ABSTRACT

Title of dissertation: MODELING SHAPE, APPEARANCE AND MOTION FOR HUMAN MOVEMENT ANALYSIS

Zhe Lin, Doctor of Philosophy, 2009

Dissertation directed by: Professor Larry S. Davis
Department of Electrical and Computer Engineering

Shape, Appearance and Motion are the most important cues for analyzing human movements in visual surveillance. Representation of these visual cues should be rich, invariant and discriminative. We present several approaches to model and integrate them for human detection and segmentation, person identification, and action recognition.

First, we describe a hierarchical part-template matching approach to simultaneous human detection and segmentation combining local part-based and global shape-based schemes. For learning generic human detectors, a pose-adaptive representation is developed based on a hierarchical tree matching scheme and combined with an support vector machine classifier to perform human/non-human classification. We also formulate multiple occluded human detection using a Bayesian framework and optimize it through an iterative process. We evaluated the approach on several public pedestrian datasets.

Second, given regions of interest provided by human detectors, we introduce an approach to iteratively estimates segmentation via a generalized Expectation-

Maximization algorithm. The approach incorporates local Markov random field constraints and global pose inferences to propagate beliefs over image space iteratively to determine a coherent segmentation. Additionally, a layered occlusion model and a probabilistic occlusion reasoning scheme are introduced to handle inter-occlusion. The approach is tested on a wide variety of real-life images.

Third, we describe an approach to appearance-based person recognition. In learning, we perform discriminative analysis through pairwise coupling of training samples, and estimate a set of normalized invariant profiles by marginalizing likelihood ratio functions which reflect local appearance differences. In recognition, we calculate discriminative information-based distances by a soft voting approach, and combine them with appearance-based distances for nearest neighbor classification. We evaluated the approach on videos of 61 individuals under significant illumination and viewpoint changes.

Fourth, we describe a prototype-based approach to action recognition. During training, a set of action prototypes are learned in a joint shape and motion space via k-means clustering; During testing, humans are tracked while a frame-to-prototype correspondence is established by nearest neighbor search, and then actions are recognized using dynamic prototype sequence matching. Similarity matrices used for sequence matching are efficiently obtained by look-up table indexing. We experimented the approach on several action datasets.

MODELING SHAPE, APPEARANCE AND MOTION FOR HUMAN MOVEMENT ANALYSIS

by

Zhe Lin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Rama Chellappa
Professor John S. Baras
Professor David W. Jacobs
Dr. David S. Doermann

Acknowledgments

I would like to acknowledge all the people who have helped me throughout my graduate study and research during the last four years. Without their help and support, the thesis would not have been completed successfully.

First, I would like to thank my advisor, Professor Larry Davis, for accepting me as his PhD student and giving me invaluable opportunities to work on challenging and interesting projects during the PhD program. It has been my great honor to work under his supervision. He has given me a lot helpful suggestions, advices, and exciting ideas that inspired me in thinking and conquering difficulties that I have faced during my study and research.

Next, I would like to acknowledge help and support from my previous advisors, Dr. David Doermann and Dr. Daniel DeMenthon, who have been great mentors on my research. I would also like to thank my PhD dissertation committee: Professor Rama Chellappa, Professor John Baras, and Professor David Jacobs for agreeing to serve on my PhD thesis committee and for sparing their invaluable time reviewing the manuscript.

My colleagues at the Language and Media Processing Laboratory and Computer Vision Laboratory have enriched my graduate life in many ways and deserve a special mention. It was very productive and pleasant experience to work with Zhuolin Jiang and Weihua Zhang, two visiting scholars from China. My interaction with lab colleagues - Guangyu Zhu, Yi Li, Xu Liu, Xiaodong Yu, Yang Yu, Ming Luo, May Huang, Aniruddha Khembavi, Ryan Farrel, Abhinav Gupta, Vlad

Morariu, Mohamed Hussein, Behjat Siddiquie, and William Schwartz has been very fruitful.

Finally, I owe my deepest thanks to my family - my mother and father, and my wife Mei Jin who have always stood by me. Words cannot express the gratitude I owe them.

Table of Contents

List of Figures

Chapter 1

Introduction

Human movement analysis is a long-studied, but still important and challenging research area in visual surveillance. It involves many fundamental problems in computer vision such as human detection, segmentation and tracking, and higher level problems such as human gesture, action and event recognition. In computer vision, shape, appearance and motion have been the most-studied and widely-used visual cues for human movement analysis. Methods to effectively represent and integrate these cues with pattern classification techniques is crucial for analyzing human movements under challenging real-world situations.

1.1 Shape-based Human Detection

Human detection is the first step for analyzing human movements. It can provide an initialization for human segmentation. More importantly, robust human tracking and identification are highly dependent on reliable detection and segmentation in each frame, since better segmentation can be used to estimate more accurate and discriminative appearance models. Although the problem of human detection has been well-studied in vision, it still remains challenging due to highly articulated body postures, viewpoint changes, varying illumination conditions, occlusion, and background clutter. Combinations of these factors result in large variability of

human shapes and appearances in images. We present a shape-based hierarchical part-template matching approach and use it to derive an articulation-insensitive feature extraction method for pedestrian classification and generic human detection. We also extend the approach to detect and segment multiple occluded humans by an iterative occlusion analysis. A preliminary version of this approach has been published in [66, 67].

1.2 Appearance-based Human Segmentation

In video surveillance, people often appear in small groups, which yields occlusion of appearances due to the projection of the 3D world to 2D image space. Given initial detections, in order to track people or to recognize them based on their appearances, it would be useful to be able to accurately segment the groups into individuals and build their appearance models. The problem is to segment images into foreground and background, and further to segment the foreground regions into individuals. When the humans of interest and the background have similar color or texture, when the humans are in a cluttered background, or when humans appear under occlusion, the segmentation problem becomes especially challenging. We present an iterative approach to appearance-based human segmentation by incorporating both local and global shape constraints. A preliminary version of this approach has been published in [68].

1.3 Appearance-based Person Recognition

Appearance-based, full-body person recognition is closely related to object recognition, and very important for understanding human movements/activities in video surveillance. Appearance information is crucial not only in tracking, but also for identifying persons across space, time, and cameras. Pose articulation, viewpoint variation, and illumination change are common factors which affect the performance of appearance recognition systems. Also, when the number of people increases, ambiguities between them become significant, and consequently, more and more sophisticated and discriminative approaches are needed. We present an approach to improve the scalability of appearance recognition systems to larger number of individuals by exploring both intra-class and inter-class invariance in a pairwise comparison framework. A preliminary version of this approach has been published in [65].

1.4 Prototype-based Action Recognition

Action recognition is also an important problem in vision and has many potential applications such as human-computer interaction, virtual reality and multimedia retrieval. Frame-to-frame matching and nearest neighbor classification-based schemes have been standard for action recognition. However, for large-scale action recognition, where the training database consists of thousands of action videos, such a matching scheme may require tremendous amount of computation due to exhaustive distance computation between a test action frame and all training action frames.

In contrast to previous work which assumes static backgrounds, recognizing actions viewed against a dynamic varying background is another important challenge. We present a very accurate and efficient approach to action recognition based on action prototypes learned in a joint shape and motion space.

1.5 Organization of the Thesis

This thesis is organized as follows. In Chapter 2, we introduce our shape-based generic human detection approach and its extension to multiple occluded human detection. In Chapter 3, we describe our iterative pose-assisted and appearance-based segmentation approach. In Chapter 4, we present our appearance-based person recognition approach. In Chapter 5, we address our approach to combine shape and motion cues for efficient and accurate action recognition. Finally, in Chapter 6, we conclude the thesis and discuss possible future extensions.

Chapter 2

Shape-based Human Detection

2.1 Introduction

Our approach to human movement modeling and recognition, describing subsequently in chapter 2 and chapter 3 involves first detecting and approximately segmenting people in each frame of a video. In this chapter, we discuss our approach to human detection. There has been a significant amount of prior research on the problem of human detection. These previous approaches can be classified into two categories: shape-based approaches and blob-based approaches. Shape-based approaches can be used for human detection in either still images or videos. Shapes have been modeled as local curve segments in [34, 82, 127, 129], modeled directly as a global shape model hierarchy in [39, 40, 138], or implicitly represented by local or global descriptors in [22, 61, 71, 98, 131]. For highly articulated objects like humans, part-based representations have been shown to be very efficient for detection. For example, Mikolajczyk *et al.* [71] use local features for part detection and assemble the part detections probabilistically. Wu and Nevatia [127] introduce edgelet features for human detection. They extend this approach to a general object detection and segmentation approach by designing local shape-based classifiers [129]. Shet *et al.* [102] propose a logical reasoning-based method for efficiently assembling part detections. One problem with these part-based detection approaches is that in very

cluttered images too many detection hypotheses may be generated, and a robust assembly method (*e.g.* boosting) is thus needed to combine these detections. On the other hand, Gavrila and Philomin [39, 40, 138] use on a more direct hierarchical template matching approach for global shape-based pedestrian detection. These shape-based detection methods can also be combined with appearance cues for simultaneous detection and segmentation [58, 126, 138]. Shape-based approaches have the advantage that they do not require background subtraction, but they need to scan whole images and can generate many false alarms in cluttered regions.

From a learning perspective, many of these shape-based approaches model human detection as a binary classification problem and rely on sliding-window scanning schemes. These approaches can be further divided into two categories in terms of shape modeling schemes. The first category models human shapes globally or densely over image locations, *e.g.* an over-complete set of Haar wavelet features in [83], rectangular features in [115], histograms of oriented gradients (HOGs) in [22], locally deformable Markov models in [131] or covariance descriptors in [113]. Global approaches such as [22, 113] are designed to tolerate certain degrees of occlusions and shape articulations with a large number of samples and have been demonstrated to achieve excellent performance with well-aligned, more-or-less fully visible training data. The second category of approaches uses local feature-based approaches to learn body part and/or full-body detectors based on sparse interest points and descriptors as in [61, 71], from predefined pools of local curve segments [82, 106], k-adjacent segments [33], or edgelets [127]. In [75], several part detectors are trained separately for each body part, and combined with a second-level classifier. Com-

pared to the global approaches, part (or local feature)-based approaches [61, 127] are more adept in handling partial occlusions, and flexible in dealing with shape articulations. Shape cues are also combined with motion cues for human detection in [23,116], simultaneous detection and segmentation in [100]. Dalal and Triggs [22] introduced HOG features and provided an extensive experimental evaluation using linear and gaussian-kernel SVMs as the test classifiers. Later, Zhu *et al.* [142] improved its computational efficiency significantly by utilizing a boosted cascade of rejectors. Recently, Tuzel *et al.* [113] reported better detection performance than [22] on the INRIA dataset. They use covariant matrices as image descriptors and classify patterns on Riemannian manifolds. Similarly, Maji *et al.* [70] also demonstrate promising results using multi-level HOG descriptors and faster (histogram intersection) kernel SVM classification. In [92], two-fold adaboost classifiers are adopted for simultaneous part selection and pedestrian classification. Ref. [128] combines different features in a single classification framework.

In contrast, blob-based approaches are computationally more efficient but have a common problem that the results depend crucially on background subtraction or motion segmentation. These approaches are mostly developed for detecting and tracking humans under occlusion. Some earlier methods [50, 110] model the human tracking problem by a multi-blob observation likelihood given a human configuration. Zhao and Nevatia [141] introduce an MCMC-based optimization approach to human segmentation from foreground blobs. They detect heads by analyzing edges surrounding binary foreground blobs, formulate the segmentation problem in a Bayesian framework, and optimize by modeling jump and diffusion dynamics

in MCMC to traverse the complex solution space. Following this work, Smith *et al.* [108] propose a similar trans-dimensional MCMC model to track multiple humans using particle filters. Later, an EM-based approach was proposed by Rittscher *et al.* [88] for foreground blob segmentation. Zhao and Tao [140] use a part-based human body model to fit binary blobs and track humans.

Few of the previous approaches explicitly model human shape articulations using part model deformations, and/or formulate human detection by unifying shape and region (or motion blob) information in a single probabilistic framework. Hierarchical template matching [39,40] is a convenient way to efficiently integrate detection and segmentation of shapes, but it is computationally expensive due to the necessity of collecting and matching with a large number of global shape templates. Moreover, previous discriminative approaches mostly train a binary classifier on a large number of positive and negative samples where humans are roughly center-aligned. These approaches represent appearances by concatenating information along 2D image coordinates for capturing spatially recurring local shape events in training data. However, due to highly articulated human poses and varying viewing angles, a very large number of (well-aligned) training samples are required; moreover, the inclusion of information from whole images inevitably makes them sensitive to biases in training data (in the worst case, significant negative effects can occur from arbitrary image regions), consequently the generalization capability of the trained classifier can be compromised.

We introduce a hierarchical part-template matching approach [67] for detecting and segmenting humans simultaneously. The approach takes advantages of both

local part-based and global template-based human detectors by decomposing global shape models and constructing a part-template tree to model human shapes flexibly and efficiently. Shape observations (edges or local gradient orientations) are matched to the part-template tree efficiently to determine a reliable set of human detection hypotheses. Shapes and poses are estimated automatically through synthesis of part detections.

Using the hierarchical part-template matching scheme, we extract features adaptively in the local context of poses, i.e. we propose a pose-invariant feature extraction method [66] for for better discriminating humans from non-humans. The intuition is that pose-adapted features produce much better spatial repeatability of local shape events. Specifically, we segment human poses on both positive and negative samples[1] and extract features adaptively in local neighborhoods of pose contours, *i.e.* in the pose context. The set of all possible pose instances are mapped to a canonical pose, such that points on an arbitrary pose contour have one-to-one correspondences to points in the canonical pose. This ensures that our extracted feature descriptors correspond well to each other, and are also invariant to varying poses.

For multiple occluded human detection problems, a set of detection hypotheses are estimated by our generic human detector and is iteratively optimized under a Bayesian MAP framework based on global likelihood re-evaluation and fine occlusion analysis. For meeting the requirement of real-time surveillance systems, we also combined the approach with background subtraction to improve efficiency, where

[1]For negative samples, pose estimation is forced to proceed even though no person is in them.

region information provided by foreground blobs is combined with shape information from the original image in a joint likelihood model.

Our main contributions are summarized as follows:

- A part-template tree model and its automatic learning algorithm are introduced for simultaneous human detection and pose segmentation. The approach combines popular local part-based object detectors with global shape template-based schemes.

- A fast hierarchical part-template matching algorithm is used to estimate human shapes and poses by matching local image cues such as gradient magnitudes and/or orientations. Human shapes and poses are represented by part-based parametric models, and the estimation problem is formulated and optimized in a probabilistic framework.

- Estimated optimal poses are used to impose spatial priors (for possible humans) for encoding pose-invariant features in nearby local pose contexts. One-to-one correspondence is established between sets of contour points of an arbitrary pose and a canonical pose.

- A Bayesian MAP framework is utilized to formulate and solve multiple occluded human detection and segmentation problems. Optimization is performed in a greedy fashion composed of iterative processes of global likelihood re-evaluation and fine occlusion analysis.

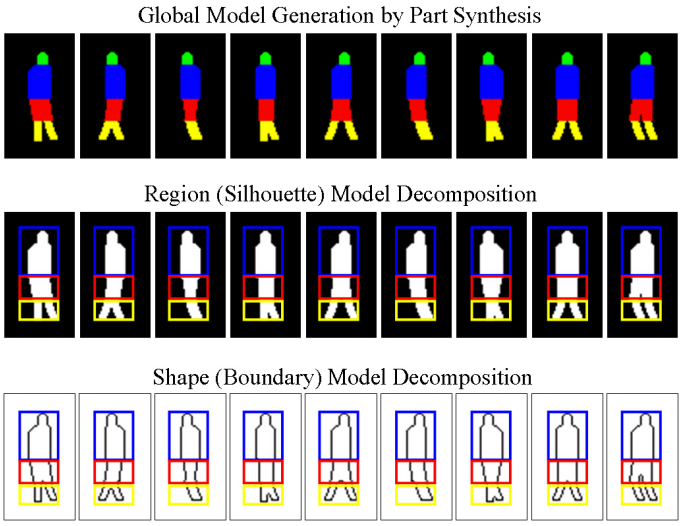


Figure 2.1: Generation of global shape models by part synthesis, decomposition of global silhouette and boundary models into region and shape part-templates.

2.2 Hierarchical Part-Template Matching

We take advantages of local part-based and global shape template-based approaches by combining them in a unified top-down and bottom-up search scheme. Specifically, we extend the hierarchical template matching method in [39, 40] by decomposing the global shape models into parts and constructing a new part template-based tree which captures appearance correlations between part models from the training database of human shapes.

2.2.1 Tree-Structured Part-Template Hierarchy

We generate a flexible set of global shape models by part synthesis using a simple pose generator and construct a part-template hierarchy using a body-part decomposer. Some examples of global generated global shape models are shown in Figure 2.1. For modeling human side views and front/back views individually, we

represent the body with six part regions - (head, torso, pair of upper-legs, pair of lower-legs). Each part region is modeled by a horizontal parallelogram (five degrees of freedom) characterized by its position, size and orientation parameters. Thus, the total number of degrees of freedom is $5 \times 6 = 30$. For initial tree construction, global shapes are modeled using only six degrees of freedom (head position, torso width, orientations of upper/lower legs) given the torso position as the reference, and other parameters can treated as hidden variables estimated only in online reasoning phases. Heads and torsos are simplified to vertical rectangles (fixed orientations) with rounded shapes at corners. The selected six parameters are discretized into $\{3, 2, 3, 3, 3, 3\}$ values. Finally, the part regions are independently collected and grouped to form $3 \times 2 \times 3 \times 3 \times 3 \times 3 = 486$ global shape models.

Next, silhouettes and boundaries are extracted from the set of generated global shape models and decomposed into three parts (head-torso, upper legs and lower legs) as shown in Figure 2.1. The parameters of the three parts ht, ul, ll are denoted as θ_{ht}, θ_{ul} and θ_{ll}, where each parameter represents the index of the corresponding part in the part-template tree. Then, the tree-structured part-template hierarchy is constructed by placing the decomposed part regions and boundary fragments into a tree as illustrated in Figure 2.2. The tree has four layers denoted as L_0, L_1, L_2, L_3, where L_0 is the (empty) root node, L_1 consists of side-view head-torso templates $L_{1,i}, i = 1, 2, 3$ and front/back-view head-torso templates $L_{1,i}, i = 4, 5, 6$, and similarly, L_2 and L_3 consists of upper and lower leg poses for side and front/back views. Hence, Each part in the tree can be viewed as a parametric model, where part location and sizes are the model parameters. As shown in the figure, the tree consists

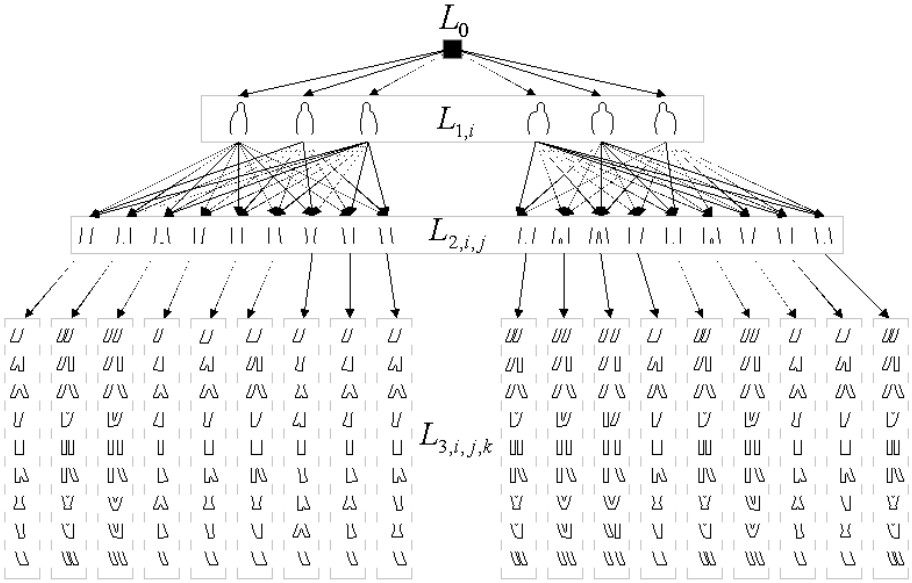


Figure 2.2: An illustration of the part-template tree model. Each part in the tree is characterized by both shape and region information.

of 186 part-templates, *i.e.* 6-12 head-torso (ht) models, 18 upper-leg (ul) models, 18 lower-leg (ll) models, and organized hierarchically based on the layout of human body parts in a top-to-bottom manner. Due to the tree structure, a fast hierarchical shape (or pose) matching scheme can be applied using the model. For example, using hierarchical part-template matching (which will be explained later), we only need to match 24 part-templates to account for the complexity of matching 486 global shape models using the method in [40], so it is extremely fast.

2.2.2 Learning the Part-template Tree

In order to more efficiently and reliably estimate human shapes and poses in the image, we learn the part-template tree model in Figure 2.2 and embed its hierarchical matching algorithm in a probabilistic optimization framework. The

learning is done by matching the tree to a set of annotated human silhouette images. Specifically, we estimate the distributions of part model parameters in each of the tree layers for handling a wider range of articulations of people.

We learn the part-template tree model in Figure 2.2 based on a training set of about 800 320×240 binary silhouette images (white foreground and black background). Each of the training silhouette images is sent through the tree from the root node to leaf nodes and the degree of coverage (both foreground and background) consistency between each part template $T_{\theta_j}, j \in \{ht, ul, ll\}$ and the observation is measured. Here, each part-template is considered to be covered by a binary rectangular image patch M (see Figure 2.5(b) for an example). The degree of coverage consistency $\rho(\theta_j|S)$ between a part-template T_{θ_j} and a silhouette image S is defined as the pixel-wise similarity of the part-template coverage image $M(\theta_j)$ and the binary sub-silhouette S_j (corresponding to the same region as the part-template), *i.e.*

$$\rho(\theta_j|S) = 1 - \frac{\sum_{\mathbf{x}} |S_i(\mathbf{x}) - M(\theta_j, \mathbf{x})|}{n}, \tag{2.1}$$

where n is the total number of pixels in the rectangular part-template region. Then, we can estimate the best set of part models $\theta^* = \{\theta_j^*\}$ for the training silhouette S by maximum likelihood estimation:

$$\theta_j^* = \arg\max_{\theta_j \in \mathbf{\Theta}_j} \rho(\theta_j|S), \tag{2.2}$$

where $\mathbf{\Theta}_j$ denotes the set of all possible part template parameters. This process is

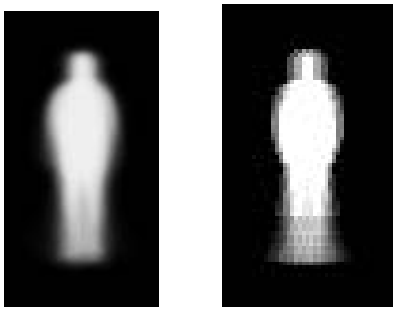


Figure 2.3: A comparison of the average of all training silhouettes (left) and the average of our 486 learned global shape models (right).

repeated for all training silhouettes and the ranges of part template models are estimated based on the statistics of each part-template's model parameters. The ranges of parameters are evenly quantized to produce the final tree model.[2] Figure 2.3 validates our tree learning approach by showing that the average of our learned global shape models (composition of parts) is very similar to the average of all training silhouettes.

2.2.3 Object Likelihood Model

We formulate the pose and shape estimation problem probabilistically as maximization of a global (pseudo) object likelihood L. In order to quickly evaluate the likelihood for a global pose (*i.e.* different parameter combinations of part models), the object likelihood is simply modeled as a *summation* of matching scores of part-template models in all tree layers. We can think of L as a log-likelihood and the summation of the matching scores over different parts is equivalent to multiplication of probabilities. Given an image I (either training or testing sample) and a candi-

[2]The learned tree model can be downloaded from `http://terpconnect.umd.edu/~zhelin/part-template-model.zip`.

date global pose model $\theta = \{\theta_j\}$ (including part-template indices and their locations and scales), in the simplest case, if we assume independence between part-template models θ_j in different layers, the object likelihood can be simply represented as follows:

$$L(\theta|I) = L(\theta_{ht}, \theta_{ul}, \theta_{ll}|I) = \sum_{j \in \{ht,ul,ll\}} L(\theta_j|I). \tag{2.3}$$

For the purpose of pose estimation, we should jointly consider different parts θ_j for optimization of L. Hence, based on the layer structure of the tree in Figure 2.1, the likelihood L is decomposed into conditional likelihoods as follows:

$$\begin{aligned} L(\theta|I) &= L(\theta_{ht}|I) + L(\theta_{ul}|\theta_{ht}, I) + L(\theta_{ll}|\theta_{ht}, \theta_{ul}, I) \\ &= L(\theta_{ht}|I) + L(\theta_{ul}|\theta_{ht}, I) + L(\theta_{ll}|\theta_{ul}, I) \\ &= L(\theta_{ht}|I) + L(\theta_{ul}|I)R(\theta_{ul}, \theta_{ht}) + L(\theta_{ll}|I)R(\theta_{ll}, \theta_{ul}), \end{aligned} \tag{2.4}$$

where $R(a, b) = 1$ if a is a descendent of b, $R(a, b) = 0$ otherwise, and the decomposition is performed in a top-to-bottom order of the layers, and independence is assumed between the two non-joining layers, ht and ll. We use Equation 2.4 as our optimization model discussed in the following.

2.2.4 Part Likelihood Model

A part template T_{θ_j} (defined by model parameters θ_j) is characterized by its boundary curve segments (see Figure 2.1) and edge orientations (or normal directions) along the boundary segments. We match individual part-templates using a

method similar to Chamfer matching [40]. Matching scores of each sample point along the part-template contour can be measure from different cues such as distance transforms or dominant edge orientations.

More formally, the likelihood $L(\theta_j(\mathbf{x}, s)|I)$ of a part template-T_{θ_j} at location $\mathbf{x}$ and scale s is modeled as follows:

$$L(\theta_j(\mathbf{x}, s)|I) = \frac{1}{|T_{\theta_j}|} \sum_{\mathbf{t} \in T_{\theta_j}} d_I^{'}(\mathbf{x} + s\mathbf{t}), \tag{2.5}$$

where $|T_{\theta_j}|$ denotes the length of the part-template, and $\mathbf{t}$ denotes the relative position of individual contour points along the template. Exact models of distances d' and part-template likelihoods are discussed in the next section.

2.2.5 Optimization

The structure of our part-template model and the form (summation) of the global object likelihood L suggest that the optimization problem can be solved by dynamic programming to achieve globally optimal solutions. But, this algorithm is computationally too expensive for dense scanning of all windows for detection. For efficiency, we perform the optimization, *i.e.* the maximization of L, by a fast k-fold greedy search procedure. Algorithm 1 illustrates the overall matching (optimization) process. We keep scores for all nodes ($k = 1, 2...K$) in the second layer (*i.e.* the torso layer) instead of estimating the best k in step 1 of the algorithm. In the following steps, a greedy procedure is individually performed for each of those K nodes (or threads).

Algorithm 1: Probabilistic Hierarchical Part-Template Matching

1) For a set of locations $\mathbf{x}$ and scales s, match all K head-torso part-templates in layer L_1 with the image and compute their part-template likelihoods $L(\theta_{ht}^k(\mathbf{x}, s)|I), k = 1, 2...K$.
2) For $k = 1...K$, repeat the following steps (3)-(4), and select $k = k^*$ and $\theta = \theta^*$ with the maximum $L(\theta|I)$.
3) According to the part-template model θ_{ht}^k of Layer L_1, estimate the maximum conditional-likelihood leg models $\theta_{ul}^*|\theta_{ht}^k$ in L_2 and $\theta_{ll}^*|\theta_{ul}^*, \theta_{ht}^k$ in L_3 using a greedy search algorithm along the tree.
4) Given the above part-template's model estimates, compute the current global object likelihood based on Equation 2.4.
5) Return the global pose model estimates $\theta^* = \{\theta_{ht}^k, \theta_{ul}^*, \theta_{ll}^*\}$.

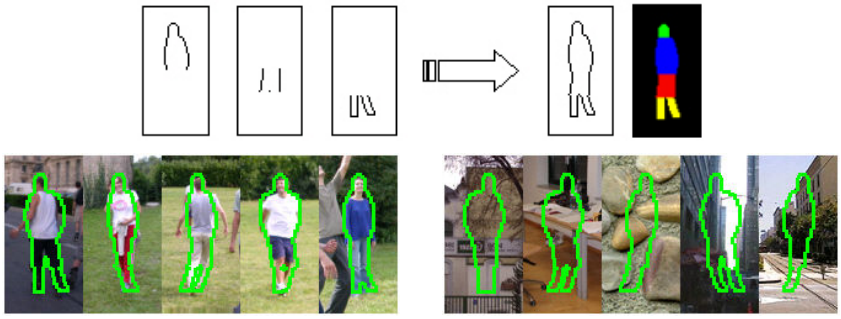


Figure 2.4: An illustration of shape/pose segmentation. Top: Best part-template estimates (three images on the left side designated by a path from L_0 to L_3) are combined to produce final global shape and pose segmentations (two images on the right side); Bottom: example pose (shape) segmentation on positive and negative examples.

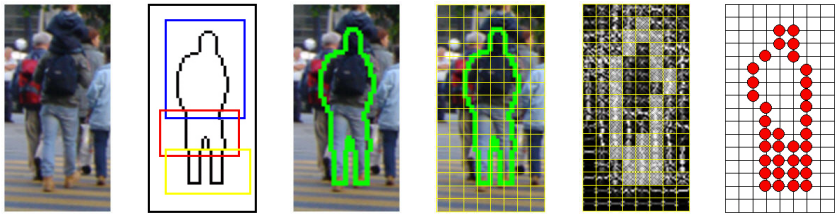
Pose model parameters estimated by the hierarchical part-template matching algorithm are directly used for pose segmentation by part-synthesis (region connection). Figure 2.4 shows the process of global pose (shape) segmentation by the part-template synthesis.

2.3 Pose-Adaptive Image Description

For applying our part-template tree model and hierarchical matching algorithm to discriminative human detection, we introduce a pose-adaptive feature computation method for detecting humans from images using standard machine learning techniques such as SVMs and Adaboost.

2.3.1 Overview of the Approach

In our training and testing datasets, training and testing samples all consist of 128×64 image patches. Negative samples are randomly selected from raw (person-free) images, positive samples are cropped (from annotated images) such that persons are roughly aligned in location and scale. For each training or testing sample, we first compute a set of histograms of (gradient magnitude-weighted) edge orientations for non-overlapping 8×8 rectangular regions (or cells) evenly distributed over images. Motivated by the success of HOG descriptors [22] for object detection, we employ coarse-spatial and fine-orientation quantization to encode the histograms, and normalization is performed on groups of locally connected cells, *i.e.* blocks. Then, given the orientation histograms, the probabilistic hierarchical part-template matching technique is used to estimate shapes and poses based on an efficient part-based synthesis approach under a probabilistic framework. Given the pose and shape estimates, block features closest to each pose contour point are collected; finally, the histograms of the collected blocks are concatenated in the order of pose correspondence to form our feature descriptor. As in [22], each block (con-



(a) (b) (c) (d) (e) (f)

Figure 2.5: Overview of our feature extraction method. a) A training or testing image, b) Part-template detections, c) Pose and shape segmentation, d) Cells overlaid onto pose contours, e) Orientation histograms and cells overlapping with the pose boundary, f) Block centers relevant to the descriptor.

sisting of 4 histograms) is normalized before collecting features to reduce sensitivity to illumination changes. The one-to-one point correspondence from an arbitrary pose model to the canonical one reduces sensitivity of extracted descriptors to pose variations. Figure 4.1 shows an illustration of our feature extraction process.

2.3.2 Low-Level Feature Representation

For pedestrian detection, histograms of oriented gradients (HOG) [22] exhibited superior performance in separating image patches into human/non-human. These descriptors ignore spatial information locally, hence are very robust to small alignment errors. We use a very similar representation as our low-level feature description, *i.e.* (gradient magnitude-weighted) edge orientation histograms.

Given an input image $\mathbf{I}$, we calculate gradient magnitudes $|G_{\mathbf{I}}|$ and edge orientations $O_{\mathbf{I}}$ using simple difference operators $(-1, 0, 1)$ and $(-1, 0, 1)^t$ in horizontal-x and vertical-y directions, respectively. We quantize the image region into local 8×8 non-overlapping cells, each represented by a histogram of (unsigned) edge orientations (each surrounding pixel contributes a gradient magnitude-weighted vote to

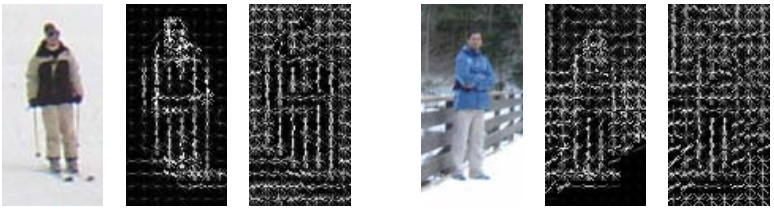


Figure 2.6: Examples of two training samples and visualization of corresponding (un-normalized and L_2-normalized) edge orientation histograms.

the histogram bins). Edge orientations are quantized into $N_b = 9$ orientation bins $[k\frac{\pi}{N_b}, (k+1)\frac{\pi}{N_b})$, where $k = 0, 1...N_b - 1$. For reducing aliasing and discontinuity effects, we also use trilinear interpolation as in [22] to vote for the gradient magnitudes in both spatial and orientation dimensions. Additionally, each set of neighboring 2×2 cells form a block. This results in overlapping blocks where each cell is contained in multiple blocks. For reducing illumination sensitivity, we normalize the group of histograms in each block using L_2 normalization with a small regularization constant ϵ to avoid dividing-by-zero. Figure 2.6 shows example visualizations of our low-level HOG descriptors.

The above computation results in our low-level feature representation consisting of a set of raw (cell) histograms (gradient magnitude-weighted) and a set of normalized block descriptors indexed by image locations. As will be explained in the following, both unnormalized cell histograms and block descriptors are used for inferring poses and computing final features for detection.

2.3.3 Computing Part-Template Likelihoods

Given the low level feature representations, the part template likelihoods are measured by magnitudes of corresponding orientation bins in local edge orientation histograms. The matching scores are measured using location-based look-up tables for speed. Magnitudes from neighboring histogram bins are weighted to reduce orientation biases and to regularize the matching scores of each template point.

Suppose the dominant orientation around contour point $\mathbf{t}$ is $O(\mathbf{t})$, its corresponding orientation bin index $B(\mathbf{t})$ is computed as: $B(\mathbf{t}) = [O(\mathbf{t})/(\pi/9)]$ ($[x]$ denotes the maximum integer less-or-equal to x), and the un-normalized (raw) orientation histogram at location $(\mathbf{x} + s\mathbf{t})$ is $H = \{h_i\}$. Then, the individual matching score $d_I^{'}$ at contour point $\mathbf{t}$ is expressed as:

$$d_I^{'}(\mathbf{x} + s\mathbf{t}) = \sum_{b=-\delta}^{\delta} w(b) h_{B(\mathbf{t})+b}, \tag{2.6}$$

where δ is a neighborhood range, and $w(b)$ is a symmetric weight distribution[3].

2.3.4 Representation using Pose-Invariant Descriptors

In our implementation, the global shape models (consisting of 3 part-template types) are represented as a set of boundary points with corresponding edge orientations. The range of the number of those model points are from 118 to 172. In order to obtain a unified (constant dimensional) description of images with those different dimensional pose models, and to establish a one-to-one correspondence be-

[3]For simplicity, we use $\delta = 1$, and $w(1) = w(-1) = 0.25, w(0) = 0.5$ in our experiments.

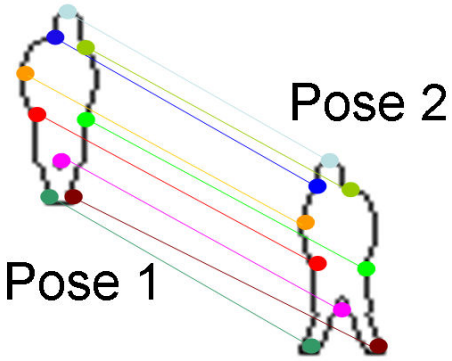


Figure 2.7: An illustration of pose alignment by one-to-one contour point correspondence. Only a subset of key contour points are shown here.

tween contour points of different poses (Figure 2.7), we map the boundary points of any pose model to those of a canonical pose model. The canonical pose model is assumed to be occlusion-free, so that all contour points are visible. For human upper bodies (heads and torso), the boundaries are uniformly sampled into 8 left side and 8 right side locations; and the point correspondence is established between poses based on vertical y coordinates and side (left or right) information. For lower bodies (legs), boundaries are uniformly sampled into 7 locations vertically with 4 locations at each y value (inner leg sample points are sampled at 5 pixels apart from outer sample points in the horizontal direction). Figure 2.5(f) shows an example of how the sampled locations are distributed).

Associated with each of those sample locations is a 36-dimensional feature vector (L_2-normalized histogram of edge orientations of its closet 2×2 block in the image). Hence, this mapping procedure generates a $(8 \times 2 + 7 \times 4) \times 36 = 1584$ dimensional feature descriptor. Figure 4.1 illustrates the feature extraction method. Note that only a subset of blocks are relevant for the descriptor, and a block might

be duplicated several times based on the frequency of contour points lying inside the block.

2.4 Detecting and Segmenting Multiple Occluded Humans

Pose-invariant descriptors discussed in the previous section are mainly developed for the purpose of detecting fully visible humans from images. However, real world images can be crowded and it is common that humans can occlude each other significantly. This is more obvious in visual surveillance scenarios where videos are usually captured in crowded public places, *e.g.* shopping malls, airports, etc. In these complex cases, our generic detector based on our pose-adaptive features should be used to provide initial sets of human hypotheses (by reducing thresholds to ensure low miss rates) and then more detailed occlusion analysis and optimization should be performed. Below, we introduce a unified Bayesian framework for detecting and segmenting multiple occluded humans in still images and videos.

2.4.1 Bayesian Problem Formulation

We model the detection and segmentation problem as a Bayesian MAP optimization:

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} P(\mathbf{c}|I), \tag{2.7}$$

where I denotes the image observation, $\mathbf{c} = \{\mathbf{h}_1, \mathbf{h}_2, ...\mathbf{h}_n\}$ denotes a human configuration (a set of human hypotheses), and n denotes the number of humans in the configuration. $\{\mathbf{h}_i = (\mathbf{x}_i, \theta_i)\}$ is an individual hypothesis which consists of

foot position[4] $\mathbf{x}_i$ and corresponding human model parameter θ_i. Using Bayes Rule, Equation 2.7 can be decomposed into a joint likelihood $P(I|\mathbf{c})$ and a prior $P(\mathbf{c})$ as follows:

$$P(\mathbf{c}|I) = \frac{P(I|\mathbf{c})P(\mathbf{c})}{P(I)} \propto P(I|\mathbf{c})P(\mathbf{c}). \tag{2.8}$$

For human detection, we assume a uniform prior, hence the MAP problem reduces to maximizing the joint likelihood. Note that the prior is non-uniform and should be modeled based on previous states in tracking problems.

Previous approaches [50, 110, 141] model the human detection and tracking problem by a multi-blob observation likelihood based on object-level and configuration-level likelihood. In [127], the joint likelihood is modeled as the probability of part-detection responses given a set of human hypotheses. We decompose the image observation, I, into shape observation I_s (edge image or edge orientation histograms) and region observation I_r (binary foreground image from background subtraction) assuming independence between the shape and region information. Then, the joint likelihood $P(I|\mathbf{c})$ is modeled as:

$$P(I|\mathbf{c}) = P(I_s|\mathbf{c})P(I_r|\mathbf{c}), \tag{2.9}$$

where $P(I_s|\mathbf{c})$ and $P(I_r|\mathbf{c})$ denote shape likelihood and region likelihood respectively. The region observation is optional and we set $P(I_r|\mathbf{c}) = 1$ or equivalently $P(I|\mathbf{c}) = P(I_s|\mathbf{c})$ when background subtraction is not used.

[4]Here, we choose the foot point as a reference to represent and search for human shapes. A foot point is defined as the bottom center point of a human bounding box.

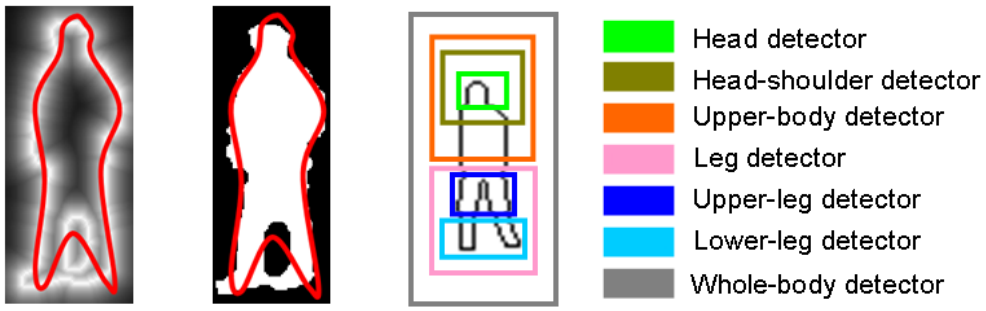


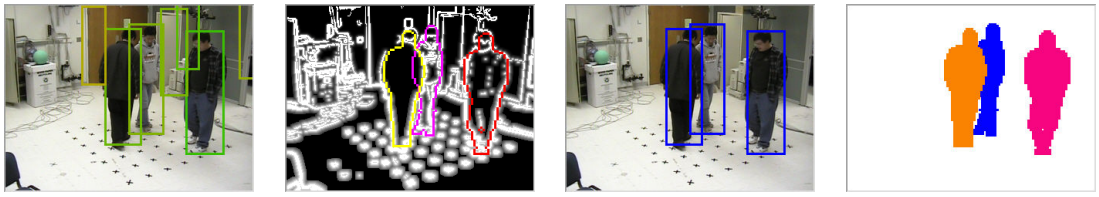
Figure 2.8: An illustration of part-template likelihood computed using multiple cues. (a) Shape information is measured by Chamfer matching, (b) region information is measured by foreground coverage density, (c) Part detectors.

2.4.2 Extended Part-Template Likelihood Model

In the multiple cue framework, a part-template is characterized by its boundary and coverage region. We match individual part-templates using both shape and region information (when region information is available from background subtraction). Shape information is measured by chamfer matching and region information is measured by part foreground coverage density. Figure 2.8(a) and 2.8(b) show how shape and region information is measured.

For a foot candidate pixel $\mathbf{x}$ in the image, the likelihood $P(I|\mathbf{x}, \theta_j)$ for a part template-T_{θ_j}, $j \in \{ht, ul, ll\}$ is decomposed into the part-shape likelihood $P(I_s|\mathbf{x}, \theta_j)$ and the part-region likelihood $P(I_r|\mathbf{x}, \theta_j)$ as follows:

$$P(I|\mathbf{x}, \theta_j) = P(I_s, I_r|\mathbf{x}, \theta_j) = P(I_s|\mathbf{x}, \theta_j)P(I_r|\mathbf{x}, \theta_j). \tag{2.10}$$



(a) (b) (c) (d)

Figure 2.9: An example of detection process without background subtraction. (a) Initial set of human detection hypotheses, (b) Human shape segmentations, (c) Detection result, (d) Segmentation result (final occlusion map).

This can be transformed to the form of log-likelihoods as:

$$L(\theta_j(\mathbf{x})|I) = L(\theta_j(\mathbf{x})|I_s) + L(\theta_j(\mathbf{x})|I_r). \tag{2.11}$$

where $L(\theta_j(\mathbf{x})|I_s)$ can be directly obtained by the method discussed earlier in Section 2.2 or by Chamfer matching, *i.e.* the likelihood can be computed either from edge orientation histograms or distance transforms of canny edge maps. If region (or blob) information is available, the part region likelihood is computed by the part foreground coverage density $\gamma(\mathbf{x}, \theta_j)$ which is defined as the proportion of the foreground pixels covered by the part-template T_{θ_j} at pixel $\mathbf{x}$, otherwise, we set $P(I_r|\mathbf{x}, \theta_j) = 1$ (or equivalently $L(\mathbf{x}(\theta_j)|I_r) = 0$).

2.4.3 Generating Initial Human Hypotheses

The generic human detectors trained using our pose-adaptive features and SVM classifier can provide reliable sets of initial human hypotheses for detecting humans from still images. However, for crowded videos, a set of simpler part detectors can be more accurate than a single full body detector due to severe occlusion

between humans. Hence, here we introduce an alternative method for generating initial human hypotheses for surveillance scenarios.

Hierarchical part-template matching provides estimates for the model parameters $\theta^*(\mathbf{x})$ for every foot candidate pixel $\mathbf{x}$ in the image. We define a likelihood function for evaluating likelihood for any part or part combinations. The object-level likelihood function $L(\mathbf{x}|I)$ for foot candidate pixel $\mathbf{x}$ is expressed as:

$$L^{\mathbf{w}}(\mathbf{x}|I) = \sum_j w_j L(\theta_j^*(\mathbf{x})|I), \tag{2.12}$$

where $\mathbf{w} = \{w_j, j = ht, ul, ll\}$ is an importance weight vector to calculate a likelihood value for different parts or part combinations. For example, $\{w_{ht} = w_{ul} = w_{ll} = 1/3\}$ corresponds to a full body detector and $\{w_{ht} = 0, w_{ul} = w_{ll} = 1/2\}$ corresponds to a leg detector. The importance weights are normalized to satisfy $\sum_j w_j = 1$. We have seven part or part-combination detectors (Figure 2.8(c), and if the head-torso is decomposed further into head-shoulder and torso, the number of detectors can be as high as 15. Suppose we use K part detectors, $D_k, k = 1, 2...K$ corresponding to K weight vectors $\mathbf{w}_k, k = 1, 2...K$ for each foot candidate pixel $\mathbf{x}$ in the image. The likelihoods for these part detectors are calculated with the object-level likelihood function (Equation 2.12).

In practice, we can use our generic human detector to reduce the search space into a small subset and boost it by searching additional hypotheses using the above part detectors. We threshold each of the final likelihood maps generated from the part detectors, merge nearby weak responses to strong responses and adaptively

select modes. This step can also be performed by local maximum selection after smoothing the likelihood image. The union of the maxima forms the set of human hypotheses:

$$O = \{o_1, o_2, ...o_N\} = \\ \{(\mathbf{x}_1, \theta^*(\mathbf{x}_1)), (\mathbf{x}_2, \theta^*(\mathbf{x}_2)), ...(\mathbf{x}_N, \theta^*(\mathbf{x}_N))\}, \tag{2.13}$$

and the corresponding likelihoods are denoted as $L(o_i), i = 1, 2...N$.

2.4.4 Optimization: Maximizing the Joint Likelihood

Suppose we have an initial set of human hypotheses $O = \{o_1, o_2, ...o_N\}$ obtained from hierarchical part template matching. The remaining task is to estimate its best subset through optimization. This is equivalent to maximizing the joint likelihood $P(I|\mathbf{c})$ (Equation 2.9) with respect to the configuration, $\mathbf{c}$.

2.4.4.1 Modeling the Joint Likelihood

If region information is not available, we set the region likelihood as $P(I_r|\mathbf{c}) = 1$, otherwise, it is calculated by the global coverage density of the binary foreground regions:

$$P(I_r|\mathbf{c}) = \frac{\Gamma(\mathbf{c})}{\Gamma_{fg}}, \tag{2.14}$$

where Γ_{fg} denotes the area of the foreground regions and $\Gamma(\mathbf{c})$ denotes the area of the foreground regions covered by the configuration $\mathbf{c}$. Intuitively, the more the

foreground is covered by the configuration $\mathbf{c}$, the higher the probability $P(I_r|\mathbf{c})$. Areas covered by the hypotheses and located outside the foreground regions are not penalized here, but considered in foot candidate region detection in Section 2.5.2. In fact, the region likelihood (Equation 2.14) has a bias towards more detections, but the bias is compensated for by the shape likelihood (Equation 2.15) (which involves a direct multiplication of individual likelihoods), since adding redundant hypotheses will decrease the shape likelihood.

The shape observation I_s now can be reduced to $o_1, o_2, ..., o_N$ since we only select the best subset from this initial set of hypotheses. This allows us to further decompose the shape likelihood as a product of likelihoods (assuming independence between each observation o_i given the configuration $\mathbf{c}$):

$$P(I_s|\mathbf{c}) = P(o_1, o_2, ..., o_N|\mathbf{c}) = \prod_{i=1}^{N} P(o_i|\mathbf{c}). \tag{2.15}$$

For evaluating the conditional probability $P(o_i|\mathbf{c})$, we need to model the occlusion status between different hypotheses in the configuration $\mathbf{c}$. For simplicity, we assume a known or fixed occlusion ordering for $\mathbf{c}$. Directly using the object-level likelihood $L(o_i)$ to model $P(o_i|\mathbf{c})$ will have problems since it only represents the strongest part response. We need to globally *re-evaluate* the object-level likelihood of each hypothesis o_i based on fine occlusion analysis; that is, we calculate the global shape likelihood only for the un-occluded parts when calculating the chamfer scores. This occlusion compensation-based likelihood re-evaluation scheme is effective in rejecting most false alarms while retaining the true detections.

Since we aim to select the best subset of O as our optimization solution, $\mathbf{c}^*$, we assume $\mathbf{h}_j \in O, j = 1, 2...n$. We can treat the individual conditional probability $P(o_i|\mathbf{c})$ as a decision likelihood with o_i as the observation and $\mathbf{c}$ as the decision. Suppose the set of hypotheses O consists of n_{tp} true positives (tp), n_{tn} true negatives (tn), n_{fp} false positives (fp), and n_{fn} false negatives (fn). The decision rules (for the detection threshold T) for each observation o_i are defined as follows:

1. $P(o_i|\mathbf{c}) = p_{tp}$ if $o_i \in \mathbf{c}$ and $L(o_i|I_{occ}) \geq T$;

2. $P(o_i|\mathbf{c}) = p_{fp}$ if $o_i \in \mathbf{c}$ and $L(o_i|I_{occ}) < T$;

3. $P(o_i|\mathbf{c}) = p_{tn}$ if $o_i \notin \mathbf{c}$ and $L(o_i|I_{occ}) \geq T$;

4. $P(o_i|\mathbf{c}) = p_{fn}$ if $o_i \notin \mathbf{c}$ and $L(o_i|I_{occ}) < T$,

where I_{occ} denotes the occlusion map generated from the configuration $\mathbf{c}$ and $L(o_i|I_{occ})$ denotes the occlusion-compensated (re-evaluated) object-level likelihood. The probabilities p_{tp}, p_{fn}, p_{fp}, and p_{tn} are set to $p_{tp} = p_{fn} = \alpha$ and $p_{fp} = p_{tn} = 1 - \alpha$ (where $\alpha > 0.5$) for the current implementation. Finally, the shape likelihood (Equation 2.15) can be expressed as: $P(I_s|\mathbf{c}) = p_{tp}^{n_{tp}} p_{fp}^{n_{fp}} p_{tn}^{n_{tn}} p_{fn}^{n_{fn}} = \alpha^{(n_{tp}+n_{fn})}(1 - \alpha)^{(n_{fp}+n_{tn})}$.

2.4.4.2 Optimization based on Likelihood Re-evaluation

We sort the hypotheses in decreasing order of vertical (or y) coordinate as in [127]. This is valid for many surveillance videos with ground plane assumption, since the camera is typically looking obliquely down towards the scene. For notational simplicity, we assume $o_1, o_2, ..., o_N$ is such an ordered list. Starting from

an empty set, the optimization is performed based on iterative addition of humans based on occlusion compensation and likelihood re-evaluation.

An example of the detection and segmentation process is shown in Figure 2.15. Note that initial false detections are rejected in the final detection based on likelihood re-evaluation, and the occlusion map is accumulated to form the final segmentation.

Algorithm 2: Optimization algorithm

Given an ordered list of hypotheses $o_1, o_2, ..., o_N$,
initialize the configuration as $\mathbf{c} = \phi$, the occlusion map I_{occ} as empty (white image), and the joint likelihood as $P(I|\mathbf{c}) = 0$.
for $i = 1 : N$
1. re-evaluate the object-level likelihood of hypothesis o_i based on the current occlusion map I_{occ}, *i.e.* calculate $L(o_i|I_{occ})$.
2. if $L(o_i|I_{occ}) \geq T$ and $P(I|o_i \cup \mathbf{c}) > P(I|\mathbf{c})$, $o_i \mapsto \mathbf{c}$.
3. update the occlusion map I_{occ} using the current configuration $\mathbf{c}$.
endfor
return the configuration $\mathbf{c}$ and occlusion map I_{occ}.

2.5 Combining with Calibration and Background Subtraction

We can also combine the shape-based detector with background subtraction and calibration in a unified system.

2.5.1 Scene-to-Camera Calibration

If we assume that humans are moving on a ground plane, ground plane homography information can be estimated off-line and used to efficiently control the search for humans instead of searching over all scales at all positions. A similar idea has been explored by Hoiem *et al.* [48] combining calibration and segmentation. To obtain a mapping between head points and foot points in the image, *i.e.*

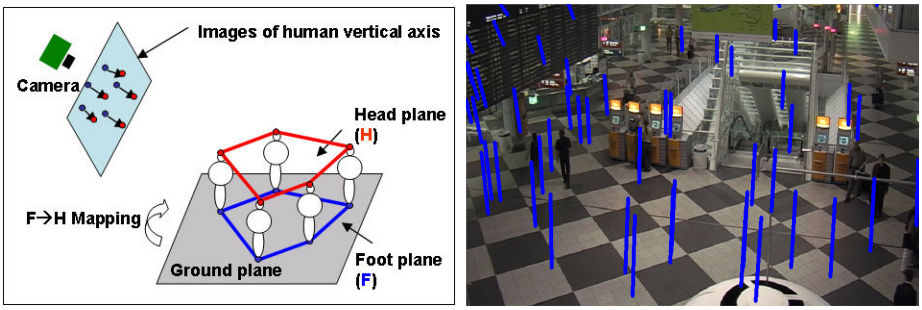
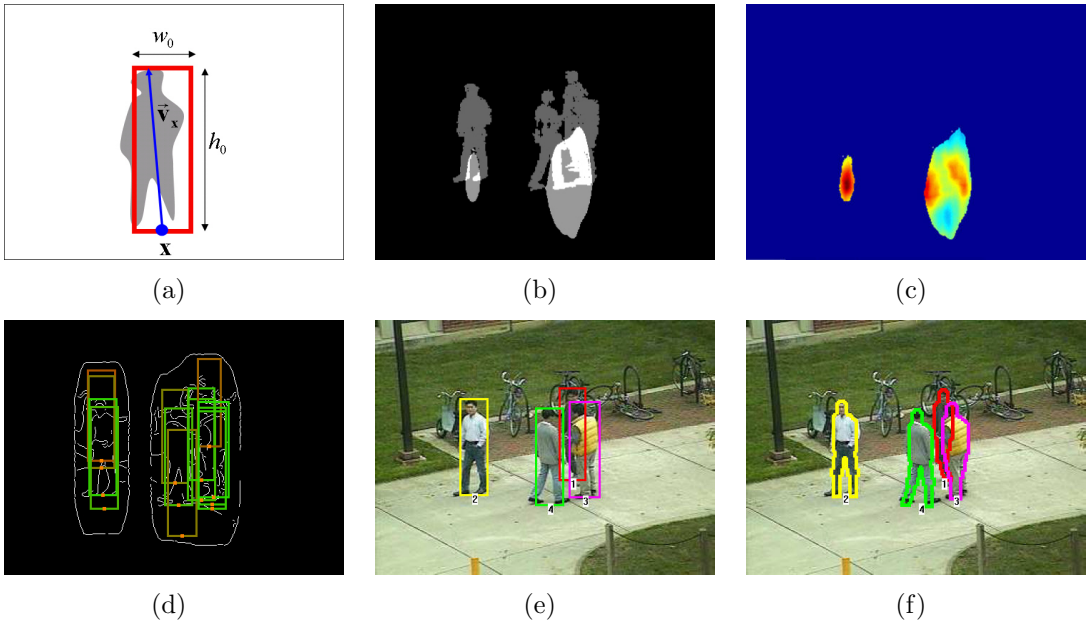


Figure 2.10: Simplified scene-to-camera calibration. Left: Interpretation of the foot-to-head plane homography mapping. Right: An example of the homography mapping. 50 sample foot points are chosen randomly and corresponding head points and human vertical axes are estimated and superimposed in the image.

to estimate expected vertical axes of humans, we simplify the calibration process by estimating the homography between the head plane and the foot plane in the image [88]. We assume that humans are standing upright on an approximate ground plane viewed by a distant camera relative to the scene scale, and that the camera is located higher than a typical person's height. We define the homography mapping as $\mathbf{f} = P_f^h : F \mapsto H$, where $F, H \in \mathbb{P}^2$. Under the above assumptions, the mapping $\mathbf{f}$ is one-to-one correspondence so that given an off-line estimated 3×3 matrix P_f^h, we can estimate the expected location of the corresponding head point $p_h = \mathbf{f}(p_f)$ given an arbitrary foot point p_f in the image. The homography matrix is estimated by the least squares method based on $L >> 4$ pairs of foot and head points pre-annotated in some frames. An example of the homography mapping is shown in Figure 2.10.

2.5.2 Combining with Background Subtraction

Given the calibration information and the binary foreground image from background subtraction, we estimate the binary foot candidate regions R_{foot} as follows:



(a) (b) (c)

(d) (e) (f)

Figure 2.11: An example of the detection process with background subtraction. (a) Adaptive rectangular window, (b) Foot candidate regions R_{foot} (lighter regions), (c) Object-level (foot-candidate) likelihood map by the hierarchical part-template matching (where red color represents higher probabilities and blue color represents lower probabilities), (d) The set of human hypotheses overlaid on the Canny edge map in the augmented foreground region (green boxes represent higher likelihoods and red boxes represent lower likelihoods), (e) Final human detection result, (f) Final human segmentation result.

we first find all foot candidate pixels $\mathbf{x}$ with foreground coverage density $\gamma_{\mathbf{x}}$ larger than a threshold ξ. Given the estimated human vertical axis $\overrightarrow{\mathbf{v}}_{\mathbf{x}}$ at the foot candidate pixel $\mathbf{x}$, $\gamma_{\mathbf{x}}$ is defined as the proportion of foreground pixels in an adaptive rectangular window $W(\mathbf{x}, (w_0, h_0))$ determined by the foot candidate pixel $\mathbf{x}$. The foot candidate regions R_{foot} are defined as: $R_{foot} = \{\mathbf{x} | \gamma_{\mathbf{x}} \geq \xi\}$. The window coverage is efficiently calculated using integral images [115]. We detect edges in the augmented foreground regions R_{afg} which are generated from the foot candidate regions R_{foot} by taking the union of the rectangular regions determined by each foot candidate pixel $p_f \in R_{foot}$, adaptively based on the estimated human vertical axes.

Figure 2.11 shows an example.

2.6 Experimental Results

We first presents results using our generic human detector on two public pedestrian datasets and then discuss results of our multiple occluded human detector on three crowded image and video datasets.

2.6.1 Detection and Segmentation using Pose-Invariant Descriptors

We evaluate our generic human detector (learned based on pose-invariant descriptors) mainly using the INRIA person dataset[5] [22] and the MIT-CBCL pedestrian dataset[6] [75, 83]. The MIT-CBCL dataset contains 924 front/back-view positive images (no negative images), and the INRIA dataset contains 2416 positive training samples and 1218 negative training images plus 1132 positive testing samples and 453 negative testing images. Comparing to the MIT dataset, the INRIA dataset is much more challenging due to significant pose articulations, occlusion, clutter, viewpoint and illumination changes.

2.6.1.1 Detection Performance

We evaluate our detection performance and compare it with other approaches using Detection-Error-Tradeoff (DET) curves, plots of miss rates versus false positives per window (FPPW).

[5] `http://lear.inrialpes.fr/data`
[6] `http://cbcl.mit.edu/software-datasets/PedestrianData.html`

Training. We first extract pose-invariant descriptors for the set of 2416 positive and 12180 negative samples and batch-train a discriminative classifier for the initial training algorithm. We use the publically available LIBSVM tool [18] for binary classification (RBF Kernel) with parameters tuned to C=8000, gamma=0.04 (as the default classifier).

For improving performance, we perform one round of bootstrapping procedure for retraining the initial detector. We densely scan 1218 (plus mirror versions) person-free photos by 8-pixel strides in horizontal/vertical directions and 1.2 scale (down-sampling) factors (until the resized image does not contain any detection window) to bootstrap false positive windows. This process generates 41667 'hard' samples out of examined windows. These samples are normalized to 128×64 and added to the original 12180 negative training samples and the whole training process is performed again.

Testing. For evaluation on the MIT dataset, we chose its first 724 image patches as positive training samples and 12180 training image images from the INRIA dataset as negative training samples. The test set contains 200 positive samples from the MIT dataset and 1200 negative samples from the INRIA dataset. As a result, we achieve 1.0% true positive rate, and a 0.00% false positive rate even without retraining. Direct comparisons on the MIT dataset are difficult since there are no negative samples and no separation of training and testing samples in this dataset. Indirect comparisons show that our result on this dataset are similar to the performance achieved previously in [22].

For the INRIA dataset, we evaluated our detection performance on 1132 pos-

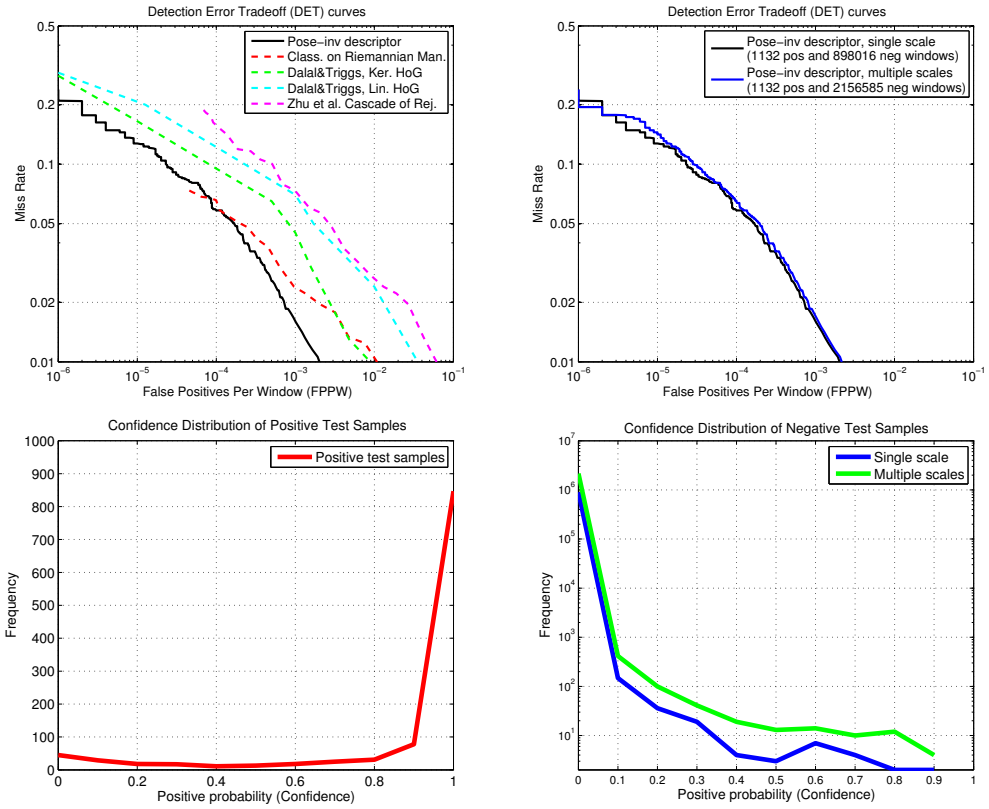


Figure 2.12: Detection performance evaluation on INRIA dataset. Top-Left: The proposed approach (testing on single scale) is compared to Kernel HOG-SVM [22], Linear HOG-SVM [22], Cascaded HOG [142], and Classification on Riemannian Manifold [113]. The results of [22] are copied from the original paper, and the results of [113,142] are obtained by running their original detectors on the same test data. Top-Right: Performance comparison w.r.t. the number of negative windows scanned. Bottom: Distribution of confidence values for positive and negative test windows.

itive image patches and 453 negative images. Negative test images are scanned exhaustively in the same way as in retraining. The detailed comparison of our detector with current state of the art detectors on the INRIA dataset is plotted using the DET curves as shown in Figure 3.7. The comparison shows that our approach is comparable to state of the art human detectors. The dimensionality of our features is less than half of that used in HOG-SVM [22], but we achieve better performance.

Another advantage of our approach is that it is capable of not only detecting but also segmenting human shapes and poses. In this regard, our approach can be further improved because our current pose model is very simple and can be extended to cover a much wider range of articulations. Figure 2.13 shows examples of detection on whole images and examples of false negatives and false positives from our experiments. Note that FNs are mostly due to unusual poses or illumination conditions, or significant occlusions; FPs mostly appeared in highly-textured samples (such as trees) and structures resembling human shapes. Figure 2.14 shows qualitative comparisons of our pose-invariant descriptors with HOG descriptors [22] on detecting humans in natural images. Our detector successfully detected very difficult poses while the HOG-based detector missed them.

2.6.1.2 Segmentation Performance

Figure 2.15 shows some qualitative results of our pose/shape segmentation algorithm on the INRIA dataset. Our pose model and probabilistic hierarchical part-template matching algorithm give very accurate segmentations for most images in the MIT-CBCL dataset and on over 80% of 3548 training/testing images in the INRIA dataset. Significantly poor pose estimation and segmentation are observed in about 10% of the images in the INRIA dataset, and most of those poor segmentations were due to very difficult poses and significant misalignment of humans.

Our detection and segmentation system is implemented in C++ and the current running time (on a machine with 2.2GHz CPU and 3GB memory) is as follows.

Figure 2.13: Detection results. Top: Example detections on the INRIA test images, nearby windows are merged based on distances; Bottom: Examples of false negatives (FNs) and false positives (FPs) generated by our detector.

Both pose segmentation and feature extraction for 800 windows takes less than 0.2 second; classifying 800 windows with the RBF-Kernel SVM classifier takes less than 10 seconds; initial classifier training takes about 10 minutes and retraining takes about two hours. The computational overhead is only due to the kernel SVM classifier which can be replaced with a much faster boosted cascade of classifiers [115] (which we have implemented recently and runs at 3 frames/second on a 320×240 image scanning 800 windows); this is comparable to [113] (reported as less than 1 second scanning 3000 windows).

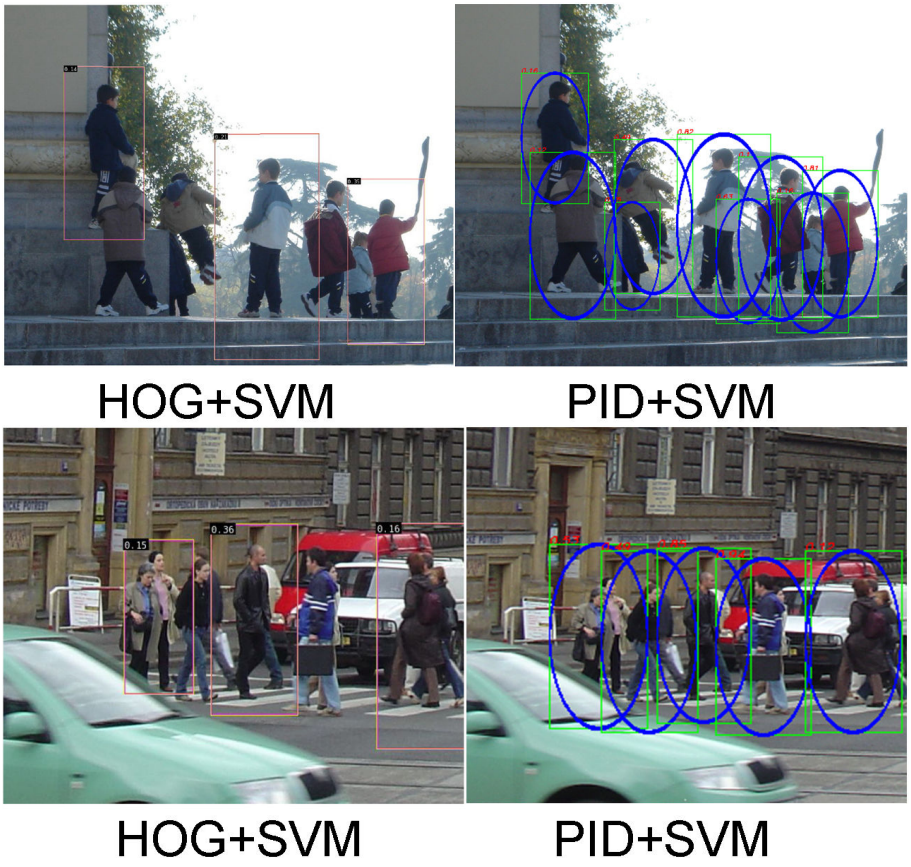


Figure 2.14: Qualitative comparisons of our pose-invariant descriptor (PID) with the HOG descriptors. Results of the 'HOG+SVM' method [22] are copied from the author's thesis.

2.6.2 Detection and Segmentation of Multiple Occluded Humans

In order to quantitatively evaluate the performance of our detector, we use the overlap measure defined in [61]. The overlap measure is calculated as the smaller value of the area ratios of the overlap region and the ground truth annotated region/detection region. If the overlap measure of a detection is larger than a certain threshold $\eta = 0.5$, we regard the detection as correct.

Figure 2.15: Example results of pose/shape segmentation.

2.6.2.1 Results without Background Subtraction

We compared our human detector with Wu and Nevatia [127] and Shet *et al.* [102] on USC pedestrian dataset-B [127] which contains 54 grayscale images with 271 humans. In these images, humans are heavily occluded by each other and partially out of the frame in some images. Note that no background subtraction is provided for these images. Figure 2.16 shows some example results of our detector and Figure 3.7(a) shows the comparison result as ROC curves. Our detector obtained better detection performance than the others when allowing more than 10 false alarms out of total of 271 humans, while detection rate decreased significantly when the number of false alarms was reduced to 6 out of 271. Proper handling of edge sharing problem would reduce the number of false alarms further while maintaining the detection rates. The running time of [127] for processing an 384×288

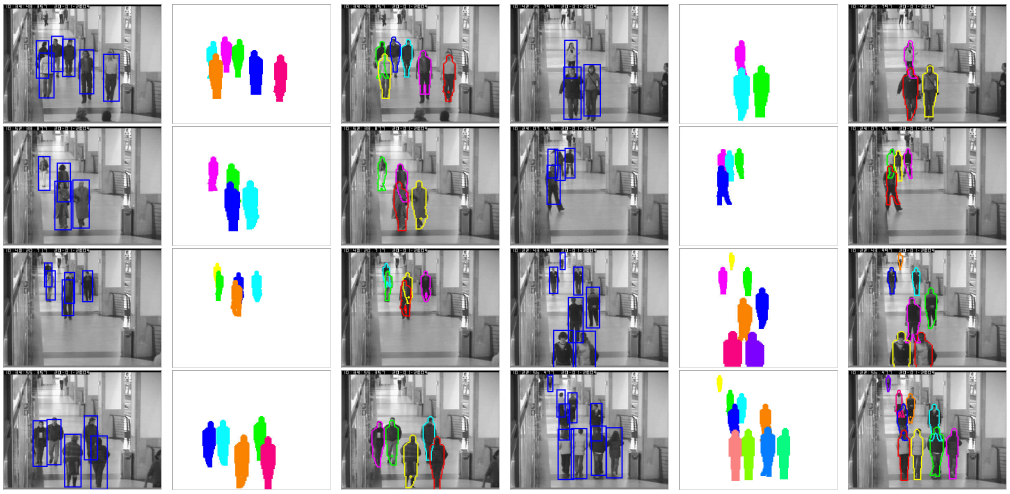


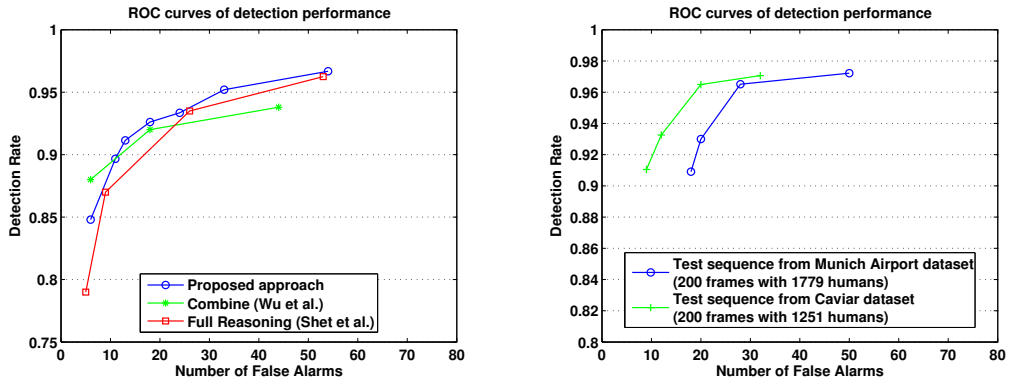
Figure 2.16: Detection and segmentation results (without background subtraction) for USC pedestrian dataset-B.

image is reported as about 1 frame/second on a Pentium 2.8GHz machine, while our current running time for a same sized image is 2 frames/second on a Pentium 2GHz machine.

2.6.2.2 Results with Background Subtraction

We also evaluated our detector on two challenging surveillance video sequences using background subtraction. The first test sequence (1590 frames) is selected from the Caviar Benchmark Dataset [1] and the second one (4836 frames) is selected from the Munich Airport Video Dataset [3].[7] The foreground regions detected from background subtraction are very noisy and inaccurate in many frames. From example results in Figure 2.18, we can see that our proposed approach achieves good performance in accurately detecting humans and segmenting the boundaries even under

[7]The selected data can be downloaded from `ftp://ftp.umiacs.umd.edu/pub/zhelin/iccv07/dataset`.



(a) (b)

Figure 2.17: Performance evaluation on three datasets. (a) Evaluation of detection performance on USC pedestrian dataset-B (54 images with 271 humans). Results of [127] and [102] are copied for the comparison purpose. (b) Evaluation of detection performance on two test sequences from Munich Airport dataset and Caviar dataset.

severe occlusion and very inaccurate background subtraction. Also, from the results, we can see that the shape estimates automatically obtained from our approach are quite accurate. Some misaligned shape estimates are generated mainly due to low contrast and/or background clutter.

We evaluated the detection performance quantitatively on 200 selected frames from each video sequence. Figure 3.7(b) shows the ROC curves for the two sequences. Most false alarms are generated by cluttered background areas incorrectly detected as foreground by background subtraction. Misdetections (true negatives) are mostly due to the lack of edge segments in the augmented foreground region or complete occlusion between humans. Our system is implemented in C++ and currently runs at about 2 frames/second (without background subtraction) and 5 frames/second (with background subtraction) for 384×288 video frames on a Pentium-M 2GHz Machine.

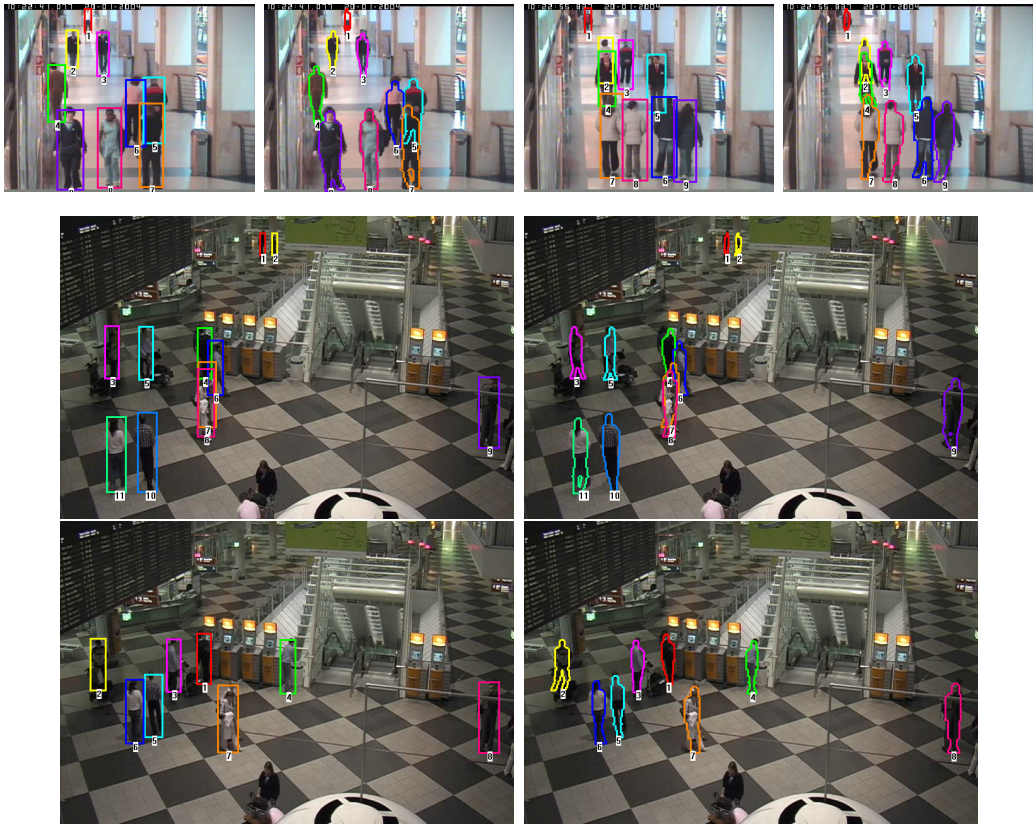


Figure 2.18: Detection and segmentation results (with background subtraction) for Caviar data [1] (first row) and Munich Airport data [3] (second and third rows).

Chapter 3

Appearance-based Human Segmentation

3.1 Introduction

Human segmentation can be regarded as a category-specific object segmentation problem and can be solved by combining traditional image segmentation techniques with high-level knowledge or constraints such as human poses. We first briefly review previous work on object segmentation, and present our approach to better solving this problem combining local and global shape constraints.

In foreground/background segmentation, pairwise potential-based approaches perform figure-ground discrimination by clustering features based on pairwise costs, *e.g.* Normalized Cut [104]. In contrast, object-centered clustering approaches group features with learned parametric or nonparametric densities; typical examples include the k-means clustering, and the EM-based clustering with mixtures of Gaussians [17]. EM-based approaches are sensitive to initialization and require appropriate selection of the number of mixture components. It is well known that finding a good initialization and choosing a generally reasonable number of mixtures for the traditional EM algorithm remain difficult problems. In [139], the KDE-EM approach is introduced by applying nonparametric kernel density estimation method in EM-based color clustering. Graph-cut approaches combine the pairwise potential-based scheme with object-centered appearance representation in a unified energy mini-

mization paradigm, *e.g.* Interactive Graph-Cuts [15], and its generalized version, GrabCut [91].

Object segmentation without any prior knowledge is well-known to be an ill-posed problem. Recently a few approaches have concentrated on enforcing global shape priors, top-down reasoning or other higher level knowledge to make the segmentation problem well posed. Object category-specific MRF [58] or pose-specific MRF [16] combines local contrast-dependent MRF with a layered pictorial structure model in [58] or a stickman model in [16] to provide strong global priors. Hence, the resulting segmentations resemble objects of interest. In [125], bottom-up cues are combined with global top-down knowledge for object class learning with unsupervised segmentation. In [87], an appearance learning-based method is proposed for articulated body segmentation and pose estimation; however it focuses on pose estimation and does not compute object segmentation explicitly. In [13], top-down shape cues are used to merge bottom-up over-segmentation to generate an object-like segmentation. In [138], the KDE-EM approach is combined with a shape template-based detection method for object segmentation. Recently, in [126], a layout-consistent random field is employed to provide a preliminary solution to segmentation in the presence of occlusion.

We propose an alternative, more efficient approach to human segmentation capable of handling inter-occlusion between humans. We incorporate local contrast-dependent MRF constraints and global shape priors iteratively into the KDE-EM framework [139] to estimate segmentations and poses simultaneously. There are four important contributions in this paper. First, we represent kernel densities of fore-

ground and background in a joint spatial and color space and update assignment probabilities *recursively* instead of using the direct update scheme in KDE-EM; this modification of feature space and update equations results in faster convergence and better segmentation accuracy. Second, we incorporate contrast-dependent MRF constraints into the KDE-EM scheme to regularize and smooth the segmentation within object and background regions. Third, we build and train a human pose model and perform pose inferences in the iterative clustering stages to enforce global shape priors throughout the segmentation process. This encourages the segmentation of human-like shapes and allows us to optimize segmentations and poses simultaneously. Fourth, and most importantly, we generalize the approach to a multiple occluded object segmentation by explicitly modeling and reasoning about occlusion.

3.2 Modified KDE-EM Approach

KDE-EM [139] uses nonparametric kernel density estimation [97] for representing feature distributions of foreground and background and performs iterative segmentation using EM. The log-likelihood objective function is similar to the one in the traditional EM-based segmentation, i.e. summation of log likelihoods of all pixels in the image, except that the likelihoods (assignment probabilities) are calculated from kernel densities.

Given a set of sample pixels $\{\mathbf{x}_i, i = 1, 2...N\}$ (with a distribution $\mathcal{P}$), each represented by a d-dimensional feature vector as $\mathbf{x}_i = (x_{i1}, x_{i2}..., x_{id})^t$, we can esti-

mate the probability $\hat{P}(\mathbf{y})$ of a new pixel $\mathbf{y}$ with feature vector $\mathbf{y} = (y_1, y_2, ..., y_d)^t$ belong to the same distribution $\mathcal{P}$ as

$$\hat{P}(\mathbf{y} \in \mathcal{P}) = \frac{1}{N\sigma_1 ... \sigma_d} \sum_{i=1}^{N} \prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.1}$$

where the same kernel function $k(\cdot)$ is used in each dimension (or channel) with different bandwidth σ_j. It is well known that a kernel density estimator can converge to any complex-shaped density with sufficient samples. Also due to its nonparametric property, it is a natural choice for representing the complex color distributions that arise in real images [19].

For enhancing the compactness and efficiency of the segmentation, we extend the color feature space in KDE-EM to incorporate spatial information. This joint spatial-color feature space has been previously explored for feature space clustering approaches such as [19, 43]. Each pixel is represented by a feature vector $\mathbf{x} = (X^t, C^t)^t$ in a 5D space, $\mathbb{R}^5$, with 2D spatial coordinates $X = (x_1, x_2)^t$ and 3D normalized rgs color[1] coordinates $C = (r, g, s)^t$. The separation of chromaticity from brightness in the rgs space allows the use of a much wider kernel with the s variable to cope with the variability in brightness due to shading effects. On the other hand, the chromaticity variables r and g are invariant to shading effects and therefore much narrower kernels can be used in these dimensions, which enables more powerful chromaticity discrimination [139]. In Equation 3.1, we assume independence between channels and use a Gaussian kernel $k(t) = 1/\sqrt{(2\pi)}exp\{-t^2\}$

[1] $r = R/(R+G+B)$, $g = G/(R+G+B)$, $s = (R+G+B)/3$

for each channel. The kernel bandwidths are estimated from the original image as in [97, 139].

KDE-EM employs a soft-labelling procedure and weighted kernel density estimation to update the assignment probabilities. For adapting the nonparametric kernel density estimation to the EM algorithm, a sampling step is substituted for the M-step in EM. In each iteration, samples are independently drawn from a *uniform distribution* and weighted by the assignment probabilities estimated from the previous iteration. The foreground/background assignment probabilities $F^t(\mathbf{y})$ and $B^t(\mathbf{y})$ are updated directly by weighted kernel densities. We modify this by updating $F^t(\mathbf{y})$ and $B^t(\mathbf{y})$ *recursively* on the previous assignment probabilities $F^{t-1}(\mathbf{y})$, $B^{t-1}(\mathbf{y})$ with weighted kernel densities (Equations 3.4 and 3.5). This modification results in faster convergence and better segmentation accuracy. An example of the modified KDE-EM approach is shown in Figure 3.5.

3.3 Pose-Assisted Segmentation

KDE-EM treats individual pixels separately, hence, the resulting segmentation usually has holes or isolated small regions. In order to obtain a coherent and object-like segmentation, we use higher-order dependencies between pixels. The higher-order dependencies can be exploited in the form of local and global MRFs. Instead of incorporating these priors in the energy function [10, 15, 16, 58, 91], we apply them iteratively and recursively in a single process to force the segmentation result to be a human-like shape. This avoids the need for an extra optimization step such as

Algorithm 3: Modified KDE-EM

Given a set of sample pixels $\{\mathbf{x}_i, i = 1, 2...N\}$ from the image, we iteratively estimate the assignment probabilities $F^t(\mathbf{y})$ and $B^t(\mathbf{y})$ $(t = 0, 1, 2...)$ of a pixel $\mathbf{y}$ belonging to the foreground $\mathcal{F}$ and background $\mathcal{B}$ as follows:
Initialization : Assign initial probabilities to pixels based on a 2D anisotropic Gaussian distribution. The parameters of the distribution are determined by the expected location and sizes (which are assigned via user interaction) of the foreground object.

$$F^0(\mathbf{y}) = e^{-1/2(Y-Y_0)^t V^{-1}(Y-Y_0)}, \tag{3.2}$$

$$B^0(\mathbf{y}) = 1 - F^0(\mathbf{y}), \tag{3.3}$$

where Y denotes the spatial coordinates of $\mathbf{y}$, Y_0 denotes expected object center coordinates, and V denotes a 2×2 (diagonal) covariance matrix. The diagonal elements of V are set proportional to the expected sizes of the object.
$\mathbf{M-Step}$: (*Random Pixel Sampling*) Randomly sample a set of pixels from the image to estimate foreground and background appearances represented by weighted kernel densities. For computational efficiency, we sample $\eta = 5\%$ of the pixels from the image for density estimation.
$\mathbf{E-Step}$: (*Soft Probability Update*)

$$F^t(\mathbf{y}) = cF^{t-1}(\mathbf{y}) \sum_{i=1}^{N} F^{t-1}(\mathbf{x}_i) \prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.4}$$

$$B^t(\mathbf{y}) = cB^{t-1}(\mathbf{y}) \sum_{i=1}^{N} B^{t-1}(\mathbf{x}_i) \prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.5}$$

where N is the number of samples and c is a normalizing factor such that $F^t(\mathbf{y}) + B^t(\mathbf{y}) = 1$.
Segmentation : The iteration is terminated when $\frac{\sum_{\mathbf{y}}\{|F^t(\mathbf{y}) - F^{t-1}(\mathbf{y})|\}}{n} < \epsilon$, where n is total number of pixels in the image. $F(\mathbf{y})$ and $B(\mathbf{y})$ denote the final converged assignment probabilities. The segmentation is finally estimated as: $\mathbf{y} \in \mathcal{F}$ if $F(\mathbf{y}) > B(\mathbf{y})$, $\mathbf{y} \in \mathcal{B}$ otherwise.

graph-cut and achieves simultaneous segmentation and pose estimation efficiently. Also, our approach maintains soft labelling throughout the optimization process, while graph-cut is a discrete (labelling) optimization scheme.

3.3.1 Incorporating Local MRF Constraints

Let $\Psi^t_{\mathcal{F}}$ and $\Psi^t_{\mathcal{B}}$ represent the probabilities of a pixel $\mathbf{y}$ being labelled as the foreground and background according to local contrast-dependent MRF constraints which are defined as:

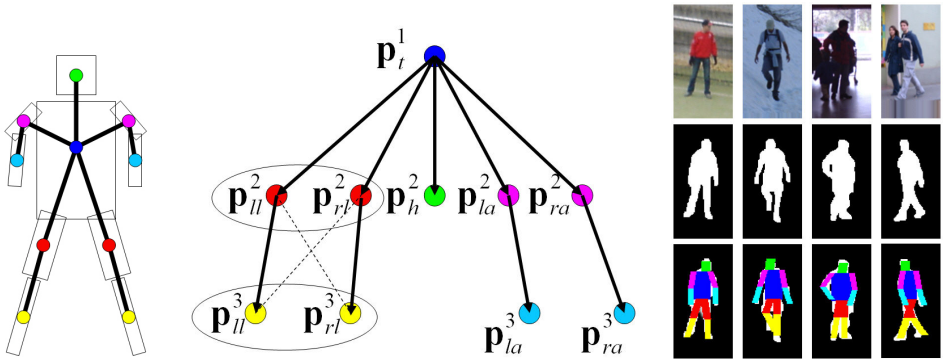
$$\Psi^t_{\mathcal{F}}(\mathbf{y}) = \sum_{\mathbf{z}\in\mathcal{N}_{\mathbf{y}}} \phi(\mathbf{I}|\mathbf{y},\mathbf{z})F^{t-1}(\mathbf{z}), \tag{3.6}$$

$$\Psi^t_{\mathcal{B}}(\mathbf{y}) = \sum_{\mathbf{z}\in\mathcal{N}_{\mathbf{y}}} \phi(\mathbf{I}|\mathbf{y},\mathbf{z})B^{t-1}(\mathbf{z}), \tag{3.7}$$

where $\mathbf{I}$ denotes the original image, $\mathcal{N}_{\mathbf{y}}$ denotes the neighborhood (8-neighborhoods) of pixel $\mathbf{y}$, and $\phi(\mathbf{I}|\mathbf{y},\mathbf{z})$ represents the contrast-dependent MRF induced likelihood for pixel $\mathbf{y}$. The likelihood $\phi(\mathbf{I}|\mathbf{y},\mathbf{z})$ is defined as:

$$\phi(\mathbf{I}|\mathbf{y},\mathbf{z}) = \frac{1}{dist(\mathbf{y},\mathbf{z})} e^{-\frac{1}{2}\left((\frac{r_{\mathbf{z}}-r_{\mathbf{y}}}{\sigma_r})^2+(\frac{g_{\mathbf{z}}-g_{\mathbf{y}}}{\sigma_g})^2+(\frac{s_{\mathbf{z}}-s_{\mathbf{y}}}{\sigma_s})^2\right)}. \tag{3.8}$$

To incorporate the local contrast-dependent MRF constraints into our iterative segmentation scheme, the recursive assignment probability update step (Equations 3.4 and 3.5) is extended by the local MRF terms (Equations 3.9 and 3.10). This can be explained as follows: the current foreground/background assignment probabilities are updated recursively by combined evidence from the spatial neigh-



(a) PS pose model (b) Tree-like structure (c) PS pose model fitting

Figure 3.1: An illustration of the pose model and training examples. (a) 10-parts PS model, (b) Simplified tree-like structure, (c) Examples of the training images, hand-segmented silhouettes, and PS pose model fitting results.

borhood and current foregournd/background appearance estimates (weighted kernel densities); in other words, the local MRF terms are incorporated to *smooth* the pixel-wise soft labelling at each iteration. We refer to this modified approach as CDMRF-KDE-EM. An example of the CDMRF-KDE-EM approach is shown in Figure 3.5.

3.3.2 Enforcing Global Shape Priors by Poses

We next describe how to incorporate prior shape information into the segmentation process. We build a Pictorial Structure (PS) pose model similar to the model in [32] and enforce global shape priors based on adaptive pose inference on soft segmentations at each iteration.

3.3.2.1 PS Pose Model

We chose 808 images from the INRIA person dataset [2] as training images (some of them are shown in Figure 3.1(c)). Human poses are modeled as a 10-part pictorial structure (Figure 3.1(a)) of which each part is represented as a horizontal parallelogram with five degrees of freedom (position $\mathbf{p}$, orientation α, sizes $\mathbf{s}$). Hence, the model (represented by parameters θ) has a total of $5 \times 10 = 50$ degrees of freedom. For simplicity, we assume independence between head, arms and legs and assume the pair of arms are also independent (the pair of legs are still correlated). This enables us to simplify the model to a tree-like structure (Figure 3.1(b)) on which the root node is chosen as the torso.

The PS model has many degrees of freedom and the parameter space is huge, while possible human poses form a low-dimensional manifold in this space. Hence, for efficiently searching the parameter space, we train the pose model and estimate its joint parameter distribution $l(\theta)$ from the set of best matching poses which are estimated using MLE by fitting the PS pose model to the binary silhouette images (obtained by manual segmentation of the training images) individually (Figure 3.1(c)). In our implementation, based on the above independence assumption, the joint distribution is marginalized as a set of individual joint distributions for different parts (head, torso, arms and legs). Hence, as a result of training, $l(\theta)$ is represented as a set of probability mass functions on low dimensional parameter spaces.

3.3.2.2 Training the Pose Model from Silhouettes

The degree of fitting $\rho(\theta|S)$ is defined as the similarity of the silhouette image S and the binary model coverage image $M(\theta)$, *i.e.* $\rho(\theta|S) = 1 - \frac{\sum_{\mathbf{x}} \|S(\mathbf{x}) - M(\theta,\mathbf{x})\|}{n}$, where n is the total number of pixels in the image. Then, the problem of model fitting can be formulated as a maximum likelihood estimation: $\theta_i^* = \arg\max_{\theta \in \mathbf{\Theta}} \rho(\theta|S_i)$, where $\mathbf{\Theta}$ denotes the set of all possible model parameters, and θ_i^* is the maximum likelihood estimate for the binary silhouette image $S_i, i \in \{1, 2, ..., N_t\}$ (N_t is the number of training images).

In training, we assume a uniform prior over $\mathbf{\Theta}$. According to the model in Figure 3.1(b), there are only loops between the pair of legs in the simplified tree-like graph structure. Optimization for matching the PS model to images is performed by belief propagation similar to [32] which is known to achieve globally optimal solutions for tree-structured acyclic graphs. In our approach, parameters for pair of legs are jointly optimized for handling the cases of occlusion between legs. Finally, the configuration corresponding to the maximum overall fitting score is returned as the estimate θ^*. Figure 3.1(c) shows some examples of PS model fitting results.

3.3.3 Pose-Assisted Segmentation

Now, we combine the modified KDE-EM scheme with local MRF constraints and global pose priors to form a single iterative algorithm: pose-assisted segmentation. The global shape prior is enforced by iteratively fitting the trained PS model to the current foreground assignment probability map and updating the probabil-

ity map with the binary model coverage image as an adaptive weighted sum. The segmentation and pose estimation are performed in an interleaved and cooperative manner (Figure 3.5).

Algorithm 4: Pose-Assisted Segmentation

Initialization : As in KDE-EM.
M − Step : As in KDE-EM.
E − Step I : Incorporating local MRFs.

$$F^t(\mathbf{y}) = cF^{t-1}(\mathbf{y})\Psi^t_{\mathcal{F}}(\mathbf{y})\sum_{i=1}^{N} F^{t-1}(\mathbf{x}_i)\prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.9}$$

$$B^t(\mathbf{y}) = cB^{t-1}(\mathbf{y})\Psi^t_{\mathcal{B}}(\mathbf{y})\sum_{i=1}^{N} B^{t-1}(\mathbf{x}_i)\prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.10}$$

E − Step II : Adaptive pose inference on the soft segmentation and assignment probability update by the estimated poses.
1. Fit the PS model $\theta \in \mathbf{\Theta}$ to the current foreground probability map F^t to find the maximum a posteriori (MAP) solution as: $\theta^*_t = \arg\max_{\theta\in\mathbf{\Theta}} l(\theta)\rho_{\mathbf{t}}(\theta)$, where $\rho_t(\theta)$ is calculated as the similarity of the foreground assignment probability F^t and the binary model coverage image $M(\theta)$:

$$\rho_t(\theta) = 1 - \frac{\sum_{\mathbf{x}} \|F^t(\mathbf{x}) - M(\theta, \mathbf{x})\|}{n}. \tag{3.11}$$

Similar to the PS model fitting scheme, we employ the belief propagation algorithm in the reduced search space for estimating the best fitting model θ^*_t.
2. Use the binary model coverage image $M^t = M(\theta^*_t)$ to update the foreground probability map F^t as follows:

$$F^t_{new} = (1 - \omega_t)F^t + \omega_t M^t,\ F^t_{new} \mapsto F^t, \tag{3.12}$$

$$B^t = \mathbf{1}_{h\times w} - F^t, \tag{3.13}$$

where $\mathbf{1}_{h\times w}$ is an all-1 matrix and $\omega_t = \beta^t \rho_t(\theta)^\gamma$ is an adaptive weight to control the iteration based on the current model fitting score.
Segmentation : As in KDE-EM.

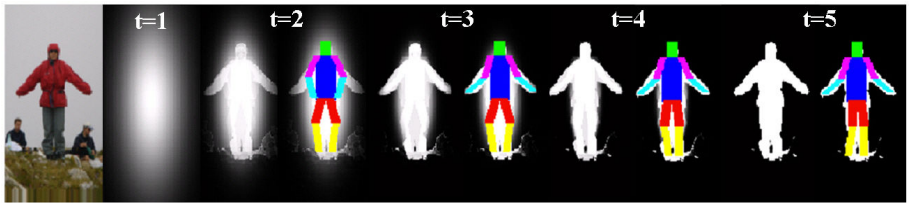


Figure 3.2: The iterative process of pose-assisted segmentation. Each frame represents the current soft segmentation overlaid with MAP fitted pose.

3.4 Segmentation of Multiple Occluded Objects

For the case of multiple objects, Elgammal and Davis [29] introduce a probabilistic framework for human segmentation assuming a single video camera. In this approach, appearance models must first be acquired and used later in segmenting occluded humans. Mittal and Davis [73] deal with the occlusion problem by a multi-view approach using region-based stereo analysis and Bayesian pixel classification. But this approach needs strong calibration of the cameras for its stereo reconstruction. Other multi-view-based approaches [35,54,56] combine evidence from different views by exploiting ground plane homography information to handle more severe occlusions. We aim to segment and build appearance models from a single view even if people are occluded in every frame.

Here, we assume an initial set of detection hypotheses (characterized by rough bounding boxes) is provided by an automatic detection system [71, 127] or interactively as in [15, 91]. Given an image $\mathbf{I}$ and a set of initial human hypotheses, $(\mathbf{x}_k, \mathbf{s}_k), k = 1, 2, ...K$, where $\mathbf{x}_k$ and $\mathbf{s}_k$ denote the location and scale of each human, the problem of segmentation is the $(K+1)$-class (K humans and background) pixel labelling problem. The label set is denoted as $\mathcal{F}_1, \mathcal{F}_2, ...\mathcal{F}_K, \mathcal{B}$. Given a pixel $\mathbf{y}$, we

denote the probability of the pixel $\mathbf{y}$ belonging to human-k as $F_k^t(\mathbf{y})$, and the probability of the pixel $\mathbf{y}$ belonging to the background as $B^t(\mathbf{y})$, where $t = 0, 1, 2...$ is the iteration index. The assignment probabilities $F_k^t(\mathbf{y})$ and $B^t(\mathbf{y})$ are constrained to satisfy the condition: $\sum_{k=1}^{K} F_k^t(\mathbf{y}) + B^t(\mathbf{y}) = 1$. When the camera is fixed, we can segment foreground regions based on background subtraction [55, 56], and the problem reduces to segment foreground into K individuals.

3.4.1 Layered Occlusion Model

We introduce a layered occlusion model into the initialization step for segmentation of multiple occluded objects. Layered representation have been used in [57] for motion segmentation. The background is assumed to be in the farthest back layer. Given a hypothesis of an occlusion ordering, we build our layered occlusion representation iteratively by calculating the foreground probability map F_k^0 for the current layer and its residual probability map R_k^0 for pixel $\mathbf{y}$. Suppose the occlusion order (from front to back) is given by $\mathcal{F}_1, \mathcal{F}_2, ...\mathcal{F}_K, \mathcal{B}$; then the initial probability map (Figure 3.3) is calculated recursively as follows:

Algorithm 5: Initialization by Layered Occlusion Model

initialize $R_0^0(\mathbf{y}) = 1$ for all $\mathbf{y} \in \mathbf{I}$
for $k = 1, 2, ...K$
– for all $\mathbf{y} \in \mathbf{I}$
– $F_k^0(\mathbf{y}) = R_{k-1}^0(\mathbf{y}) e^{-1/2(Y-Y_0)^t V^{-1}(Y-Y_0)}$
– $R_k^0(\mathbf{y}) = 1 - \sum_{j=1}^{k} F_j^0(\mathbf{y})$
endfor
return $F_1^0, F_2^0, ..., F_K^0$ and $B^0 = R_K^0$

3.4.2 Pose-Assisted Segmentation for Multiple Occluded Objects

We generalize the single-human segmentation scheme presented in the previous sections. We first incorporate the contrast-dependent MRF to regularize the probability maps in the $E-Step\ I$, and perform the PS pose model inference on individual probability maps F_k^t for each object and update the probability maps in the $E-Step\ II$. Based on the pose inference on individual probability maps, we explicitly reason about occlusion status between humans by comparing the assignment probabilities of the pixels in the occluded regions. Our pose-assisted segmentation approach performs segmentation, pose estimation and occlusion reasoning simultaneously in an interleaved, iterative process where occlusion reasoning is applied as a prior to update the assignment probability maps at each iteration.

Occlusion reasoning: the initial occlusion ordering is determined by sorting the hypotheses by their vertical coordinates and the layered occlusion model is used to estimate initial assignment probabilities. The occlusion status is updated at each iteration after the $E-step\ I$ by comparing the evidence of occupancy in the overlap area between different object hypotheses. For two object hypotheses H_i and H_j, if they have overlap area O_{H_i,H_j}, we estimate the occlusion ordering between the two as: H_i occlude H_j if $\sum_{\mathbf{x}\in O_{H_i,H_j}} F_i(\mathbf{x}) > \sum_{\mathbf{x}\in O_{H_i,H_j}} F_j(\mathbf{x})$ (*i.e.* H_i better accounts for the pixels in the overlap area than H_j), H_j occlude H_i otherwise, where F_i^t and F_j^t are the foreground assignment probabilities of H_i and H_j. At each iteration, every pair of hypotheses that have a non-empty overlap area is compared in this way. The whole occlusion ordering is updated by exchanges if and only if the estimated

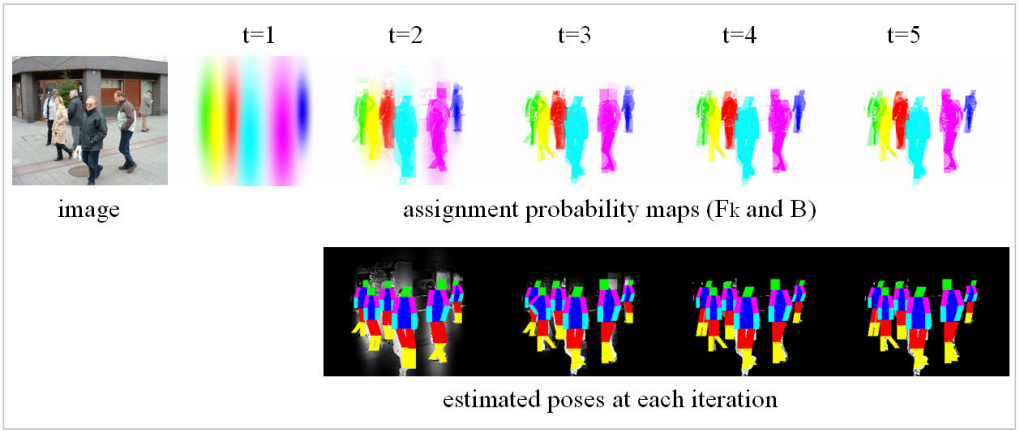


Figure 3.3: The process of pose-assisted segmentation for multiple occluded objects.

pairwise ordering differs from the previous ordering. Similar reasoning scheme have been explored in [57] using $\alpha\beta$-swap and α-expansion algorithms.

3.5 Experiments and Evaluation

In this section, we first present experiments on initialization sensitivity and then discuss qualitative and quantitative results for both single and multiple occluded human segmentation. In the experiments, the segmentation accuracy $\gamma[\%]$ is defined as the proportion of pixels correctly classified as foreground or background by comparing the binary segmentation result with the ground truth: $\gamma = \left(1 - \frac{\sum_{\mathbf{x}} |F(\mathbf{x}) - H(\mathbf{x})|}{n}\right) \times 100\%$, where F is the binary segmentation image and H is the hand-segmented ground truth. The constants β and γ are set to $\beta = 0.9$, $\gamma = 4$, and remained constant during the experiments.

Algorithm 6: Pose-Assisted Segmentation for Multiple Occluded Objects

Initialization : By the layered occlusion model.

M − Step : As in KDE-EM.

E − Step I : Assignment probability updates for multiple foreground objects and background.

$$F_k^t(\mathbf{y}) = cF_k^{t-1}(\mathbf{y})\Psi_{\mathcal{F}_k}^t(\mathbf{y})\sum_{i=1}^{N} F_k^{t-1}(\mathbf{x}_i)\prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.14}$$

$$B^t(\mathbf{y}) = cB_k^{t-1}(\mathbf{y})\Psi_{\mathcal{B}}^t(\mathbf{y})\sum_{i=1}^{N} B^{t-1}(\mathbf{x}_i)\prod_{j=1}^{d} k(\frac{y_j - x_{ij}}{\sigma_j}), \tag{3.15}$$

where c is a normalizing constant such that $\sum_{k=1}^{K} F_k^t(\mathbf{y}) + B^t(\mathbf{y}) = 1$.

E − Step II : Adaptive pose inference on the soft segmentation and assignment probability update by the estimated poses.

1. Update the occlusion ordering

2. Fit the PS pose model $\theta \in \mathbf{\Theta}$ to the current foreground probability map F_k^t to find the maximum a posteriori (MAP) estimation as:
$\theta_{k,t}^* = \arg\max_{\theta\in\mathbf{\Theta}} l(\theta)\rho_{k,t}(\theta)$, where

$$\rho_{k,t}(\theta) = 1 - \frac{\sum_{\mathbf{x}} \|F_k^t(\mathbf{x}) - M(\theta, \mathbf{x})\|}{n}. \tag{3.16}$$

We perform MAP optimization for each hypothesis to estimate the set of best fitting models $\theta_{k,t}^*, k = 1, 2, ...K$ for the current iteration step t.

3. Use the set of binary model coverage images $M_k^t = M(\theta_{k,t}^*)$, $k = 1, 2, ...K$ to update the foreground probability maps F_k^t, $k = 1, 2, ...K$ as follows:

$$F_{k_{new}}^{\ t} = (1 - \omega_t)F_k^t + \omega_t M_k^t,\ F_{k_{new}}^{\ t} \mapsto F_k^t, \tag{3.17}$$

$$B^t = \mathbf{1}_{h\times w} - \sum_k F_k^t, \tag{3.18}$$

where $\omega_t = \beta^t \rho_{k,t}(\theta)^\gamma$.

Segmentation : The iteration is terminated when $\frac{\sum_k \sum_{\mathbf{y}} \{|F_k^t(\mathbf{y}) - F_k^{t-1}(\mathbf{y})|\}}{n} < \epsilon$. We denote $F_k(\mathbf{y})$ and $B(\mathbf{y})$ as the final converged assignment probabilities. Then the final segmentation is determined as: pixel $\mathbf{y}$ belong to human-k, *i.e.* $\mathbf{y} \in \mathcal{F}_k, k = 0, 1, ...K$ (where $k = 0$ corresponds to background $\mathcal{F}_0 = \mathcal{B}$), if $k = \arg\max_{k\in\{0,1,...K\}} F_k^t(\mathbf{y})$.

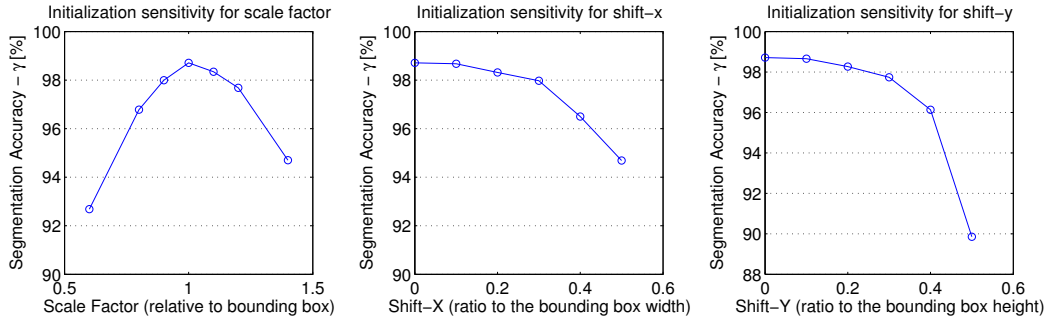
3.5.1 Initialization Sensitivity

The sensitivity of segmentation accuracy with respect to the initialization bias (scale, shift-x, shift-y) is tested for various images and results for a typical example are shown in Figure 3.4. (Note that results for other examples are very similar). The sensitivity curves show that segmentation accuracy decreases monotonically (but very slowly) with respect to scale and horizontal/vertical shifts. Specifically, the best segmentation accuracy is above 98% which is achieved with the true bounding box, and the accuracy is above 96% when the scale factor is in the range [0.75 1.25], when the horizontal shift factor is below 0.4, and when the vertical shift factor is below 0.42. Also, the accuracy remains above 90% when the scale factor is in the range [0.5 1.5], and remains above 92% and approximately above 90% when the horizontal and vertical shift factors increase from 0 to 0.5. We only consider sensitivity in these intervals since the initialization rectangle will have less than 50% overlap with the object region for more severe biases. Horizontal shifts tend to be less sensitive than scale change and vertical shifts.

3.5.2 Results on Single-human Segmentation

We have tested our approach to single-human segmentation on the INRIA person dataset [2]. Figure 3.5 shows comparison of the segmentation performances for GrabCut, KDE-EM, CDMRF-KDE-EM, and the proposed approach. KDE-EM resulted in a very inaccurate segmentation with many holes and isolated small regions. GrabCut obtained coherent segmentations but the results are very sensitive

(a) Initialization and Ground Truth Segmentation



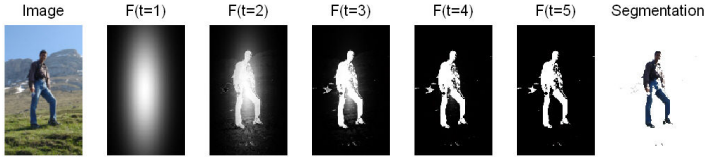
(b) Initialization Sensitivity Analysis

Figure 3.4: Experiments on initialization sensitivity. (a) Ground truth and biased bounding boxes, (b) Sensitivity w.r.t. scale, shift-x, and shift-y.

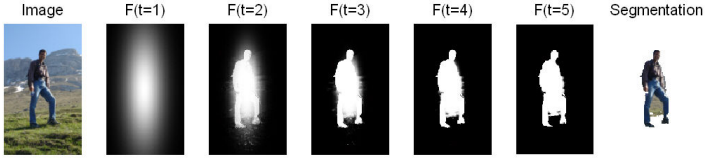
to the initialization and does not guarantee a human-like segmentation. CDMRF-KDE-EM obtained a coherent segmentation but incorrectly included background regions in the segmentation. In contrast, with the local MRF and global shape priors provided by the PS pose model inference, our approach achieved the best result, and the segmentation accuracy almost reached the ground truth (98.71%) for this example. Results for more difficult examples are shown in Figure 3.6.

We also quantitatively evaluated the proposed segmentation approach on a subset of 100 test images from the INRIA person dataset [2] and compared it with KDE-EM [139]. The set of test images are chosen to avoid redundancies of mirror images and overlap with the training set. Figures 3.7(a) and 3.7(b) show some examples of test images and the quantitative comparison results. The distribution of

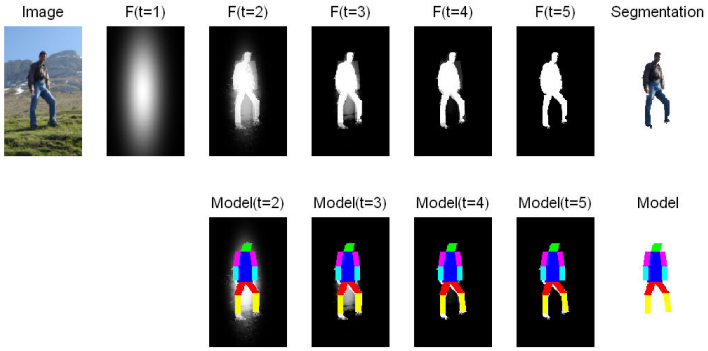
(a) GrabCut



(b) KDE-EM



(c) CDMRF-KDE-EM



(d) Pose-Assisted Segmentation

Figure 3.5: Example processes of segmentation approaches. (a) GrabCut [91] segmentation for three different initializations, (b) KDE-EM: EM soft-labelling using weighted kernel density estimation, (c) CDMRF-KDE-EM: KDE-EM combined with local contrast-dependent MRF constraints, (d) Pose-assisted segmentation.

the performance is evaluated by sorting the images by segmentation accuracy and number of iterations. The result shows that our proposed approach outperformed KDE-EM significantly in segmentation accuracy. For the number of iterations to convergence 3.7(c), our approach achieved slightly better convergence (fewer iterations) than KDE-EM (this is mainly due to the recursive soft probability update).

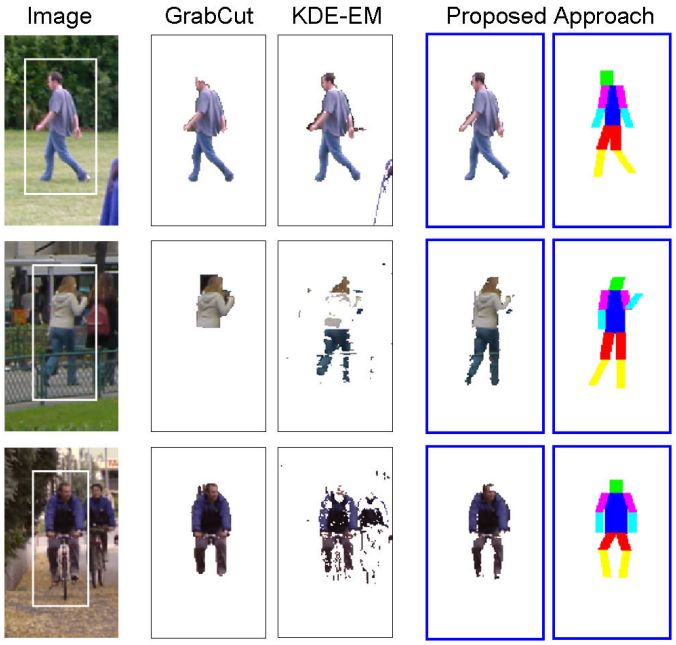
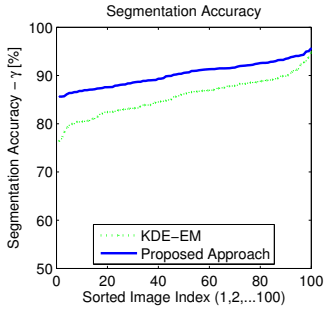


Figure 3.6: Results for more test images with increasing complexity. From left to right are original image with selected bounding boxes, result using GrabCut, result using KDE-EM, and segmentation and pose estimation results using our proposed method. Note that in these examples, we assume there is single foreground object and only segment the human in the center of the image.

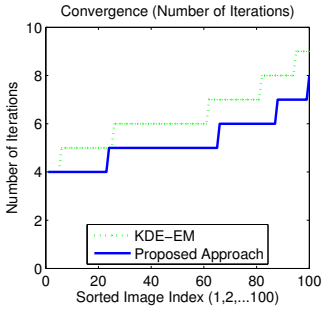
3.5.3 Results on Multi-human Segmentation

We compared our multi-human segmentation approach to G-KDE-EM (KDE-EM generalized to the case of multiple objects) on a variety of test images. Figure 3.8 shows some results on our segmentation and pose estimation results for images with multiple occluded humans. Our approach achieved good segmentation and pose estimation results even with severe inter-occlusions between humans, while KDE-EM resulted in poor segmentations with few human-like segmentations. This is as expected since KDE-EM does not enforce any prior knowledge in the segmentation. Finally, the running time and the number of iterations needed for our multi-human segmentation algorithm are similar to the cases of single human segmentation.

(a)



(b)



(c)

Figure 3.7: Quantitative performance evaluation. (a) Sample test images, (b) Comparison of segmentation accuracy, (c) Comparison of convergence rates.

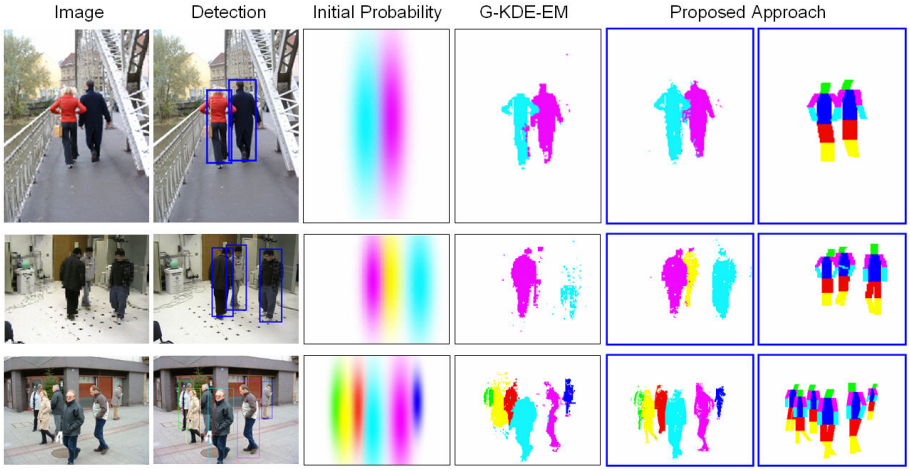


Figure 3.8: Comparison of segmentation and pose estimation for occluded cases.

Chapter 4

Appearance-based Person Recognition

4.1 Introduction

A central problem in multi-camera surveillance system is tracking people through gaps in observation - possibly due to occlusion or to people moving through areas not within the field of regard of any camera. This problem is typically addressed by building models of people's appearance or gait, since in most surveillance situations there is insufficient resolution on face to utilize face recognition. We formulate appearance-based full-body person matching as a multiclass learning and classification problem where each person (or appearance) is considered to be a class and instances of his/her appearances are considered to be class samples.

Learning discriminative classifiers such as LDA and SVM for a large number of classes is a challenging problem due to increased ambiguities between classes. Instead of building multiclass discriminative classifiers, we aim to learn invariance between classes in a discriminative manner, and apply it to classification. Specifically, we propose a multiclass learning and classification framework to maximally explore such inter-class information present in training data for improving the scalability of classifiers to larger number of categories. In order to better handle the scalability (*i.e.* number of classes) problem, we propose a pairwise coupling-based multiclass learning and classification framework, and apply it to appearance-based

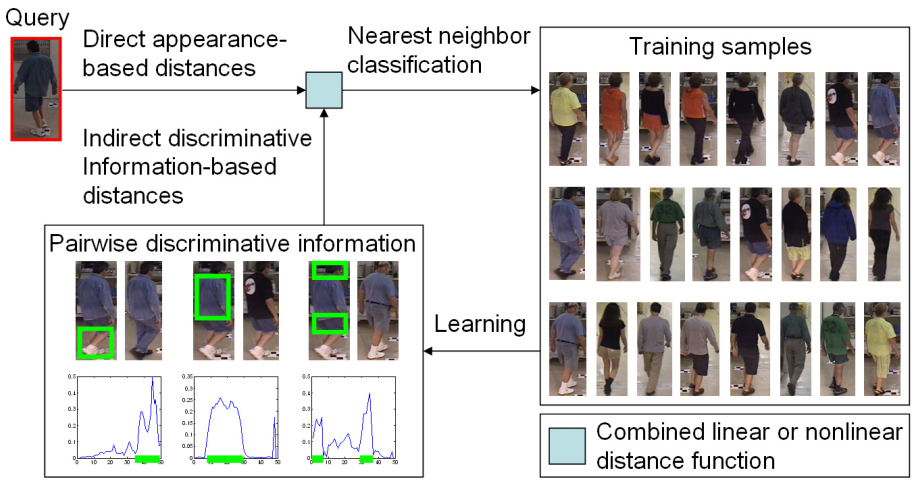


Figure 4.1: Outline of the approach. In learning, normalized invariant profiles are estimated for every pair of training samples in a discriminative way. In recognition, (direct) appearance-based distances are combined with (indirect) discriminative information-based distances for nearest neighbor classification.

person recognition (Figure 4.1).

Our motivation is summarized as follows. (1) A small region (or a feature) can be crucial in recognition because it might be the only distinguishing element to discriminate two otherwise very similar appearances, (2) Discriminative features are much easier to train in a pairwise scheme than in a one-against-all scheme, (3) Discriminative features are generally different for different pairs of persons, and (4) Pairwise discriminative properties remain invariant under pose, viewpoint and illumination changes. Based on these observations, we aim to incorporate pairwise discriminative information-based evidence into a traditional nearest neighbor classifier to reduce the distances of a query to prototypes from the same class, while magnifying distances to prototypes from different classes, *i.e.* to increase the expected relative margins.

There are numerous approaches to multiclass learning and recognition. In

instance-based learning, nearest neighbor (NN) and k-NN [21] are the most commonly used nonparametric classifiers. These classifiers are simple and perform well for large number of training samples. In discriminative learning, linear discriminant analysis (LDA) [8] is a well-known linear dimensionality reduction technique which has been successfully applied to many problems, including face recognition. Support vector machine (SVM) [84] is a popular discriminative technique to linearly separate multiclass patterns in a high-dimensional feature space through nonlinear kernel mapping. There are several schemes to decompose a multiclass classification problem into a set of binary classification problems. For example, one-against-all [76], pairwise coupling [46, 90, 130], error correcting output codes [24, 137], decision trees [6], probabilistic boosting trees [112], etc. Different schemes for combining binary SVM classifiers were tested in [76] and performance was compared to k-NN classifiers. In [74], a multiclass classification problem was transformed into a binary classification problem by modeling intra-personal and extra-personal distributions.

Recently, more sophisticated approaches have been developed for applying discriminative learning techniques to image and object category classification. k-NN and SVM are combined in [136] by performing a multiclass SVM only for a set of neighbors and query. Random forests and ferns classifier [14] and local ensemble kernel SVM [64] employ multiclass-SVM for object category recognition based on the one-against-all scheme. Also, a number of approaches have been proposed in the machine learning community for learning locally or globally adaptive distance metrics [26, 47, 99, 122, 132] by weighting features based on labeled training data. To simplify the one-against-all discrimination task for a huge number of categories, a

triplet-based distance metric learning framework was proposed in [96], and later was extended by learning weights for every prototype in [36, 37] for image retrieval and recognition. However, fixed weighting for each prototype can be inefficient for a very large number of classes, especially when ambiguities between classes are significant. Only a few works [42,45,76] have addressed multiclass classifiers for full-body person recognition, but these approaches have focused on comparing traditional classifiers such as k-NN, complex RBF neural networks, and one-against-all multiclass-SVMs.

Most previous approaches to multiclass classification have focused on designing good classifiers with large separation margins and good generalization properties, or on learning discriminative distance metrics. Our work is different in that, instead of building discriminative classifiers between categories, we explore invariant information for each pair of categories in a discriminative way, and apply it to classification by calculating distances of a query to training samples from both intra-class and inter-class invariant information. Here, the intra-class invariance is based on common appearance information in each class, *i.e.* appearance information that does not change dramatically for each individual as pose, viewpoint and illumination change. The inter-class invariance is based on the observation that any pair of examplars E_A and E_B from two different classes A and B share certain common discriminative properties. For example, in the context of person appearance recognition, if person 1 and person 2 have different jacket colors, then any variants of person 1 and person 2 will probably also have different jacket colors.

4.2 Appearance Representation and Matching

4.2.1 Appearance Model

Color and texture are the primary cues for appearance-based object recognition. For human appearances, the most common model is the color histogram [20]. Spatial information can be added by representing appearances as a function of height [56,73] or in a joint color-spatial feature space [28,135]. Other representations include color structure descriptors [45], spatial-temporal appearance modeling [41] by interest points and model fitting-based methods, spatial and appearance context modeling [119], part-based appearance modeling [63], panoramic appearance map-based modeling [38], and gait/motion-based modeling [9].

We build appearance models of individuals based on nonparametric kernel density estimation [97]. It is well known that a kernel density estimator can converge to any complex-shaped density with sufficient samples. Also due to its nonparametric property, it is a natural choice for representing the complex color distributions that arise in real images.

Given a set of sample pixels, represented by d-dimensional feature vectors $\{\mathbf{s}_i = (s_{i1}...s_{id})^t\}_{i=1...N_s}$, from a target appearance a, we estimate the probability of a new feature vector $\mathbf{z} = (z_1, z_2, ..., z_d)^t$ from the same appearance a using multivariate kernel density estimation as:

$$\hat{p}^a(\mathbf{z}) = \frac{1}{N_s \sigma_1 ... \sigma_d} \sum_{i=1}^{N_s} \prod_{j=1}^{d} k(\frac{z_j - s_{ij}}{\sigma_j}), \tag{4.1}$$

where the same kernel function $k(\cdot)$ is used in each dimension (or channel) with different bandwidth σ_j. The kernel bandwidths can be estimated as in [97]. We assume independence between channels and use a Gaussian kernel for each channel. The kernel probability density function (PDF) $\hat{p}^a(\mathbf{z})$ in (Equation 4.1) is referred to as the model of the appearance a.

As in [28], we extend the color feature space to incorporate spatial information in order to preserve color structure in appearances. Assuming people are in approximate upright poses, we encode each pixel by a feature vector $(\mathbf{c}, h)^t$ in a $4D$ joint color-height space, $\mathbb{R}^4$, with $3D$ color feature vector $\mathbf{c}$ and $1D$ height feature h (represented by vertical image coordinate y). We decide to use only the y coordinate instead of using $2D$ spatial coordinates (x, y) for handling viewpoint and pose variations, while preserving vertical color structures. For dealing with illumination changes, we use the following two illumination insensitive color features.

Normalized Color Feature : [28] $3D$ normalized rgs color[1] coordinates are commonly used as illumination insensitive features since the separation of chromaticity from brightness in the rgs space allows the use of a much wider kernel with the s variable to cope with the variability in brightness due to shading effects.

Color Rank Feature: [135] The features are encoded as the relative rank[2] of intensities of each color channel R, G and B for all sample pixels. Color rank (rR, rG, rB) features ignore the absolute values of colors by reflecting relative color rankings instead. Ranked color features are invariant to monotonic color transforms

[1] $r = R/(R+G+B)$, $g = G/(R+G+B)$, $s = (R+G+B)/3$.
[2] The rank is quantized in the interval $[1, 100]$.

and are very stable under a wide range of illumination changes.

4.2.2 Appearance Matching

Appearance models represented by kernel PDFs (Equation 4.1) can be compared by information theoretic measures such as the Battacharyya distance [20] or the Kullback-Leibler (KL) divergence (or distance) [28] for tracking and matching objects in video.

Suppose two appearances a and b are modeled as kernel PDFs $\hat{p}^a$ and $\hat{p}^b$ in the joint color-height space. Assuming $\hat{p}^a$ as the reference model and $\hat{p}^b$ as the test model, the similarity of the two appearances can be measured by the KL distance as follows:

$$D_{KL}(\hat{p}^b||\hat{p}^a) = \int \hat{p}^b(\mathbf{z}) log \frac{\hat{p}^b(\mathbf{z})}{\hat{p}^a(\mathbf{z})} d\mathbf{z}. \tag{4.2}$$

Note that the KL distance is a nonsymmetric measure in that $D_{KL}(\hat{p}^b||\hat{p}^a) \neq D_{KL}(\hat{p}^a||\hat{p}^b)$. For efficiency, the distance is calculated using only samples instead of the whole feature set. Given N_a samples $\{\mathbf{s}_i\}_{i=1...N_a}$ from appearance a and N_b samples $\{\mathbf{t}_k\}_{k=1...N_b}$ from appearance b, Equation 4.2 can be approximated by the following form given sufficient samples from the two appearances:

$$D_{KL}(\hat{p}^b||\hat{p}^a) = \frac{1}{N_b} \sum_{k=1}^{N_b} log \frac{\hat{p}^b(\mathbf{t}_k)}{\hat{p}^a(\mathbf{t}_k)}, \tag{4.3}$$

where

$$\hat{p}^a(\mathbf{t}_k) = \frac{1}{N_a}\sum_{i=1}^{N_a}\prod_{j=1}^{d} k(\frac{t_{kj}-s_{ij}}{\sigma_j}), \tag{4.4}$$

$$\hat{p}^b(\mathbf{t}_k) = \frac{1}{N_b}\sum_{i=1}^{N_b}\prod_{j=1}^{d} k(\frac{t_{kj}-t_{ij}}{\sigma_j}). \tag{4.5}$$

Let Φ^{ab} denote the log-likelihood ratio function, *i.e.* $\Phi^{ab}(\mathbf{u}) = log\frac{\hat{p}^b(\mathbf{u})}{\hat{p}^a(\mathbf{u})}$, where $\mathbf{u}$ is a d-dimensional feature vector in the color-height space. Since we sample test pixels only from appearance b, $\hat{p}^b$ is evaluated by its own samples, so $\hat{p}^b$ is generally equal to or larger than $\hat{p}^a$ for all test samples $\mathbf{t}_k$. Note that a few noisy (ambiguous) samples $\mathbf{t}_k$ from appearance b can be better matched by the reference model PDF $\hat{p}^a$ than the test model pdf $\hat{p}^b$ so that $\hat{p}^b(\mathbf{t}_k) < \hat{p}^a(\mathbf{t}_k)$, and, consequently, the log-likelihood ratio $\Phi^{ab}(\mathbf{t}_k)$ can be slightly less than zero. For minus log-likelihoods generated from those noisy samples, we force them to zeros (positive correction). Then, Equation 4.3 can be written as:

$$D_{KL}^{+} = \frac{1}{N_b}\sum_{k=1}^{N_b}[\Phi^{ab}(\mathbf{t}_k)]_+, \tag{4.6}$$

$$[\Phi^{ab}(\mathbf{t}_k)]_+ = max\left(\Phi^{ab}(\mathbf{t}_k), 0\right). \tag{4.7}$$

The positively corrected KL distance D_{KL}^{+} is guaranteed to be nonnegative for all samples: $D_{KL}^{+}(\hat{p}^b||\hat{p}^a) \geq \mathbf{0}$, where equality holds if and only if two density models are identical for all test samples.

The direct appearance-based distance $D_a(q, p_j)$ from a query q to a prototype

p_j is defined as:

$$D_a(q, p_j) = D^+_{KL}(\hat{p}^q || \hat{p}^j). \tag{4.8}$$

In conventional NN classification, $D_a(q, p_j)$ is evaluated for all prototypes $\{p_j\}_{j=1...N}$ and the minimum is chosen for classification.

4.3 Discriminative Learning of Pairwise Invariant Properties

For classifying large number of classes, it has been noted previously in [46, 90, 137] that a one-against-all scheme has difficulty separating one class from all of the others and often very complex classification models (probably leading to overfitting) are used for that purpose. In contrast, pairwise coupling is much easier to train since it only needs to separate one class from an other. For handling the scalability problem, we perform more detailed analysis of discriminative features between classes by estimating invariant information from pairwise comparisons.

4.3.1 Learning Pairwise Invariant Profiles

As discussed above, the KL distances are calculated as an average of log-likelihood ratios (Equation 4.3) over all samples from the test appearance. In addition to averaging the log-likelihood ratios to evaluate distances between two appearances, we can observe an interesting property. As seen in Figure 4.2, if we densely sample pixels in the target appearance b, the resulting log-likelihood ratio function $\Phi^{ab}(x, y)$[3] exactly reflects differences of two appearances; that is, the log-

[3]We can treat Φ^{ab} as a function of image pixel location (x, y) as $\Phi^{ab}(\mathbf{u}) = \Phi^{ab}(\mathbf{u}(x, y)) = \Phi^{ab}(x, y)$, where $\mathbf{u}$ is the feature vector of pixel (x, y).

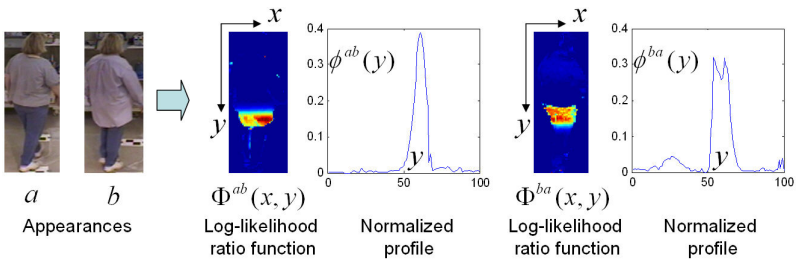


Figure 4.2: The log-likelihood ratio from a to b is calculated for all pixels (x, y) in the test appearance b to obtain the log-likelihood ratio function (or image) $\Phi^{ab}(x, y)$. And, $\Phi^{ab}(x, y)$ is marginalized over the x-axis and normalized to obtain an invariant profile $\phi^{ab}(y)$ from a to b. The profile $\phi^{ba}(y)$ from b to a is obtained in the same way. Here, normalized color-height features are used to generate the profiles.

likelihood ratio function quantitatively reflects discriminating regions (or features) between two appearances. This motivates us to conjecture that if we had variations of these two appearances, denoted by a' and b', which might be captured from different cameras or at different times, the difference between those new appearances would be very similar to the case of a and b. Consequently, the log-likelihood ratio function would be similar, *i.e.* $\Phi^{a'b'}(x, y) \simeq \Phi^{ab}(x, y)$. For dealing with shape variations due to viewpoint and pose variations and to estimate invariant information between two appearances, we project the $2D$ log-likelihood ratio function $\Phi^{ab}(x, y)$ onto the y-axis (or marginalize the function over the x-axis), and normalize the projected $1D$ function to have unit length (Figure 4.2).

$$\phi^{ab}(y) = C \int_0^{x_0} [\Phi^{ab}(x, y)]_+ dx, \tag{4.9}$$

where x_0 is the width of appearance b and C is a constant such that

$$\|\phi^{ab}\|^2 = \int_0^{y_0} [\phi^{ab}(y)]^2 dy = 1, \tag{4.10}$$

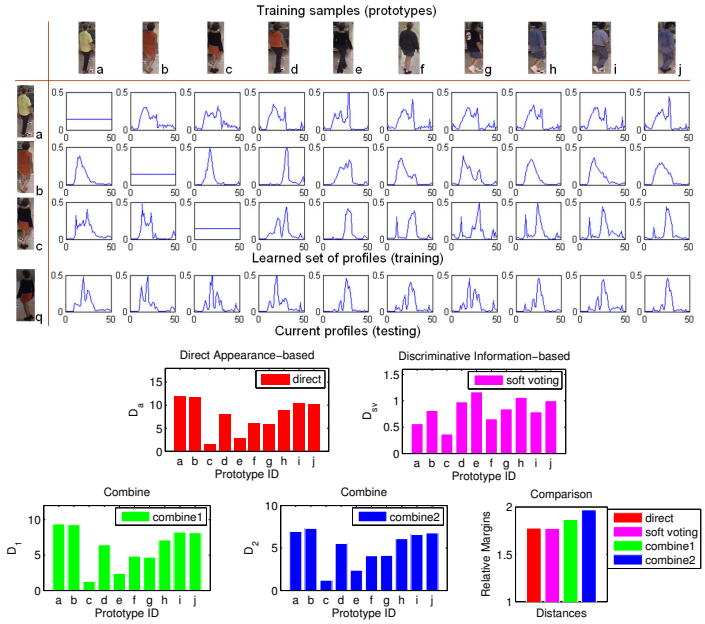


Figure 4.3: An illustration of the invariance of pairwise normalized profiles, and a comparison of direct, indirect, and combined distances for an example of 10 prototypes (with different labels) and one query. Top: it can be observed that the current test profiles of appearance q are very similar to the learned profiles of appearance c (which is the true classification of q) while largely different for the case of other appearances such as a and b. Bottom: distances D_a, D_{sv}, D_1, D_2 from q to all prototypes are evaluated and the relative margins are compared. We can see that all distance measures result in correct top one recognition, while the combined distance measures D_1 and D_2 result in larger relative margins than D_a and D_{sv}.

and y_0 is the height of appearance b. We define the $1D$ function ϕ^{ab} as the normalized invariant profile from a to b (Figure 4.2).

4.3.2 All-Pairs Training

Suppose we have N training samples (prototypes) $\{p_i\}_{i=1...N}$ labeled as n different appearances. We learn normalized invariant profiles for every pair of the prototypes. Hence, we produce $N \times N$ normalized invariant profiles indexed by $\{(i,j)\}_{i,j=1...N}$. Note that for two identical prototypes with index $i = j$, the log-

likelihood ratio function $\phi^{ii}(y) = \phi^{jj}(y) = 0$ for $\forall y \in (0, y_0)$, hence we set the profiles as uniform $\phi^{ii}(y) = \phi^{jj}(y) = 1/\sqrt{y_0}$ for this case. While we can see that ϕ^{ij} and ϕ^{ji} are different for $i \neq j$, they are very similar in shape.

We next discuss why we calculate all N^2 normalized profiles and how they can be used for classification.

4.4 Discriminative Information-based Distances

Given a query q, we want to match it to prototypes $\{p_i\}_{i=1...N}$ using the learned set of profiles $\{\phi^{ij}\}_{i,j=1...N}$. We first calculate a likelihood ratio function Φ^{iq} from every prototype p_i to query q, and perform normalization (as in the learning step) to obtain a set of query profiles $\{\phi^{iq}\}_{i=1...N}$. The idea is to vote for the ID of q (which is unknown) by matching a query profile ϕ^{iq} to a learned profile ϕ^{ij} for which the corresponding ID j is known. The distance $D(j, q|i)$ between the query profile ϕ^{iq} and the learned profile ϕ^{ij} is defined as follows:

$$D(j, q|i) = \int_0^{y_0} [\phi^{iq}(y) - \phi^{ij}(y)]^2 dy. \tag{4.11}$$

The intuition here is that the smaller the distance $D(j, q|i)$, the more similar the two profiles are, consequently, the more confident to vote for j as the ID of q. For each i, we calculate $D(j, q|i)$ for all $j = 1...N$ and then vote for the ID of q. We perform such a voting procedure for all training samples $\{p_i\}_{i=1...N}$.

We can vote for the ID of q based only on the best matching profile ϕ^{ij^*} for each prototype p_i, *i.e.* the one for which $j_i^* = \arg\min_j D(j, q|i)$, then assign one

vote for j_i^*: $V(j_i^*) + 1 \longmapsto V(j_i^*)$. This is referred to as hard voting.

The voting can be performed either in a soft manner, *i.e.* we vote for the ID of q based on the evidence from all profiles $\{\phi^{ij}\}_{i,j=1...N}$, instead of only choosing the best matching ID j^* (corresponding to the lowest profile distance) for each i. The soft voting-based distance $D_{sv}(q, p_j)$ from query q to prototype p_j is defined as:

$$D_{sv}(q, p_j) = \frac{1}{N} \sum_{i=1}^{N} D(j, q|i). \tag{4.12}$$

Compared to hard voting, soft voting is less sensitive to ambiguities and noise effects since it collects all possible evidence for calculating final voting-based distances instead of choosing the top one match as in hard voting. From experiments, we verify that soft voting gives better performance in recognition rate. Based on the above reasoning, we define the (indirect) discriminative information-based distance from the a query q to a prototype p_j as:

$$D_d(q, p_j) = D_{sv}(q, p_j). \tag{4.13}$$

4.5 Classification and Recognition

As discussed previously, traditional nearest neighbor or k-NN methods directly use D_a as the distance from a query to a prototype, and the classification is performed by finding the minimum distance or by majority voting for k nearest neighbors. D_a only considers information between a query and a prototype, while D_d only considers inter-relations between different training samples; that is, the

two distances are based on independent information. This leads to the idea that combining the two would boost recognition performance.

We tested two parameterized distance measures D_1 and D_2 involving linear and nonlinear combinations:

$$D_1(q, p_j) = (1 - \alpha)D_a(q, p_j) + \alpha D_d(q, p_j), \tag{4.14}$$

$$D_2(q, p_j) = D_a^{1-\beta}(q, p_j)D_d^{\beta}(q, p_j). \tag{4.15}$$

We learn the parameters α and β by evaluating the overall recognition rates using a large number of labeled testing samples. Experiments shows that the optimal parameter estimates are as listed in Table 4.1. The parameters can be selected flexibly around the optimal values (± 0.1) without performance degradation.

Given a query q, we want to estimate its unknown class label (person ID) by calculating the distances between the query and all labeled prototypes. Classification is done by the nearest neighbor rule using one of the combined distance measures D_1 and D_2 as:

$$j^* = \arg\min_j D_{combine}(q, p_j). \tag{4.16}$$

Feature Space	Training (cam1) Testing (cam2)	Training (cam2) Testing (cam1)
Normalized Color + Height	$\alpha = 0.23$ $\beta = 0.18$	$\alpha = 0.25$ $\beta = 0.22$
Color Rank + Height	$\alpha = 0.90$ $\beta = 0.40$	$\alpha = 0.60$ $\beta = 0.20$

Table 4.1: Learned optimal parameter values for combined distance measures D_1 and D_2. Total of 180 labeled test samples are used against 180 training samples of different appearances for learning the parameters α and β.

Feature Space	Channel Bandwidth
Normalized Color + Height	$\sigma_r = \sigma_g = 0.02, \sigma_s = 20$ $\sigma_h = 1$
Color Rank + Height	$\sigma_{rR} = \sigma_{rG} = \sigma_{rB} = 4$ $\sigma_h = 1$

Table 4.2: Bandwidths estimated and used for our experiments.

4.6 Implementation Details

The bandwidths for each dimension of the $4D$ feature spaces are listed in Table 4.2. The bandwidths are generally estimated as 2% of the ranges of the corresponding channel and adjusted slightly by repeated trial-and-error procedures. We resize image patches of a person to have fixed height of 50 pixels $y_0 = 50$. Note that all the parameters including bandwidths for different feature spaces, height sampling rate, distance model parameters α and β are fixed to the listed values throughout the experimental evaluation.

For efficiently matching people in video, we select key frames (appearances) as prototypes of training and testing sequences and recognize appearances based on the prototypes. The purpose is to represent all of the appearance information from

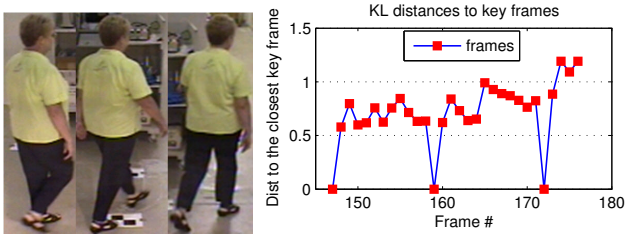


Figure 4.4: Left: three key frames (147, 159, 172) are obtained for a 30 frame example sequence. Right: the plot shows the KL distances of each frame to the closest key frame.

a person's track using as few representative frames as possible. The process is as follows. The first frame ($t = 0$) is automatically selected as the first key frame K_1. Then, we calculate the symmetric KL distances of the subsequent frames ($t = 1, 2...$) to all current key frames $\{K_j\}_{j=1...i}$. The symmetric KL distance is defined as:

$$D_{sKL}(\hat{p}^b, \hat{p}^a) = \min(D^+_{KL}(\hat{p}^b||\hat{p}^a), D^+_{KL}(\hat{p}^a||\hat{p}^b)). \tag{4.17}$$

For the current frame $t \geq 1$, if all symmetric KL distances $\{D_{sKL}(\hat{p}^t, \hat{p}^{K_j}), j = 1...i\}$ are greater than a fixed threshold[4] $\tau = 1.5$, frame t becomes the next key frame K_{i+1}, and is added to the set of current key frames. In this way, those frames with large information gain or having new information are selected, and those not selected can be explained by the key frames with a bounded deviation from one of the key frames in the symmetric KL distance. Figure 4.4 shows key frames selected from an example sequence. We preprocess the videos by background subtraction [55] and neighborhood-based noise removal to obtain single connected component for each frame. Person sub-images (rectangular patches) are extracted from bounding box

[4]the threshold is estimated such that on average, three key frames are selected for each track of about 30 frames.

Appearances from camera1

Appearances from camera2

Figure 4.5: List of sample appearances taken under two overlapping and widely separated cameras.

of foreground regions.

4.7 Experimental Results

We use the Honeywell appearance datasets for experimental evaluation. The classification and recognition performance is quantitatively analyzed in terms of Cumulative Match Curve (CMC) and Expected Relative Margin. The expected relative margin δ is defined as a geometric mean of relative margins for all test samples:

$$\delta = (\prod_{i=1}^{N_T} \frac{d_2^i}{d_1^i})^{1/N_T}, \tag{4.18}$$

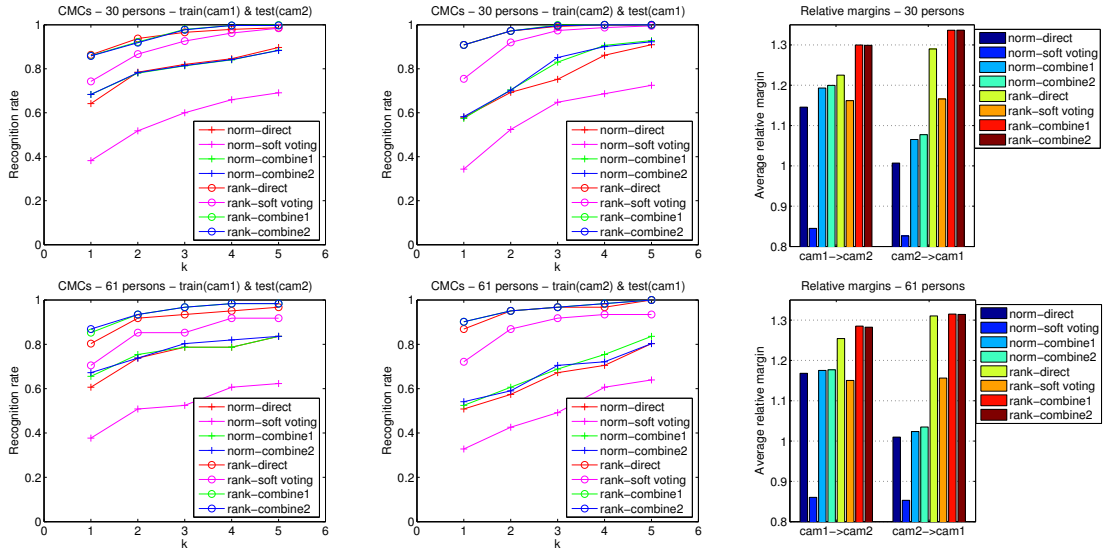


Figure 4.6: Recognition performance with respect to the increasing number of persons ($N = 10, 20, 30, 40, 50, 60, 61$) involved in training and testing. Here, only the cases for 30 and 61 persons are listed. ('norm': normalized color feature, 'rank': color rank feature, 'direct': appearance-based distance D_a, 'soft voting': discriminative information-based distance $D_d = D_{sv}$, 'combine1' and 'combine2': combined distances D_1, D_2.)

where N_T denotes the number of test samples, d_1^i denotes the distance of query q_i to the correct (same class) prototype, and d_2^i denotes the distance of query q_i to the closest incorrect (different class) prototype. Our test data include videos of 61 individuals taken by two overlapping cameras widely separated in space. Figure 4.5 shows samples from all 61 appearances. We can observe that the dataset has many appearance ambiguities because a limited number of people were used to create a large number of 'appearance' classes by a partial change of clothing for each person. Also there are significant illumination, viewpoint and pose changes across the two cameras.

Scalability: To show the scalability of our voting-based and combined approaches, we evaluated cumulative recognition rates and expected relative margins

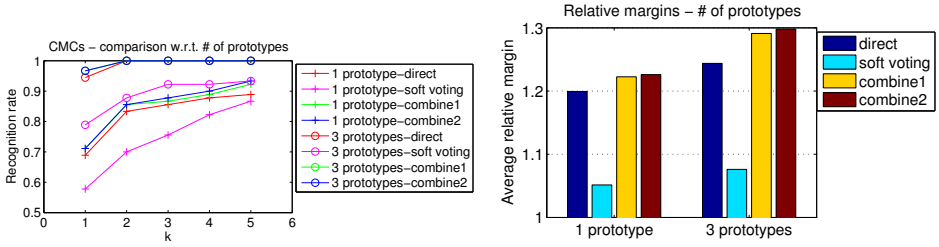


Figure 4.7: Recognition performance with respect to the changing number of training samples (prototypes) per class.

for increasing number ($N = 10, 20, 30, 40, 50, 60, 61$) of individuals involved in training and testing (Figure 4.6). For each of the 61 individuals, we select one key frame from a camera for training and one key frame from another camera for testing, and evaluated the performance using different features and different training-testing data (cam1 to cam2, cam2 to cam1). The results show that our combined approaches consistently perform better than the direct appearance-based method. More importantly, we note that our voting-based approach and combined approaches only have very small degradations with increasing numbers of people, while direct methods resulted in large degradation in recognition performance. This is more obviously seen for the case of color rank features. We can see that our discriminative information-based combined distances are more useful in recognizing large number of classes than the direct appearance-based distances.

Effects of Number of Training Samples: We compared recognition performance over the number of training samples per class (Figure 4.7). We select three key frames for each of 30 individuals from camera 2 as training samples and three key frames for each of 30 individuals from camera 1 as test samples. The figure shows that increasing the number of training samples per class improves cumulative

Approaches	Avg. Recog. Rate
direct comparisons	-
norm-direct-appearance 61 persons	0.56
norm-soft voting 61 persons	0.35
norm-combine1 61 persons	0.59
norm-combine2 61 persons	0.61
rank-soft voting 61 persons	0.71
rank-direct-appearance 61 persons	0.84
rank-combine1 61 persons	**0.88**
rank-combine2 61 persons	**0.89**
[135] rank-path length 61 persons	0.85
indirect comparisons	-
[41] model fitting 44 persons	0.59
[119] shape & appear. c. 99 persons	0.82

Table 4.3: Direct and indirect comparisons of top one recognition rates to state of the art work. Our combined approaches with (color rank + height) features are marked as 'bold'.

recognition rates and relative margins significantly.

Comparison with Other Approaches: We also compared the performance of our voting-based and combined approaches to the direct appearance-based approach in terms of average top one recognition rates on 61 individuals. Both cases of (Normalized Color + Height) and (Color Rank + Height) feature spaces show that our combined approaches improve top one recognition rates of direct appearance-based method by 5-6% (equivalently, the error rate is reduced by 31%) (Table 4.3). Results on the same dataset with a same number of individuals are compared to Yu *et al.* [135][5]. Using similar number of pixels (500 samples) per appearance, our combined approach obtained 4% better top one recognition rate (equivalently, the

[5]The result of [135] is obtained from their most recent experiments.

error rate is reduced by 27%) and is 5-10 times faster than [135]. This is because computing path-length features for all samples significantly slows down the process. We used the much simpler normalized height feature to achieve better performance. Indirect comparisons (Table 4.3) to Gheissari *et al.* [41] and Wang *et al.* [119][6] on datasets with similar number of people show that our approach is comparable to state of the art work on person recognition.

Computational Complexity: The computational complexity[7] of our learning algorithm is $O(N^2)$, while the complexity of our testing algorithm for a single query image is $O(N)$, which is the same as the traditional nearest neighbor method.

[6] In [41, 119], Datasets (which are publicly unavailable) of 44 and 99 individuals are used.

[7] Using about 500 samples per appearance, the learning time for 61 appearances is about 2 minutes, and the time for matching a single query image to 61 prototypes is less than 2 seconds in C++ on a Intel Xeon CPU 2.40GHz machine.

Chapter 5

Prototype-based Action Recognition

5.1 Introduction

Recently, action recognition has been a popular research topic in the vision community due to its wide applicability to multimedia analysis and video surveillance. It is the most primitive element in human movement analysis and event/activity analysis. Shape and motion have been shown to be the most important and useful visual cues for human action recognition.

Many studies have been performed on visual cues and features for robust action recognition. They can be roughly classified into three categories: geometry-based [62], motion-based [27, 31, 120, 121], appearance-based [28, 62, 111], and space-time feature-based [52, 79, 80, 89, 95, 101, 133]. The geometry-based approaches recover information about human body configuration, but they heavily rely on object segmentation and tracking, which is typically difficult and time consuming. The motion-based approaches extract optical flow features for recognition, but they rely on segmentation of the foreground for reducing effects of background flows. The appearance-based approaches use shape and contour information to identify actions, but they are vulnerable to cluttered complex backgrounds. The space-time feature-based approaches either characterize actions using space-time volumes [12,44,52,53,89,101,133] or using space-time interest points [25,59,60,79,80,95].

Acquisition of the space-time volumes usually requires foreground/background segmentation, which is itself a difficult problem in real scenarios (e.g. moving cameras). Space-time interest points are suitable only when there is significant variation of image intensity values in space and time dimension, so it is not particularly suitable for capturing smooth motions. Ref. [5] used trajectories of a finite number of (manually assigned) body interest points (called landmarks) to model and recognize human actions, however, it is hard to automatically obtain these landmarks in practical applications.

Recently, there have been approaches, *e.g.* [4, 49, 51, 60, 69, 72, 77, 94, 103], combining multiple features to detect and recognize actions. Laptev and Perez [60] used shape and motion cues to detect drinking and smoking actions. Jhuang *et al.* [51] introduced a biologically inspired action recognition system which used a hierarchy of spatial-temporal feature detectors. Liu *et al.* [69] combined quantized vocabularies of local spatial-temporal volumes and spin images. Shet *et al.* [103] combined shape and motion exemplars in a unified probabilistic framework to recognize gestures. Holte *et al.* [49] presented a view-invariant gesture recognition approach using depth and intensity information. Schindler and Gool [94] extracted both form and motion features from an action snippet to modeling and recognizing actions. Niebles and Fei-Fei [77] introduced a hierarchical model and a hybrid usage of static shape features and spatial-temporal features for action classification. Ahmad and Lee [4] combined shape and motion flows to classify actions from multi-view image sequences. Mikolajczyk and Uemura [72] extracted a large set of low dimensional local features to learn many vocabulary trees to allow efficient action recognition

and perform simultaneous action localization and recognition.

On the other hand, some approaches represented a human action as a sequence of basic units called *exemplars* [28, 103, 123, 124] or *primitives* [111]. Elgammal *et al.* [28] viewed an action as a sequence of leaned shape exemplars and imposed temporal constraints between different exemplars by Hidden Markov Model (HMM) [85]. Later, Shet *et al.* [103] extended this approach by including motion cues to improve recognition results. Weinland and Boyer [123] represented an action as a set of distances from silhouette exemplars to the frames of the action sequence and then classified the action sequence using a standard Bayes classifier. In Weinland *et al.* [124], 3D exemplars are projected onto a 2D image plane so that they can be directly compared to image observations. Thurau *et al.* [111] represented actions by histograms of pose primitives and then performed action recognition by matching histograms of pose primitives. Souvenir and Babbs [109] provided a compact representation of primitive action for view-variant action recognition using manifold learning. Wang *et al.* [121] introduced a hierarchical probabilistic model of human actions based on a bag of motion words, where each frame corresponds to a motion word.

Categorization methods are mostly based on machine learning or pattern recognition techniques. Ref. [28, 103] incorporated temporal constraints between examplars using HMM, but it required a large training set and it is difficult to choose appropriate examplars and HMM parameters (such as number of states for each action category). Holte *et al.* [49] used an edit distance to identify which of the learned gestures best explains a query sequence of pose primitives, but the require-

ment of pose estimation and primitive matching limits the approach's applicability. The method in [111] used n-Gram models to represent local temporal context and recognized actions based on histogram comparisons. However, local temporal context modeling of the n-Gram model is insufficient for handling complex actions since temporal consistency cannot be globally guaranteed. Efros *et al.* [27] computed motion-to-motion similarity matrices and then used a k-NN classifier for classifying a query action. Schuldt *et al.* [95] combined local space-time features with a Support Vector Machine (SVM) classifier for action recognition. Fanti *et al.* [30] presented a hybrid probabilistic model for human motion recognition and modeled the human motion as a triangulated graph. But this approach can be highly dependent on the quality of point tracking. Sminchisescu *et al.* [107] integrated discriminative Conditional Random Field (CRF) and Maximum Entropy Markov Models (MEMM) to recognize human actions. Shi *et al.* [105] combined semi-Markov model and HMM to conduct human action segmentation and recognition. Vitaladevuni *et al.* [117] presented a Bayesian framework for action recognition from psych-kinesiological observations that ballistic movements are the basic units of human movements. Wang and Suter [118] used kernel principal component analysis (KPCA) to get a low-dimensional embedded space by performing nonlinear dimensionality reduction and then explored factorial conditional random field (FCRF) to classify human actions in the embedded space.

Previous work mostly relied on high dimensional descriptors for modeling action frames and used nearest neighbor (NN) or k-nearest neighbor (k-NN) classifiers for database frame (or examplar) retrival and action recognition, *e.g.* Gorelick and

Blank *et al.* [11,44] and Efros *et al.* [27]. This can be expensive when the size of the examplar database is large. On the other hand, existing approaches [86,134] mostly assumed simple backgrounds or static cameras, and did not explicitly consider the challenging cases of dynamic backgrounds and the presence of other moving objects.

In contrast, we introduce a very efficient, prototype-based approach for action recognition. Our approach extracts rich information from observations but performs recognition efficiently via prototype matching and look-up table indexing. In addition, it has an advantage of tolerating complex dynamic backgrounds by probabilistic model fitting instead of brute force search of pose space.

The block diagram of our system is shown in Figure 5.1. During training, background subtraction segments a person and localizes an action interest region around the person from which shape-motion descriptors are computed. Next, action prototypes are learned via k-means clustering. During testing, humans are detected and tracked using appearance information, and a frame-to-prototype correspondence is established by nearest neighbor search. Finally, actions are recognized based on dynamic prototype sequence matching. Similarity matrices used for the matching are rapidly obtained by look-up table indexing, which is an order of magnitude faster than the brute-force computation of frame-to-frame distances. Our main contributions are three-fold:

- Actions are modeled by learning their prototypes in a joint shape-motion space via k-means clustering.

- Frame-to-frame distances are rapidly estimated via fast look-up table indexing.

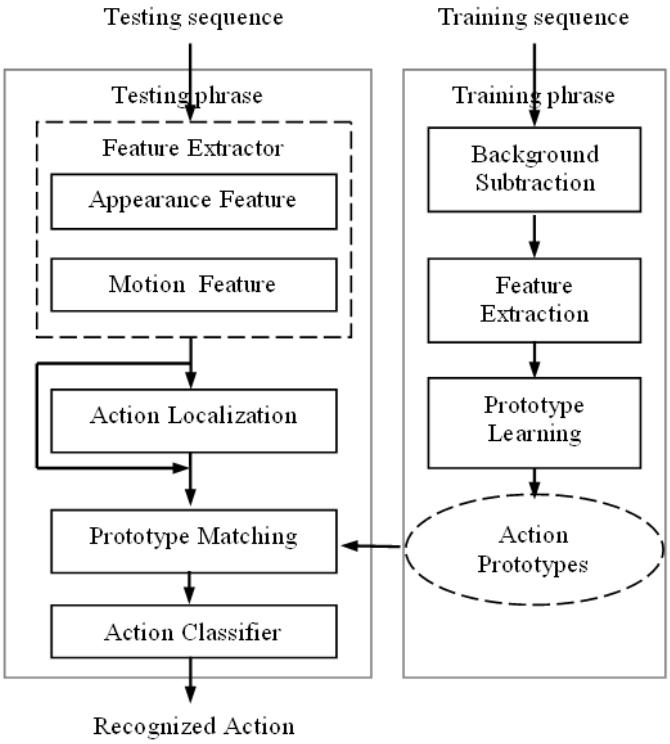


Figure 5.1: Overview of our approach.

- A probabilistic framework is introduced for robustly detecting and matching prototypes, and recognizing actions against dynamic backgrounds.

5.2 Action Representation

For representing and describing actions, action interest regions should be defined precisely around an actor (or human) so that the representation can be location and scale invariant.

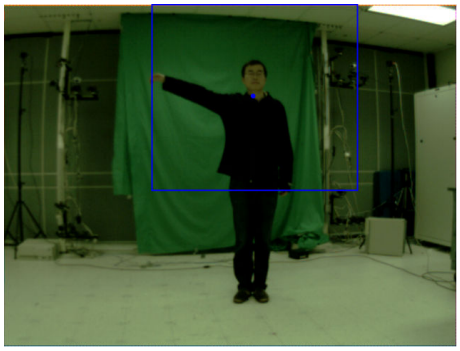
5.2.1 Action Interest Region

We define an action (or gesture) interest region as a (square) region around a reference point (*e.g.* the center of the square) in the human body. For full-body actions, the reference point is defined as the center of the bounding box[1] around the localized person. The side-length of the square region is proportional to the height of the bounding box. For gestures, similarly, the reference point is defined as a point on the bounding box's center-vertical axis. For the gesture dataset, the reference point is at a distance of 1/8 of the box's height from the top of the box, while it is at the center of the bounding box for the action dataset. The side-length of the square region is proportional to the height of the bounding box. Since we compute action descriptors in each frame solely from its action interest region, robustly detecting and tracking human bounding boxes (under dynamic backgrounds) is critical for overall action recognition performance. We will discuss details of localization and tracking in Sec. 5.4. Examples of action interest regions are illustrated in Figure 5.2.

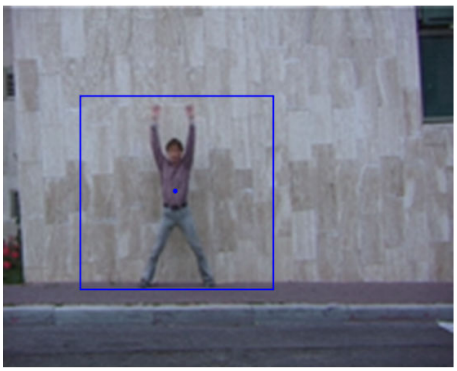
5.2.2 Shape-Motion Descriptor

A shape descriptor for an action interest region is represented as a feature vector $D_s = (s_1...s_{n_s}) \in \mathcal{R}^{n_s}$ by dividing the action interest region into n_s square grids (or sub-regions) $R_1...R_{n_s}$. The i-th component of a raw shape descriptor measures occupancy moments (averages of intensities) of that region R_i. In the training phase,

[1]Note that a "bounding box" here means a human bounding box, so is different from action interest region. An action interest region is derived from a "bounding box" by a one-to-one mapping.



(a) A gesture interest region from the Keck gesture dataset



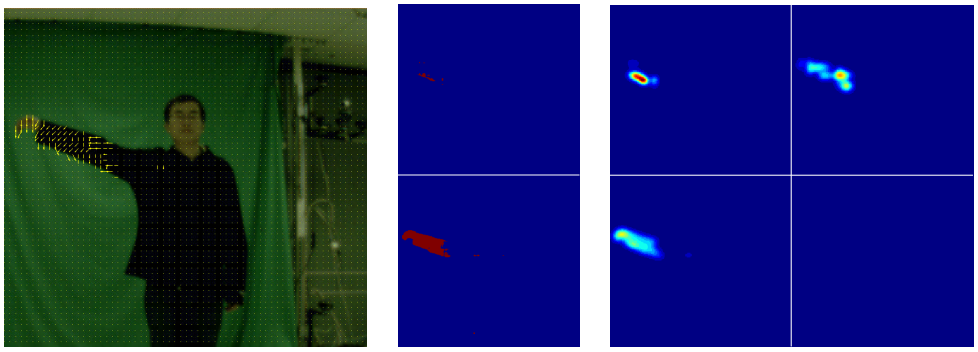
(b) Action interest region from the Weizmann action dataset

(c) Action interest region from the KTH action dataset

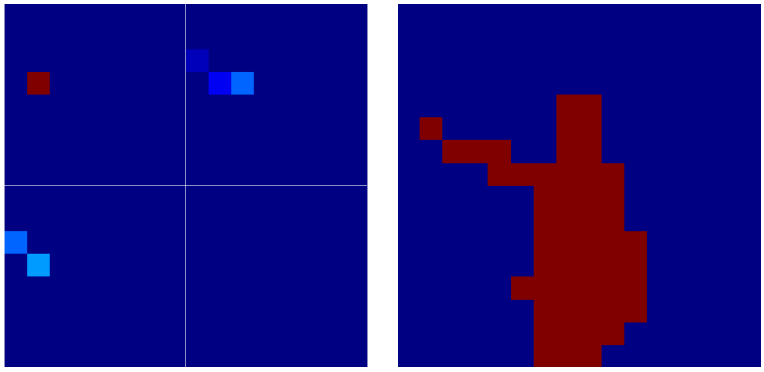
Figure 5.2: Examples of gesture and action interest regions.

shape observations are binary silhouettes obtained by background subtraction; and in the testing phase, the shape observations are either binary silhouettes from background subtraction (under static backgrounds) or normalized part-appearance-based likelihood maps (under dynamic backgrounds).

A motion descriptor for an action interest region is represented as a feature vector $D_m = (QFb_x^+, QFb_x^-, QFb_y^+, QFb_y^-) \in \mathcal{R}^{n_m}$, where '$QFb$' refers to quantized, blurred flow. We use the robust motion flow feature introduced in [27] to compute the motion descriptor as follows. The optical flow field F of an action interest region is divided into horizontal and vertical components, F_x and F_y, each of which is then half-wave rectified into four non-negative channels $F_x^+, F_x^-, F_y^+, F_y^-$. These



(a) Optical flow F (b) F_x, F_y (c) $Fb_x^{+,-}, Fb_y^{+,-}$



(d) $QFb_x^{+,-}, QFb_y^{+,-}$ (e) G_s

Figure 5.3: Visualization of the shape and motion descriptors. (a) An optical flow field of an action interest region. (b) Motion flow features in horizontal and vertical directions. (c) Gaussian blurred motion observation for four channels. (d) The motion descriptor. (e) The shape descriptor.

channels are blurred with a Gaussian kernel to form the low-level motion observations $(Fb_x^+, Fb_x^-, Fb_y^+, Fb_y^-)$. Then, as in computing shape descriptors, we map each channel of the motion observations into low resolution by averaging them inside uniform grids overlaid on the interest region. The resulting four channel descriptors are L_2 normalized independently and concatenated to form the motion descriptor $G_m = (QFb_x^+, QFb_x^-, QFb_y^+, QFb_y^-)$. Figure 5.3(a), 5.3(b), and 5.3(c) show the process of computing motion observations, and Figure 5.3(d) and 5.3(e) illustrate the final motion and shape descriptors, respectively.

We concatenate shape and motion descriptors D_s and D_m to form joint shape-

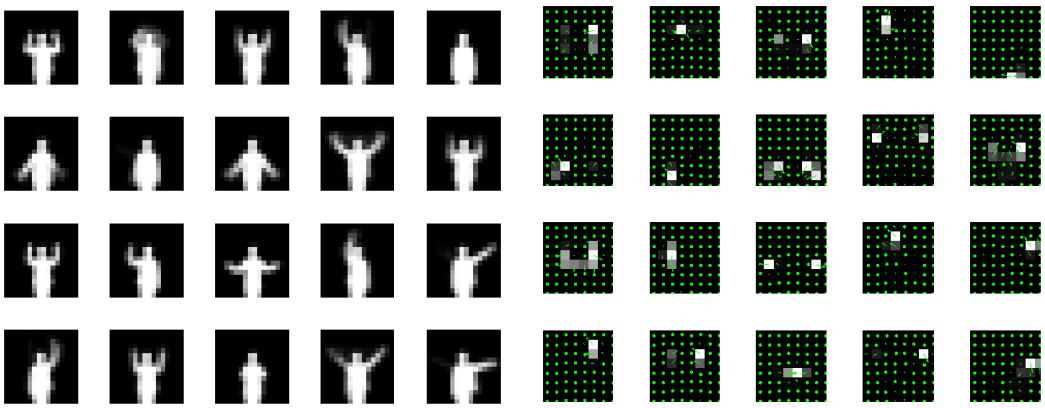
motion descriptors: $D_{sm} = (D_s, D_m) \in \mathcal{R}^{n_{sm}}$, where $n_{sm} = n_s + n_m$ is the dimension of the combined descriptor. The distance between two gesture frames, *i.e.* two shape-motion descriptors, D^a_{sm} and D^b_{sm}, is computed using the Euclidean distance metric.

Based on the relative importance of shape and motion cues, we can learn a weighting scheme for the shape and motion components of $D_{sm} = (w_s D_s, w_m D_m)$ (where the weights are chosen such that $w_s^2 + w_m^2 = 1$), where the optimal weights w_s, w_m can be estimated using a validation set by maximizing the recognition rate.

5.2.3 Learning Shape-Motion Prototypes

Following the idea of [28, 111], we represent an action as a set of basic action units. We refer to these action units as action prototypes. More formally, an action, G of length n (the number of frames in G), is represented as a sequence of prototypes $G = (g_1 ... g_n)$, where each prototype g_i is represented as a high-dimensional feature vector consisting of a shape component g_{si} and a motion component g_{mi}.

Given the set of shape-motion descriptors from all action frames of the training set, we perform k-means clustering in a joint shape-motion space using the Euclidean distance. Since both of our shape and motion descriptors are vector quantization of original shape and motion observations, the Euclidean distance metric is reasonable for clustering the joint shape-motion descriptors. From experiments, we found that a simple k-means clustering algorithm is sufficient for learning prototypes in the joint shape and motion space. The cluster centers are then used as the action prototypes,



(a) Shape components of prototypes (b) Motion components of prototypes

Figure 5.4: Visualization of shape and motion components of learned prototypes for $k = 20$. The shape component is represented as 16×16 grids and the motion component is represented as four (orientation channels) 8×8 grids. In the motion component, grid intensity indicates motion strength and 'arrow' indicates the dominant motion orientation at that grid.

hence action prototypes are also called shape-motion prototypes. Examples of the shape and motion components of our learned action prototypes are visualized in Figure 5.4. A merit of representing an action with the above representation is that we can construct a prototype-to-prototype distance matrix (computed off-line in the training phase) use it as a look-up table to speed up the action recognition process.

5.3 Action Recognition

The action recognition process is divided into two steps. The first step is frame-to-prototype matching, and the second step is prototype-based sequence matching.

5.3.1 Probabilistic Framework for frame-to-prototype matching

Let random variables $V = (V^{ap}, V^{mf})$ (where V^{ap} and V^{mf} denote appearance and motion flow observations, respectively) be observations from an image frame, θ be a prototype random variable chosen from a set of k learned shape-motion prototypes $\Theta = (\theta_1, \theta_2 ... \theta_k)$, and α denote the location (x, y) and scale s parameters of the actor. Then, the frame-to-prototype matching problem is equivalent to maximizing the joint likelihood $p(V, \theta, \alpha)$ of the image observation, prototype, and location. Assuming V has a uniform prior (*i.e.* $p(V)$ is a uniform distribution), based on the Bayes rule, we can decompose the joint likelihood $p(V, \theta, \alpha)$ into likelihoods of person localization term and frame-to-prototype matching as follows:

$$\begin{aligned} p(V, \theta, \alpha) &\propto p(\theta, \alpha | V) \\ &= p(\theta | V, \alpha) p(\alpha | V) \\ &= p(\theta | V(\alpha)) p(\alpha | V), \end{aligned} \tag{5.1}$$

where $V(\alpha)$ denotes localized observation, *i.e.* observation specified by the localized interest region. For simplicity, we maximize $p(V, \theta, \alpha)$ sequentially by separating optimization of localization and action prototypes. Maximization of this likelihood $p(\alpha | V)$ is exactly the problem of object localization, and given α, maximization of $p(\theta | V(\alpha))$ is the problem of frame-to-prototype matching.

5.3.2 Independent Frame-to-Prototype Matching

Then, given the localized observations $V(\alpha)$ (assuming known localization α), the prototype likelihood $p(\theta|V(\alpha))$ is evaluated and maximized via shape-motion prototype matching. Assuming α is given, we perform the likelihood decomposition as follows:

$$p(\theta|V(\alpha)) \propto p(V(\alpha)|\theta)p(\theta). \tag{5.2}$$

In practice, the prior probability $p(\theta)$ can be explicitly estimated by counting occurrences of prototypes in the training set. Here, for simplicity we assume $p(\theta)$ is uniform, hence we estimate the optimal match, shape-motion prototype θ^*, according to maximum likelihood estimation (MLE) as:

$$\theta^* = \arg\max_{\theta\in\Theta} p(\theta|V(\alpha)) = \arg\max_{\theta\in\Theta} p(V(\alpha)|\theta). \tag{5.3}$$

We only search using the space (set) of learned action prototypes $\theta \in \Theta$ instead of the entire high-dimensional pose space, making the method computationally efficient. Maximizing the prototype likelihood $p(\theta|V(\alpha))$ is equivalent to minimizing the distance d between the descriptor determined by observations $V(\alpha)$ and a prototype precomputed in the training phase, since the prototype likelihood can be modeled as $p(\theta|V(\alpha)) = exp(-d)$, where the distance d is directly computed as the Euclidean distance between joint shape-motion descriptor $D(V) = (D_s(V), D_m(V))$ and a prototype descriptor $D(\theta) = (D_s(\theta), D_m(\theta))$.

5.3.3 Prototype-based Sequence Matching

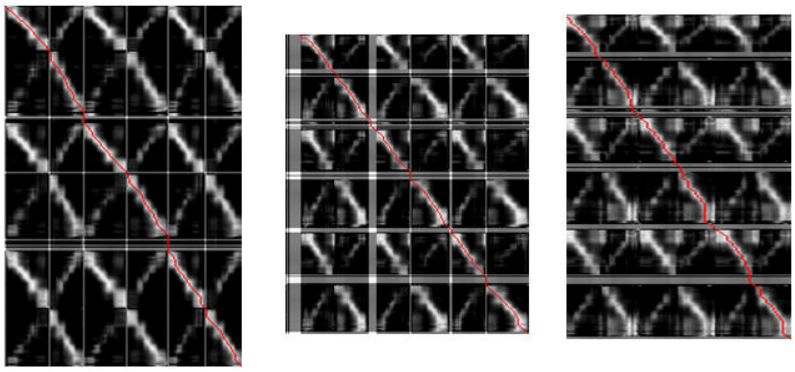
5.3.3.1 Dynamic Time Warping

We use the dynamic time warping algorithm [93] to measure the distance between actions. Suppose we have two actions G_x and G_y of lengths $|X|$ and $|Y|$, $G_x = x_1, x_2, ..., x_{|X|}$ and $G_y = y_1, y_2, ..., y_{|Y|}$.

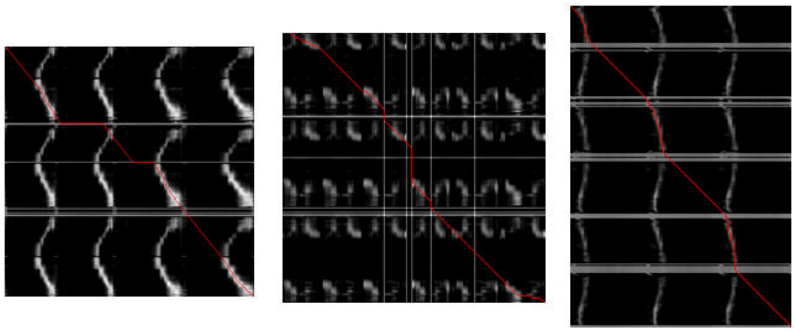
Finding the best match of these two sequences is equivalent to finding a minimum-cost path through a cost matrix to align these two sequences in time. The warping path is defined as $W = w_1, w_2, ..., w_K$, where K is the length of path and satisfies $\max(|X|, |Y|) \leq K \leq |X| + |Y|$. The cost matrix is constructed by computing all the distance $d(x_i, y_j)$ between every pair of frames x_i and y_j from the action sequences G_x and G_y. The distance $Dist(G_x, G_y)$ between two actions G_x and G_y is given as the average of distances on the minimum-cost-path obtained from dynamic time warping:

$$Dist(G_x, G_y) = \min \left(\sum_{k=1}^{K} dist(x_{k,i}, y_{k,j})/K \right), \tag{5.4}$$

where $dist(x_{k,i}, y_{k,j})$ is the distance between two frames $x_{k,i}$ and $y_{k,j}$ at the k-th element of the warping path. Distance $d(x_{k,i}, y_{k,j})$ can be computed via direct Euclidean distance or look-up table of distances between any two prototype. This is also the reason for the computation difference between feature-to-feature distance-based approach and prototype-based approach.



(a) Same gestures performed by different persons



(b) Different gestures performed by different persons

Figure 5.5: Gesture matching results by dynamic time warping. Columns and rows with low variation in intensities indicate that the corresponding frame is static, *i.e.* the average magnitude of motion flows is very small.

5.3.3.2 Sequence Matching

We first compute the cost (or frame-to-frame distance) matrix between a test action G and each of the model actions $\{M_t\}_{t=1...n}$ in the training set, and then use dynamic time warping to compute the distances $\{D_t\}_{t=1...n}$, where $D_t = D(G, M_t)$. Then, we find the optimal match G_t^* using the minimum distance (or k-nearest neighbor) classification.

To efficiently compute the frame-to-frame distance matrices, we represent a test action G as a sequence of closest (nearest neighbor) shape-motion prototypes learned in the training phase. Then, the distance between two actions can be repre-

sented as the distance between two sequences of prototypes instead of two sequences of feature descriptors; and consequently the look-up table constructed in the training phase can be used to speed up the computation of descriptor-based frame-to-frame distance matrices.

Figure 5.5 shows examples of gesture matching using the dynamic time warping algorithm. The result shows that if the test action is matched correctly to a model action, the red (dynamic time warping) path in the Figure 5.5(a), as expected, is close to diagonal; otherwise, it can be arbitrary, as shown in Figure 5.5(b).

Two different versions of our approach used to obtain action-to-action distance $d(x_i, y_i)$ are: (1) Descriptor distance-based approach directly computes frame-to-frame distances, (2) Prototype-based approach approximates frame-to-frame distances by indexing the look-up table (of prototype-to-prototype distances) precomputed during training. Frame-to-prototype distances are only used to find nearest prototypes for test frames in the prototype-based approach. We reject non-modeled gestures by thresholding action-to-action distances, where the threshold is estimated via cross-validation.

5.4 Action Localization

For different test datasets, we use different available appearance information for detecting and tracking the person (or gesturer) in the sequence. We first detect the person in the first frame and then track his bounding box through subsequent frames.

More formally, assuming observations V^{ap}, V^{mf} are independent, and $p(\alpha)$ is uniform, the location probability is represented as:

$$\begin{aligned} p(\alpha|V) &\propto p(V|\alpha) \\ &\propto p(V^{ap}|\alpha)p(V^{mf}|\alpha) \\ &\propto p(V^{ap}|\alpha). \end{aligned} \tag{5.5}$$

In order to maximize $p(\alpha|V)$, now we only need to maximize $p(V^{ap}|\alpha)$. We use foreground segmentation maps or foreground color-likelihood maps to model the location likelihood $p(V^{ap}|\alpha)$.

For test datasets created with fixed cameras, such as the Weizmann dataset and the KTH dataset, we use the codebook model-based background subtraction [55] to obtain binary foreground images, and then use integral images [115] to evaluate the coverage information for arbitrary rectangular regions. The resulting foreground images might be noisy depending on the quality and contrast of input video data.

In contrast, for test datasets created with moving cameras, such as the Keck gesture dataset, tracking is primarily based on appearance information. The location likelihood $P(V^{ap}|\alpha)$ is expressed as:

$$p(V^{ap}|\alpha) = \prod_{j \in \{h,t,l\}} p(V^{ap}(j)|\alpha), \tag{5.6}$$

where h, t, and l denote the three parts, head, torso, and legs, respectively. Taking

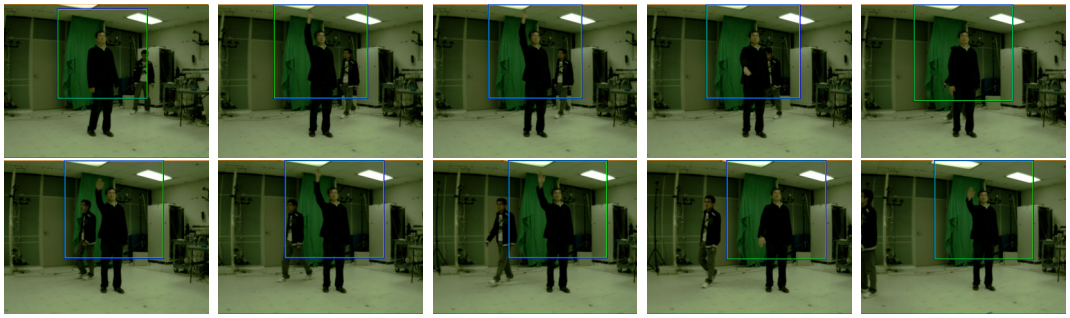


Figure 5.6: Examples of action localization result on the Keck Gesture dataset. Our localization method can avoid the influence of a sceondary person moving around in the background and a moving camera.

logs to both sides, the equation becomes:

$$L(V^{ap}|\alpha) = \sum_{j \in \{h,t,l\}} L(V^{ap}(j)|\alpha), \tag{5.7}$$

where, $L(V^{ap}(j)|\alpha)$ is modeled as the difference of average appearance-based likelihood between the inside and the outside of a rectangle surrounding a hypothetical part location. Intuitively, this is like a generalized Laplacian operator and favors situations in which the person matches well inside a detection window, but not coincidentally because the image locally mimics the color distribution of the person. We efficiently compute the coverage information using integral images [115]. We use a generic human detector such as [22] for locating the person in the first frame, then build the color appearance model of the person and use it throughout. We employ a tracker such as particle filter [7, 81] for tracking the person in 3D (location and scale) space. The likelihood in Equation 5.7 is used as the observation model of the tracker.

For building human appearance models, we divide the human body into three

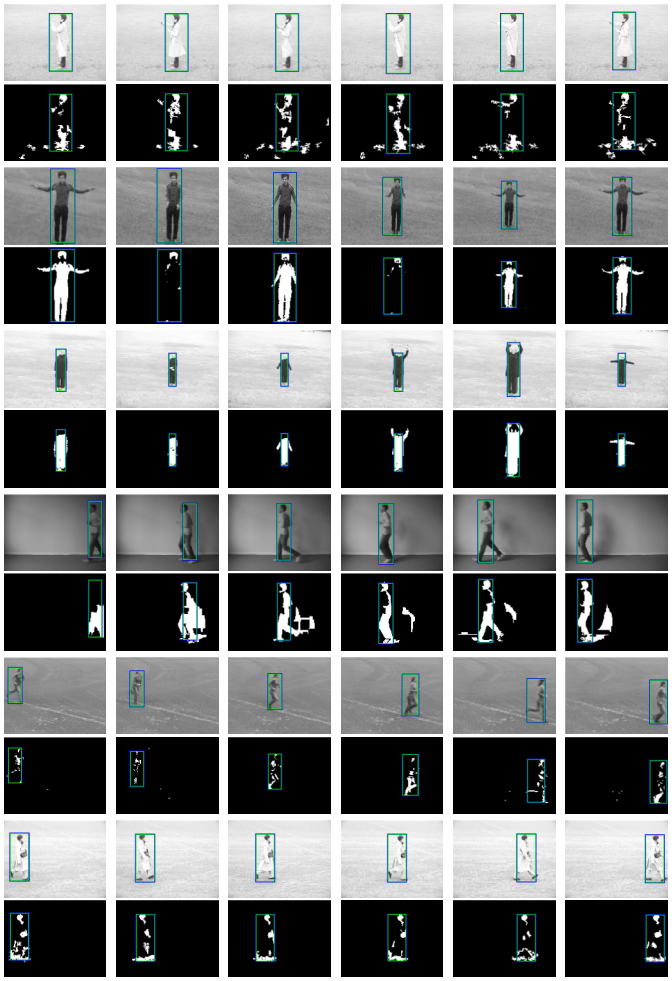
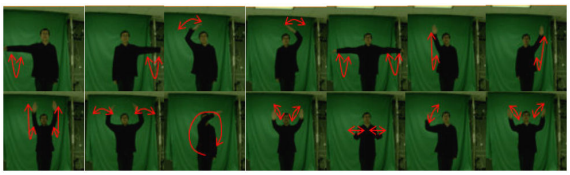


Figure 5.7: Examples of action localization and tracking results on the KTH dataset. Our localization method effectively handled influences of shadows, fast camera movements, low contrast, poor background subtraction, and even missing human silhouettes for a short period of time.

parts, head, torso, and legs. Each part's appearance model is represented by a color/grayvalue distribution. Here, we assume color information is available and build color-appearance models for tracking. To allow multi-modality of the underlying density, kernel density estimation is used to obtain the color distribution. Given a set of pixels $\{x_i\}_{i=1...N}$ from an image region, where x_i is a d-dimensional feature vector representing a color, we can calculate the probability that an observed



(a) The Keck Gesture Dataset

(b) The Weizmann Action Dataset

(c) The KTH Action Dataset

Figure 5.8: Datasets. (a) The Keck gesture dataset consisting of 14 different gestures, (b) The Weizmann action dataset consisting of 10 different actions, (c) The KTH action dataset consisting of 6 different actions collected under 4 different scenarios.

pixel y belongs to that image region as [56]:

$$p(y) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d} K_{\sigma_j}(y_j - x_{ij}). \tag{5.8}$$

where the function $K(\cdot)$ is a kernel function and dimension j of the feature vector has bandwidth σ_j. To tolerate illumination variability, the normalized `rgs` color space($r = \frac{G}{R+G+B}, g = \frac{G}{R+G+B}, s = \frac{R+G+B}{3}$) is used. We chose the Gaussian as $K(\cdot)$ in our experiment, and the channel bandwidths are set to $\sigma_r = 0.02$, $\sigma_g = 0.02$ and $\sigma_s = 20$.

Figure 5.6 shows some examples of our localization results on the Keck Gesture Dataset. There is a secondary person continuously moving around in the background and the camera is often moving fast. Our localization approach can get rid of these influences. Figure 5.7 presents some examples of the localization results on the KTH

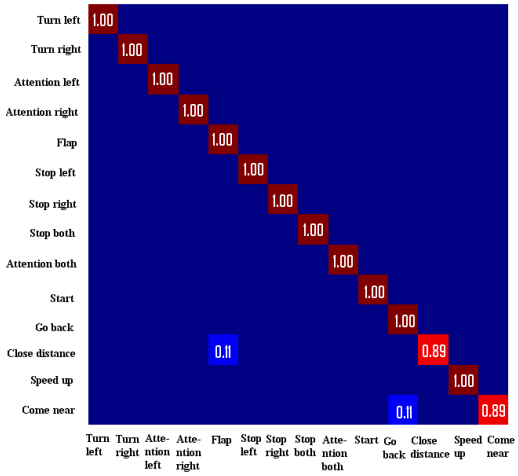


Figure 5.9: Confusion matrix for gesture recognition against a static background.

Table 5.1: Feature-based recognition result (leave-one out procedure).

method	recog. rate (%)
motion only	92.86
shape only	92.86
joint shape and motion	95.24

dataset. From our experiments, we found that our localization method can detect and track objects even with very poor silhouettes from background subtraction due to the robustness of our tracking algorithm. That is to say, our approach can be applied to poor binary silhouettes, which is very useful for practical applications.

5.5 Implementation details

In this section we give implementation details of our recognition approach in order to guarantee the reproducibility of results of our approach. The standard deviation σ of the gaussian kernel for blurring the four non-negative channels

Table 5.2: Prototype-based recognition result (leave-one out procedure). Joint motion and shape descriptors are used for the evaluation. The average time is computed as the average of computing an action-to-action similarity matrix.

method	recog. rate(%)	avg. time (ms)
Our method: descriptor dist.	95.24	154.5
Our method: look-up(20 prot.)	90.48	21.8
Our method: look-up(40 prot.)	90.48	22.2
Our method: look-up(60 prot.)	90.48	22.6
Our method: look-up(80 prot.)	90.48	23.2
Our method: look-up(100 prot.)	92.86	25.6
Our method: look-up(120 prot.)	90.48	22.3
Our method: look-up(140 prot.)	92.86	22.7
Our method: look-up(160 prot.)	95.24	23.2
Our method: look-up(180 prot.)	95.24	25.6
Shet *et al.* [103]	83.7	N/A

Table 5.3: Feature-based recognition result using a moving camera viewing a dynamic background.

method	recog. rate (%)
motion only	87.5
shape only	53.57
joint shape and motion	91.07

$F_x^+, F_x^-, F_y^+, F_y^-$ is set to 5. Before concatenating a shape descriptor G_s and a motion descriptor D_m to form a joint shape-motion descriptor D_{sm}, the shape descriptor D_s and motion descriptor D_m are normalized by L_2 norm, respectively. We found that this independent channel normalization scheme is crucial for obtaining high recognition rates. In the training phase, we exclude frames (or descriptors) with no motion or shape information from the k-means clustering input data, *i.e.* those frames of which L_2-norm of raw shape and motion descriptors are excluded.

In the testing phase, computation of shape descriptor D_s is different for dif-

Table 5.4: Prototype-based recognition performance using a moving camera viewing a dynamic background. The average time is the average of the time needed to compute an action-to-action similarity matrix.

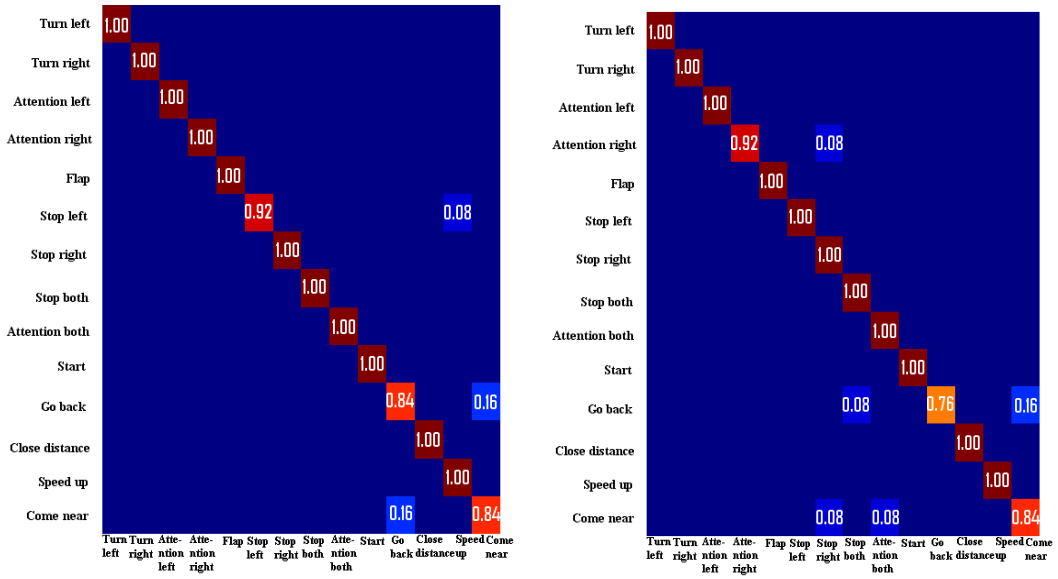
method	recog. rate (%)	avg. time (ms)
descriptor dist.	91.07	96.5
look-up(20 prot.)	55.36	7.2
look-up(40 prot.)	75	7.3
look-up(60 prot.)	76.79	7.4
look-up(80 prot.)	82.14	7.2
look-up(100 prot.)	80.36	7.2
look-up(120 prot.)	89.29	7.2
look-up(140 prot.)	82.14	7.3
look-up(160 prot.)	82.14	7.7
look-up(180 prot.)	89.29	7.8

Table 5.5: Feature-based recognition result on the Weizmann dataset.

method	recog. rate (%)
motion only	88.89
shape only	81.11
joint shape and motion	100

ferent datasets used in our experiments. For the Keck gesture dataset, we use the normalized color-part-appearance-based likelihood maps to compute the shape descriptors because this dataset can not simply use background subtraction to obtain binary silhouettes. For the Weizmann action dataset and the KTH action dataset, we perform background subtraction for each action sequence and simply compute the shape descriptors using the resulting binary silhouettes.

Our recognition approach classifies an action by matching it to any of the model actions in the training set and then performing k-NN classification. The range of k in k-means clustering was set by cross-validation on a validation set



(a) Descriptor-based approach (b) Prototype-based approach ($k = 180$)

Figure 5.10: Confusion matrices for gesture recognition using a moving camera viewing gestures against dynamic backgrounds.

during training. Typically, the optimal k is assigned to the value which achieves the best overall recognition rate on the validation set. It is highly dependent on the characteristics of individual datasets. For the Keck gesture dataset and Weizmann dataset, varying k from 80 to 180 results in stable recognition rates from our experiments, while for the KTH dataset, the optimal range of k is from 200 to 300.

Because the action-to-action similarity matrix is symmetric, we only compute its elements in the upper or lower triangle for speeding up the recognition process.

5.6 Experiments

We evaluated our approach both on a gesture dataset and two action datasets in terms of recognition rate and computation time. The shape-motion descriptor is 512-dimensional which consists of a 256-dimensional shape descriptor and a $64 \times 4 =$

256-dimensional motion descriptor.

5.6.1 Evaluation on the Keck Gesture Dataset

Similar to [103], we collected a dataset consisting of 14 different gesture classes which are a subset of military signals [114]. Because we collected the dataset in our Keck laboratory, we named it the Keck Gesture Dataset. Figure 5.8(a) shows sample training frames of the gesture data. Compared to the dataset used in [103] which assumes a static camera and simple background both for training and testing, our dataset has the same classes of gestures but much more challenge due to moving cameras, moving objects and dynamic backgrounds. These challenges are very common for human-robot interaction.

The gesture dataset is collected using a color camera with 640×480 resolution. Each of the 14 gestures is performed by three people. In each sequence, the same gesture is repeated three times by each person. Hence there are $3 \times 3 \times 14 = 126$ video sequences for training which are captured using a fixed camera with the person viewed against a simple, static background. There are 168 video sequences for testing which are captured from a moving camera and in the presence of background clutter and other moving objects.

5.6.1.1 Gesture Recognition against a Static Background

We evaluated our recognition approach based on a leave-one-out experiment using the training data. The confusion matrix is shown in Figure 5.9. Table 5.1

shows that the recognition rate of our approach using the joint shape-motion descriptor outperforms using the shape only feature descriptor or motion only feature descriptor. The recognition rate of our approach is 95.24% which is much better than Shet *et al.* [103] which reported an overall recognition rate of 83.7% on a simpler gesture dataset than ours.

Table 5.2 shows that the prototype-based approach achieves the same recognition rate as using the more computationally demanding descriptor-based approach. When $k = 160 or 180$, the prototype-based approach obtained 95.24% recognition rate which is the same as the result of the descriptor-based approach, but the computational cost of the prototype-based approach is much lower that of the descriptor-based approach.

5.6.1.2 Gesture Recognition against a Dynamic Background

This experiment was performed using a moving camera viewing the gesturer against a dynamic background, where one person (who is regarded as the gesturer) performed the specified fourteen gestures in a random order and the other person (who is regarded as 'noise') moved continuously around the gesturer making recognition more challenging. The experimental results using different features are shown in

Table 5.3. The joint shape-motion descriptor outperforms pure shape and motion-based features. Pure shape-based features obtained poor performance. This is because appearance-based likelihood maps were too noisy, probably due to the

strong color similarity between the gesturer and some parts of the background and the distracting person at times. On the other hand, because the distracting person did not always overlap with the gesturer, the camera motion was not always rapid. So there are many frames with small background flows. the pure motion-based approach obtained quite good performance due to the robustness of the motion descriptors to small background flows in many gesture frames.

As shown in Table 5.4, the prototype-based approach achieved an accuracy similar to the descriptor-based approach, but is an order of magnitude faster. Figures 5.10(a) and 5.10(b) show the confusion matrices for both the descriptor-based and the prototype-based approaches.

5.6.2 Evaluation on the Weizmann Action Dataset

The Weizmann dataset [44] contains 90 videos of 10 actions performed by 9 different people. Example frames of the this dataset are shown in Figure 5.8(b). We used background subtraction to obtain a single largest foreground blob and localize the person. We performed leave-one-out experiments using nearest neighbor classification to evaluate both the joint shape-motion descriptor and the prototype-based recognition method. Table 5.5 shows comparative results of our joint shape-motion descriptors with 'shape only' and 'motion only' descriptors in terms of average leave-one-out recognition rate. The joint shape-motion descriptors obtained 100% recognition while 'shape only' and 'motion only' descriptors obtained much lower recognition rates.

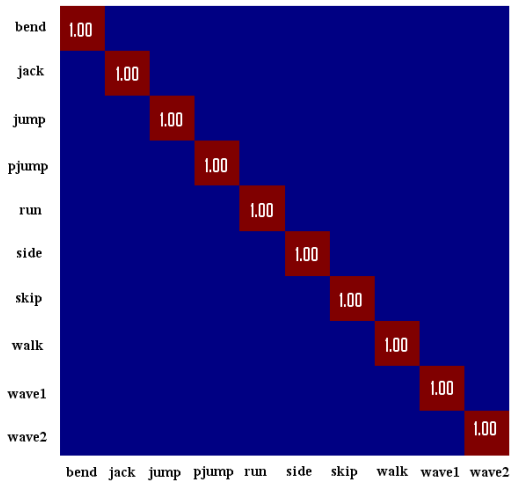


Figure 5.11: Confusion matrices on the Weizmann dataset using the prototype-based approach ($k = 180$).

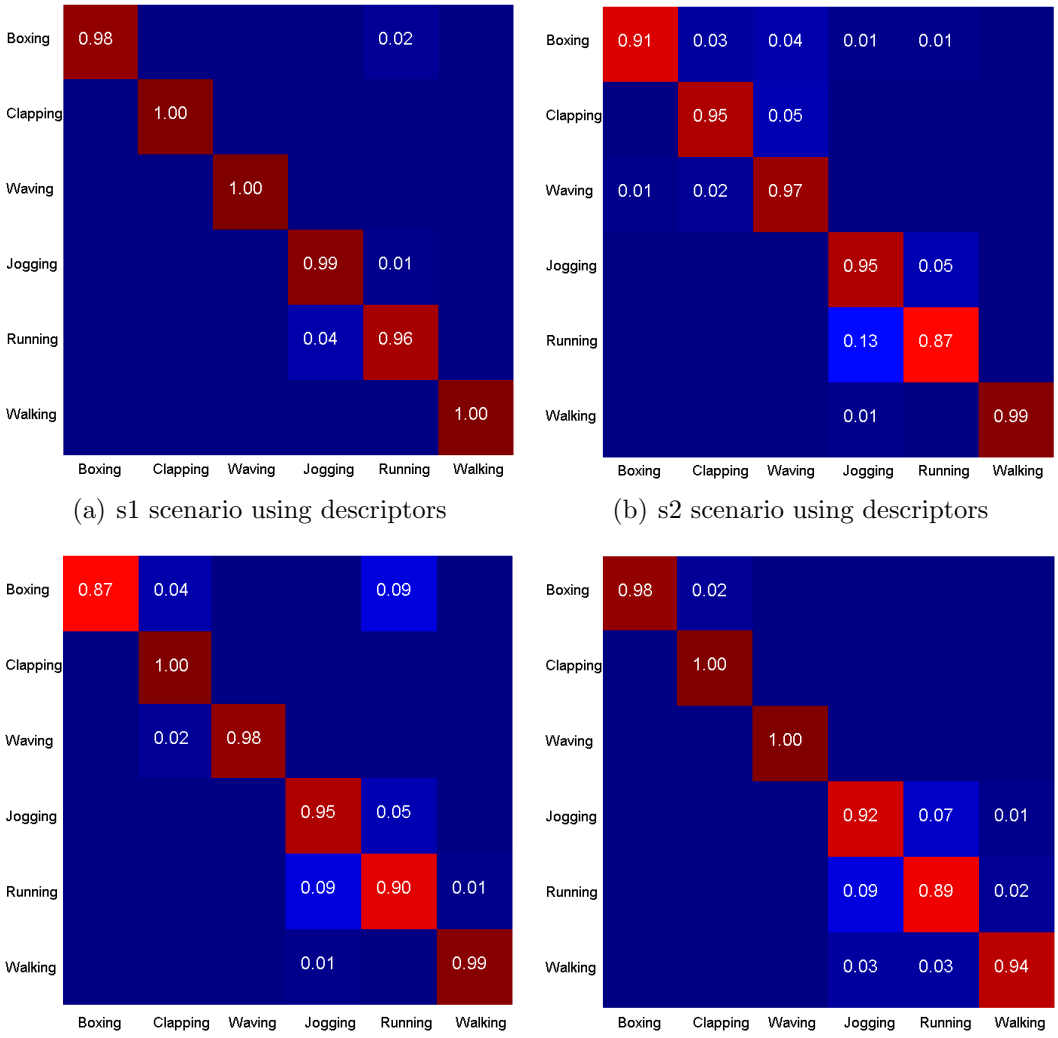
We also evaluated the performance of the prototype-based approach with respect to the number of prototypes k from 20 to 180, and compared these to the brute force descriptor-based apporach. As shown in Table 5.6, the prototype-based approach achieved an average recognition rate - 98.52%, and is robust to the selection of k. The recognition rate reached 100% at $k = 140, 180$ which is the same as the descriptor-based approach. Comparing the computation times, the prototype and look-up table-based method is almost 26 times faster than the brute force descriptor-based approach but with only a slight $1 - 2\%$ degradation of recognition rate. The confusion matrix of look-up table indexing-based approach is shown in Figure 5.11. We have compared the experimental results with state of the art action recognition approaches [5, 11, 31, 51, 69, 77, 79, 111, 118] in Table 5.6. Our approach achieved the same perfect recognition rate as Fathi and Mori [31] and outperformed all the other approaches significantly.

Table 5.6: Prototype-based recognition performance on the Weizmann dataset. The recognition rate is averaged based on leave-one-out experiments and the average time is computed as the average of computing an action-to-action similarity matrix. The results of [5, 11, 31, 51, 69, 77, 79, 111, 118] are copied from the original papers.

method	recog. rate (%)	avg. time (ms)
Our method: descriptor dist.	100	13.4
Our method: look-up(20 prot.)	82.22	0.5
Our method: look-up(40 prot.)	91.11	0.6
Our method: look-up(60 prot.)	94.44	0.5
Our method: look-up(80 prot.)	96.67	0.5
Our method: look-up(100 prot.)	97.78	0.5
Our method: look-up(120 prot.)	97.78	0.6
Our method: look-up(140 prot.)	100	0.5
Our method: look-up(160 prot.)	98.89	0.6
Our method: look-up(180 prot.)	100	0.5
Fathi & Mori [31]	100	N/A
Thurau *et al.* [111]	94.40	N/A
Niebles *et al.* [79]	90	N/A
Ali *et al.* [5]	92.6	N/A
Jhuang *et al.* [51]	98.8	N/A
Liu *et al.* [69]	89.26	N/A
Niebles & Fei-Fei [77]	72.8	N/A
Wang *et al.* [118]	97.78	N/A
Blank *et al.* [11]	99.61	N/A

5.6.3 Evaluation on the KTH Action Dataset

The KTH Action Dataset [95] includes 2391 sequences of six action classes: 'boxing', 'hand clapping', 'hand waving', 'jogging', 'running' and 'walking', performed by 25 actors in four scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). Example images from this dataset are shown in Figure 5.8(c). The KTH dataset is known to be more challenging than the Weizmann dataset due to non-colored grayscale input videos, low contrastness, frequent shadows, scale variations, etc. Previous work regarded the dataset either as a single large set (all scenarios in one) or as four different sub-



(a) s1 scenario using descriptors

(b) s2 scenario using descriptors

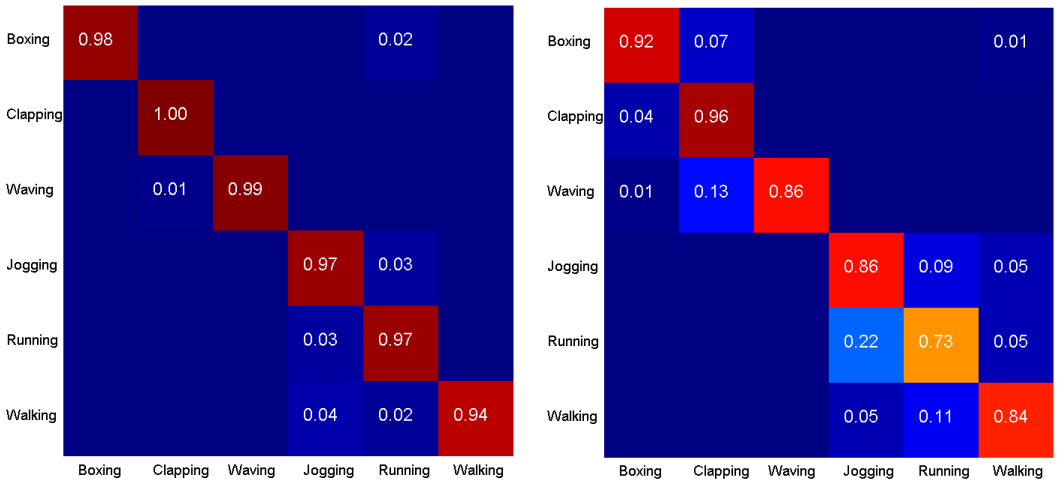
(c) s3 scenario using descriptors

(d) s4 scenario using descriptors

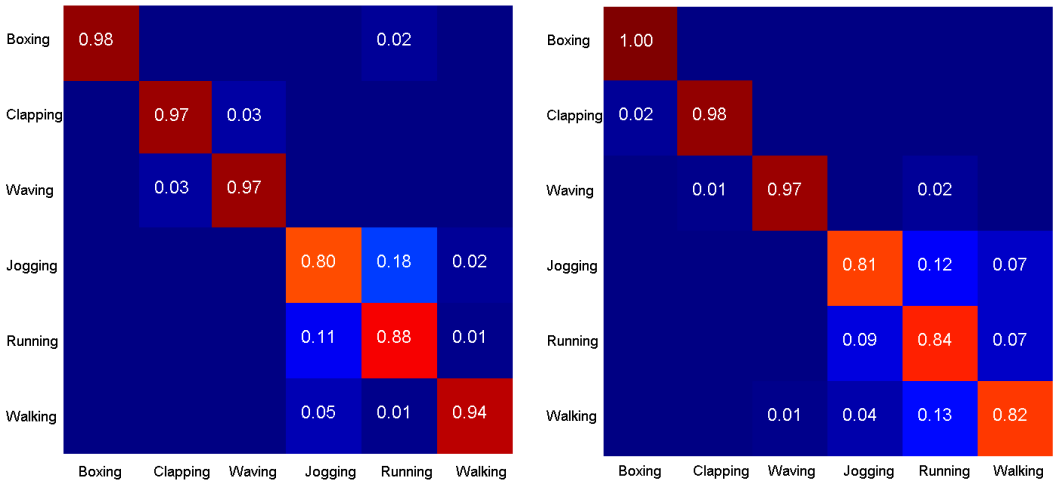
Figure 5.12: Confusion matrices for individual scenarios using the descriptor-based approach.

datasets (individual scenarios as one sub-dataset trained and tested separately). We perform experiments using both of these settings for better evaluating our approach in the context of other results reported on this dataset.

We evaluated our descriptor-based approach and prototype-based approach using leave-one-out experiments, where one action sequence performed by an actor is as the test sequence and all remaining action sequences performed by other ac-



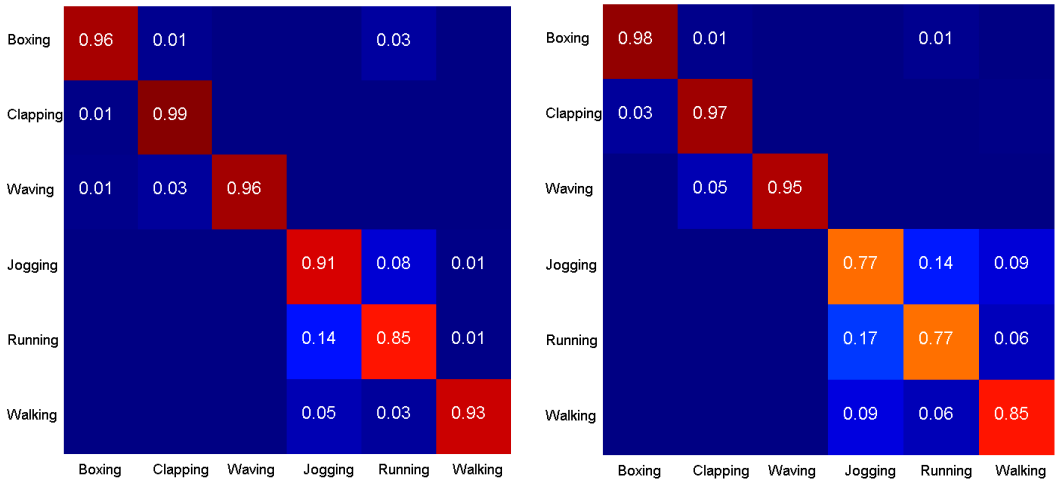
(a) s1 scenario using prototypes ($k = 240$) (b) s2 scenario using prototypes ($k = 300$)



(c) s3 scenario using prototypes ($k = 200$) (d) s4 scenario using prototypes ($k = 240$)

Figure 5.13: Confusion matrices for individual scenarios using the prototype-based approach.

tors are as the training sequences. Table 5.7 presents that our recognition results under four different scenarios using joint shape-motion descriptor, 'shape only', and 'motion only' descriptor. As a result, joint shape-motion descriptor achieved better recognition rates than 'shape only' and 'motion only' descriptors in all four scenarios. Figure 5.12 shows the confusion matrices for action recognition using our descriptor-based approach. From this figure, we can observe that most recognition



(a) Descriptor-based approach (b) Prototype-based approach (k=220)

Figure 5.14: Confusion matrices for the 'all-in-one' experiments.

errors are produced by 'Jogging' and 'Running' actions. This is reasonable because even human eyes are difficult to discriminate the two. Misclassifications between 'Boxing', 'Running', and 'Jogging' are possibly caused by inaccurate detection and localization of actors.

In addition, we evaluated the performance of our prototype-based approach using different number of prototypes, $k = 200, 220, 240, 260, 280, 300$, and compared it to the descriptor-based approach. The experimental results in Table 5.8 show that our prototype-based approach can get similar recognition rates as the descriptor-based approach, but is approximately 17 times faster. We also compared our results with state of art action recognition approaches [4, 51, 94]. Both versions of our method achieved the highest recognition rates under the s1, s2 and s3 scenarios, and the results are comparable to these approaches under the s4 scenario. Moreover, our average recognition rate for all four scenarios is 95.77%. To the best of our knowledge, this is the highest among all published results on the KTH dataset. Fig-

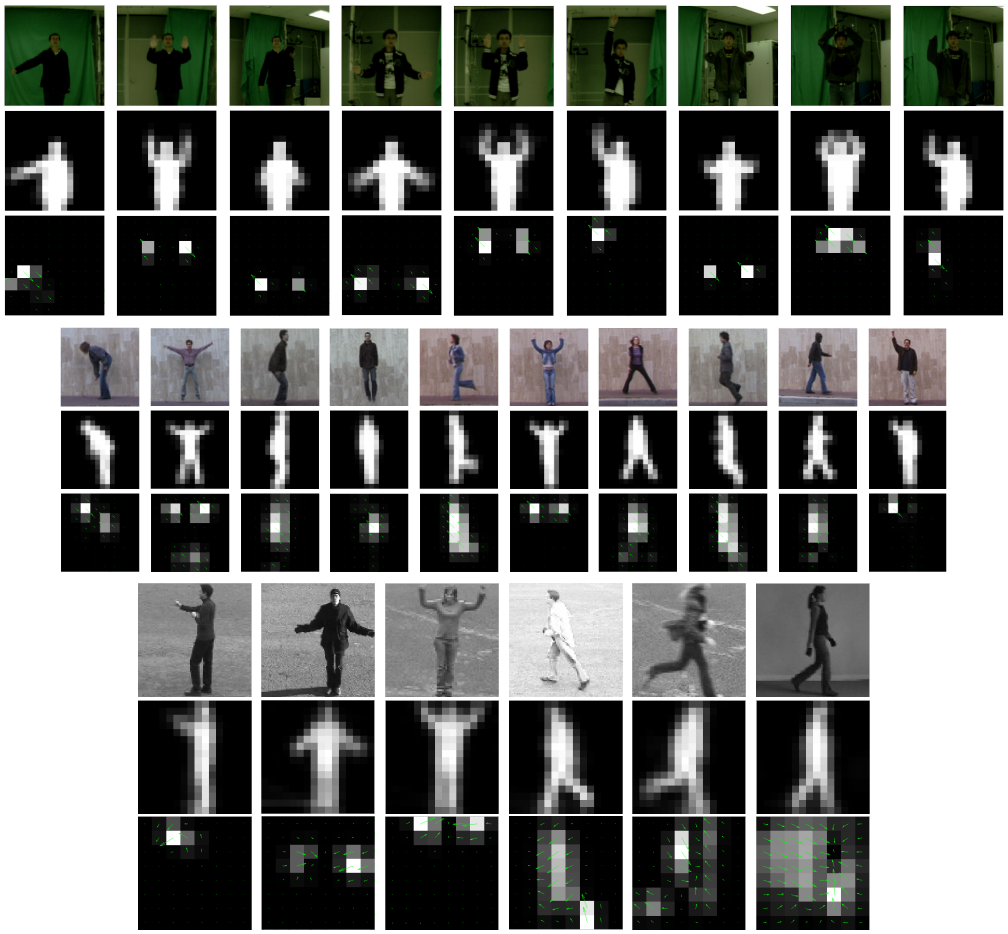


Figure 5.15: Examples of frame-to-prototype matching. Top: The Keck gesture dataset. Notice that the background against which the gesturer is viewed changes as we move through the figure, as does the location of the gesturer in the frame. Middle: The Weizmann dataset. Bottom: The KTH dataset.

ure 5.13 presents confusion matrices for action recognition using the prototype-based approach. Misclassified cases are similar to that of the descriptor-based approach.

Finally, we evaluated our approach in the context of 'all-in-one', *i.e.* all scenarios in a single set. We compared our approach with state of art approaches [25, 31, 52, 78, 80, 95, 121] in terms of recognition accuracy, as shown in Table 5.8. The results show that our descriptor-based approach achieved the highest recognition rates. Compared to descriptor-based approach, the prototype-based approach re-

Table 5.7: Feature-based recognition result on the KTH dataset.The unit for recognition rate is percentage.

method/recog. rate/scenario	s1	s2	s3	s4
motion only	92.82	78.33	89.39	83.61
shape only	71.95	61.33	53.03	57.36
joint shape and motion	98.83	94	94.78	95.48

sulted in slight degradation (about $5-6\%$) in the recognition rate, but it is almost 14 times faster and outperformed most of the current state of art approaches. Figure 5.14 shows confusion matrices of both version of our method from the 'all-in-one' experiments. Similar to the experiments on the individual scenarios, misclassifications here are also mainly occurred between 'Jogging', 'Running', and 'Walking', which is reasonable considering similarity of these three actions.

Figure 5.15 shows some qualitative results of frame-to-prototype matching for the Keck gesture dataset, the Weizmann action datset, and the KTH action dataset.

5.6.4 Discussions

We experimented with three different datasets in order to show the effectiveness and the robustness of our approach. Our method is shown to be quite successful for recognizing actions under dynamic backgrounds. While it has several limitations to be handled in our future work. Firstly, it performs frame-to-prototype matching using nearest-neighbor search which is based on the assumption that frame observations are independent and identically distributed. But for many applications, this

Table 5.8: Prototype-based recognition performance on the KTH dataset. The recognition rate is averaged based on leave-one-out experiments and the average time is computed as the average of computing an action-to-action similarity matrix. The results of [4, 25, 31, 51, 52, 69, 78, 80, 94, 95, 121] are copied from the original papers.

	recognition rate (%) / time (ms)					
method	s1	s2	s3	s4	avg.	all-in-one
Our method: descriptor dist.	98.83 / 15.2	94 / 19.3	94.78 / 14.5	95.48 / 16.7	95.77 / 16.43	93.43 / 15.2
Our method: look-up(200 prot.)	96.83 / 0.9	85.17 / 1.2	92.26 / 0.8	85.79 / 1.1	90.01 / 1.0	87.54 / 1.1
Our method: look-up(220 prot.)	96.33 / 0.9	83.33 / 1.3	92.09 / 0.8	86.79 / 1.1	89.76 / 1.0	88.04 / 1.1
Our method: look-up(240 prot.)	97.50 / 0.9	83.50 / 1.3	91.08 / 0.8	90.30 / 1.1	90.6 / 1.0	87.70 / 1.1
Our method: look-up(260 prot.)	96.33 / 0.9	84.17 / 1.2	90.74 / 0.8	87.96 / 1.1	89.8 / 1.0	87.37 / 1.1
Our method: look-up(280 prot.)	96.83 / 0.9	85.67 / 1.2	90.40 / 0.8	86.79 / 1.1	89.92 / 1.0	87.29 / 1.2
Our method: look-up(300 prot.)	96.66 / 0.9	86.17 / 1.2	90.07 / 0.8	89.97 / 1.1	90.72 / 1.0	87.49 / 1.2
Schindler & Gool snip.1 [94]	90.9 / N/A	78.1 / N/A	88.5 / N/A	92.2 / N/A	87.43 / N/A	88 / N/A
Schindler & Gool snip.7 [94]	93.0 / N/A	81.1 / N/A	92.1 / N/A	96.7 / N/A	90.73 / N/A	90.9 / N/A
Ahmad & Lee [4]	90.17 / N/A	84.83 / N/A	89.83 / N/A	85.67 / N/A	87.63 / N/A	88.83 / N/A
Jhuang *et al.* [51]	96.0 / N/A	86.1 / N/A	89.8 / N/A	94.8 / N/A	91.68 / N/A	N/A
Liu & Shah [69]	N/A	N/A	N/A	N/A	94.15 / N/A	N/A
Niebles *et al.* [78]	N/A	N/A	N/A	N/A	N/A	81.5 / N/A
Dollar *et al.* [25]	N/A	N/A	N/A	N/A	N/A	81.17 / N/A
Schuldt *et al.* [95]	N/A	N/A	N/A	N/A	N/A	71.72 / N/A
Ke *et al.* [52]	N/A	N/A	N/A	N/A	N/A	62.96 / N/A
Fathi & Mori [31]	N/A	N/A	N/A	N/A	N/A	90.50 / N/A
Nowozin *et al.* [80]	N/A	N/A	N/A	N/A	N/A	87.04 / N/A
Wang *et al.* [121]	N/A	N/A	N/A	N/A	N/A	92.43 / N/A

assumption is a poor one. This is the reason why matching results for some action sequences are non continuous and possibly random. Secondly, in the training phase, we learned the shape-motion prototype by k-means clustering in the joint shape-motion space. When the dataset and the value k are very large, k-means clustering is very slow. Thirdly, in the testing phase, when there is no color information for a person of interest, we detect and track him using part's appearance likelihood maps or the silhouette information from background subtraction. When the person's cloth color is very similar to background or there is no pure background available, the background subtraction is not applicable. Finally, although good overall recognition performance is achieved, our feature representation still has

difficulties differentiating some classes of actions due to large variability of actions performed by different individuals. Discriminative analysis of features between different actions might mitigate this issue to some degree. In spite of these limitations, the experimental results have shown that our approach can meet the needs of action recognition in practical applications.

Chapter 6

Conclusion and Future Work

We described a probabilistic hierarchical part-template matching approach to match human shapes with images to detect and segment humans simultaneously. Local part-based and global shape-template based approaches are combined to detect and segment humans from images. Based on the shape matching approach, we first introduced a pose-invariant (articulation-insensitive) image descriptor for learning a discriminative classifier for challenging problems of detecting and segmenting humans in generic photos. The descriptor is computed adaptively based on human poses instead of concatenating features along 2D image locations as in previous approaches. Specifically, we estimate the poses using a fast hierarchical matching algorithm based on a learned part-template tree. Given the pose estimate, the descriptor is formed by concatenating local features along the pose boundaries using a one-to-one point correspondence between detected and canonical poses. The pose-adaptive descriptors are trained using kernel SVM classifiers to discriminate humans from nonhumans. For applying the tree matching approach to multiple occluded human detection in crowded surveillance scenarios, we also introduce a Bayesian approach to iteratively optimize human configurations in images.

The results demonstrate that the proposed part-template tree model captures the articulations of the human body, and detects humans robustly and efficiently.

Although our approach can handle the majority of standing human poses, many of our misdetections are still due to pose estimation failures. This suggests that the detection performance could be further improved by extending the part-template tree model to handle more difficult poses and to cope with alignment errors in positive training images. We are also investigating the addition of color and texture information to our local contextual descriptor to improve the detection and segmentation performance.

The KDE-EM framework has fast convergence and achieves accurate results for color-based segmentation. The incorporation of local contrast-dependent MRF and PS pose model inference shows the combined local and global priors give very accurate segmentations, while human poses are estimated simultaneously. The pose-assisted segmentation approach is also generalized to the case of multiple occluded human segmentation based on a layered occlusion model and a probabilistic occlusion reasoning method. Experiments show that our approach improves KDE-EM to a large extent while preserving the basic computational cost and running time. Currently our approach can deal with most standing human poses (front/back and side views) but has limitations on handling self-occlusion and performing inference on more difficult poses. We need to extend our system to incorporate in-process user adjustments (*e.g.* correcting orientation of arms) to handle pose inference in these these cases. Another future direction is to generalize the approach to the cases of other object categories.

We introduced a pairwise comparison-based learning and classification framework for appearance-based recognition. We used a learned set of pairwise invari-

ant profiles to adaptively calculate distances from a query to prototypes so that the expected relative margins can be improved. The combined distances from appearance and discriminative information lead to significant improvements over pure appearance-based nearest neighbor classification. We also experimentally validated the scalability of our approach to larger number of categories. We are currently generalizing the framework for matching and recognizing people under occlusion. Also, we want to apply the approach to real-time person recognition in multi-camera systems by considering specific appearance database management schemes including appearance addition, removal and update schemes.

The experimental results demonstrate that our gesture and action recognition approach is both accurate and efficient, even when the action is viewed by a moving camera and against a possibly dynamic background. Our future work includes investigation of more robust frame-to-prototype matching methods. For example, using HMM to incorporate temporal constraints on the prototype sequence estimation. Although we introduced a fast and efficient action recognition approach, it is still based on separate and independent detection and recognition schemes. Hence, we are extending it to simultaneous action detection and segmentation in order to handle more challenging cases such as the presence of multiple different actions performed simultaneously by multiple actors. We are also exploring discriminative feature learning algorithms for improving our recognition performance.

Even though robust performance can be obtained in these fundamental problems, there are still many unsolved problems in integrating these fundamental components into a robust surveillance system capable of automatic human movement

analysis and event/activity analysis. Fore example, the incorporation of scene-specific cues or high-level spatial or temporal contexts would make human movement analysis more reliable and accurate. In the future, we aim to combine our approaches to human detection, segmentation, person and action recognition in a unified system framework for application in challenging real-world scenarios.

Bibliography

[1] Caviar Dataset: `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`.

[2] INRIA person dataset, http://pascal.inrialpes.fr/data/human/.

[3] Munich Airport Video Dataset, Siemens Corporate Research.

[4] M. Ahmad and S. Lee. Human action recognition using shape and clg-motion flow from multi-view image sequences. *Pattern Recognition*, 41(7):2237–2252, 2008.

[5] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. *ICCV*, 2007.

[6] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. PAMI*, 19(11):1300–1305, 1997.

[7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002.

[8] V.I. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. PAMI*, 19(10):711–720, 1997.

[9] C. BenAbdelkader and L. S. Davis. Motion-based recognition of people in eigengait space. *FGR*, 2002.

[10] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. *ECCV*, pages Vol I:428-441, 2004.

[11] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *ICCV*, 2005.

[12] A. Bobick and J. Davis. The recognition of human movement using tempral templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

[13] E. Borenstein and J. Malik. Shape guided object segmentation. *CVPR*, 2006.

[14] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. *ICCV*, 2007.

[15] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *ICCV*, 2001.

[16] Matthieu Bray, Pushmeet Kohli, and Philip H.S. Torr. PoseCut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. *ECCV*, 2006.

[17] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using em and its application to image querying. *IEEE Trans. PAMI*, 24(8):1026–1038, 2002.

[18] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[19] D. Comaniciu and P. Meer. Mean-Shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5), 2002.

[20] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Int'l J. Computer Vision*, 25(5):64–577, 2003.

[21] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory*, IT(13):21–27, 1967.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[23] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *ECCV*, 2006.

[24] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, pages 263–286, 1995.

[25] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *VS-PETS*, 2005.

[26] C. Domeniconi and D. Gunopulos. Adaptive nearest neighbor classification using support vector machines. *NIPS*, 2001.

[27] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, volume 2, pages 726-733, 2003.

[28] A. Elgammal and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. *ICCV*, 2003.

[29] A. M. Elgammal and L. S. Davis. Probabilistic framework for segmenting people under occlusion. *ICCV*, 2001.

[30] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. *CVPR*, 2005.

[31] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. *CVPR*, pages 1-8, 2008.

[32] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

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

[34] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. *ECCV*, 2006.

[35] F. Fleuret, R. Lengagne, and P. Fua. Fixed point probability field for complex occlusion handling. *ICCV*, 2005.

[36] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. *NIPS*, 2006.

[37] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retreval and classification. ICCV, 2007.

[38] T. Gandhi and M. Trivedi. Panoramic appearance map (PAM) for person re-identification. *AVSS*, 2006.

[39] D. M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Trans. PAMI*, 29(8):1408–1421, 2007.

[40] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. *ICCV*, 1999.

[41] N. Gheissari, T. Sebastian, P. Tu, J. Rittscher, and R. Hartley. Person reidentification using spatiotemporal appearance. *CVPR*, 2006.

[42] L. Goldmann, M. Karaman, J.T. Minquez, and T. Sikora. Appearance-based person recognition for surveillance applications. *WIAMIS*, 2006.

[43] S. Gordon, H. Greenspan, and J. Goldberger. Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations. *ICCV*, 2003.

[44] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. PAMI*, 29(12):2247–2253, 2007.

[45] M. Haehnel, D. Kluender, and K.-F. Kraiss. Color and texture features for person recognition. *IJCNN*, 2004.

[46] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *NIPS*, 1997.

[47] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Trans. PAMI*, 18:607–616, 1996.

[48] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *CVPR*, 2006.

[49] M. B. Holte, T. b. Moeslund, and P. Fihl. Fusion of range and intensity information for view invariant gesture recognition. *CVPR Workshop*, pages 1-7, 2008.

[50] M. Isard and J. MacCormick. BraMBLe: A bayesian multiple-blob tracker. *ICCV*, 2001.

[51] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. *ICCV*, 2007.

[52] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. *ICCV*, 2005.

[53] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. *ICCV*, 2007.

[54] S. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. *ECCV*, 2006.

[55] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[56] K. Kim and L. S. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. *ECCV*, 2006.

[57] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentations of video. *ICCV*, 2005.

[58] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj cut. *CVPR*, 2005.

[59] I. Laptev, M. Marszalek, and C.Schmid B. Rozenfeld. Learning realistic human actions from movies. *CVPR*, 2008.

[60] I. Laptev and P. Perez. Retrieving actions in movies. *ICCV*, 2007.

[61] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. *CVPR*, 2005.

[62] H. Li and M. Greenspan. Multi-scale gesture recognition from time-varying contours. *ICCV*, volume 1, pages 236-243, 2005.

[63] J. Li, S. K. Zhou, and R. Chellappa. Appearance context modeling under geometric context. *ICCV*, 2005.

[64] Y. Lin, T. Liu, and C. Fuh. Local ensemble kernel learning for object category recognition. *CVPR*, 2007.

[65] Z. Lin and L. S. Davis. Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance. *ISVC*, 2008.

[66] Z. Lin and L. S. Davis. A pose-invariant descriptor for human detection and segmentation. *ECCV*, 2008.

[67] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon. Hierarchical part-template matching for human detection and segmentation. *ICCV*, 2007.

[68] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon. An interactive approach to pose-assisted and appearance-based segmentation of humans. *ICCV Workshop on Interactive Computer Vision*, 2007.

[69] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. *CVPR*, 2008.

[70] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. *CVPR*, 2008.

[71] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *ECCV*, 2004.

[72] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. *CVPR*, 2008.

[73] A. Mittal and L. S. Davis. M2 tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *I*, 51(3):189–203, 2003.

[74] B. Moghadam, T. Jebara, and A. Pentland. Bayesian face recognition. *MERL Technical Report*, TR-2000-42, Febrary, 2002.

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

[76] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. Full-body person recognition system. *Pattern Recognition*, 36(9):1997–2006, 2003.

[77] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. *CVPR*, pages 1-8, 2007.

[78] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporalwords. *BMVC*, 2006.

[79] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int'l J. Computer Vision*, 79(3):299–318, 2007.

[80] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. *ICCV*, 2007.

[81] K. Nummiaro, E. K. Meier, and L. V. Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.

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

[83] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. *In Proc. of Intelligent Vehicles*, 1998.

[84] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. *NIPS*, 1999.

[85] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition, 1989.

[86] S. Rajko, G. Qian, T. Ingalls, and J. James. Real-time gesture recognition with minimal training requirements and on-line learning. *CVPR*, pages 1-8, 2007.

[87] D. Ramanan. Learning to parse images of articulated bodies. *NIPS*, 2006.

[88] J. Rittscher, P. H. Tu, and N. Krahnstoever. Simultaneous estimation of segmentation and shape. *CVPR*, 2005.

[89] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recogntion. *CVPR*, 2008.

[90] V. Roth and K. Tsuda. Pairwise coupling for machine recognition of hand-printed japanese characters. *CVPR*, 2001.

[91] C. Rother, V. Kolmogorov, and A. Blake. GrabCut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics: SIGGRAPH*, 2004.

[92] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. *CVPR*, 2007.

[93] S. Salvador and P. Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. *KDD Workshop on Mining Temporal and Sequential Data*, pages 70-80, 2004.

[94] K. Schindler and L. V. Gool. Action snippets: How many frames does human action recognition require? *CVPR*, 2008.

[95] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *ICPR*, 2004.

[96] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. *NIPS*, 2003.

[97] D.W. Scott. Multivariate density estimation. *Wiley Interscience*, 1992.

[98] E. Seemann, Bastian Leibe, and Bernt Schiele. Multi-aspect detection of articulated objects. *CVPR*, 2006.

[99] S. Shalev-Shwartz, Y. Singer, and A. Ng. Online and batch learning of pseudo metrics. *ICML*, 2004.

[100] V. Sharma and J. W. Davis. Integrating appearance and motion cues for simultaneous detection and segmentation of pedestrians. *ICCV*, 2007.

[101] E. Shechtman and M. Irani. Space-time behavior-based correlation. *IEEE Trans. PAMI*, 29(11):2045–2056, 2007.

[102] V. D. Shet, J. Neumann, V. Ramesh, and L. S. Davis. Bilattice-based logical reasoning for human detection. *CVPR*, 2007.

[103] V. D. Shet, V. S. N. Prasad, A. Elgammal, Y. Yacoob, and L. S. Davis. Multi-cue exemplar-based nonparamtric medel for gesture recognition. *ICVGIP*, pages 656-662, 2004.

[104] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8), 2000.

[105] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. *CVPR*, 2008.

[106] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. *ICCV*, 2005.

[107] C. Sminachisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. *ICCV*, 2005.

[108] K. Smith, D. G. Perez, and J. M. Odobez. Using particles to track varying numbers of interacting people. *CVPR*, 2005.

[109] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. *ICCV*, 2008.

[110] H. Tao, H. Sawhney, and R. Kumar. A sampling algorithm for detecting and tracking multiple objects. *ICCV Workshop on Vision Algorithms*, 1999.

[111] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. *CVPR*, pages 1-8, 2008.

[112] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. *ICCV*, 2005.

[113] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifold. *CVPR*, 2007.

[114] US-ARMY. Visual signals. Field Manual FM 21-60, 1987.

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

[116] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *ICCV*, 2003.

[117] S. N. Vitaladevuni, V. Kellokumpu, and L. S. Davis. Action recognition using ballistic dynamics. *CVPR*, 2008.

[118] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. *CVPR*, 2007.

[119] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. *ICCV*, 2007.

[120] Y. Wang and G. Mori. Learning a discriminative hidden part model for human action recognition. *NIPS*, 2008.

[121] Y. Wang, P. Sabzmeydani, and G. Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. *ICCV Workshop on Human Motion*, 2007.

[122] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *NIPS*, 2005.

[123] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. *CVPR*, 2008.

[124] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. *ICCV*, 2007.

[125] J. Winn and N. Jojic. LOCUS: Learning object classes with unsupervised segmentation. *ICCV*, 2005.

[126] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. *CVPR*, 2006.

[127] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *ICCV*, 2005.

[128] B. Wu and R. Nevatia. Optimizing discrimination-efficientcy tradeoff in integrating heterogeneous local features for object detection. *CVPR*, 2008.

[129] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. *CVPR*, 2007.

[130] T. Wu, C. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

[131] Ying Wu, Ting Yu, and Gang Hua. A statistical field model for pedestrian detection. *CVPR*, 2005.

[132] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side information. *NIPS*, 2002.

[133] P. Yan, S. M. Khan, and M. Shah. Learning 4d action feature models for arbitray view action recognition. *CVPR*, 2008.

[134] G. Ye, J. J. Corso, and G. D. Hager. Gesture recognition using 3d appearance and motion features. *CVPR Workshop on Realtime Vision for HCI*, pages 160-167, 2004.

[135] Y. Yu, D. Harwood, K. Yoon, and L. S. Davis. Human appearance modeling for matching across video sequences. *Mach. Vis. Appl.*, 18(3-4):139–149, 2007.

[136] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *CVPR*, 2006.

[137] H. Zhang and J. Malik. Learning a discriminative classifier using shape context distances. *CVPR*, 2003.

[138] L. Zhao and L. S. Davis. Closely coupled object detection and segmentation. *ICCV*, 2005.

[139] L. Zhao and L. S. Davis. Iterative figure-ground discrimination. *ICPR*, 2004.

[140] Q. Zhao, J. Kang, H. Tao, and W. Hua. Part based human tracking in a multiple cues fusion framework. *ICPR*, 2006.

[141] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. *CVPR*, 2004.

[142] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *CVPR*, 2006.