a change of a section of the image only affects the features of the corresponding

components so that the postures can still be recognized from the remaining components. An evaluation of contemporary local-sensitive visual features can be found in [50].

There are many types of features that have been proposed or adopted to characterize humans. In general, they can be categorized into three classes: shape-based features, appearance-based features and spatio-temporal joint features. Note that the use of these features is not mutually exclusive, they can be combined together to exploit their respective advantages (as shown in [76] and [77]).

Shape-based feature

The shape-based feature describes the shapes of ROIs. Due to their dependency on ROI extraction their usage was limited. The extraction of the shape-based features often relies on the extraction of ROI. However, due to their simplicity they are often used in real-time detection from video and stereo images (recently this usage has declined).

Contour and silhouette are two basic shape-based features. The former is often represented by a closed curve that encircles the object (usually this curve is a piecewise linear curve or spline curve denoted by a couple of key points). The latter is often represented by a binary map which has value 1 inside the object contour and value 0 outside the object contour [84][85][86][87]. In [10], the Hu moment descriptor of the binary map, consisting of 14 coefficients, was used to achieve rotation and scale invariance.

A shape histogram is another shape-based feature that describes the distribution of silhouette or contour. It divides an image plane into several vertical/horizontal lines

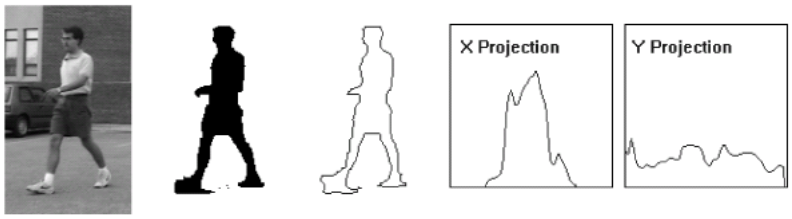


Figure 2.3: From left to right: input image, silhouette, contour, and histograms of x and y axes[36]

or blocks and counts the number of the contour's key points or silhouette's pixels with value 1 in each block or line. The shape histogram can reduce the feature dimensionality and increase its robustness with respect to small variations of the object shape. In [56], shape histograms in vertical and horizontal lines were used as a shape descriptor. In [4], discrete cosine transform (DCT) coefficients of the shape histogram were used as a shape descriptor.

Appearance-based feature

The appearance-based feature usually describes pixel-level information of an image. It is often extracted directly from the patch without extraction of ROIs. Therefore, the appearance-based feature is the most commonly used type of features. According to the survey of [50], appearance-based features can be categorized into spatial-frequency features, differential-based features and distribution-based features.

Spatial-frequency features are derived from the 2D spectrum of an image and usually constructed by correlating with different scales of spectral basis. Motivated by their success in signal processing field, the spatial-frequency features has been extensively studied. Today the most common spatial-frequency feature is probably the

wavelet-based features. A wavelet transform provides a multi-resolution decomposition of the image. Lower-rank coefficients capture the large-scale brightness distribution and higher-rank coefficients capture the local brightness variations. Wavelet features are invariant to scale transform but are neither locally sensitive nor invariant to illumination changes. The Haar wavelet is the most widely used wavelet basis due to its computational efficiency. It was firstly used by Oren et al. [55] for whole body detection, then in [52] for local body part detection. In [57] and [58], a reduced set of Haar wavelet coefficients was manually selected that encodes the outline of the human body, which greatly increased the detection speed at the cost of small decrease in detection rate. In [17], Gabor wavelet coefficients were used as feature vectors. The author observed that most of these coefficients are close to zero, and a set of 'principal' basis coefficients can be chosen from these coefficients by dynamically reducing the reconstruction error. In this way a compact feature vector with enough discriminative power can be selected.

The rectangular feature is very similar to the Haar wavelet feature. It was first proposed in Viola and Jones face detection [74], and later applied to the pedestrian detection task [75] [47]. The feature is represented as the correlation of the image and simple rectangular templates in different positions and sizes (see Figure 2.2.2 for illustration); these features are then chosen by Adaboost [28] to have the maximum discriminative ability. The rectangular feature can be calculated quickly by a few addition and subtraction operations on the integral image that sums up the brightness of all pixels from one corner.

A differential-based feature is computed from 2D derivatives of an image patch,

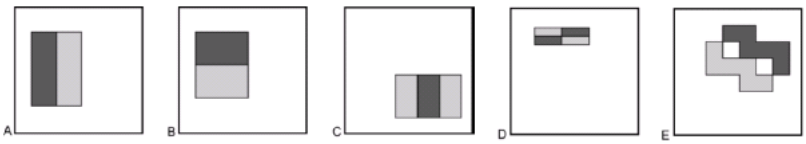


Figure 2.4: Examples of rectangular features used in [75]

which can be obtained by convolving the image patch with a discrete differential operator (e.g. a Sobel operator). The differential-based feature is designed to extract crude shape information of objects from image (since object boundaries always feature strong derivatives) and be invariant to brightness and illumination changes. The most widely-used differential based feature is the edgemap, which is a binary map denoting local maximum of the gradient intensity of the image. It is invariant to brightness and illumination changes. However, many important information that is essential to discriminate humans is also discarded, e.g. intensity and orientation of edges. Note that a threshold that filters out weak edge responses is hard to specify for all situations. Edgemap was used in [33] [29] [32] [31] [30] and [46].

The disadvantage arising from discarding edge intensity or orientation can be partially compensated by attaching intensity and/or orientation information to each edge pixel; this results in a binary edgemap in 3D or 4D space. In [21], edge orientations were introduced into the score function of the Chamfer matching, inconsistent orientations between the edgemap and the template are penalized. Also, features in several consecutive frames are matched against a temporally changing template. Both improvements assured that fewer false positives can be achieved.

The edgelet feature aims to construct a compact differential based feature without thresholding. In this feature the most distinctive edge pixels are learned from examples

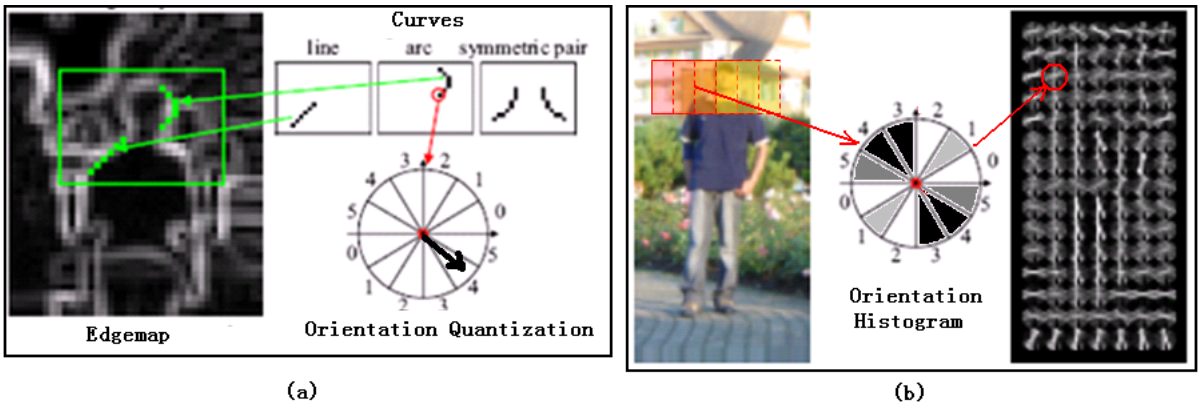


Figure 2.5: Generation of the edgelet feature (a) [80] and the HOG feature (b) [19]

in training stage and are fixed in detection stage. It is constructed by first generating a set of all possible curves that lie on the image (usually some constraints on the shapes of the curves will be imposed, e.g. only lines and arcs), pixel-level information on each curve, including edge intensity and direction of all pixels on it, is considered as an edgelet candidate. All candidates are fed into AdaBoost and only those with the best discriminative ability are selected (see Figure 2.5 for illustration). Only a few candidates that lie in the ROIs will be selected, this idea was used by others[47, 78, 79].

The third type of appearance-based features, the distribution-based features, were proposed at an early stage [27] but were extensively studied only recently in the last 5 years. Most of the state-of-the-art detectors are based on these features and finding more effective distribution-based features is dominating the research work in this field. The distribution-based features are designed to represent the statistical properties (e.g. histogram and covariance) of pixels in small regions. Compared to other types of features they have the following advantages. First, they are the only type of appearance-based features that are robust to image and object deformations, and as a

result invariance to small degree of articulation and viewpoint changes is guaranteed (Local descriptors, or 'bag of features' also possess the same ability, they will be discussed in the next section). Second, the impacts of noise and outlying pixels on the statistical properties are minimized, making them invariant to image degradation and low resolution.

The Histogram of Oriented Gradient (HOG) feature is currently the most important distribution-based feature in human detection. It is constructed by dividing the image into many overlapping square blocks, the size of each block and overlap between two adjacent blocks are usually fixed (e.g. 4×4 or 8×8 with 50% overlap), and quantizing the the gradient orientations of all pixels in each block into nine directions. The HOG feature is the histograms of gradient orientations of all blocks. This feature is first proposed by Dalal et al. and gained its popularity due to its outstanding performance in human detection [81] [20] [82] [83] [88] [68]. In the benchmark of [50], it attained the highest performance compared to other local features. Nevertheless, it has a number of drawbacks. First, it has very large dimensionality, and is costly to handle if all coefficients are used. Second, if the blocks are too small (e.g. when image resolution is low), the histogram will become meaningless and cannot reflect the real image structure. Third, the HOG was primarily used as a robust edge descriptor, its discriminative power mainly comes from the blocks located on the edge of human body. When edges are largely missing or contaminated by cluttered background, the HOG's performance will consequently deteriorate.

For the first problem, many efforts have been made to obtain a more compact subset of HOG representation. In [61], the components of HOG are selected by AdaBoost,

resulting in a classification rule that uses only a small subset of HOG (called 'shapelet' in the literature). A similar framework was used in [16], where an improved AdaBoost select components from an HOG-rectangular joint feature. In [6] [5] and [13] the HOG components in vertical gradient direction was used to detect the symmetry of vertical edges. The second problem can be avoided by using an alternative feature on small patches. For example, in [82] the HOG was replaced by edgemap on small patches. And in [20], the HOG was replaced by histogram of optical flow orientation.

A method of overcoming the third problem was proposed in [76], in this method, the edge representation ability of the HOG feature was combined with the texture representation of local binary pattern (LBP) feature. The LBP feature is another block-histogram based feature constructed by following steps. First, the image patch is resized and bilinear interpolation is used to convert it into a continuous 2-D function. Then this function is converted into LBP map. For each pixel, the signs of its difference with eight pixels around it are measured and quantized into 0 and 1. Together they form a 8-bit number that denotes the corresponding pixel value of LBP map. In the last step, the LBP map is divided many non-overlapping cells and the histograms of pixel values in all cells are concatenated to form the feature vector. To reduce its dimensionality and increase its robustness to cluttering background, the 'uniform pattern' constraint is imposed, that is, for each 8-bit pixel value of the LBP map, if the number of its '0-1' transition is more than two, then this pixel will be categorized as 'nonuniform' and all 'nonuniform' pixels are voted into one bin of the histogram. According to the experiments of [76], a concatenation of HOG and LBP forms the state-of-the-art feature for human detection that has achieved the highest detection

rate so far on the INRIA database.

A covariance feature [71] is another differential-based feature that was initially proposed for image matching and texture classification, but has been adopted in human detection recently and achieved results comparable to the HOG feature [19]. The covariance feature is a collection of 8×8 covariance matrices denoting statistics of pixels in numerous subwindows. Subwindows are selected exhaustively inside the image patch. For each subwindow, the covariance matrix is obtained from a set of vectors, each denoting vertical and horizontal position, first and second derivatives in both directions, and gradient intensity and orientation of each pixel. Like the rectangular feature, extraction of the covariance feature can also be accelerated by the integral image. The initial set of features is highly redundant, only a small number of the most discriminative subwindows will be selected in the following logitboost-based learning step and be used in the detection stage.

Spatio-temporal joint feature

The last type of feature, the Spatio-temporal joint feature, is a combination of appearance and/or shape-based feature and motion information. The motion information can be obtained by comparing two or more frames. For instance, the difference between two frames can infer the movement of human body in the scene and provide cues to identify posture patterns. In [10], the timed motion history image (tMHI) is used to represent the blob motion in short time. It is constructed by iteratively weakening the brightness of the previous tMHI and placing the silhouette of the newest frame over it. Such representation allows easy extraction of motion trajectory by image gradient. In

[39], the recurrent motion image (RMI) was used. Its pixel value denotes the changes of intensity of brightness . This makes RMI invariant to the phase of cyclic motions. In [25] and [20], the optical flow of the human body is used as a pixel-level feature of the detected postures. In [75], the second frame was shifted in four directions and compared to the previous frame. The difference between them was used as part of the feature and to coarsely indicate the direction of movement of each pixel.

2.2.3 Classification

The classification step decides the posture class of a given test image patch based on its feature. Based on the decision rule used a classifier can be categorized into three types: example-based classifier which make the decision by comparing to each of the training example, boundary-based classifer which defines the boundaries between classes and the likelihood-based classifier which models the distribution of each class independently and makes the decision by maximizing a likelihood criterion.

K nearest neighbor (KNN) is the most widely used example-based classifier. It enjoys popularity because of its simplicity and the lack of necessity of training step. In KNN the decision for an unknown posture candidate is made by threshoulding the average distances from the feature vector of the test image to its K nearest training samples. The distance is a measure of dissimilarities between the test image and the training examples and can be in many forms, including Euclidean distance [9], Mahalanobis distance [10] and the number of similar components [24].

For instance, KNN (often 1-NN) is widely used in Chamfer matching which is based on the Chamfer distance from the contour of the training examples to the edgemap of

the test image patch. The Chamfer distance can be efficiently calculated by applying distance transform (DT) on the edgemap and correlating it with the contour templates. The DT converts an edgemap into a greyscale image whose intensity of each pixel is proportional to its distance to the nearest edge point. A major drawback of Chamfer matching is its lack of generalization power. Each training example is indexed independently, so a very large training dataset is often required to densely cover the possible posture patterns which is computationally infeasible. Such an efficiency problem can be overcome by introducing a decision tree-based dynamic search technique. In [33], examples were indexed by a coarse-to-fine hierarchy by distance-based clustering. The approach first search for the nearest examples in coarse scale, then successively narrow down the search range in finer scales for better examples. In each tier of search only a few examples are matched, this greatly reduce the time for finding the nearest examples. This idea was inherited in [29] and combined with a radial basis function (RBF) classifier applied to the positive image patches as a verification step. A Bootstrapping procedure was used in the selection of negative examples, that is, whenever a false positive result is manually detected, it was added to the negative dataset. This detection approach was further expanded in [32] by using alpha-beta tracking on each detected posture blob which significantly increases the speed of the detection.

In the work of Gavrila et al. [30] they aimed to further increase the efficiency of the approach in [33] by rejecting non-posture test images at earlier stages of the hierarchical matching. A decision threshold was set up for each node of the coarse-to-fine hierarchy. If the Chamfer matching on one node is negative, it is rejected by all of its child nodes. This threshold can be obtained by collecting the statistical information

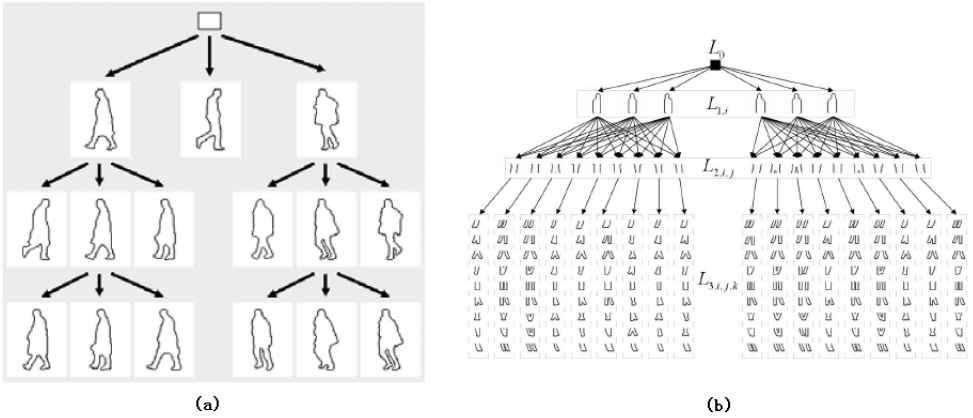


Figure 2.6: Hierarchical structure of the decision tree used in [33] and [46]

of its sub-nodes.

In [46], the idea of hierarchical template matching in [30] was combined with a body part detector, since the variation of torso-arm and legs are more than the variation of head-shoulder contour, it is reasonable to first find the nearest example of head-shoulder contour, then detect the torso and legs. The results of the template matching are fed into a validation algorithm that employs an occlusion map derived from the vertical positions of detected head-shoulders.

Another major drawback of Chamfer matching is that its robustness is severely affected by a cluttered background. Densely distributed edge points generated by a cluttered background greatly increase the possibility of close matches and may result in many false positives. In [21], this effect was mitigated by using chamfer matching along multiple consecutive frames, and only if the matching distance in all frames are less than a threshold, can a positive match is detected. This strategy reduces the possibility of false alarms, which frequently happens in edgemap-based detection from a single frame.

It appears that KNN is the only widely example-based classifier without a need for a generalization from the training examples. The other 2 types of classifiers: boundary-based classifiers and likelihood-based classifiers, require a training step which seeks to find the patterns that characterize the training examples. Learning and classification through patterns have many advantages, for example, the classification can be made much faster, and the results can be less affected by bad training examples.

Support vector machine (SVM) [8] is a typical boundary-based classifier. It finds the separating hyperplane that maximizes the margin width between two classes in the Euclidean space. There are two major improvements on the traditional SVM: soft margin SVM and kernel SVM. The soft margin SVM is designed to overcome the problem caused by outliers in the training set. It sets up a 'slack' parameter to govern the trade-off between the misclassified examples and the boundary width, hence, excluding the influence of the outliers. The kernel SVM aims to classify linear inseparable data by mapping them into a higher-dimensional kernel space, where they become linear separable and can be classified by a linear SVM. The SVM was first used in posture detection by [55] for classification of the Haar wavelet feature, then was further expanded in [57] and [58]. In [57], the posture detection from single frames was temporally filtered by comparing the detected position and predicted position from previous frames so as to reduce false positives. In [58], the idea was generalized to object detection. In [17], the SVM was used on a reduced set of Gabor wavelet coefficients and in [19] and [20] the SVM was applied on HOG and optical flow features.

In almost all cases a kernel SVM outperforms a linear SVM due to its ability to output non-linear decision hyperplane, but a linear SVM has an advantage. The decision

function of a linear SVM is a linear combination of components of the training feature vectors. Therefore, if the feature is local sensitive, the linear SVM will be position-independent. This would allow the occluded part to be automatically identified by negative response of the SVM, this providing a new mechanism for occlusion handling. This idea is the key contribution of the work by Wang et al. [76]. In their work, if the decision score is ambiguous (i.e. the score falls in the SVM classification margin), then the response of the linear SVM to each HOG-LBP block would be extracted independently. The sign of each response will form a binary pixel of a 'likelihood image', then a mean-shift based image clustering algorithm is applied on the 'likelihood image' resulting in potential regions with an overall negative response. The remaining HOG-LBP blocks can be classified by a body part detector.

Adaboost [28] is another widely used boundary-based classifier for multiple classes. Here, the test image patch is first classified by a number of simple but weak classifiers, the results are then merged together by a combination rule learned from the training dataset. Adaboost is essentially the optimizer of the linear combination rule, that is, a weighted sum of all responses of the weak classifiers. Its objective is to minimize the exponential loss of the response of the combination function to the training data. It works by incrementally adding new classifiers to the combination function in a 'greedy' manner: each newly added classifier is chosen and weighted to minimize the loss function of the combination function. Therefore, the Adaboost is particularly suitable for very long features such as the rectangular feature and covariance feature.

The Adaboost was first employed in the Viola and Jones detector [73] for face and object detection, in which weak classifiers were simply binary thresholding of one

component of the rectangular features. This method was directly used in [43] for face and body part detection. However, due to the complexity of the whole human body, this method requires further improvement when employed to posture detection. In [74] and [75], the algorithm was improved by introducing a new visual feature. The rectangular feature was extracted from both brightness and the shifted difference image of two frames. In [61], the rectangular feature was replaced with an HOG feature and the chosen component of the HOG feature was called 'shapelet'. In [81], the Adaboost algorithm was employed to select components of the HOG features and construct a strong classifier. The classifier was first used on single frame, then the detection results along multiple frames were verified mutually to remove false positive and negative detection by constructing a motion estimation among the detection results of consecutive frames.

In [47], the Viola and Jones detector was improved by using edgelet features. The improved detector was applied for both whole body detection and body part detection. The detector first scanned for positives using whole body detector, then for each positive region, body part detectors were used to search for three parts: head-shoulder, torso and legs. The cues from all detectors were then combined together by weighted sum.

In [16], the Viola and Jones detector was improved from two aspects. First, rectangular feature was extracted from HOG map instead of intensity. Second, a new boosting technique called feed-forward Adaboost cascade was used instead of the traditional Adaboost in the training step. Once each new weak classifier was added to the strong classifier, an 'ad-hoc' classifier was constructed for misclassified examples

by a linear SVM, This new SVM was also added to the strong classifier like other weak classifiers. The temporarily constructed SVM represents the best linear weak classifier available for next boosting round, thus provides a 'fail safe' mechanism if all the weak classifiers are not optimal.

The classifier used in [88] can be seen as a tradeoff between the speed of the Adaboost [75] and accuracy of the SVM [19]. It divides the HOG feature of the patch into subwindows of different size, and trains an Adaboost cascade of SVM classifiers on the HOGs of subwindows. Finally the strong classifier will only use a small subset of all HOG components, compared to the original HOG+SVM framework [19] this effectively increases the speed of classification without a significant loss of performance.

LogitBoost [72] is similar to Adaboost, but a logistic loss function is used instead of exponential loss function. In this framework, each covariance matrix is converted into a point on a Riemannian manifold, and weak classifiers are linear classifiers trained from examples' mapping onto different tangent spaces of the Riemannian manifold. The tangent spaces are adaptively chosen so each new tangent point will be closer to the points that are misclassified, which means linear classifier on the new tangent space will have the best discriminative power around these misclassified points.

In the likelihood-based classifier, a probabilistic model is often defined for each class in the feature space so each patch of the sliding window is classified into one of the classes which gives the highest likelihood. Compared to the boundary-based classifier, the likelihood-based classifier can usually describe more complex probabilistic pattern thanks to the highly developed theory of probabilistic modeling and inference. For our multiple posture cases, it also provides superior scalability, the training of each posture

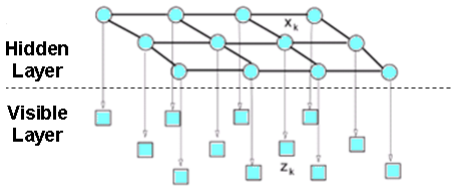


Figure 2.7: The Graph indicating causality of a 2D markov random field

is independent so when a new posture is added to the task, only the corresponding model needs to be retrained.

In [82], a two-layer Markov random field (MRF) was used to encode the 2D variation pattern of HOG feature, it consists of two layers: the status of nodes in the visible layer represent the HOG vector of overlapping grids and is assumed to be generated from the nodes in the hidden layer (see Figure 2.7 for illustration), the status of nodes in the hidden layer represents a low-resolution binary map denoting the contour of the human body. By imposing the assumption that the joint probability of the hidden layer is a Boltzmann distribution, it is possible to train the parameters of two MRF with enough examples, encoding the probabilistic patterns of posture's HOG and non-posture's HOG respectively. They can be later used for maximum-likelihood classification for an unknown image patch. This paper also proposed the idea of using probabilistic variational analysis to accelerate this process since the training and probabilistic analysis on a generative model with many loops is very slow.

Occlusions between postures are commonly observed in crowded scenes, and this poses hard problems for posture detection because some postures may be significantly occluded (up to 70%) and completely lose their structural information. In this case, it is

possible to use the position of the occluding object (which is another posture) as a priori information. An attempt was made in [23], where the probabilistic model was defined for the whole scene rather than for each individual patch. In this model, each person was represented by an ellipsoid vertically separated into 3 blobs, denoting head, torso and leg respectively, the colors distribution of each blob was determined at initialization by kernel density estimation. Each new model was initialized when the person made its first appearance in the video. For a subsequent frame, a scene hypothesis consisting of several overlapping ellipses is constructed and iteratively optimized through the EM algorithm, by maximizing its likelihood given by the product of the likelihood of all pixels in the ellipses. Finally the best explanatory hypothesis is obtained that gives the position of all human in the scene.

2.2.4 Merging

The merging of multiple spatially neighboring positive responses is a simple but necessary step in most detection by patterns approaches. Typically the merging can be achieved by clustering adjacent positives in the 3D space of x-position/y-position/scale and denoting each cluster with its local maximum. The merging step merges the spatially overlapped positive responses to give the final detection results.

2.2.5 Summary

Detection by patterns is the most popular framework for the problem of posture detection (and arguably the most popular one for any object detection problem). It aims to detect the human posture directly as an entire object, in which all types of features can

be used. The framework is intuitive, robust, and can be easily combined with other cues (e.g. results from an infrared camera or sound sensor). The major drawback of this framework is that the feature selection and classification steps will be iterated for many times at different scales and different positions. So the computational cost is usually high, especially when it is used in detection of multiple postures.

2.3 Detection by local descriptors

Given an image with an unknown number of postures of interest presented, a typical detection by local descriptors approach consists of the following two steps: First, local descriptors are found and detected using a simple detector. Then, the postures are inferred from the local descriptors. This framework is illustrated in Figure 2.8.

Local descriptors are special features that aim to represent the image as a structural combination of visual features of a number of local interest regions. The local descriptors are gaining popularity because according to the intuition of human perception, structural information plays a more important role than appearance information in identifying an object holistically. For example, a human face is characterized by two eyes, one mouth and one nose in their corresponding positions; their individual appearances and shapes, however, are of less importance. In other words, a local descriptor captures only the detailed appearance in local scale and structural information in global scale. They have been proved to be successful in numerous computer vision problems. They are also distinctive, robust to occlusion, and do not require segmentation. Recent work has concentrated on making these descriptors invariant to image transformations.

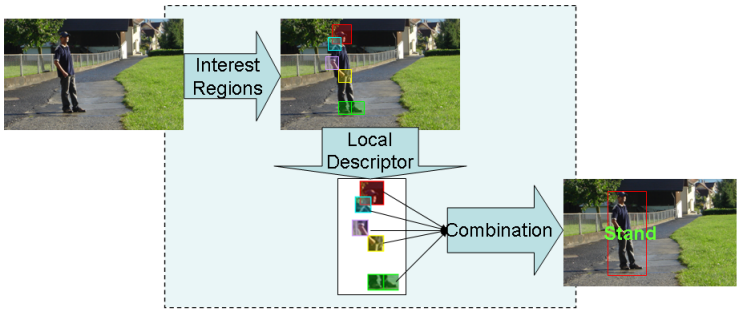


Figure 2.8: Detection by local descriptors CITE

Considering the difficulty of modeling appearance variations of postures directly, detecting rigid body parts may be easier. More importantly, the occlusion of body parts will not cause the total loss of discriminative power of the posture feature, which may possibly happen in the detection by patterns framework. Even when some parts are missing, we can still utilize information from other parts to detect the partially occluded postures.

2.3.1 Local descriptors

Similar to the feature extraction step in the detection by patterns framework, two most important criteria for constructing useful local descriptors are being discriminative and invariant. To maximize the discriminativeness, the local interest regions are specifically chosen to meet two conditions: (a) they must be distinctive enough as parts of the human body (so they can be easily identified), and (b) their position and appearance should be more crucial to the posture configuration than the other regions. Based on these conditions there are two ways to define the local interest regions for our purpose: either defined manually to cover the most crucial body parts, like face, arms, hands,

legs and feet, or learned in an unsupervised manner from the interest points (e.g. edge points, ridge points, corners) from a sufficient number of training examples.

The first way is more demanding for the training stage, because it is hard to tell which part of the human body is more discriminative and crucial to the a specific posture configuration. Manual marking of these regions is tedious and challenging. However, manual definition is still possible if the number of representative local regions is small. In [52], [48], [68], [78], [79], [67] and [81], the regions are fixed rectangles denoting bounding box of head-shoulder, torso and legs. Their size and relative positions with regard to the posture's bounding box were pre-determined and the variations of these regions were learned in the same way as learning the full body. In [53], these regions are the contours of limbs, extracted manually from the image examples. There are also a few local descriptors that do not require a training stage, but assume homogeneity of color or simple intensity patterns [59] [69] of the limb areas.

The second way aims to index all interest points inside the contour of postures using body parts based codebooks without human intervention. This can be done by a clustering algorithm. In [44], [66] and [65], the codebook is learned by applying agglomerative clustering on all detected scale invariant derivative of Gaussian (DoG) interest points of the training examples. In [60], the interest points are equally sampled from the contour of the training examples and the codebook is learned by K-means clustering on local shape context descriptors.

After learning the descriptors, the body parts can be detected by scanning the image at different position and different scales followed by a classification step. The scanning step searches through numerous potential interest points or regions where body parts

may exist, and the classification step convert these points/regions into local descriptors and classify them into different types of body parts. A typical example is given in [52], where SVM is used for both body part detection and combination rule of the decision of body part detectors: The response or each body part detector is regarded as a component of a four dimensional feature vector, which can be classified by another SVM.

The scanning can be either exhaustive in the entire image [48], [68] [78], [79] and [80] [59] [67] or selectively guided by the ROI extraction using the techniques described previously, for instance, background subtraction was used in [60]. Other characteristics of body parts are also helpful to limit the scanning. In [44], [66] and [65], the body parts candidates are chosen from all scale invariant DoG interest points. In [53], the homogeneity of color and intensity inside the skin and clothes regions (e.g. face, hand, upper torso and limbs) were used to select body parts candidates. A normalized cut-based image segmentation was first performed in two scales, the segment of a coarser scale (segmentation map) is used as candidates of half-limbs and torsos, and the segment of a finer scale ('superpixels') was used to extend the detection result of half-limbs to full limbs.

To classify each scanning window into body parts, the classification techniques reviewed previously can be employed. For instance, example-based or template matching was used in the part detection step of implicit shape model (ISM) [66] [65] and [44]. In the ISM, the body parts are located by the key point selection algorithm in the SIFT feature. The region around these points are then converted into distance transform

(DT) map and are classified according to its correlation with various body part examples from a trained codebook (this is equivalent to chamfer matching). In [60], the example based classifier was employed to the shape context coefficients of contours of the body parts.

In [67], examples were clustered into nine classes. For each class a soft-margin SVM was trained on 2×2 HOG of 13 different interest regions (in total 117 SVM are trained). In [52], SVM is trained on haar wavelet coefficients of five interest regions. In [48], the body part classifier is the same as [75], where AdaBoost was applied on rectangular feature for detection of face, torso, legs and hand, and part of the detection results were validated by color cue. In [78] [79] and [80], the body part classifier is similar to [47], where a multi-class Adaboost was applied on the likelihood of 14-dimensional edgelet features in different positions. In [68], the body part classifier is the same as [88], where AdaBoost was applied on HOGs in blocks of different sizes.

The likelihood-based classification was adopted in [59], [69] and [53]. In [69], the likelihood function consists of three terms, denoting edge response, ridge response and optical flow consistency respectively. In [59], the likelihood was measured by the homogeneity of pixels inside and outside the region of body part. In [53], the likelihood function consisted of four terms, denoting contour integrity, shape correspondence, intensity pattern and focus cue (since the background is usually out of focus) respectively.

2.3.2 Combination of local cues

Basically there are two approaches to infer a salient posture from scattered cues of body parts: either through reasoning or learning. It is widely known that all the body

parts form a flexible kinematic model that follows a head-torso-upper limb-lower limb structure: each two parts are linked by their mutual joints near their end. In addition, if the posture can be determined then the location of each part is relatively fixed, and can be learned from examples. And sometimes, after the salient postures have been inferred from local cues, they should be validated by their holistic appearance to ensure they are actual human bodies, instead of a false positive caused by combinations of irrelevant interest points. This is important because body parts are less discriminative than whole body, as limbs and trunk sometimes look like pillars, so they will have higher false positive rate and must be suppressed thereafter. Obviously, the combination through reasoning can accommodate more variations of the postures. However, since this approach requires a 2D kinematic model to be inferred from body parts, and it poses a few restrictions on kinematic configurations (many of which are counter-anatomic and counter-dynamic) and increases the difficulty of combination and, hence, the chance of false positives.

The combination through reasoning was used in [53] and [59]. In [53], a hierarchical combination rule was set up. The orientation of the detected torso was first determined by detecting the most plausible head around the torso region using the same likelihood-based detector as in limbs and torso detection. Then, adjacent candidates of three half-limbs and torsos were linked together by imposing global constraints, which limited the relative width, length, adjacency and clothes color homogeneity of the four combined body parts. The best body configuration was thus chosen by dynamic programming. In [59], the links between torso and arms were constrained by

two factors: the distance between corresponding joints of torso and arms, and the similarity of the two symmetric body pairs. In this way, two likelihood functions were set up to penalize its plausibility when this distance exceed a certain value and when the divergence between the two symmetric body parts was too large. These likelihoods were multiplied with the likelihood of individual body parts to estimate the global plausibility of the candidates.

In the learning approach, the postures have to be learned from the training examples instead using prior knowledge of the composition of the human body. The information on the relative positions of the body parts plays a critical role in identifying the postures. There are two ways to use the relative position information. One is to infer the position of body parts from posture candidates, and validate the detector's response to these assumed body parts, then combine them by a multivariate combination function that gives the joint likelihood of the whole body from the detection result of body parts. For instance, this approach was adopted in [52] and [67], where the body parts were defined using respectively five and nine fixed subwindows of the posture's sliding window, and the combination function was trained by SVM and AdaBoost respectively. A more complicated strategy was employed in [48], in which a likelihood function was defined as the multiplication of body parts likelihood obtained in body part detection, and a combination likelihood learned from training. The random sample consensus (RANSAC) technique and a loose spatial constraint is used to iteratively search the optimal combination of corresponding body parts to maximize the likelihood of the resultant combination.

When multiple persons appear in the scene, mutual occlusions will become a serious

problem. Even though the local descriptors based framework itself provides some occlusion tolerance, it fails to handle situations when the occlusion is too large. In this case one probably needs to build up the combination criterion of the whole scene instead of an individual person. Hence, occluded body parts could not decrease the likelihood of a candidate being covered by another person. In [68], this combination function was defined as logical reasoning rules, consisting of a bilattice-based reasoning network that synthesizes multiple binary event detection results (e.g. presence and absence of body parts and occlusions). The network was defined semi-manually, in other words, some of the 'obvious' parameters (e.g. the influence of occlusions and relations between the position and scale of the human body determined through camera calibration) were manually set up while the rest of them was learned from training data. In [78], [79] and [81], the heads were first detected. Then according to their vertical position in the image, their distances to the camera were determined and lead to an occlusion map for each person. Hence, cues from body parts were dynamically chosen according to this map. If one body part was missing without occlusion, it would cause the rejection of the whole person. To prevent false positives from the noise in the occlusion map, they were removed in the subsequent step using a greedy algorithm.

Another way of using the relative position information is to infer the position of the whole body from individually detected body parts regardless of their relevance. This idea was proposed in implicit shape model (ISM) [44] [60] [66] [65]. The original ISM was used in [44], where body part detection was accumulated through spacial voting on the relative position of the center of mass of the human bodies from each detected body part (this relative position from body part to body center can be learned in training).

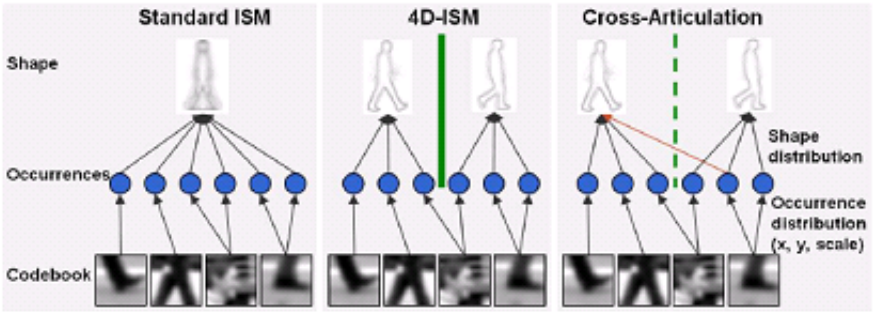


Figure 2.9: Comparison between the combination process of the implicit shape model in [44],[65] and [66]

After the density of the voting result was obtained, the local maxima of the voting density, denoting a possible center of mass of the human body, can be extracted by a mean-shift optimization as a preliminary detection result. Then the 'sources of voting' were validated by eliminating the voting from irrelevant regions of the detected human body (e.g. background regions). The pruned voting was then recalculated, leading to a new density map in which local maxima represent a more accurate detection result. The accuracy of the detection was further increased by validating the detection result by chamfer matching of the whole body's edgemap. The detection was performed for several times in different scales, allowing postures at different sizes to be detected.

In [44], the appearance and relative position of each body part is fixed, this is not true if multiple postures or postures in different directions are being detected. Such assumptions may cause irrelevant body parts being assembled together, leading to a high rate of false positives. To overcome this, an improvement of ISM was proposed in [66], in which the appearance of the body parts and combination rules for different types of postures from different viewpoints were independently trained. In the detection stage, the voting from detected body parts was only accumulated if they belong to the

same posture type and viewpoint. This is equivalent to expanding the voting space from 3D (x-coordinate, y-coordinate, scale) to 4D (plus the type of the posture and viewing aspect). However, this improvement seems to jump from one extreme to another. It assumes that each class of body part's appearance can only belong to one posture type and viewpoint and disregards the fact that some body parts may look similar in two or more posture configurations. This problem has been dealt with in [65] by introducing cross-articulation voting in which a vote from a body part was no longer cast on one point in the 4D space, instead, it was cast on multiple posture configurations based on the probability of the body parts appearing in these posture configurations. An illustration of different combinations of rules in [44][66][65] are shown in Figure 2.9.

2.3.3 Summary

Recent local descriptors based approaches has demonstrated promising results. But it is still arguable whether local descriptors based detection would consistently outperform the pattern based one if the same feature and classifier are used to detect body parts and whole body respectively. The framework itself is complex, but its speed is generally not a drawback. This may be because the patterns of local parts are not complex and can be detected by faster features and classifiers. The framework is specifically designed to have high robustness in partial occlusion conditions, and in fact many works have indeed shown such an ability [66] [44] [68]. However, the relationship between different body parts has not been fully utilized, despite the advantages and disadvantages. On one hand, the button-up representation allows new instance of postures to be detected even if its illumination condition, articulation and overall texture are different from

the training examples. Therefore, a small training dataset could train a detector with enough discriminative power and flexibility. On the other hand, discarding of global appearance information of posture will cause a great loss of performance, and false alarms may be generated due to the combination of irrelevant local descriptors. Though this can be partially alleviated by applying validation using global cues, the ability of using a very small number of training samples will be lost at the same time. Another disadvantage of this framework is its sensitivity to image degradations (blur or noises) because local features are usually damaged in such cases.

2.4 Performance Evaluation

This section presents the commonly used datasets and techniques to evaluate the performance of a posture detection method.

2.4.1 Datasets

Many different datasets were constructed to train the detectors and test their performances under different conditions and assumptions with various categories of postures. These datasets were usually captured under controlled environments or modified to mimic practical conditions, such as indoor/outdoor surveillance, the use of a hand-held camera and lab-condition motion capture. Ground truth was also manually labelled. Datasets are crucial for performance evaluation and comparison. Comparison between two detectors is only meaningful if they are trained and tested on the same dataset.

It should be pointed out that most datasets were originally designed and captured

for action recognition and human detection rather than for posture detection. However, they can be also used for posture detection if the postures are labelled properly. Table 2.1 lists the datasets that may be used for evaluating posture detection, where MP refers to multiple person inside one image, and O refers to the grade of occlusion (1-no occlusion, 2-self occlusion, 3-occluded by objects, 4-occluded by other persons).

Name	data	MP	Classes	Viewpoint	Scene	O.	C.	Used by
INRIA	902 images	yes	walking/standing	Horizontal	Outdoor	4	3	[81] [16] [19] [25] [47] [61]
MIT	924 images	no	walking	Front/Back	Outdoor	2	2	[81] [19]
USC-A	313 images	yes	walking/standing	Front/Back	Outdoor	3	2	
USC-B	271 images	yes	pedestrian	Upper Front/Back	Indoor	3	2	[68] [46]
USC-C	232 images	yes	walking/standing	Horizontal	Both	3	2	
PETS2001	10 videos	yes	pedestrian	Upper	Outdoor	4	2	[39]
Caviar-INRIA	28 videos	yes	6 actions	Upper	Indoor	3	2	[46] [78] [79] [80]
Caviar-Lisbon	26 videos	yes	pedestrian	Upper	Indoor	3	2	
MunichAirport	1779 frames	yes	pedestrian	Upper	Indoor	3	2	[46]
Weizmann	90 videos	no	10 actions	Side	Outdoor	2	2	
CMU Mobo	600 videos	no	4 walking types	8 Surrounding	Treadmill	2	2	
UCF sport	197 videos	no	10 sport actions	Horizontal	Both	2	2	
MuHAVi	119 videos	no	7 actions	Upper	Indoor	2	2	

Table 2.1: The datasets that may be used for evaluating posture detection algorithms.

2.4.2 Evaluation criteria

The performance of any posture detection algorithm should be evaluated in terms of detection accuracy, time and space complexity, and an ability to tolerate variations in camera and illumination conditions. In this section, we will present the commonly used evaluation criteria including detection rate (DR), false positive per window (FPPW), receiver operating characteristic curve (ROC), precision-recall (PR) and decision error trade-off (DET)

The DR and FPPW are two basic benchmark criteria for a posture detector. The DR refers to the ratio of successfully detected postures with respect to all postures and the FPPW is the ratio of false positives to the number of iterations of a sliding window.

$$DR = \frac{\text{number of true positive}}{\text{number of real postures}} \tag{2.1}$$

$$FPPW = \frac{\text{number of false positive}}{\text{number of iterations of sliding window}} \tag{2.2}$$

The receiver operating characteristic (ROC) curve is derived from the detection rate and false positives. For any detector, the detection rate can always be adjusted by tuning a threshold that governs the 'sensitivity' of the classification step or combination step as described above, that is, the tradeoff between detection rate and false alarm rate. For example, in an SVM or boosting-based classifier the threshold is set on the value of the decision function and in an ISM the threshold is set on the accumulated density of the spatial voting cast. Generally the threshold is set on the point where the detector has the highest overall rate, but in practice the importance of the two

rates might be different. If numerous patches are generated from an image which contains a small number of postures, then a reduction of FPPW should be emphasized to prevent a large number of false alarms. However, if the detector is combined with a motion-based validation step or tracking step, the FPPW will become less important because most of the false alarms can be filtered thereafter. Therefore, the ROC curve is usually used to give a comprehensive evaluation of the overall performance. In a ROC curve, the x-coordinate denotes the false positive rate and the y-coordinate denotes the detection rate. The ROC curve is an intuitive illustration on how the detector performs at different thresholds. Another criterion derived from a ROC curve is the area of region under the curve, which coarsely denotes the mean detection rate under different FPPW values.

A Precision-recall (PR) curve is similar to the ROC curve but slightly different in its definition. The PR curve is also a curve representing a detector's performance with respect to different threshold settings. However, the x-axis and y-axis now are used to represent recall and precision respectively. These two measurements are defined as follows:

$$recall = \frac{\text{number of true positive}}{\text{number of real postures}} \tag{2.3}$$

$$precision = \frac{\text{number of true positive}}{\text{number of all positive alarms}} \tag{2.4}$$

The PR curve indicates the performance purely based on detector's positive response, so it covers the situations where the ROC curve fails to give an objective

evaluation because no sliding windows are employed. For detectors based on local descriptors or detectors heavily relying on the accuracy of segmentation, the PR curve is a better choice. The general characteristic of the PR curve is featured by an equal error rate (EER), referring to the $recall/precision$ rate when $recall = precision$.

The decision error trade-off (DET) curve is a third benchmark criterion that has a similar functionality to the ROC curve, its only difference is that the detection rate is replaced by the miss rate:

$$MR = 1 - precision \tag{2.5}$$

It should be noted that the benchmark criteria are usually calculated before the merging step.

In addition, it is also important to judge whether a positive response constitutes a successful detection. If the detection result is evaluated manually, then a single response will be marked as successful if its position lies just in the bounding box of a true posture. If multiple postures are defined, the class of the posture should also be specified correctly. However, the ground truth bounding boxes of the postures are manually labelled. A certain level of deviation from the position of labelled ground truth should also be tolerated. On the other hand, sometimes (e.g. in object retrieval) this level of deviation should also be measured as part of the evaluation system because it reflects the ability of the detector to precisely identify the position of the postures. In [44], three criteria were defined: relative distance, cover and overlap. The relative distance is defined as the distance between the geometric center of the detected region and its nearest ground truth in relation to the length and width of the bounding box.

Detection Rate	[10] [56] [85] [87] [86] [9] [39] [12] [33] [29] [23] [55] [57] [58] [48] [60] [44] [68] [78] [79] [80]
ROC	[46] [32] [30] [16] [20] [47] [52] [55] [57] [58] [67] [75] [82] [48] [60] [68] [78] [80]
PR	[65] [66] [44]
DET	[19] [20]
complexity	[23] [55] [68]

Table 2.2: Summary of the key literature

The cover and overlap measure how much of a bounding box of the ground truth is covered by that of the positive detection and vice versa. Each positive detection is considered to be true only when all of these values are less than a threshold.

In terms of efficiency, the commonly used measurements are the execution time of the algorithm and the big-O notation of the algorithm. The execution time depends on the language (e.g. C/C++ or Matlab) used to code the algorithm and machine on which the program runs. The big-O notation indicates how the computational cost will be increased when the problem size increases.

Table 2.2 shows evaluation criteria adopted in some key literature.

2.5 Major Challenges

This section discusses the key challenges that influence a detector's performance, including occlusion, articulation, clothing, illumination, image quality and cluttered background.

2.5.1 Occlusion

Occlusion is probably the biggest challenge in the detection of any object. The occluded part of a body can undermine the structural integrity of the human body and cause the

total loss of its global features. Recently many approaches were proposed to overcome this problem. In general, occlusions can be categorized into the following four cases as shown in Figure 2.10 and different frameworks should be adopted in order to address each case:

1. No occlusion: Every articulated part of the human body is visible and not occluded. The assumption of no occlusion can only be enforced in human-machine interactive applications where the user of the application is facing the camera. If multiple calibrated cameras arranged in different viewpoints are available, their information can mutually compensate the occluded parts. Since the articulation of the body is fully visible, it allows kinematic information to be used unambiguously for detection.

2. Self-Occlusion: A part of the human body is obstructed by another part of the same person. Common self-occlusion cases can be observed in the profiles of a human body, where a limb can be obstructed by the torso or other limbs. Since most of the reviewed literature learned appearance information directly rather than kinematic information, the self-occlusion is usually relatively easy to address.

3. Occluded by other objects: A part or all of the human body is obstructed by other objects (e.g. when a pedestrian walks behind a tree). This condition usually appears in outdoor scenes that contain a lot of foreground obstacles. Detection of postures occluded by objects relies on extensive exploitation of local features

Figure 2.10: The four cases of occlusion

and tracking techniques. Local features will only change partially with the occlusion but will not change completely as global features. Tracking techniques like Kalman filters or particle filters are able to predict the continuous movement of the human body by using motion information of adjacent frames, regardless of whether the body is fully occluded. However tracking techniques are not within the scope of this thesis and, hence, will not be discussed.

4. Occluded by other person: Multiple persons are in the scene and some of them are partially occluded by other persons. In a crowded scene such occlusion is almost inevitable. This condition, however, is easier than previous cases because as long as the foremost posture can be detected, an occlusion model over the whole scene can be incrementally constructed and used to predict which part of the occluded person is missing. This technique has been explored in [23], [46], [68], [78], [79], and [81].

2.5.2 Articulation

Articulation refers to changes of the torso and limb's configuration in one posture class, caused by continuous movements of limbs, or by the differences of individuals and situations (also known as gait differences), or by continuous movements of camera's position. Articulation does not obviously affect detection of local descriptors, but unfortunately the majority of the literature aims to detect the whole posture. Many distribution-based features and likelihood-based classifiers are designed to tolerate small range of articulation. This problem may be dealt with by a sufficient number of examples so that the articulation can be encoded by piecewise representation of the examples.

2.5.3 Clothing

Different clothing will greatly affect both the appearance and contour of the human postures. Unlike face detection, they should not be treated as occluding objects because they usually cover more than 50% of the body, and their appearances contain important structural information. Due to these reasons, clothing is always learned as part of the posture patterns in the training stage, this further increases the difficulty of learning because the variations of clothing are far more unpredictable than the variations of articulation. The general conditions of clothing can be categorized into four cases as shown in Figure 3.1:

1. Tight clothes: This only appears in lab condition and athletic scenes. Tight clothes are easy to handle because they only merely change the contour and brightness patterns of the torso and limbs. In particular, the tight clothes can be in a specified color so the segmentation of human body becomes easier.

2. Monochrome casual clothing: This is the most common condition that will be encountered in most applications. These clothing will slightly change the color and contour of human body but in general their variations are small, so they do not pose a significant problem to the learning process as long as we can guarantee the same kind of clothing appear in both the training and test dataset.

3. Clothing with complex textures and patterns: textures and patterns will provide unexpected interest points or edges with high contrast, which will interfere with some of the local visual features. However, compared to monochrome clothing the global features and histogram-based features will usually not be affected because textures and patterns do not contribute significantly to low-frequency information.

4. Formal clothing and costumes: Formal clothing like a robe, gown or scarf and costumes such as a big hat and unusual clothing will severely affect both the appearance and contour of human body. Their variations are numerous but they do not appear frequently. They are most challenging to the detectors.

2.5.4 Illumination

Environments that are either too dark or too bright will lower the contrast of images or videos and increase the difficulty of detection. However the greatest challenge comes from the variations of appearance caused by uneven lighting and shadows. Uneven lighting will change the overall distribution of brightness patterns and will negatively affect the performance of the detector if it is trained in one lighting condition and tested

Figure 2.11: The four cases of clothing variations

on others. This problem can be partially solved by using illumination-invariant features like homomorphic map, edgemap or HOG. Shadows cast on human body (including self-cast shadows and shadows of other objects) will cause unexpected strong edges and large dark regions that may interfere with the contour of human body. So far, there is no good solution for this problem. Illumination-invariant features are not effective in removing local strong edges, and any attempt to predict cast shadows based on modeling of projection of light and obstruction of objects will be time consuming. Fortunately, obvious shadows are not common in daylight or indoor scenes. Shadows cast on the ground by the human body will also interfere with the function of some motion-based segmentation techniques, because a person and attached shadow will be extracted from the background as one moving object. To help overcome this, a shadow removal step was applied in [39] and [86], where shadows were removed using their shape and color cues.

2.5.5 Image degradation

Image degradation refers to loss of visual information caused by low resolution, badly calibrated camera and noise. Usually local, detailed, high frequency information will be severely damaged, but global features are less affected. Therefore human are still able to identify objects from a degraded image even when they are unable to clearly identify the details. We conjecture that the detection by patterns framework has a better performance in tolerating image degradation because it uses more global features. But this claim has not yet been verified.

2.5.6 Cluttered background

A cluttered background may fit the learned patterns of the postures but may not actually correspond to one of them. The false alarm rate is arguably the most important evaluation criterion of a detector's discriminating power and is the the main reason why oversimplified features should not be used. For example, detection based on pattern matching of edgemap will frequently generate false alarms in a cluttered background. So some approaches [21] [29] [32] combine the edgemap with other cues (edge intensity, direction, etc.) to achieve a better discriminating power.

Negative examples are sometimes required in the training stage of some classifiers like SVM and Adaboost. They are used as posture examples to draw a line between true postures and false alarms. Negative examples are generated by randomly cropping landscape pictures. They must be selected from as many sources as possible and in a large quantity because they represent the rest of the world. The number of required negative examples varies for different features. In [19], this number is almost ten times

the number of positive examples. A large number of negative examples will sometimes lower the algorithm's computational efficiency, but in many situations it improves its performance.

2.6 Summary and Discussion

As in many other detection problems, the presented challenges are either caused by in-class variations that are too big or cross-class variations that are too small, and solutions to these problems rely on both statistical learning and heuristic reasoning. In our case, the challenges of articulation and clothing are particularly important, because they are unique in posture detection.

There are enormous possibilities of combining different features and classification techniques to create a novel posture detector, and indeed many of these techniques have never been used in this field. However, considering the uniqueness of the posture detection problem, we generally do not expect it is effective to simply adopt existing features and classification techniques. Instead, we believe that the solution should be to design new features and classification techniques specifically to overcome these two challenges.

Chapter 3

Kernel PCA for open-set classification

This chapter presents an open-set classification framework which is built upon manifold learning using Kernel PCA (KPCA). In particular, it intorduces an improved KPCA learning algorithm that avoids the problem of numerical instability that may exist for high dimensional data. The open-set classification is achieved by measuring the reconstruction error. To overcome the problem of high computational cost associated with conventional KPCA, we propose in this chapter a new approximatioin algorithm that aims to find a reduced KPCA to approximate the kernel mapping. The algorithm works by greedily choosing a subset of the training samples that minimizes the mean square error of the kernel mapping between the original KPCA and the reduced KPCA. Experimental results on both real and simulated data and comparsions to existing methods have verified the effectiveness and efficiency of the classification framework and the approximation algorithm.

3.1 Introduction

In this thesis, we are particularly interested in the multi-class posture detection problem, that is, detecting and identifying the postures that have been learned from images

or videos. This problem is different from the problem of human/pedestrian detection in three apspects. First, unlike the human detection, the detection can no longer be formulated as a binary classification problem, instead, it is a multi-class and open-set classification problem in which one of the classes corresponds to unknown postures or backgrounds. Second, we assume that a limited or small amount of training samples are available for each known postures due to the difficulty and tediousness of acquiring the training samples. Third, it is assumed that there is no negative training samples representing postures that are not in the interest set or background. In human detection negative examples can be easily cropped from landscape pictures containing no human, but in our case there won't be any negative examples for the uninterested postures.

Under such conditions, we believe that one of the key issues is the effective learning and representation of the posture patterns from a limited amount of training data. A traditional classifier like SVM or Boosting is not applicable because they require negative examples for references. On the other hand, a generative statistical model may be used to model each class and make decision by probabilistic inferences to achieve detection. However, statistical models often rely on certain assumptions which may not necessarily be satisfied. For example, the implicit shape model (ISM) makes the inference from local descriptors under the assumption that the appearance of these descriptors is independent, and a Hidden Markov Model (HMM) is used to represent temporal signals under the assumption of Markovian property.

A recent study on the problem of estimating three dimensional (3D) human poses from images or image sequences has provided us with greater knowledge on posture

representation. The essence of the pose estimation is to seek an effective representation of the pose appearance in the images and the 3D joint configuration of the body that is most likely to generate the appearance. An effective representation of the appearance for a given 3D configuration would reduce the ambiguity of estimation. In [34][2], the appearance is approximated by a cluster represented as a Gaussian mixture model (GMM) as illustrated in Figure 3.1, and in [11][1][22], it is represented by a non-linear smooth manifold. The intrinsic relations between the posture configuration and the appearance are obtained through a regressor. According to the results provided by these literatures, manifold representation usually has a better performance than cluster representation. This can be explained by the fact that the body articulation that causes the major in-class (within-posture) variations is concentrated on a few joint angles and these variations can be effectively enclosed in a lower-dimensional manifold. In contrast, the piecewise GMM cluster representation will become too sparse and hard to be accurately estimated when the number of examples is too small or the dimensionality of the feature space is too high (the curse of dimensionality). In addition, the ability of a manifold representation in dealing with cluttered background has been further demonstrated by the work [3] on estimating upper-body poses by combining the regressor with HOG feature [19] and a non-negative matrix factorization (NMF) based background suppression scheme.

Motivated by the success of manifold learning in posture estimation we adopt the idea and extend it to solve the proposed posture detection problem. Specifically, we adopt kernel PCA as the manifold learning tool as analysed in the next section.

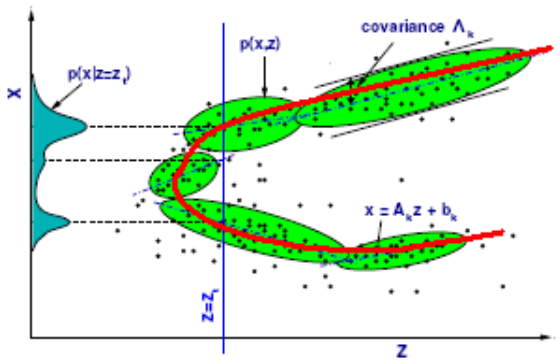


Figure 3.1: Manifold representation (red solid curve) versus cluster representation (green ellipses) of a scattering dataset, this picture illustrates the simplicity and fidelity of the manifold representation comparing to its opponent.

3.2 Overview of manifold learning

Manifold learning (also known as manifold embedding or dimensionality reduction) refers to the problem of finding a lower-dimensional submanifold of a data space, which can best enclose a set of given data. High-dimensional data are not convenient for either visualization or processing (due to the curse of dimensionality). If they can be approximated by new data that lies on the embedded submanifold, they can be represented by shorter vectors. In adidtion, manifold learning can also reveal the latent statistical structure that is hiden beneath the high-dimensionality of the data. The manifold span that encloses the data variation can be seen as a generalization of the examples. Unlike a generative model, this generalization relies on no specific assumption.

A numerical solution to the manifold learning problem relies on how we limit the smoothness of the manifold. If the manifold is close to linear (a hyperplane), then finding its tangent directions that may indicate the major variations is equivalent to

a matrix factorization process, and many tools like PCA, LPP, ICA and NMF can be used. These tools, however, can only be applied when the variations are small, otherwise the linearity assumption cannot be fulfilled. An alternative way to overcome this problem is to adopt a piecewise manifold learning, i.e. approximating the manifold by mixtures of numerous small linear pieces. For example, in [2] and [34] a Gaussian mixture model (GMM) is learned from examples as a representation of the manifold. In [38] examples are unsupervisedly clustered into small groups and each of them is learned by a local PCA. There are two problems in these approaches. First the piecewise representation is not very accurate. Second, it is too complex and requires too many parameters and iterative steps to tune its performance. Another type of manifold learning technique relies on explicit parameterization of the manifold. For example, principal curves [15] aim to iteratively update a curve so it passes through the middle of nearby examples. A nonlinear autoencoder [40] creates a multi-layer neural network to approximate a nonlinear mapping from a high-dimension space to a low-dimensional space and optimizes its parameters by reducing the reconstruction error of the training set. Again these techniques involve time-consuming iterative optimization. Due to these concerns we chose Kernel PCA as the manifold learning tool.

3.3 Kernel PCA

The kernel PCA is a manifold learning technique proposed first in 1998 and has quickly become one of the most popular tools. Kernel PCA (KPCA) is a non-linear dimensionality reduction technique that can be regarded as a kernel expansion of the conventional linear PCA [49]. Let $X = \{x_1, x_2...x_n\}$ be a set of n samples, each sample

$x_i, i = 1, 2, \cdots, n$ being a D dimensional feature vector. KPCA maps the samples into a kernel feature space using the mapping $\phi(x)$ and then performs PCA in the feature space. It is based on the fact that a vector mapping $\phi(x)$ can always be found if its inner-product kernel $k(x, y) =< \phi(x), \phi(y) >$ is defined and positive semi-definitive [62]. Given the n samples, a kernel PCA model can be trained in two steps [63]:

1. Define the kernel $k(x, y)$, commonly used types of kernel include:

   (a) linear kernel: $k(x, y) =< x, y >$ (KPCA that uses a linear kernel is equivalent to PCA)

   (b) Gaussian kernel: $k(x, y) = e^{\frac{||x-y||^2}{\tau}}, \tau \in R^+$ (the most commonly used kernel)

   (c) polynomial kernel: $k(x, y) = (1+ < x, y >)^p, p > 1$

   (d) Three other manifold learning tools: Isomap, Laplacian Eigenmaps and Locally Linear Embedding has been proved to yield the same result to KPCA with a graph-based kernel [35].

2. Calculate the kernel matrix

$$K = [K_{ij}],\ K_{ij} = k(x_i, x_j)$$

   and centered kernel matrix

$$\hat{K} = HKH$$

   where $H = I_n - 1_n$, I_n is an $n \times n$ identity matrix and 1_n denotes a $n \times n$ matrix in which each element takes the value $1/n$.

3. Compute d largest eigenvalues $\lambda_1 ... \lambda_d$ and the corresponding eigenvectors

$$A = [a_1, \cdots, a_d]$$

of $\hat{K}$. where $a_i, i = 1, 2, \cdots, d$ are all n dimensional column vectors and scaled such that $|a_i| = \frac{1}{\lambda_i}$.

Given a new sample z, its mapping onto the PCA subspace can be calculated by:

$$\begin{aligned} y(z) &= P^T \hat{\phi}(z) && (3.1) \\ &= P^T(\phi(z) - \Phi_X 1_{n,1}) \\ P &= \Phi_X H_A && (3.2) \end{aligned}$$

where P is the column matrix of the first d principal components, Φ_X is the column matrix of $\phi(x_i)$, $1_{n,1}$ is a column vector with each elements being $1/n$ and $\hat{\phi}(z) = \phi(z) - \Phi_X 1_{(n,1)}$ is the centered $\phi(z)$.

Equation 3.2 can be rewritten as

$$\begin{aligned} y(z) &= w(z) - b && (3.3) \\ w(z) &= (H_A)^T(k(x_1, z), \cdots, k(x_n, z))^T && (3.4) \\ &= (H_A)^T k(X, z) \\ b &= (H_A)^T K 1_n && (3.5) \end{aligned}$$

where $H_A = HA$ and both $w(z)$ and b are d-dimensional vectors representing the projection of $\phi(z)$ onto the principal components and the mean of $\phi(x)$ over the n samples.

Note that in cases where d is close or equal to n, $\hat{K}$ is often not a full rank matrix, thus some λ_i could be very small or equal to zero. Consequently, scaling of the a_i can lead to infinite $|a_i|$ and numerical instability would occur in the training. To avoid this, we first reconstruct the feature vectors in an n-dimensional interim space by employing

a multidimensional scaling (MDS) technique [41], in which the relative distances and inner-products between vectors are preserved. Then, PCA is performed in the interim space. Details of the algorithm are given below:

1. Obtain the kernel matrix K of the dataset $X = x_1, x_2, \cdots, x_n$.

2. Find the n-dimensional representation $U_X = (u(x_1)u(x_2)...u(x_n))^T$ such that $< u(x_i), u(x_j) >= k(x_i, x_j)$,

$$U_X = \check{D}^{\frac{1}{2}}\check{B}^T$$

   where $\check{D}$ and $\check{B}$ are the diagonal matrix of the eigenvalues and column matrix of eigenvectors of K respectively. Notice that K is not the centered $\hat{K}$.

3. Perform PCA on U_X, and obtain the projections of U_X on all the principal components, the result is W_X.

4. Calculate H_A by solving

$$w(z) = (H_A)^T\Phi_X^T\phi(z) = (H_A)^T W_X^T w(z),$$

   The result is $H_A = W_X^+$, where $^+$ denotes Moore-Penrose pseudoinverse.

5. calculate b by $b = (H_A)^T K 1_n$.

Although the algorithm requires twice eigen-decomposition, it avoids numerical instability when d is high.

3.4 Open-set Classification based on Reconstruction Error

Open-set classification aims to classify an unknown observation into $N+1$ classes where the training samples are only available for N classes, the extra one class is dedicated for outlier or unknown observations that do not belong to any of the learned classes. This problem is meaningful not only for our posture detection, but also for many other applications when negative examples cannot be acquired.

The most simple case of open-set classification is the 2-class case ($N = 1$) which has been studied in previous literature. Due to the success of SVM as a traditional classifier where every class has training examples, attempts have been made to apply its principle to the open-set classification and the result is one-class SVM [64]. However, it is argued that the one-class SVM gives a larger decision boundary than necessary due to the fact that it has no tolerance to outliers or noisy examples.

A recent work in [37] has shown the potential of manifold representation in open-set classification. In this thesis, classification is achieved by simply measuring the reconstruction error of the observation z, which is the Euclidean distance between $\phi(z)$ and its projection $\phi(z')$ onto the linear subspace of the principal components in the feature space. Let c be the manifold of the class with training samples, the classification of z can be defined as

$$c_z = \begin{cases} c, & r(z|c) \leq d_T; \\ unknown, & r(z|c) > d_T. \end{cases} \tag{3.6}$$

where $r(z|c) = ||\phi(z) - P_d P_d^T \phi(z)||$ is the reconstruction error of z when projecting

onto the class manifold c, d_T is a threshold, P_d is the column matrix of the first d principal components in the feature space. Here $r(z|c)$ can be further simplified as:

$$r(z|c) = \sqrt{|||[y(z)_{d+1}, y(z)_{d+2}, \ldots, y(z)_n]||^2 + k(z,z)^2 - ||w(z)||^2} \tag{3.7}$$

where $y(z)_{d+1}, \ldots, y(z)_n$ is the projection of $y(z)$ onto all non-principal components, and $k(z,z)^2 - ||w(z)||^2$ is the squared distance from the infinite-dimensional $\phi(z)$ to its projection onto the n-d mapping space.

Though initially designed for two-class cases, this approach can be easily extended to multi-class cases:

$$c_z = \begin{cases} \arg\min_{\forall i} r(z|c_i), & min_{\forall i} r(z|c_i) \leq d_T; \\ unknown, & min_{\forall i} r(z|c_i) > d_T. \end{cases} \tag{3.8}$$

where $c_i, i = 1, 2, \cdots, N$ are the N class manifolds.

3.5 Greedy Approximation by Minimizing the Mapping Error

A major drawback in applying KPCA is the requirement to keep both H_A and all of the n training samples in X, and the time required to calculate $k(X, z)$ for any given new sample z . When n is very large, KPCA often becomes impractical to use. This problem becomes even worse in posture detection, where a trained KPCA model is going to be used on numerous patches of sliding window. One way to solve the problem is to find a reduced KPCA model, given by $\tilde{X}$ and $\tilde{H_A}$, such that the mapping

of a new observation, z in the reduced KPCA model is sufficiently close to the mapping of the sample in the original KPCA space, given by X and H_A, that is, to minimize the mean squared error of the mapping:

$$\varepsilon = E(||\tilde{y}(z) - y(z)||^2) \tag{3.9}$$

where $\tilde{y}(z)$ is the mapping function of the reduced KPCA model:

$$\tilde{y}(z) = \tilde{P}^T \hat{\phi}(z) \tag{3.10}$$

P follows the definition of Equation 3.3 and $\tilde{P} = \Phi_{\tilde{X}} \tilde{H}_A$, where $\tilde{H}_A$ and $\tilde{X}$ are a $m \times d$ matrix and a column matrix of m examples that are used to approximate H_A and X respectively, m is a predefined scalar and $m << n$.

3.5.1 The existant methods

The minimization of Equation 3.9 can be regarded as an optimization problem, in which the difference between the reduced model and the original model is minimized. A solution to linear and polynomial kernels was first proposed in the context of kernel SVM [14] and subsequently extended to kernel PCA [63]. Two generalised solutions were subsequently developed for non-polynomial kernels by Scholkopf et. al.[62] and Franc [26]. These works represent the state-of-the-art of the kernel approximation and are widely adopted in applications involving large dataset.

Since Equation 3.9 is the expectation of an yet undetermined function, Both existant methods proposed in [62] and [26] replace it with an equivalent objective function. In

[62], the new objective function is:

$$\varepsilon = ||P - \tilde{P}||^2 \tag{3.11}$$

which is to approximate principal components in the feature space with the approximated principal components $\tilde{P}$. In [26], the new objective function is slightly different:

$$\varepsilon = ||\Phi_X^T \Phi_X - \Phi_{\tilde{X}}^T \Phi_{\tilde{X}}||^2 \tag{3.12}$$

which is to approximate the mapping of X on the linear subspace of $\tilde{X}$. The purpose of this objective is to reduce the size of X prior to KPCA training so both training and mapping of the model can be accelerated.

Despite on being significantly simplified, both Equation 3.11 and 3.12 are ill-posed optimization problem and an analytical solution would be intractable. As a result, greedy solvers are used, which optimizing Equation 3.9 by incrementally adding new $\tilde{x}$ into X and expanding its linear span. In [62], the new $\tilde{x}$ is the pre-image of the mean vector of columns of the $P - \tilde{P}$, denoting the projections of principal components on the null space of $\phi_{\tilde{X}}$. After the $\tilde{x}$ is determined, the entire coefficients matrix $\tilde{H_A}$ is recalculated to yield the minimum $||P - \tilde{P}||^2$. In [26], the new $\tilde{x}$ is chosen from the training set X that maximally decreases the approximation error of Equation 3.12. This is achieved by Gram-Schmidt orthogonalization of all columns of X with $[\tilde{X}, \tilde{x}]$ and choose $\tilde{x}$ such that the mean length of the column vectors of the orthogonalized X is as small as possible.

However, in [62], the greedy selection of the examples is not optimal since the examples are incrementally selected to expand the linear span of the kernel space without considering the correlation between a newly selected example and the previous

examples in the kernel feature space. Although this problem is addressed in [26], it uses a different objective function that pursues the training speed at a cost of significant drop in the accuracy of approximation. Both methods do not guarantee that the mapping error is minimized. Additionally, both methods define their objective functions in the implicitly-defined kernel feature space which unnecessarily increases the computational cost.

3.5.2 The proposed method

In this chapter, we propose a new solution to the problem of Equation 3.9 that keeps a balanced trade-off between accuracy and efficiency. The objective function is designed to best reflect the purpose of the KPCA and the proposed algorithm performs the minimization in the mapping space instead of the kernel feature space so as to ensure that the reduced KPCA is an optimal approximation of the original KPCA.

Equation 3.9 can be written as:

$$\begin{aligned} \varepsilon &= E(||(\tilde{P}-P)^T\hat{\phi}(z)||^2) \\ &= E(tr((\tilde{P}-P)^T\hat{\phi}(z)\hat{\phi}(z)^T(\tilde{P}-P))) \\ &= tr((\tilde{P}-P)^T E(\hat{\phi}(z)\hat{\phi}(z)^T)(\tilde{P}-P)) \end{aligned} \tag{3.13}$$

where tr denotes the trace of matrix. If the factor of $\hat{\phi}(z)$ is ignored, then minimizing Equation(3.9) is equivalent to minimizing $||P-\tilde{P}||$ [62]. Note that $E(\hat{\phi}(z)\hat{\phi}(z)^T)$ is the covariance of the sample population in the kernel feature space which can be estimated from the training samples.

$$E(\hat{\phi}(z)\hat{\phi}(z)^T) \approx cov(\hat{\phi}(x)) \tag{3.14}$$

where $cov(\hat{\phi}(x))$ is the covariance matrix of $\hat{\phi}(x)$ in the feature space, also considering:

$$P_n^T cov(\hat{\phi}(x))P_n = D_n \tag{3.15}$$

where D_n is an $n \times n$ diagonal matrix of all n eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\hat{K}$, and P_n is the column matrix of all n principal components. Equation 3.13 can be more effectively written as:

$$\begin{aligned}
\varepsilon &= tr((\tilde{P} - P)^T P_n D_n P_n^T (\tilde{P} - P)) \\
&= tr((\tilde{P}^T P_n - I_{d,n}) D_n (\tilde{P}^T P_n - I_{d,n})^T) \\
&= ||(\tilde{P}^T P_n D_n^{\frac{1}{2}} - D_{d,n}^{\frac{1}{2}})||^2 \\
&= ||(\tilde{H_A}^T \hat{\Phi}_{\tilde{X}}^T P_n D_n^{\frac{1}{2}} - D_{d,n}^{\frac{1}{2}})||^2 \\
&= ||(D_n^{\frac{1}{2}}[w(\tilde{x}_1)|\ldots|w(\tilde{x}_m)]\tilde{H}_A - D_{d,n}^{\frac{1}{2}})^T||^2 \\
&= ||(D_n^{\frac{1}{2}} W_{\tilde{X}} \tilde{H}_A - D_{d,n}^{\frac{1}{2}})||^2
\end{aligned} \tag{3.16}$$

where $I_{d,n}$ is the first d rows of $I_{n,n}$, $D_n^{\frac{1}{2}}$ is the $n \times n$ diagonal matrix of $\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}$ and $D_{d,n}^{\frac{1}{2}}$ is the first d rows of $D_n^{\frac{1}{2}}$. Unlike [26] and [62] which define the objective function in kernel feature space, our objective function Equation 3.16 is defined in the KPCA mapping space which provides significant computational advantages.

Finding all vectors in $\tilde{H_A}$ and $\tilde{X}$ simultaneously according to Equation 3.16 is not trivial. We propose a greedy algorithm that selects $\tilde{x}$ one-by-one by iteratively executing the following two steps [62] [26]:

1. select one sample $\tilde{x}$ from X such that $\varepsilon(\tilde{X} \cup \tilde{x})$ is minimized

2. add the sample x into $\tilde{X}$, $\tilde{X} = \tilde{X} \cup \tilde{x}$

To simplify the first step, assuming $D_n^{\frac{1}{2}} W_{\tilde{X}}$ can be QR decomposed into a column matrix of orthogonalized vector Q and a right triangular square matrix R: $D_n^{\frac{1}{2}} W_{\tilde{X}} = QR$, ε then becomes:

$$\varepsilon = ||(QR\tilde{H}_A - (D_{d,n}^{\frac{1}{2}})^T)||^2 \tag{3.17}$$

This is a typical quadratic programming problem, the solution of $\tilde{H_A}$ for a given Q is:

$$\begin{aligned} \tilde{H_A} &= (QR)^+ (D_{d,n}^{\frac{1}{2}})^T \\ &= R^{-1} Q^T (D_{d,n}^{\frac{1}{2}})^T \end{aligned} \tag{3.18}$$

Hence Equation 3.17 becomes:

$$\begin{aligned} \varepsilon &= ||((QQ^T - I)(D_{d,n}^{\frac{1}{2}})^T)||^2 \\ &= \sum_{i=1}^{d} ||QQ^T v_i - v_i||^2 \\ &= \sum_{i=1}^{d} (||v_i||^2 + ||QQ^T v_i||^2 - 2 v_i^T QQ^T v_i) \\ &= \sum_{i=1}^{d} (\lambda_i - ||Q^T v_i||^2) \end{aligned} \tag{3.19}$$

where v_i is the i^{th} column of $(D_{d,n}^{\frac{1}{2}})^T$. So minimizing ε is equivalent to maximizing:

$$\zeta = ||(Q^T(D_{d,n}^{\frac{1}{2}})^T)||^2 \tag{3.20}$$

This optimization term is easy to calculate, and most importantly, when a new $\tilde{x}$ is added to $\tilde{X}$, matrix Q would change little because the new Q can be easily obtained by $Q = [Q|q]$, where q is the $\phi(\tilde{x})$ being Gram-Schmidt orthogonalized with the rest of the column vectors of Q. Therefore, the increase of Equation 3.20 for each new $\tilde{x}$ being added to $\tilde{X}$ can be expressed as:

$$\begin{aligned} \Delta\zeta &= ||([Q|q]^T(D_{d,n}^{\frac{1}{2}})^T||^2 - ||Q^T(D_{d,n}^{\frac{1}{2}})^T||^2 \\ &= ||q^T(D_{d,n}^{\frac{1}{2}})^T||^2 \end{aligned} \tag{3.21}$$

Iterative maximization of this term forms the main idea of the proposed greedy approximation algorithm. Although it is possible to find the new $\tilde{x}$ in the entire input space by gradient descent optimization, such optimization will be slow and unstable because Equation 3.21 may have many local maxima with respect to $\tilde{x}$. Nevertheless, finding $\tilde{X}$ from X can be considered as a problem of sampling points from the clusters defined by X [62]. Therefore, minimizing Equation 3.17 becomes a finite-state searching problem and we propose the following algorithm:

1. Given a dataset X in the input space and a kernel function $k(.,.)$, obtain the image of X in the KPCA mapping space, denoted by W_X. Initialize the orthonormalized candidate set $V = D_n^{\frac{1}{2}} W_X$.(Complexity: $O(n^2)$)

2. Calculate the inner-product matrix $N = V^T D_{d,n}^{\frac{1}{2}}$, find one of its rows with the largest norm. (Complexity: $O(2nd)$)

3. Add the i^{th} column vector of V to the reduced set $\tilde{X} = \tilde{X} \cup v_i$. v_i can be removed from V.

4. Update V by orthonormalizing its column vectors V: $V = V - v_i v_i^T V$. (Complexity: $O(n^2)$)

5. Repeat step 3 to 5 until m examples are chosen or the mapping error is lower than a threshold. (Complexity: $O(n^2) + O(2nd) + O(n^2) = O(m(2n^2 + 2nd))$)

6. Obtain $\tilde{H_A}$ by $\tilde{H_A} = (D_n^{\frac{1}{2}} W_{\tilde{X}})^+$.

The complexity of the proposed algorithm is on the order of $O(m(2n^2+2nd))$. This is slightly faster than the algorithm proposed in [26], whose complexity is $O(mpn^2)$ (p is the search depth that usually equals to a quarter of n).

3.6 Performance Evaluation

3.6.1 Mapping and Reconstruction

Six real datasets from the intelligent data analysis (IDA) benchmark repository [54] were used to evaluate the performance of the proposed algorithm. These datasets are *banana, breast cancer, diabetes, flare, german and heart*. Samples in each dataset have been manually labeled into two classes. Since each dataset contains large number of samples, we randomly chose a subset of the samples in each experiment due to the limitation of the available computing resources. The numbers of samples chosen are same as the ones used in [26]. Table. 3.1 lists the numbers of samples used in the experiments and the dimensions of the six datasets.

Name	Dimension	No. of samples
Banana	2	400
Breast Cancer	9	200
Diabetes	8	468
Flare	9	666
German	20	700
Heart	13	170

Table 3.1: Number of samples chosen and dimensions of the six test datasets

KPCA is primarily used as an effective tool for manifold learning or data compression. In manifold learning, the KPCA aims to compute the projection of an unknown sample into the kernel mapping space so its distance to the manifold can be determined. In data compression, the unknown sample is reconstructed from a compact set of vectors defining the kernel mapping space. Due to these reasons, the following two sets of experiments were designed to evaluate the proposed algorithm by measuring the mapping and reconstruction errors between the reduced KPCA and original KPCA. The reconstruction error is calculated between the original example and the reconstructed one from the mapping of the example in the kernel feature space. Gaussian kernels were adopted in all the experiments. The results were compared to those obtained by the Franc's method [26]. Notice that since the proposed Greedy approximation only uses W_X and is kernel-independent the results presented in this chapter can potentially be extended to arbitrary kernels and even graph-based ones.

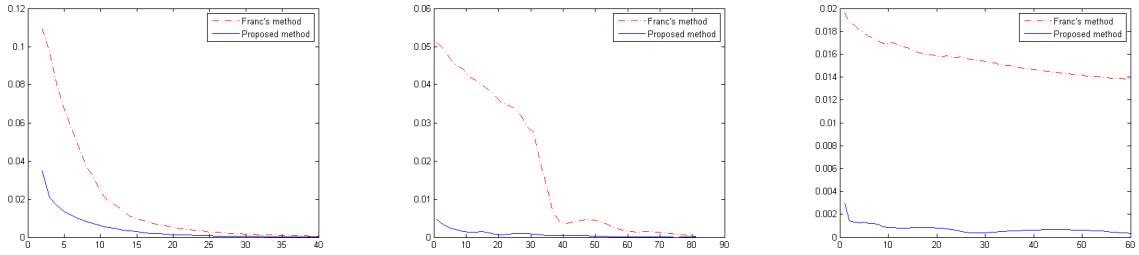
Figures 3.2 and 3.3 show the mean squared mapping and reconstruction errors versus the number of samples in the reduced set. The errors were averaged over 20 runs; in each run, the number of samples specified in Table. 3.1 were randomly selected from the original datasets. Since in Franc's method the maximum number of principal components are reduced to m, the mapping and reconstruction errors were calculated using

the first m principal components. In other words, when $m = 1$, the errors were calculated using the first principal component. The mapping errors of the reduced KPCA obtained by the proposed method produced substantially lower mapping errors in all cases than Franc's method. This indicates that the proposed method is particularly useful for the cases when KPCA is used as a manifold learning tool.

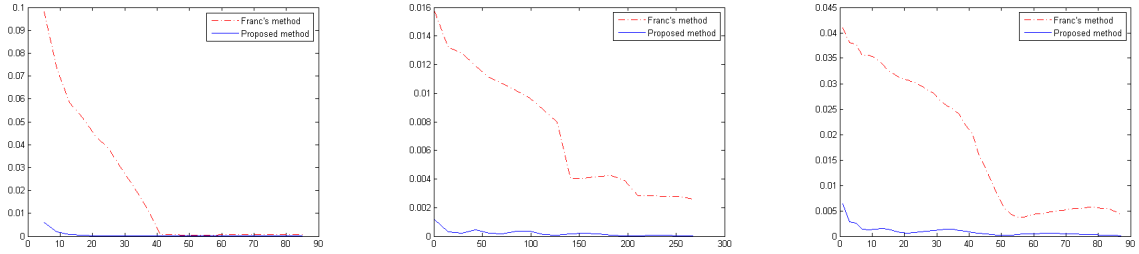
In terms of the reconstruction error, the proposed method performed comparable to Franc's method. Since the kernel reconstruction is a reverse mapping from the kernel feature space to the input space and all principal components are used, the reconstruction result is solely dependent on the configuration of the kernel feature space but not relevant to the direction of the principal components. Therefore we can conclude that the proposed method has comparable capability of approximating the kernel feature space as Franc's method. However, the directions of the principal components of the reduced set obtained by Franc's method are not necessarily aligned with the direction of the original KPCA.

3.6.2 Open-set Classification

In the first experiment, we compared the performance between the classifiers derived from the original KPCA and the reduced KPCA. The dataset used in this experiment consists of six synthesized objects in 2D and 3D space. Table 3.2 lists its specifications. The low-dimentionality of the six objects allows us to visualize the decision manifolds of the classifiers. Figure 3.4 shows the decision manifolds trained for the six objects using the original KPCA and reduced KPCA obtained by the proposed approximation method and Franc's method [26] at given m as listed in the Table 3.2. For the reduced

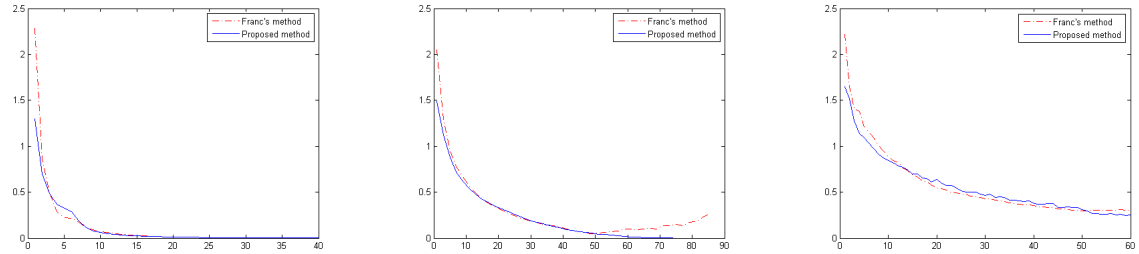


(a) banana (n=400) (b) breast cancer (n=200) (c) diabetis (n=468)

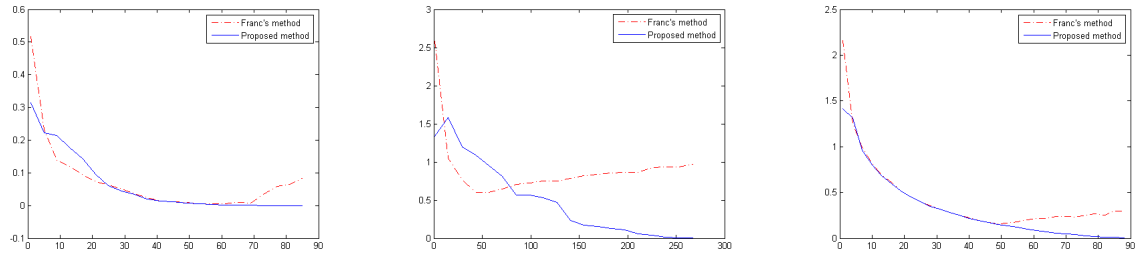


(d) flare (n=666) (e) German (n=700) (f) heart (n=170)

Figure 3.2: Mapping errors (y axis) w.r.t. m (x axis), the results obtained by Franc's algorithm and the proposed algorithm are represented by red and blue curves respectively



(a) banana (n=400) (b) breast cancer (n=200) (c) diabetis (n=468)



(d) flare (n=666) (e) German (n=700) (f) heart (n=170)

Figure 3.3: Reconstruction errors (y axis) w.r.t. m (x axis), the results obtained by Franc's algorithm and the proposed algorithm are represented by red and blue curves respectively

Name	Dimensionality	No. of Examples	d	m
Square	2	200	8	18
8-Shape	2	400	8	18
Spiral	2	500	8	45
3D 8-Shape	3	400	8	18
3D Helix	3	300	8	45
3D Swiss Roll	3	1100	8	60

Table 3.2: Specification of 6 datasets and corresponding parameters for the reduced KPCA

KPCA, d was chosen to be 8. Decision threshold of each classifier is tuned to have 5% missing rate on training data.

It can be seen that while the original KPCA successfully obtained smooth and tight decision manifolds that enclose the data of only the same class, the proposed approximation achieved almost identical results with significantly higher computational efficiency. On the other hand, Franc's method did not provide the results as we expected, its decision surface was generally deformed and even broken into pieces when applied to the Spiral and Helix objects. The experimental results partially verified the correctness of our motivation, that is, by combining the manifold learning with the proposed approximation algorithm, it is feasible to create an open-set classifier that is both effective and efficient.

In the second experiment, we evaluated the detection rate of the open-set classifier based on the reduced KPCA using the proposed approximation method. We used the 1000 examples of '0' from the USPS handwriting digit database for training and 10001 examples of all classes from the rest of the dataset for testing. The performance of the classifier using the proposed approximated KPCA model was compared with the original KPCA model and Franc's greedy KPCA model. We set the parameters $\sigma = 4$ and $d = 20$, according to experimental setup of [37]. The areas under ROC curves of

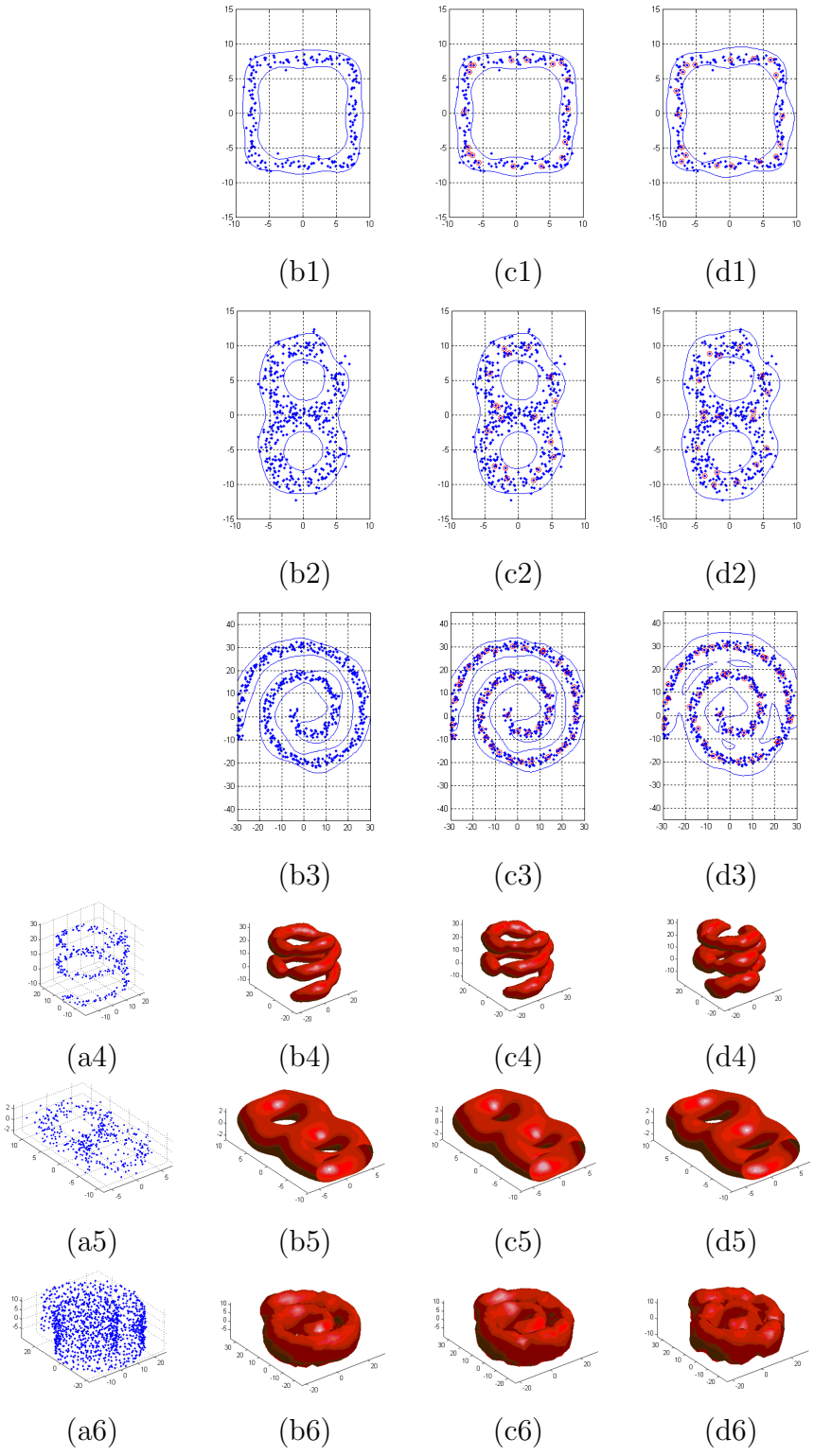


Figure 3.4: Decision manifolds of the open-set classifiers trained for the six synthesized objects. Column (a) shows training data in 3D data space, for 2D data this column is missing because the data are illustrated in other columns. Column (b), (c)and (d) show the classification boundaries obtained by the original KPCA, the proposed KPCA approximation method and the Franc's approximation method respectively.

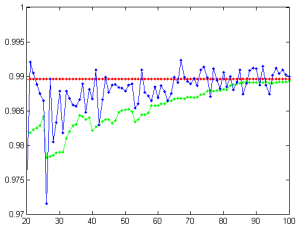
these three open-set classifiers with respect to m, the number of selected samples in the reduced KPCA, are shown in Figure 3.5. Since in Franc's method the maximum number of principal components has to be under m, we omitted all cases when $m < d$ in this experiment.

It has been demonstrated that the proposed method achieved a much higher area under ROC curve than Franc's method for most m. Both curves converge to the area under ROC curve of the original KPCA based classifier, and the proposed method converges much faster.

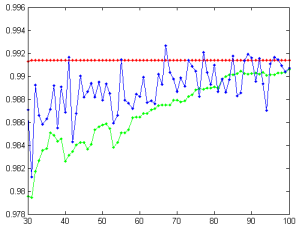
Further experiments have shown that the proposed method is not sensitive to the number of principal components d. But when the kernel width σ becomes very small, all examples will become almost orthogonal to each other and the difficulty of being approximated by a lower-dimensional space will be increased. In this case, performance of both approximations deteriorates and becomes close to each other. Figure 3.5 (a) and (b) illustrate the case of small σ, ($\sigma = 1.2$). As seen, the convergence becomes much slower and the rate of both approximations are very close. Practically, σ should be relatively large since small σ will lead to overfitting and also decrease the performance of the original classifier as well.

3.7 Discussion

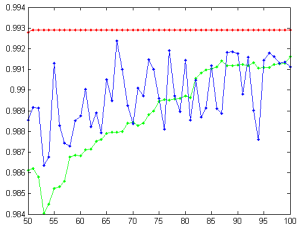
The main contribution of this chapter is the greedy approximation algorithm, which substantially increases the efficiency of the kernel PCA mapping with very small mapping error (compared to previous methods). Further improvement of the proposed method is possible. KPCA mapping and reconstruction have so far been implemented



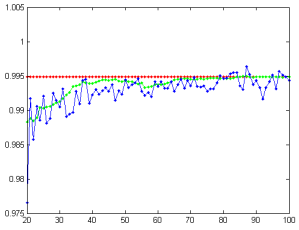
(a) $\sigma = 1.2, d = 20$



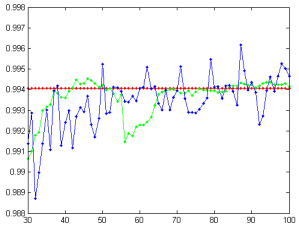
(b) $\sigma = 1.2, d = 30$



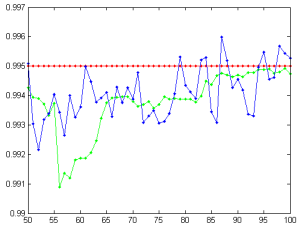
$\sigma = 1.2, d = 50$



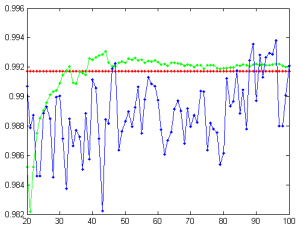
(c) $\sigma = 4, d = 20$



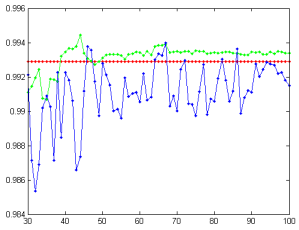
(d) $\sigma = 4, d = 30$



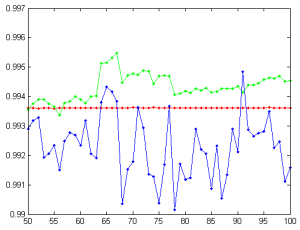
$\sigma = 4, d = 50$



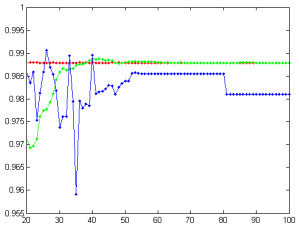
(a) $\sigma = 10, d = 20$



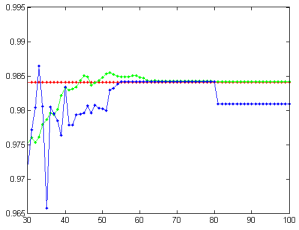
(b) $\sigma = 10, d = 30$



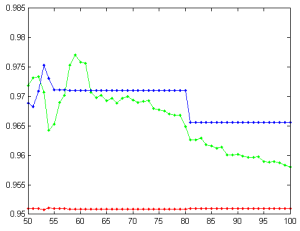
$\sigma = 10, d = 50$



(c) $\sigma = 100, d = 20$



(d) $\sigma = 100, d = 30$



$\sigma = 100, d = 50$

Figure 3.5: The areas under ROC curves (AUC) of the three open-set classifiers w.r.t. m (horizontal axis): Red-Original KPCA, Green-Proposed approximation method, Blue-Franc's method

by comparing the new sample with a fixed set of chosen examples. However, intuition suggests that the examples that are close to the new sample will be more important, which could be adaptively selected after comparing the new sample with very few number of examples. Incorporation of this may probably lead to a new fast mapping and reconstruction algorithm that follows a coarse-to-fine heuristics.

Chapter 4

Posture Detection by Kernel PCA

In this chapter, we propose a new approach for detecting human postures from single images. This approach follows the detection by pattern framework. In the training stage, KPCA is employed to learn the manifold span of a set of HOG-represented examples that can effectively represent a posture. In the detection stage, the open-set classifier presented in the previous chapter was iteratively applied on the HOG of every detection window of the image to identify and locate the postures. Experimental results have shown that the proposed method can achieve promising detection rates with a relatively small amount of positive only training data.

4.1 System Description

The proposed posture detection system adopts the HOG as the features and consists of two phases: training and detection. In the training phase, the manifold is learned through KPCA for each posture to be detected from its training samples. In the detection phase, HOG features are extracted from the test image and projected to the manifold of every learned posture. The reconstruction error between the original HOG features and the preimage of the HOG features of the test image in the KPCA space

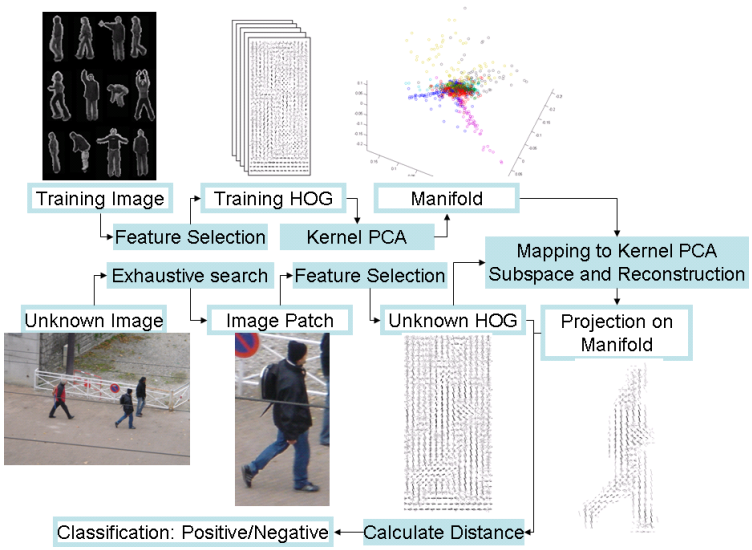


Figure 4.1: Schematic of the proposed posture detection system

is used to determine whether the test image contains a human in a specific posture. Details of the proposed method is presented in subsequent sections of this chapter.

4.2 Extraction of HOG

We construct the HOG feature [19] by dividing the image into multiple overlapping blocks of the same size and quantizing the gradient direction of all pixels into 9 directions. For each block, the histogram is formed such that the occurrence of each direction represents the total gradient magnitude of the gradient along the direction and the histogram reflects the weighted gradient distribution within each block. The HOG of each block is then normalized so the sum of the HOG elements is unity, and the HOG of the entire image is formed by concatenating the block HOGs. If the gradient magnitudes of all pixels in one block are zero, the corresponding block HOG will be normalized to uniformly distributed (each of its elements are set to 1/9). In

our approach, each block is an 8×8 image patch that has 50% overlapping with its neighbors; in total 31×15 blocks are used to cover a 128×64 image window. The HOG for the image window is a $31 \times 15 \times 9$-dimensional vector, and the sum of all elements in one HOG is 31×15. Finally, the square root of all components of the HOG are determined, this is to allow Bhattacharyya distance between two HOGs to be quickly evaluated. We will explain the reason of doing so later.

According to the report [50], HOG outperforms other local descriptors in locality and invariance characteristics. The HOG has also been proved effective in many previous works [20] [82] [3]. Its dimensionality increases linearly with the resolution of the image. Though several other features such as covariance features [72] could achieve similar performance, their dimensionality increases exponentially with the image resolution and, hence, they are not suitable for kernel based methods, especially when the amount of training data is small.

4.3 Training

A non-linear manifold is able to capture variation that can occur within a class and in our case the postures of interest. Thus we employ the KPCA described in the previous chapter to learn the manifold span of each of the types of postures.

The training of the classifier requires a set of training images for each class of postures. We assume each training example should only contain one posture without any background. Therefore, the background of all training images are first removed. Since a small number of training examples for each posture class is assumed and no negative examples are available, the influence of the background cannot be automatically

'weighted down' by finding the weak covariance of features between background and the classes as pointed out in [19] and [75]. In addition, all training samples are resized to the same resolution. In this thesis, 128×64 pixels was adopted.

The training of the class manifold for each posture class is independent. Hence, if a new posture class is defined and added to an existing detection system, we only need to train the manifold for this new class. Given a set of HOGs $X = \{x_1...x_n\}$ extracted from the n training samples for a specific class, where x_i is the HOG for the $i'th$ sample, the class manifold, denoted by matrix H_A and vector b, can be learned from X by using the kernel PCA-based learning approach proposed in the last chapter. Three parameters are required to be determined, the fist parameter is the kernel function k. According to the experiments of [38], the RBF kernel achieved the best result in open-set classification:

$$k(x, y) =< \phi(x), \phi(y) >= e^{-\frac{||x_1 - X_2||^2}{2\sigma^2}}$$

The second parameter is the number of principal components d; it can be determined such that the mean reconstruction error of all training samples is less than a specific threshold. The last parameter is the number of reduced examples m for kernel PCA approximation, as a tradeoff between speed and performance this parameter is determined according to the requirement of the system.

4.4 Detection

The detection of trained postures is formulated as a problem of open-set classification which classifies a test image window into one type of the trained postures or as negative

(background/unknown posture). Let z be the test image and $p(c_i|z)$ be the probability of posture c_i given z. Here, z is considered containing posture c_k, if

$$\begin{cases} k = \arg\max_{\forall i} p(c_i|z) \\ p(c_k|z) > th \end{cases} \tag{4.1}$$

where th is a threshold, and $p(c_i|z)$ is the decision function of the classifier. Using Bayesian rule it becomes:

$$p(c_i|z) \propto \frac{p(z|c_i)}{p(z)} \tag{4.2}$$

where $p(z|c_i)$ is the conditional probability of z given posture c_i and $p(z)$ is the prior probability of z. As each c_i is represented by a manifold in the HOG space, the closest HOG to z that lies in the manifold span of c_i can be readily expressed as the KPCA reconstruction result of z [49] [42], which is the reverse mapping from $\phi_P(z)$ in the feature subspace to the original HOG space:

$$r(z) = \phi^{-1}(\phi_P(z)) \tag{4.3}$$

where $\phi_P(z)$ is the projection of $\phi(z)$ onto the first d principal components. The obtained $r(z)$ can be regarded as a HOG of reference that illustrates 'what z should look like when z is assumed to be in the class of c_i. Meanwhile, since the HOG is a histogram-based feature, we can assume that the conditional probability $p(z|c_i)$ is proportional to the exponential of Bhattacharyya distance between the histogram of z and $r(z)$. Thus we can write:

$$p(z|c_i) = e^{\alpha_1 D_B(z,r(z))} \tag{4.4}$$

where D_B is the Bhattacharyya distance and α_1 is a constant. The KPCA reconstruction is a recursive and slow optimization process, however, referring back to the fact that each component of z is actually the square root of the original HOG component, $D_B(z, r(z))$ becomes:

$$\begin{aligned} D_B(z, r(z)) &= \langle z, r(z) \rangle \\ &= \frac{1}{2}(|z|^2 + |r(z)|^2 - |z - r(z)|^2) \end{aligned} \tag{4.5}$$

Also since both $|z|$ and $|r(z)|$ are constant, and $|z - r(z)|^2$ can be calculated from the kernel function (in our case only the Gaussian RBF kernel is used), the distance becomes:

$$D_B(z, r(z)) = \alpha_2 + \frac{1}{2} ln(2\sigma^2 \langle \phi(z), \phi_P(z) \rangle) \tag{4.6}$$

where $\alpha_2 = 15 \times 31$ and σ is the width of the kernel $k(.,.)$. This distance function is similar to the decision function of the novelty detector proposed by Hoffmann [37]. But the difference is that our distance function measures the reconstruction error in the HOG space while Hoffmann's function measures it in the feature space. Furthermore, since $\phi_P(x)$ is in the principal component subspace, $\langle \phi(z), \phi_P(z) \rangle$ can be replaced by $\langle (y_n(z) + b_n), (y(z) + b_n) \rangle$, where $y(z)$ is the mapping of z onto the KPCA subspace, and $y_n(z)$ and b_n are the mapping result $y(z)$ and centering offset b respectively when all n eigenvectors of $\hat{K}$ are used as principal components.

The $p(z|c_i)$ can be a good decision function in many open-set classification problems. But, in our case, $p(z|c_i)$ alone cannot provide enough discriminative power. This

is because each block of the HOG is a 9-bin histogram vector, and the intrinsic distribution of the binned gradient direction can be regarded as a uniform categorical distribution in most cases. The prior distribution of each block HOG is actually a multinomial distribution and will peak when its components are equal to the mean vector. Therefore, $p(z)$ in Equation 4.2 will not be a constant and must be taken into account. As the multinomial distribution is hard to calculate, again we approximate $p(z)$ by the exponential of the Bhattacharyya distance between the HOG of the image and the mean HOG vector:

$$p(z) = e^{\alpha_1 D_B(z,(\frac{1}{3})_{15\times31\times9})} \tag{4.7}$$

$$= e^{\alpha_1(\alpha_2-\frac{1}{2}|z-(\frac{1}{3})_{15\times31\times9}|^2)} \tag{4.8}$$

Where $(\frac{1}{3})_{15\times31\times9}$ denotes the mean HOG vector with each element being 1/3. Combining Equation(4.2), Equation(4.4), Equation(4.6) and Equation(4.7) together we will have:

$$p(c_i|z) \propto e^{\frac{\alpha_1}{2}[ln(2\sigma^2\langle\phi(z),PP^T\phi(z)\rangle)+|z-(\frac{1}{3})_{15\times31\times9}|^2]} \tag{4.9}$$

Substitute $p(c_i|z)$ with Equation(4.9) the decision function Equation(4.1) will become:

$$\begin{cases} k = \arg\max_{\forall i} ln(\langle\phi(z),\phi_P(z)\rangle) + |z-(\frac{1}{3})_{15\times31\times9}|^2 \\ ln(\langle\phi(z),\phi_P(z)\rangle) + |z-(\frac{1}{3})_{15\times31\times9}|^2 > \frac{2}{\alpha_1}ln(th) \end{cases} \tag{4.10}$$

Note that parameter α_1 governs the tradeoff between false positives and false negatives and can be tuned to attain the best detection result.

4.5 Experimental Results

4.5.1 Detection of humans

In our first experiment we tested our approach on a human detection task, though this approach was not initially designed for this task, we used the INRIA database for training and testing. 489 images and their mirror images from the INRIA positive training dataset were used for training; their background were manually removed. The removal of background is crucial to the performance of the classifier since there were no negative examples. If the background is included in the positive examples the learning approach will be unable to automatically 'weight down' the background regions as in SVM, and HOG components from these regions will be learned as part of the posture manifold. Such effect can only be avoided if the examples densely cover every possible HOG pattern of the background, but this is against our assumption of a small amount of training data and is often impractical. On the other hand, if the background regions are removed, the corresponding HOG components will become uniformly distributed, which coincides with our assumption that the binned HOG of one block in the background region should follow the uniform categorical distribution. According to our experiments, the detection results would not make sense if the background of the examples were not removed.

In test stage, the INRIA positive test dataset consisting of 589 human images is used as positive test samples and 9060 images randomly cropped from the INRIA negative test dataset (mainly consisting of landscape images) are used as negative test samples. Background in test images are not removed. The result is compared to Dalal's work

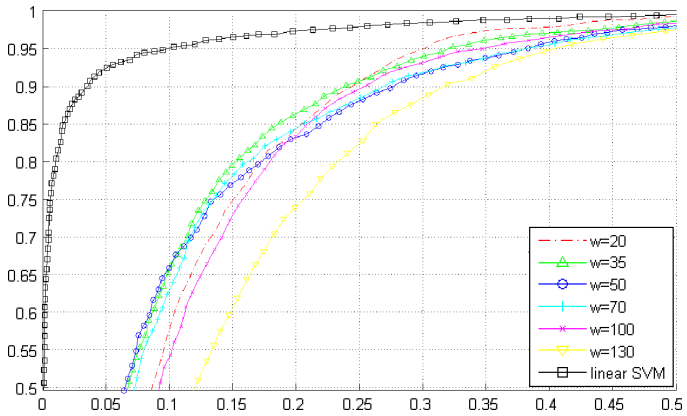


Figure 4.2: The ROC curves of the proposed detector and Dalal's detector [19] on human detection task with different kernel width w

[19] that uses a linear SVM for classification. The ROC curves of the linear SVM-based detector and the proposed detector are shown in Figure 4.2.

The performance is worse than the SVM-based detector if under the condition that no negative samples were used. Since in human detection or pedestrian detection, the variations of HOGs of human bodies are usually not continuous or not smooth, in which case the manifold representation is not particularly suitable. Even if it can be enclosed in a lower-dimensional manifold, its dimensionality will still be too high and cannot be easily estimated from a relatively small training set. Hence it is more important to find discriminative features rather than representative features. Introducing a new negative class ('Non-human' class) denoted by an extra KPCA model learned from the negative training dataset will lead to a close-set classifier, of which performance is almost identical to the detector using linear SVM (the ROC curves of both classifiers are shown in figure 4.3). As stated before, this experiment was not used to show the performance of our method as it is not designed for the human detection task, but the capability of the proposed method capturing the representative features.

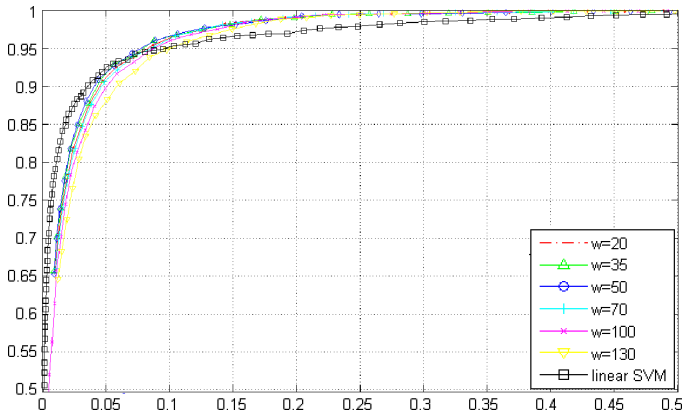
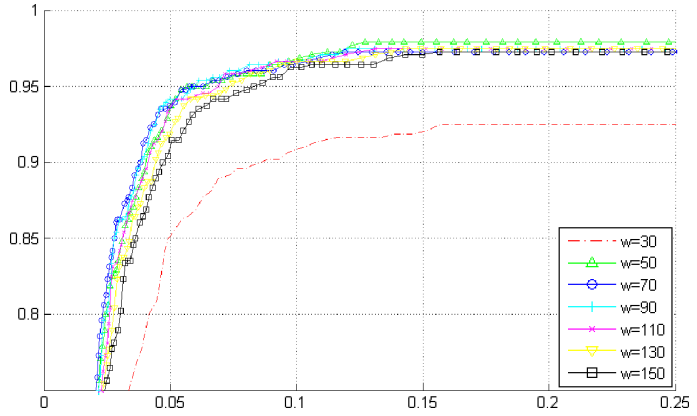


Figure 4.3: ROC curves of the proposed detector in human detection task with different kernel width w when negative examples are introduced, the result is compared to Dalal's detector.

Figure 4.4: Typical images for the 12 postures from the Weizmann action dataset [7]

4.5.2 Detection of postures

In the following experiments, the proposed approach was tested in multi-posture cases where training and test samples were extracted from the Weizmann action dataset [7]. The dataset contains 93 low resolution video (188×144, 25 fps) sequences for 10 actions. Nine subjects played each action once. Over 2000 images were manually cropped and divided into 12 postures to form the posture corpus. Figure 4.4 shows typical images for the 12 postures. The corpus was randomly divided into training and testing sets at a ratio of 7:3, and the same negative test dataset used in previous experiment were used. On average, there are 150 training samples for each posture. A number of experiments were conducted to evaluate the performance of the proposed method.



(a)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	NG
P1	22	0	0	0	0	0	0	0	0	0	0	0	0
P2	1	66	0	0	0	0	2	0	0	0	0	0	1
P3	0	0	19	0	0	0	0	0	0	0	0	0	5
P4	0	0	0	23	0	0	0	0	0	0	0	0	6
P5	0	0	0	0	41	0	0	0	0	0	0	0	0
P6	0	0	0	0	0	22	0	0	0	0	0	0	3
P7	0	0	3	0	0	0	66	0	0	0	0	0	1
P8	0	0	0	0	0	0	0	16	0	0	0	0	12
P9	0	0	0	0	0	0	0	0	27	0	3	0	2
P10	0	0	7	0	0	0	0	3	0	40	0	0	3
P11	0	0	0	0	0	0	0	0	2	0	47	0	9
P12	0	0	0	0	0	0	0	0	0	0	0	26	10
NG	38	38	55	1	1	16	24	0	0	0	0	0	8887

(b)

Figure 4.5: (a) ROC curve of the proposed detector on Weizmann action database with different kernel width w (b) The confusion matrix of the proposed detector. P1-P12 represents the 12 postures and NG represents the negative samples

This experiment aims to find the posture detection rate of the proposed KPCA-based detector. Each test image is classified into 13 classes: 12 trained postures and 1 background including unknown postures. The experiment was carried out for several times, each time with a different RBF kernel width σ, The ROC curves of the detector and corresponding confusion matrix at the best detection/false alarm ratio are shown in Figure 4.5 (a) and (b) respectively. This result starts to show the potential of the proposed detector. It is able to achieve 94% detection rate at 0.05 false positive rate. We believe it can be further improved if higher resolution training samples were used.

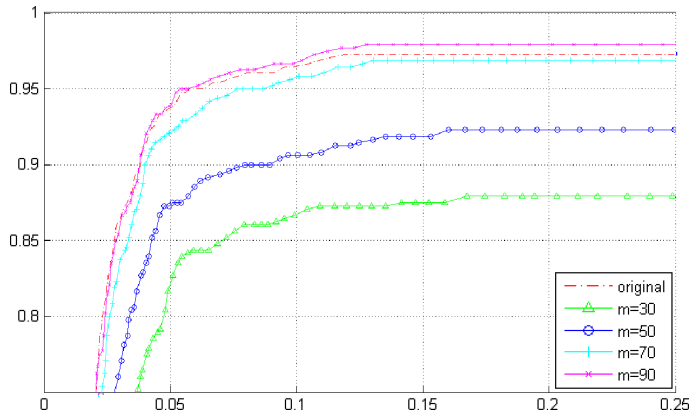


Figure 4.6: ROC curve of the proposed detector on Weizmann action database with different size of the reduced set m in KPCA approximation

4.5.3 Posture detection with KPCA approximation

The third experiment aims to test the accuracy of the proposed KPCA approximation algorithm and how it will affect the performance of the detector. The experiment was carried out for several times, each time with a different number of selected examples m. The ROC curves of the detector using a reduced model is compared to that of the detector using original and full KPCA model. These ROC curves are shown in Figure 4.6. As expected, if m, the number of selected examples for the approximated KPCA model increases, the performance increases. When m is 90, the performance of the approximated KPCA model is almost as good as the original full model trained with all examples.

4.5.4 Detection of posture in videos

In the last experiment, we employed the detector trained from the Weizmann's dataset to detect the postures of interest from MPEG-7 ETRI video sequences and videos in the INRIA human dataset. The detection was conducted frame-by-frame using a sliding

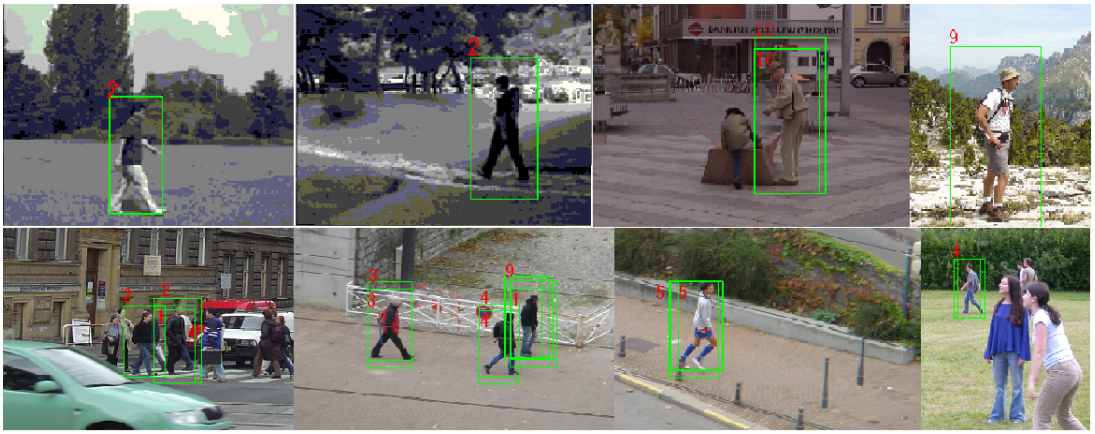


Figure 4.7: Some detection results on the images from ETRI and INRIA videos

window. Each window image is classified into either one of the trained postures or non-posture. Figure 4.7 shows several results where the bounding boxes indicate detected postures related to walking, running and standing. In the last image, the standing posture of the central person was missed. This is because the training samples for the standing posture were all from side viewpoint rather than a frontal view.

4.6 Discussion

The contribution of the chapter is a new method for detecting postures from single images using KPCA based manifold learning. The performance is quite promising considering only 90 training samples per posture was used without involving any negative samples. Due to lack of literature for similar problems, the performance of the proposed system was not compared with others. Further experiments are required to explore the potential of the method to deal with multiple viewpoints and occlusion.

Chapter 5

Conclusion

5.1 Summary

Posture detection is usually a crucial step in human motion analysis. Despite its significance, it has not been extensively studied. Most literature has basically focused on human/pedestrian detection with assumption of upright standing/walking humans and has also formulated the problem as a two-class classification problem. While human/pedestrian detection is useful in many applications like video surveillance, there are also many applications that require the identification of specific postures. In this thesis, we have studied the problem of posture detection.

Compared to detection of humans and faces, the key challenge of posture detection is the complex in-class variations of the body appearance, particularly those caused by body articulation and changes of viewpoints. Previous research has shown that as a posture undergoes variations due to articulation, their appearance changes smoothly and traces out a smooth manifold. As long as this smoothness assumption holds, we are able to obtain the manifold with enough accuracy from a small number of examples. This representative manifold interpolates and generalizes the limited examples and

creates a pattern for identifying the posture.

In this thesis, we adopted HOG as features to describe the postures and employed KPCA, a non-linear manifold learning technique, to learn the posture manifolds and developed an open-set classifier for posture detection that can be trained only using positve examples. With the support of the proposed KPCA approximation, the system is compact, fast, and requires much smaller memory space for storing training dataset in the classification stage than the standard KPCA. The experiments on both synthetic and real datasets have verified that the proposed KPCA approximation and posture detection methods are effective.

5.2 Future Work

In the manifold representation, selection of the features is important. Though HOG has behaved well in our system, it may be too detailed in representing appearance and contains little global structural information. According to [3], a manifold representation will have poor performance for a 'bag of features' consisting of local descriptors extracted from interesting points, because the variations of the 'bag of features' may not be smooth and can not be traced out as a manifold. Thus the SIFT-like features will not be applicable in the proposed framework. A possible improvement is to construct a local descriptor that contains enough structural information.

The computational efficiency of the system is another possible direction where improvement can be made. In the proposed system we have significantly increased the

efficiency by using the reduced-set approximation algorithm compared with the standard kernel PCA and reconstruction. However, it still needs to be improved for real-time application. Potential improvements are possibly made in three aspects. First, reconstruction in kernel PCA is much slower than mapping. But in our framework the reconstruction error has to be measured in the input space. If these measurements can be done in the feature space or KPCA subspace, then the reconstruction can be avoided and the computation will be substantially lowered. Second, the proposed kernel PCA approximation algorithm can be improved as discussed in Section 3.7, resulting in a tree-based, less compact but more efficient manifold representation. This would accelerate the mapping process in the same way as a decision tree accelerates the template matching process. Third, in this thesis, the exhaustive search strategy was used in detecting the postures in an image. Here, a sliding window has to iterate through different positions and at various scales which generates large number of patches that require to be classified. This process may be accelerated by using heuristic search strategy.

Bibliography

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Proc IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–882–II–888, 2004.

[2] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Proc. IEEE Workshop on Vision for Human-Computer Interaction*, page 72, 2005.

[3] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *Proc. Asian Conference on Computer Vision*, volume 3851/2006, pages 50–59, 2006.

[4] L.H.W. Aloysius, G. Dong, H. Zhiyong, and T. Tan. Human posture recognition in video sequence using pseudo 2-d hidden markov models. In *Proc. Control, Automation, Robotics and Vision Conference*, volume 1, pages 712–716, 2004.

[5] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. Shape-based pedestrian detection and localization. In *Proc. Intelligent Transportation Systems*, volume 1, pages 328–333, 2003.

[6] M. Bertozzi, A. Broggi, A. Fascioli, and P. Lombardi. Vision-based pedestrian detection: will ants help? In *Proc. Intelligent Vehicle Symposium*, volume 1, pages 1–7, 2002.

[7] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *The Tenth IEEE International Conference on Computer Vision*, pages 1395–1402, 2005.

[8] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. Annual Workshop on Computational learning theory*, pages 144–152, 1992.

[9] B. Boulay, F. Bremond, and M. Thonnat. Human posture recognition in video sequence. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 23–29, 2003.

[10] G.R. Bradski and J.W. Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13(3):174–184, 2002.

[11] M. Brand. Shadow puppetry. In *Proc International Conference on Computer Vision*, volume 2, pages 1237–1244, 1999.

[12] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–574, June 1997.

[13] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. Shape-based pedestrian detection. In *Proc. Intelligent Vehicle Symposium*, pages 215–200, 2000.

[14] C.J.C. Burges. Simplified support vector decision rules. In *Proc. International Conference on Machine Learning*, pages 71–77, 1996.

[15] K.Y. Chang and J. Ghosh. Principal curves for nonlinear feature extraction and classification. *SPIE Applications of Artificial Neural Networks in Image Processing*, 3(3307):120–129, 1998.

[16] Y. Chen and C. Chen. A cascade of feed-forward classifiers for fast pedestrian detection. In *Proc. Asian Conference on Computer Vision*, volume 4843, pages 905–914, 2007.

[17] H. Cheng, N. Zheng, and J. Qin. Pedestrian detection using sparse gabor filter and support vector machine. In *Proc. Intelligent Vehicles Symposium*, pages 583–587, 2005.

[18] J.C. Cheng and J.M.F. Moura. Automatic recognition of human walking in monocular image sequences. *The Journal of VLSI Signal Processing*, 20(1):107–120, 1998.

[19] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[20] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. European Conference on Computer Vision*, volume 3952, pages 428–441, 2006.

[21] M. Dimitrijevic, V. Lepetit, and P. Fua. Human body pose recognition using spatio-temporal templates. In *Proc. ICCV workshop on Modeling People and Human Interaction*, volume 2, page (No page information), 2005.

[22] A. Elgammal and C.S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–681–II–688, 2004.

[23] AE Elgammal and LS Davis. Probabilistic framework for segmenting people under occlusion. In *Proc. International Conference on Computer Vision*, volume 2, pages 145–152, 2001.

[24] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *Proc. Intelligent Vehicles Symposium*, pages 500–504, 2003.

[25] R. Fablet and M.J. Black. Automatic detection and tracking of human motion with a view-based representation. In *Proc. European Conference on Computer Vision*, volume 2350, pages 476–491, 2002.

[26] V. Franc. *Optimization algorithms for kernel methods*. PhD thesis, Centre for Machine Perception, Czech Technical University, 2005.

[27] William T. Freeman, William T. Freeman, Michal Roth, and Michal Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1994.

[28] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[29] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. European Conference on Computer Vision*, volume 1843, pages 37–49, 2000.

[30] D.M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *Pattern Analysis and Machine Intelligence*, 29(8):1408–1421, 2007.

[31] DM Gavrila, J. Giebel, S. Munder, D.C. Res, and G. Ulm. Vision-based pedestrian detection: the protector system. In *Proc. Intelligent Vehicles Symposium*, pages 13–18, 2004.

[32] DM Gavrila, J. Giebel, M. Perception, D.C. Res, and G. Ulm. Shape-based pedestrian detection and tracking. In *Proc. Intelligent Vehicle Symposium*, volume 1, pages 8–14, 2002.

[33] DM Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. International Conference on Computer Vision*, volume 1, pages 87–93, 1999.

[34] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *Proc. International Conference on Computer Vision*, pages 641–647, 2003.

[35] J. Ham, D.D. Lee, S. Mika, and B. Schoelkopf. A kernel view of the dimensionality reduction of manifolds. In *Proc. International Conference on Machine Learning*, 2004.

[36] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[37] H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[38] H. Hoffmann and R. Moller. Unsupervised learning of a kinematic arm model. In *Proc. Artificial Neural Networks and Neural Information Processing*, volume 2714, page 176, 2003.

[39] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *Proc European Conference on Computer Vision*, volume 2353, pages 343–357, 2002.

[40] M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[41] J.B. Kruskal and M. Wish. *Multidimensional scaling*. Sage Publications, Inc, 1978.

[42] J.T.Y. Kwok and I.W.H. Tsang. The pre-image problem in kernel methods. *IEEE Trans. Neural Networks*, 15(6):1517–1525, 2004.

[43] M.W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *Proc. Computer Vision and Pattern Recognition*, volume 2, pages 334–341, 2004.

[44] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 878–885, 2005.

[45] W. Li, Z. Zhang, and Z. Liu. Expandable data-driven graphical modeling of human actions based on salient postures. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1499, 2008.

[46] Z. Lin, L.S. Davis, D. Doermann, and D. DeMenthon. Hierarchical part-template matching for human detection and segmentation. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007.

[47] X. Mao, F. Qi, and W. Zhu. Multiple-part based pedestrian detection using interfering object detection. In *Proc. Natural Computation*, volume 2, pages 165–169, 2007.

[48] A. Micilotta, E. Ong, and R. Bowden. Detection and tracking of humans by probabilistic body part assembly. In *Proc. British Machine Vision Conference*, volume 1, pages 429–438, 2005.

[49] S. Mika, B. Schoelkopf, A. Smola, K.R. Muller, M. Scholz, and G. Ratsch. Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems*, 11:536–542, 1999.

[50] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[51] A. Mittal and L.S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *Proc. European Conference of Computer Vision*, volume 2350, pages 18–36, 2002.

[52] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[53] G. Mori, X. Ren, AA Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–326–II–333, 2004.

[54] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Schoelkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Networks*, 12(2):181–201, 2001.

[55] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[56] L. Panini and R. Cucchiara. A machine learning approach for human posture detection in domotics applications. In *Proc. International Conference on Image Analysis and Processing*, pages 103–108, 2003.

[57] C. Papageorgiou and T. Poggio. Trainable pedestrian detection. In *Proc. International Conference on Image Processing*, volume 4, pages 35–39, 1999.

[58] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[59] T.J. Roberts, S.J. McKenna, and I.W. Ricketts. Human pose estimation using learnt probabilistic region similarities and partial configurations. In *Proc. European Conference of Computer Vision*, pages 291–303, 2004.

[60] M.D. Rodriguez and M. Shah. Detecting and segmenting humans in crowded scenes. In *Proc. International conference on Multimedia*, pages 353–356, 2007.

[61] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[62] B. Schoelkopf, P. Knirsch, A. Smola, and C. Burges. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. *Mustererkennung*, 20:124–132, 1998.

[63] B. Schoelkopf, S. Mika, A. Smola, G. Ratsch, and K.R. Muller. Kernel pca pattern reconstruction via approximate pre-images. In *Proc. International Conference on Artificial Neural Networks, Perspectives in Neural Computing*, pages 147–152, 1998.

[64] B. Schoelkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12:582–588, 2000.

[65] E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *Proc Computer Vision and Pattern Recognition*, volume 2, pages 1582–1588, 2006.

[66] E. Seemann and B. Schiele. Cross-articulation learning for robust detection of pedestrians. *Pattern Recognition*, 4174:242–252, 2006.

[67] A. Shashua, Y. Gdalyahu, and G. Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *Proc. Intelligent Vehicles Symposium*, pages 1–6, 2004.

[68] V.D. Shet, J. Neumann, V. Ramesh, and L.S. Davis. Bilattice-based logical reasoning for human detection. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[69] H. Sidenbladh and M.J. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1):183–209, 2003.

[70] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:747–757, 2000.

[71] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. European Conference on Computer Vision*, volume 2, pages 589–600, 2006.

[72] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30:1713–1727, 2008.

[73] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[74] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. International Conference on Computer Vision*, volume 2, pages 734–741, 2003.

[75] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[76] X. Wang, T.X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Proc. International Conference on Computer Vision*, page (No page information), 2009.

[77] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *Proc. 30th DAGM symposium on Pattern Recognition*, volume 5096, pages 82–91, 2008.

[78] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proc. International Conference onComputer Vision*, volume 1, pages 90–97, 2005.

[79] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *Proc Computer Vision and Pattern Recognition*, volume 1, pages 17–22, 2006.

[80] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[81] C. Wu and S. Lai. Temporally integrated pedestrian detection from non-stationary video. In *Proc. International Conference on MultiMedia Modeling*, volume 4351, pages 188–197, 2007.

[82] Y. Wu and T. Yu. A field model for human detection and tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(5):753–764, 2006.

[83] L. Zhang, B. Wu, and R. Nevatia. Detection and tracking of multiple humans with extensive pose articulation. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007.

[84] T. Zhao and R. Nevatia. Stochastic human segmentation from a static camera. In *Proc. Motion and Video Computing*, pages 9–14, 2002.

[85] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *Proc. Computer Vision and Pattern Recognition*, volume 2, pages 459–466, 2003.

[86] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26:1208–1221, 2004.

[87] T. Zhao, R. Nevatia, and B. Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30:1198–1211, 2008.

[88] Q. Zhu, S. Avidan, M.C. Yeh, and K.T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. Computer Vision and Pattern Recognition*, volume 2, pages 1491–1498, 2006.