Extracting Moving People and Categorizing their Activities in Video

Juan Carlos Niebles Duque

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Electrical Engineering
Adviser: Fei-Fei Li

January 2011

Abstract

The ability to automatically detect and track human movements, recognize actions and activities, understand behavior and predict goals and intentions has captured the attention of many computer vision scientists. One of the main motivations is the great potential impact that this technology can make on many applications such as video search and indexing, smart surveillance systems, medical research, video game interfaces, automatic sport commentary, human-robot interaction, among others.

In this work, we focus on two important questions: given a video sequence, *where* are the moving humans in the sequence? *what* actions or activities are they performing?

We first discuss the problem of extracting human motion volumes from video sequences. We present a fully automatic framework to detect and extract arbitrary human motion volumes from challenging real-world videos. We have explored a purely top-down methodology that estimates body configurations at every frame to achieve the extraction. We also present a much more efficient approach that carefully combines bottom-up and top-down cues, which enables fast extraction in near real time.

We are not only interested in finding where the humans are in a given sequence, but also in understanding what they are doing. We present statistical models for the task of simple human action recognition based in spatial and spatio-temporal local features. First, we show that by adapting latent topic models we can achieve competitive simple action categorization performance in an unsupervised setting. We also present a hierarchical model for simple actions that can be characterized as a constellation-of-bags-of-features. This model leverages the spatial structure of the human body to improve action recognition.

While these models are successful at the task of simple action recognition, their performance suffers when the actions of interest are more complex. We propose a discriminative model for complex action recognition capable of leveraging the temporal

structure and composition of simpler motions into complex actions. We show that the contextual information provided by the temporal structure in our model greatly improves the complex action classification accuracy over state-of-the art models for simple action recognition.

Acknowledgements

I would like to thank my advisor, Professor Fei-Fei Li for all the help and advice along this journey.

Thanks all my collaborators (in chronological order) Hongcheng Wang, Bohyung Han, Andras Ferencz and Chih-Wei Chen, who positively influenced the outcome of my research. I also thank my collaborators Andrey Del Pozo, Silvio Savarese, Jia Li and Bangpeng Yao, with whom I had the chance to do research not related to this work. Thanks to all the members of the Vision Lab for the many great moments and research discussions: Min Sun, Barry Chai, Jia Deng, Hao Su and Olga Russakovsky.

Thanks to the Fulbright Commission, *Colciencias*, the *Departamento Nacional de Planeación* and *Universidad del Norte* for their financial support during my graduate studies.

Thanks to my family for their continuous support and encouragement.

To my family.

Contents

List of Tables

List of Figures

Part I

Introduction

One of the Holy Grails in computer vision is creating algorithms for automatic analysis of human behavior in video sequences. The ability to automatically detect and track human movements, recognize actions and activities, understand behavior and predict goals and intentions has being studied by many computer vision scientists [35, 91]. One of the main motivations is the great potential impact that this technology can make on a large variety of applications such as video search and indexing, smart surveillance systems, medical research, video game interfaces, automatic sport judging and commentary, human-robot interaction, among others.

In this work, we focus on two important questions: given a video sequence, *where* are the moving humans in the sequence? *what* actions or activities are they performing? These are challenging vision problems, mostly because human bodies are highly articulated, people tend to wear clothing with complex texture, and each actor has a different pace and style to perform certain actions. In addition, background clutter, occlusions, illumination changes and unconstrained camera motions create significant variations and uncertainties.

We first discuss the problem of extracting human motion volumes from video sequences in Part II. We present a fully automatic framework to detect and extract arbitrary human motion volumes from challenging real-world videos collected from *YouTube*. We introduce our framework and review some of the related work in Chapter 1. In Chapter 2, we explore a purely top-down methodology that estimates body configurations at every frame with a pictorial structure model. The resulting pose estimations indicate the spatio-temporal volume that encloses each person. Our method relies on a technique that effectively reduces the search space, reducing the computation time of the pictorial structure measurement process. We present experimental evaluation of this method in Chapter 3. In spite of the large reduction in computation by pruning the search space, applying the top-down model on every frame remains computationally intensive for some applications that require real-time or faster pro-

cessing. In Chapter 4, we present a much more efficient approach that carefully combines bottom-up and top-down cues, which enables fast extraction in near real time. The algorithm sparsely applies a top-down driven person segmentation in a few frames, and efficiently propagates the estimated human regions into other frames in a bottom-up fashion. Finally, we present promising experimental results in Chapter 5.

Once a computer vision system is able to localize and extract moving people from video sequences, it is natural to ask the question of what activities are being performed by these actors. We first approach the problem of simple action recognition in Part III. We present statistical models for this task that are based in spatial and spatio-temporal local features. We introduce the topic and review some of the recent work in Chapter 6. In Chapter 7, we show that by adapting latent topic models we can achieve competitive simple action categorization performance in an unsupervised setting. While these models achieve good accuracy, they lack any understanding of the structure of the human body or the spatio-temporal structure of the human actions. In Chapter 8, we address this issue by presenting a hierarchical model for simple actions that can be characterized as a constellation-of-bags-of-features. This model leverages the spatial structure of the human body which is empirically shown to improve action recognition accuracy. Finally, experimental results are presented in Chapter 9.

While these models are successful at the task of simple action recognition, their performance suffers when the actions of interest are more complex. In Part IV, we present our approach to the problem of complex action recognition by exploiting temporal structures. An introduction to the topic is presented in Chapter 10, where we also discuss the related prior work. In Chapter 11, we present the details of our discriminative model for complex action recognition, which is capable of leveraging the temporal structure and composition of simpler motions into complex actions.

We show that the contextual information provided by the temporal structure in our model greatly improves the complex action classification accuracy over state-of-the art models for simple action recognition. Lastly, experimental evaluation is covered in Chapter 12.

Part II

Extracting Human Motion Volumes from Video Sequences

Chapter 1

Introduction

Human motion analysis is notoriously difficult because human bodies are highly articulated and people tend to wear clothing with complex textures that obscure the important features needed to distinguish poses. Uneven lighting, clutter, occlusions, and camera motions cause significant variations and uncertainties. Hence it is no surprise that the most reliable person detectors are built for upright walking pedestrians seen in typically high quality images or videos.

Nevertheless, extracting moving humans from video is a critical in many applications that require accurate and efficient human motion estimation. For example, a mobile agent that navigates the world by interacting with humans in real-time needs to identify and track people in its surroundings. Also, tasks such as video indexing, search, and intelligent surveillance would benefit greatly by accurate human behavior understanding. Traditionally, research in this area has been done mostly from a tracking perspective [35]; however, tracking humans in natural videos is still a challenging problem.

Our goal is to be able to *automatically* and *efficiently* carve out spatio-temporal volumes of humans with arbitrary motions and poses from videos taken in unknown settings. In particular, we focus our attention on videos that are typically present on

Internet sites such as *YouTube*. These videos are representative of the kind of real-world data that is highly prevalent and important. As the problem is very challenging, we do not assume that we can find every individual. Rather, our aim is to enlarge the envelope of upright human detectors by tracking detections from typical to atypical poses. Sufficient data of this sort will allow us in the future to learn even more complex models that can reliably detect people in arbitrary poses.

Part II is organized as follows. We review related prior work in Section 1.1. We first introduce a purely top-down methodology that estimates body configurations at every frame to achieve the extraction in Chapter 2, with the experimental results covered in Chapter 3. We also present a much more efficient approach that carefully combines bottom-up and top-down cues in Chapter 4, with the experimental validation in Chapter 5.

Earlier versions of this work appeared in ECCV 2008 [67] and CVPR 2010 [66].

1.1 Related Work

1.1.1 Human body tracking

The most straightforward method to track humans is to consider them as blobs and use generic object tracking methods. General object tracking has a long history in computer vision, see [106] for a recent survey. Two approaches with great influence in the field are the Lucas-Kanade tracker [62], and the mean-shift tracking algorithm [16]. The Lucas-Kanade is a template based tracker, which tends to be more suitable for tracking rigid objects in sequences with minor view point change. Mean-shift tracking, on the other hand, is more robust to view point changes, as long as the global color distribution of the target remains roughly constant. Other interesting general object tracking methods are based on discriminative classifiers that aim to select features on the target object which are most distinctive to the surrounding

background [15, 36]. However, this type of tracking methods still consider the target as a rectangular (or elliptical) blob and do not provide a segmentation of the tracked object.

Alternatively, contour-based representations of deformable shape are also often used to describe the human body region efficiently. The level-set framework is certainly one of the most common approaches [18]. Cremers [17] presents a methodology to incorporate dynamical shape priors to a level-set tracker, which enables tracking under very noisy or corrupted conditions. In [107], Yilmaz et al. extend a level-set tracking framework to handle occlusions of the target. Most recently, Bibby and Reid present a real-time level-set tracker that is formulated probabilistically and allows the use of pixel-wise posteriors for improved tracking accuracy [5]. While, these methods tend to be fast, they ignore the structure of the human body and/or impose very strong priors, which may lead to critical limitations when estimating articulated and flexible human poses.

Another interesting angle is to cast the estimation of the human body region as a figure/ground segmentation problem. In [76], Ren tackles this formulation with the help of multiple low-level cues, though the algorithm seems to rely on objects having a high contrast with the background.

Model-based methods, or top-down models, encode the articulation and movements of the human body with an *a priori* structure model [22, 58, 84, 82, 88, 86, 12, 42]. A popular method to encode the structure of the body is the pictorial structure model [29, 75]. Lan and Huttenlocher [51] present a spatio-temporal pictorial structure for tracking walking motions. Han et al. [41] present an articulated body tracker that uses an efficient non parametric belief propagation inference algorithm. Most of these methods rely on manual initialization, strong priors to encode the expected motion, a controlled or very simple environment with good foreground/background separation, and/or seeing the motion from multiple cameras. Learning and inference

procedures attempt to fit the image evidence to the best configuration of the model, but typically involve many degrees of freedom, large search space, and complex observations. They are, in general, prohibitively slow, due to the large amount of complex computations.

1.1.2 Pedestrian detection and human pose estimation

Several fairly reliable algorithms for pedestrian detection in still images have been developed recently [53, 59, 19, 92, 104]. Most of these methods do not consider the intrinsic properties of pedestrians, but are instead general object detection algorithms. Dalal and Triggs presented a discriminative detector of upright humans based on the Histogram of Oriented Gradient feature (HOG) [19]. The detection is done in a sliding-window manner, where each scanned window is classified with a Support Vector Machine (SVM) classifier. An alternative sliding-window detector is proposed by Laptev in [53]. This detector uses features that are similar to HOG, but the classification is done by AdaBoost with Fisher linear discriminats as weak classifiers. More recently Felzenszwalb et at. [30] have proposed a deformable part based model which builds upon the HOG detector [19] and the pictorial structure model [29]. Their method has greatly influenced many of the most recent object and pedestrian detection systems. In spite of their relative success, these methods typically deal with upright persons only, and the detection accuracy is significantly reduced by even moderate pose variations. Furthermore, these algorithms offer little segmentation of the human, providing only a bounding box around the body.

Another body of work deals with pedestrian detection in video sequences [95, 102, 20]. These methods incorporate additional cues from motion patterns available in the sequence. However, they are still restricted to upright people only.

To model body configurations, tree shaped graphical models have shown promising results [29, 73]. In [29], Felzenszwalb and Huttenlocher show an algorithm for efficient

inference in pictorial structure models, and apply it to estimate human poses in still images. Ramanan [73] extends the method to include template-based body part detectors and person-specific color models, which improve pose estimation accuracy. More recently in [1], classifier-based body part detectors show further improvement. A couple of recent methods have also extended the use of pictorial structures for pose estimation in video sequences. Ramanan et al. [75] use pictorial structures for tracking. Ferrari et al. [34] use temporal information to reduce the search space progressively in applying pictorial structures to videos. In general, these generative models are often able to find an accurate pose of the body and limbs. However, they are less adept at making a discriminative decision: is there a person or not? They are typically also very expensive computationally in both the measurement and inference steps.

In the following chapters, we present a framework capable of automatically extracting moving humans from video. We build on several of the techniques mentioned here, and obtain efficient algorithms with promising results.

Chapter 2

Extracting People with Top-Down Pictorial Structures

In this chapter, we present a system for extracting moving people with top-down pictorial structures. Unlike much of the previous work, our system is capable of fully automatic extraction. Furthermore, it achieves computational efficiency by exploiting temporal information and smoothness. Our representation of body part and pose estimation distributions are based on a efficient semi-parametric Gaussian mixture representation, which is key for effectively maintaining the accuracy of the system while reducing the computation complexity.

Our first objective is to find moving humans automatically. In contrast to much of the previous work in tracking and motion estimation, our framework does not rely on manual initialization or a strong *a priori* assumption on the number of people in the scene, the appearance of the person or the background, the motion of the person or that of the camera. To achieve this, we improve a number of existing techniques for person detection and pose estimation, leveraging on temporal consistency to improve both the accuracy and speed of existing techniques. We initialize our system using a state-of-the-art upright pedestrian detection algorithm [53]. While this technique

Figure 2.1: Two example results of our human motion extraction algorithm with top-down pictorial structures. Our input videos are clips downloaded from *YouTube* and thus are often low resolution, captured by hand-held moving cameras, and contain a wide range of human actions. In the top sequence, notice that although the boundary extraction is somewhat less accurate in the middle of the jump, the system quickly recovers once more limbs become visible

works well on average, it produces many false positive windows and very often fails to detect. We improve this situation by building an appearance model and applying a two-pass constrained clustering algorithm [50] to verify and extend the detections.

Once we have these basic detections, we build articulated models following [29, 73, 75] to carve out arbitrary motions of moving humans into continuous spatio-temporal volumes. The result can be viewed as a segmentation of the moving person, but we are not aiming to achieve pixel-level accuracy for the extraction. Instead, we offer a relatively efficient and accurate algorithm based on the prior knowledge of the human body configuration. Specifically, we enhance the speed and potential accuracy of [73, 75] by leveraging temporal continuity to constrain the search space and applying semi-parametric density propagation to speed up evaluation. Two example sequences and the system output are shown in Figure 2.1.

2.1 System Architecture

Our system consists of two main components. The first component generates object-level hypotheses by coupling a human detector with a clustering algorithm. In this

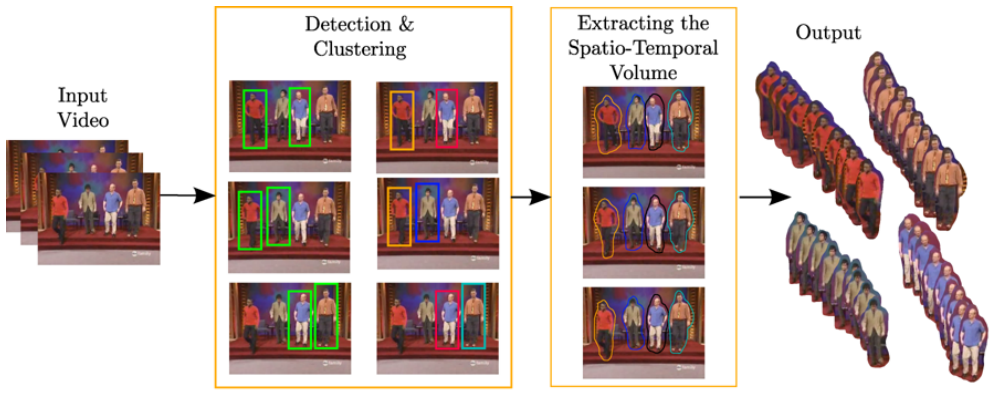


Figure 2.2: Overview of our human motion volume extraction system

part, the state of each person, including location, scale and trajectory, is obtained and used to initialize the body configuration and appearance models for limb-level analysis. Note that in this step two separate problems – detection and data association – are handled simultaneously, based on the spatio-temporal coherence and appearance similarity.

The second component extracts detailed human motion volumes from the video. In this stage, we further analyze each person's appearance and spatio-temporal body configuration, resulting in a probability map for each body part. We have found that we can improve both the robustness and efficiency of the algorithm by limiting the search space of the measurement and inference around the modes of the distribution. To do this, we model the density function as a mixture of Gaussians in a sequential Bayesian filtering framework [2, 24, 40].

The entire system architecture is illustrated in Figure 2.2. More details about each step are described in the following two sections.

The focus of our work is to extract arbitrarily complex human motions from *YouTube* videos that involve a large degree of variability. We face several difficult challenges, including:

1. Compression artifacts and low quality of videos

2. Multiple shots in a video

3. Unknown number of people in each shot or sequence

4. Unknown human motion and poses

5. Unknown camera parameters and motion

6. Background clutter, motion and occlusions

We will refer back to these points in the rest of the discussion as we describe how the components try to overcome them.

2.2 People Detection and Clustering

As Figure 2.2 shows, our system starts by estimating the location, scale, and trajectories of the moving persons in the video. This step is composed of the following two parts.

2.2.1 Initial hypotheses by detection

We first employ an human detection algorithm [53] to generate a large number of hypotheses for persons in a video. This method, which trains a classifier cascade using boosting of HOG features to detect upright standing or walking people, has serious limitations. It only detects upright persons and cannot handle arbitrary poses (challenge 4). The performance is degraded in the presence of compression artifacts (challenge 1). Moreover, since it does not use any temporal information, the detection is often inconsistent and noisy, especially in scale. It is, therefore, difficult to reject false positives and recover miss-detections effectively. The complexity increases dramatically when multiple people are involved (challenge 3). This step, therefore, serves only as an initial hypotheses proposal stage. Additional efforts are required to handle various exceptions.

2.2.2 People clustering

The output of the person detector is a set of independent bounding boxes; there are no links for the same individual between detections. The detections also have significant noise, false alarms and miss-detections especially due to the low quality of the video (challenge 1). In order to recover from these problems, we incorporate a clustering algorithm based on the temporal and appearance coherence of each person. The goal of clustering in our system is to organize all correct detections into groups, where each corresponds to a single person in the sequence (challenge 3), while throwing away false alarms. To achieve this, we apply a constrained clustering paradigm [50] in two hierarchical stages, adding both positive (should link) edges and negative (can not link) constraints between the detections. See Figure 2.3 for an example.

Stage 1

In the first stage, we focus on exploiting the temporal-coherence cue by associating detections from multiple frames with the help of a low-level tracking algorithm [16]. When the first detection is observed, a low-level tracker is initialized with the detected bounding box. A new detection in a consequent frame is assigned to an existing track if it coherently overlaps with the tracker predictions. In this case, we reinitialize the tracker with the associated detection bounding box. When no existing track can explain the new detection, a new track is created. Due to the complexity of the articulated human body, a low-level tracker is susceptible to drift from the person. We thus limit the temporal life of the tracker by counting the number of frames after the last detection and terminating the track at the last detection if the maximum gap (e.g. 100 frames) is surpassed. Very small clusters with few detections are discarded. The clusters produced in this first stage are almost always correct but over-segmented tracks (see Figure 2.3 (b)). This is because the person detector often fails to detect a

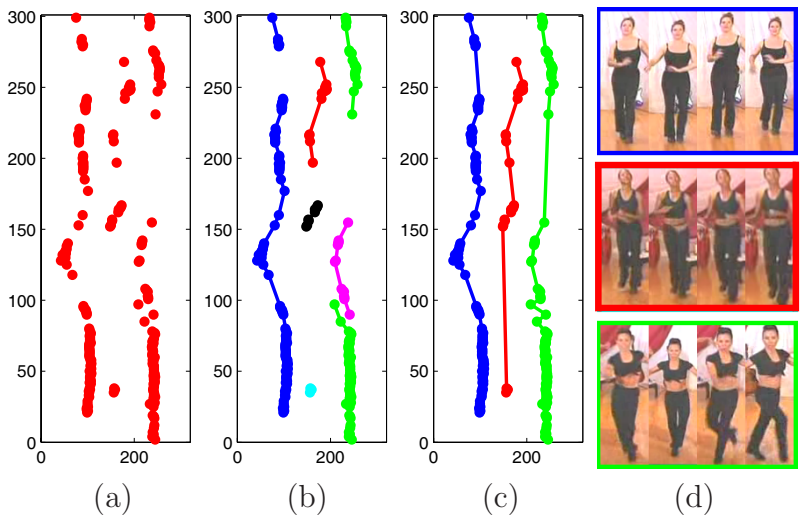


Figure 2.3: Example results of the human detection and clustering stage. From noisy detections, three tracks of people are identified successfully by filling gaps and removing outliers. (In this figure, the horizontal and vertical axis are the x locations and frame numbers, respectively.) (a) Original detection (b) Initial clusters after step 1 (c) Final clusters (d) Example images of three similar people that correctly clustered into different groups

person in the video for many frames in a row – especially when the person performs some action that deviates from an upright pose.

Stage 2

The stage 2 agglomerative constrained clustering views the stage 1 clusters as atomic elements, and produces constraints between them with positive weights determined by appearance similarity and negative constraints determined by temporal/positional incompatibility.

For the appearance similarity term, we select multiple high-scoring detection windows for each stage 1 cluster, and generate probability maps for the head and torso locations using a simple two-part pictorial structure [73]. We use these results to (1) remove false detections by rejecting clusters that have unreliable head/torso estimation results (e.g., high uncertainty in the estimated head and torso locations), and (2) generate a weighted mask for computing color histogram descriptors for both the

head and the torso. The appearance of the person in each cluster is then modeled with the color distributions of head and torso.

After the second pass of our hierarchical clustering, we obtain one cluster per person in the sequence. Figure 2.3 (c) illustrates the final clustering result, which shows that three different persons and their trajectories are detected correctly, despite the fact that the appearance of these individuals are very similar (Figure 2.3 (d)).

2.3 Extracting Spatio-Temporal Human Motion Volumes

We now have a cluster for each person, with a detection bounding box giving the location, scale, and appearance in some subset of the frames. Our goal is to find the body configuration for all the frames of the cluster (challenge 4), both where we have detections and where we do not. In this section, we discuss how to extract human body pose efficiently in every frame.

The existing algorithms for human motion analysis based on belief propagation such as [29, 75] typically require exhaustive search of the input image because minimal (or no) temporal information is employed for the inference. Our idea is to propagate the current posterior to the next frame for the future measurement.

2.3.1 Overview

We summarize here the basic theory for the inference algorithm based on belief propagation [29, 73]. Suppose that each body part p_i is represented with a 4 dimensional vector of $(x_i, y_i, s_i, \theta_i)$ – location, scale and orientation. The configuration of the entire human body B is composed of the location of each of m body parts, i.e. $B = \{p_1, p_2, \ldots, p_m\}$. Then, the log-likelihood of a configuration B given the mea-

surement from the current image I is:

$$L(B|I) \propto \sum_{(i,j)\in E} \Psi(p_i - p_j) + \sum_i \Phi(p_i), \tag{2.1}$$

where $\Psi(p_i - p_j)$ is the relationship between two body parts i and j, and $\Phi(p_i)$ is the observation for body part i. E is a set of edges that relate directly connected body parts, which we restrict to form a tree. In particular, $\Phi(p_i)$ is the measurement, or response map, for the i-th body part. It indicates the agreement between the appearance model of the i-th part and the location p_i. On the other hand, the geometric constraints are encoded by $\Psi(p_i - p_j)$, which indicates the agreement between parts i and j when placed at locations p_i and p_j. This term acts as a prior over body poses and encodes the plausible body configurations.

Inference in this tree model can be done exactly with a message passing algorithm. The messages are given by:

$$M_i(p_j) \propto \sum_{p_j} \Psi(p_i - p_j) O(p_i) \tag{2.2}$$

$$O(p_i) \propto \Phi(p_i) \prod_{k\in C_i} M_k(p_i), \tag{2.3}$$

where $M_i(p_j)$ is the bottom-up message from part p_i to p_j, $O(p_i)$ is the measurement of part p_i, and C_i is a set of children of part p_i. The top-down message from part p_j to p_i is defined by:

$$P(p_i|I) \propto \Phi(p_i) \sum_{p_j} \Psi(p_i - p_j) P(p_j|I), \tag{2.4}$$

which generates the probability map of each body part in the 4 dimensional state.

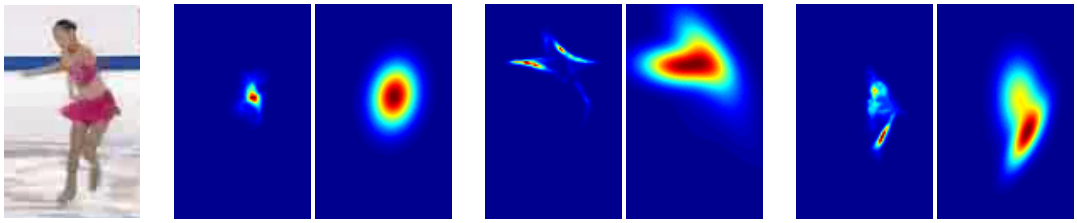
Based on this framework, we propose a method to propagate the density function in the temporal domain in order to reduce search space and produce temporally consistent results. The rest of the section describes the details of our algorithm.

2.3.2 Initialization

The first step for human body extraction is to estimate an initial body configuration and create a reliable appearance model. The initial location of the human is given by the method presented in Section 2.2. Note that the bounding box produced by the detection algorithm does not need to be very accurate since most of the background area will be removed by further processing. Once a potential human region is found, we apply a pose estimation technique [73] based on the same pictorial structure model. After inference, we then obtain a probability map for the configuration of each body part. In other words, the output of this algorithm is the probability map $P_p(u, v, s, \theta)$ for each body part p, where (u, v) is location, s is scale and θ is orientation. A sample probability map is presented in Figure 2.4 (b)-(d). Although this method creates accurate probability maps for each human body part, it is too computationally expensive to be used in every frame of a video sequence. Instead, we use this algorithm only for initialization and rely on temporal propagation of the estimation to reduce the computational load.

2.3.3 Representation of probability map

The original probability map P_p is represented by a non-parametric discrete distribution in the 4 dimensional space for each body part. There are several drawbacks with the use of such representation. First of all, it requires a significant amount of memory space, which is proportional to the image size and granularity of the orientations and scales. The memory requirements do not change even if most of the pixels in the image have negligible probabilities. Second, it is more desirable to propagate



(a) (b) torso (c) upper-left arm (d) upper-left leg

Figure 2.4: Approximation of pose estimation probability maps with Gaussian mixtures. Comparison between the true probability map for the pose estimation (left in each sub-figure) and its Gaussian mixture approximation (right) for each body part. The approximated density functions are propagated for the measurement in the next time step. Note that our approximation results look much wider since different scales in the color palette are applied for better visualization

a smooth distribution to the next time step instead of a discrete spiky density. For example, a distribution with many spikes could tend to ignore a significant number of potentially good candidate pose configurations due to their relatively low probability compared to nearby spikes.

Instead of using the non-parametric probability map, we employ a representation based on a parametric density function. However, finding a good parametric density function is not straightforward, especially when the density function is highly multi-modal as it is the case in human body pose estimates. In our problem, we observe that the probability map for the orientation of each body part is mostly uni-modal and close to a Gaussian distribution. [1] We employ a mixture of N Gaussians for the initialization of human body configuration, where N is the number of different orientations.

Denote by $\mathbf{x}_i^{(k)}$ and $\omega_i^{(k)}$ $(i = 1, \ldots, n)$ the location and weight of each point in the k-th orientation probability map. Let $\theta^{(k)}$ be the orientation corresponding the k-th orientation map. The mean ($\mathbf{m}^{(k)}$), covariance ($\mathbf{P}^{(k)}$) and weight ($\kappa^{(k)}$) of the

[1] Arms occasionally have significant outliers due to their flexibility. A uni-modal Gaussian fitting may result in more error here.

Gaussian distribution for the k-th orientation map is then given by:

$$\mathbf{m}^{(k)} = \begin{pmatrix} \mathbf{x}^{(k)} \\ \theta^{(k)} \end{pmatrix} = \begin{pmatrix} \sum_i \omega_i^{(k)} \mathbf{x}_i^{(k)} \\ \theta^{(k)} \end{pmatrix} \tag{2.5}$$

$$\mathbf{P}^{(k)} = \begin{pmatrix} \mathbf{V_X} & \mathbf{0} \\ \mathbf{0}^\top & V_\theta \end{pmatrix} = \begin{pmatrix} \sum_i \omega_i^{(k)} (\mathbf{x}_i^{(k)} - \mathbf{m}^{(k)})(\mathbf{x}_i^{(k)} - \mathbf{m}^{(k)})^\top & \mathbf{0} \\ \mathbf{0}^\top & V_\theta \end{pmatrix} \tag{2.6}$$

$$\kappa^{(k)} = \sum_i \mathbf{x}_i^{(k)} / \sum_k \sum_i \mathbf{x}_i^{(k)} \tag{2.7}$$

where $\mathbf{V_X}$ and V_θ are (co)variance matrices in spatial and angular domain, respectively. The representation of the combined density function based on all the orientation maps is given by:

$$\hat{f}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{N} \frac{\kappa^{(k)}}{\mid \mathbf{P}^{(k)} \mid^{1/2}} \exp\left(-\frac{1}{2} D^2 \left(\mathbf{x}, \mathbf{x}^{(k)}, \mathbf{P}^{(k)} \right) \right) \tag{2.8}$$

where $D^2\left(\mathbf{x}, \mathbf{x}^{(k)}, \mathbf{P}^{(k)}\right)$ is the Mahalanobis distance from $\mathbf{x}$ to $\mathbf{x}^{(k)}$ with covariance $\mathbf{P}^{(k)}$.

Even after simplifying the density functions for each orientation as Gaussians, it is still difficult to manage them in an efficient way. When the density is propagated temporally, the number of Gaussian components will increase exponentially if the densities are treated naively. We therefore adopt Kernel Density Approximation (KDA) [39] to further simplify the density function with little sacrifice in accuracy. KDA is a technique to approximate a multimodal density function with a mixture of Gaussians. The algorithm finds the mode locations of the underlying density function by an iterative procedure. Each mode is then approximated by a Gaussian distribution centered around that mode. Finally, the algorithm produces a compact mixture of Gaussians which is used to represent the original distribution.

Figure 2.4 presents the original pose estimation probability map and our approximation using a mixture of Gaussians for each body part. Note that the approximated density function is similar to the original and the multi-modality of the original density function is well preserved.

2.3.4 Measurement, inference and density propagation

Fast and accurate measurement and inference are critical in our algorithm. As shown in Equation (2.2) and (2.3), the bottom-up message depends on the relative configuration of the current part with its parent, the propagated measurements from the children parts and the observation for the current part. Exhaustive search is good for generating the measurement information at all possible locations, scales and orientations. However, computing measurements in the entire search space is slow, and more importantly, the accuracy in the inference process may be affected by spurious observations. Consider for instance that noisy measurements might be incurred due to other objects in the scene, and these may corrupt the inference process. A desirable reduction of search space not only decreases computation time, but also improves the accuracy. Therefore, instead of computing measurements at the entire search space, we restrict the measurements by a probability density function that characterizes the potential pose configuration of the human body. This probability function is represented by a mixture of Gaussians obtained from the KDA process and is propagated temporally in a sequential Bayesian filtering framework [2, 24, 40].

In our method, we perform local search based on propagated spatio-temporal information. We first diffuse the pose estimation posterior density function from the previous frame. The propagation can be done analytically since we represent the density as a Gaussian mixture. The diffused density indicates the restricted search space, so that measurements are done densely in local neighborhoods. Finally, a non-parametric density function for the measurement is constructed. Note that inference is

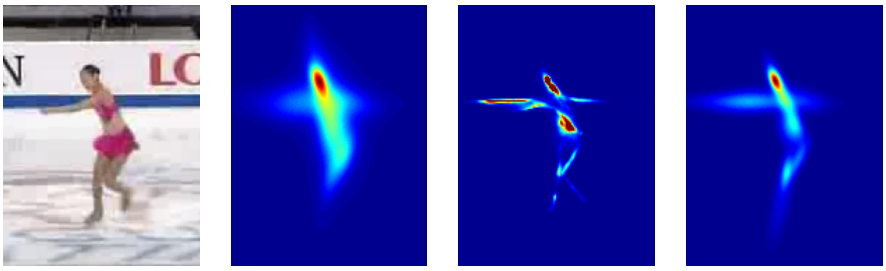
performed using the latter non-parametric density function; however, the diffusion to the next frame is done with the parametric representation. Given the non-parametric representation of the measurement, we can do inference with the message passing algorithm outlined in Section 2.3.1, which produces the pose estimate posteriors in non-parametric form. After inference, the non-parametric pose estimation density function is converted to a mixture of Gaussians using KDA, as described in Section 2.3.3. The posterior is given by the product of the diffused density and the pose estimation density function in the current frame. This step is conceptually similar to the integration of the measurement and inference history (temporal smoothing). We denote by $\mathbf{X}$ and $\mathbf{Z}$ the state and observation variables in the sequential Bayesian filtering framework, respectively. The posterior at the time step t of the state is given by the product of two Gaussian mixtures as follows:

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) \propto p(\mathbf{Z}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) \tag{2.9}$$

$$= \underbrace{\left(\sum_{i=1}^{N_1} \kappa_i \cdot \mathcal{N}(\mathbf{x}_i, \mathbf{P}_i)\right)}_{\text{diffused pose}} \underbrace{\left(\sum_{j=1}^{N_2} \tau_j \cdot \mathcal{N}(\mathbf{y}_j, \mathbf{Q}_j)\right)}_{\text{current pose}}, \tag{2.10}$$

where $\mathcal{N}(\cdot)$ represents a Gaussian distribution parametrized by its mean, and covariance. The first and second terms in the right hand side represent diffusion and pose estimation density function, respectively. Note that the product of two Gaussian mixtures is still a Gaussian mixture. This would cause an exponential increase of the number of components if at each time step we naively compute the product of the diffused and current pose densities. Instead, we apply KDA to the posterior density after each time step in order to maintain a compact representation of the density function.

The density propagation algorithm for inference is summarized in Algorithm 2.1, and illustrated in Figure 2.5.



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

Figure 2.5: Density functions in one time step of the human motion extraction process. (a) Original frame (cropped for visualization) (b) Diffused density function (c) Measurement and inference results (d) Posterior (Note that the probability maps for all orientations are shown in a single image by projection)

Algorithm 2.1 Extraction of human motion volumes with pictorial structures

1: Apply pedestrian detection to the input sequence
2: Apply clustering algorithm on the detections to obtain the number of people in the sequence.
3: **for** each person found **do**
4: Estimate the initial body pose and appearance at the first detection.
5: Construct a parametric representation of the pose estimation density function for each body part using KDA. The resulting mixture of Gaussians is also used as the posterior for this first frame.
6: **while** not at end of sequence **do**
7: Go to the next frame
8: **if** there exists a detection of the same person **then**
9: Optionally reinitialize the appearance and pose estimation.
10: **end if**
11: Diffuse the posterior from the previous frame
12: Perform measurement and pose estimation inference with the restricted search space as indicated by the diffused density.
13: Construct a parametric representation of the resulting pose estimation distribution using KDA.
14: Compute the pose estimation posterior by multiplying the diffusion and pose estimation densities.
15: **end while**
16: **end for**

Chapter 3

Experimental Results

In order to evaluate our extraction algorithm with top-down pictorial structures, we have collected a dataset of 50 sequences containing moving humans downloaded from *YouTube*. The sequences contain natural and complex human motions and various challenges mentioned in Section 2.1. Many videos have multiple shots (challenge 2), so we divide the original videos into several pieces based on the shot boundary detection, which is performed by global color histogram comparison with threshold [61]. We deal with each shot as a separate video. We have made this dataset public and it can be found at `http://vision.cs.princeton.edu/projects/extractingPeople.html`.

Instead of a 4 dimensional state space for human body configuration, we use a 3 dimensional state space for x and y location and orientation, and fix the scale depending on the pedestrian detection size. Although the estimated scale from the person detector is not very accurate, the extraction algorithm is robust enough to handle some variations in the scale. Also, the gaps between detections are generally not very long, and it is not often the case that we observe significant change in scale between two detections.

The measurement process is based on edge templates and color histogram for each body part, as in [73]. However, the search space for the measurement is significantly

Table 3.1: Performance evaluation of our algorithm for human motion volume extraction

	Detection only			Detection & Clustering			Full model		
	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
Rate	0.89	0.31	0.46	0.89	0.30	0.45	0.83	0.73	**0.78**
	0.90	0.25	0.39	0.91	0.24	0.38	0.87	0.62	**0.72**
	0.92	0.19	0.32	0.92	0.19	0.32	0.86	0.51	**0.64**
	0.93	0.16	0.27	0.94	0.15	0.27	0.92	0.43	**0.58**
	0.94	0.13	0.24	0.94	0.13	0.23	0.88	0.32	**0.46**

reduced as presented in the previous chapter. Figure 2.5 (b) illustrates the search space reduction, where low density areas are not sampled for the observations.

We first evaluate our system in terms of its capability to find the moving people in the video sequences. For each sequence, we have generated ground-truth by manually labeling every human present in each frame with a bounding box. Under this setting, we evaluate the retrieval performance of our system in terms of the precision-recall measures. We compare the precision-recall rates at three stages of our system: pedestrian detection only [53], people detection and clustering, and the full model. For a fixed threshold/operating point of the pedestrian detection algorithm, we obtain the three precision-recall pairs in each row of Table 3.1. Our full system provides the highest performance in terms of the F-measure. Recall that the F-measure is defined [93] as $2 \cdot (precision \cdot recall)/(precision + recall)$. This reflects the fact that our system achieves much higher recall rates by extracting non-upright people beyond the pedestrian detections.

We also evaluate the performance of our system in terms of the segmentation of the moving people. In this setting, we are interesting in finding the pixels that belong to the moving humans, while ignoring the pixels associated to the background scene. We create ground-truth for the spatial support of the moving people in the form of binary masks. We have labeled a random sample of 122 people from our 50 sequences. The evaluation of the pose estimation is performed at frames t_d, t_d+5 and

Figure 3.1: Example extraction results for various sequences. Each row corresponds to a separate sequence. More example sequences are available at the website: `http://vision.cs.princeton.edu/projects/extractingPeople.html`

$t_d + 10$, where t_d is a frame containing a pedestrian detection, and no detections are available in $[t_d+1, t_d+10]$. The average accuracies are 0.68, 0.68 and 0.63 respectively. Note that the accuracy decrease in the extracted person mask is moderate, and the temporal error propagation is small.

The results for several *YouTube* videos are presented in Figure 3.1, with some failure examples in Figure 3.2. Various general and complex human motions are extracted with reasonable accuracy, but there are some failures that are typically

Figure 3.2: Examples of failed extraction results

caused by inaccurate measurements. In a PC with a 2.33 GHz CPU, our algorithm requires around 10-20 seconds for the measurement and inference per person per frame, one order of magnitude faster than the full search method of [73].

3.1 Discussion

We presented a method to *automatically* extract human motion volumes from natural videos. The proposed method first identifies the location, trajectory and appearance of each person in the scene during the person detection and clustering stage. It then finds the configuration of the human body using a top-down pictorial structure model. Our method dramatically reduces the search space for the measurement and inference processes, by means of effective temporal propagation using a semi-parametric density function. Our system achieves promising results although many improvements can still be made.

A possible future direction is related to improving the body part appearance models to more robust body part detectors [1]. Also worth exploring is a closer integration between the top-down estimation and the people clustering stage, to create a closed-loop interaction that can potentially improve the accuracy of both stages.

The measurement process in the current algorithm is based solely on the top-down pictorial structure, however, the integration of bottom-up cues can potentially enable more robust and efficient processing, as we will show in Chapters 4 and 5.

Chapter 4

Efficient Extraction of Human Motion Volumes By Combining Top-Down and Bottom-up Cues

In this chapter, we present a method that achieves a balance between efficiency and accuracy for extracting human motion volumes from uncontrolled videos. We observe that a combination of top-down and bottom-up modeling can extract accurate motion volumes with only a relatively small computational load. Our idea is simple: given a video sequence, we apply top-down human models in a very sparse set of key frames. The bottom-up algorithm then bootstraps this detailed human information to complete the rest of the extraction through a temporal propagation and a global optimization procedure. Our experiments show that the proposed method achieves a near real-time human tracking in natural videos. Our contributions can be summarized as follows:

- A system is designed to automatically extract human motion volume from challenging videos by combining the top-down and the bottom-up method.

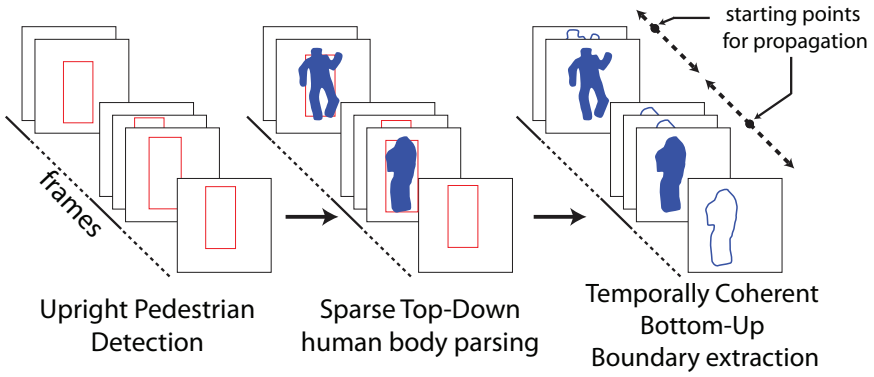


Figure 4.1: Overview of our efficient algorithm for extraction of human motion volumes. For each identified human obtained from pedestrian detection, a probabilistic human body shape on a sparse set of frames is computed by integrating top-down segmentations of detected pedestrian windows, which are driven by upright human pose templates (Section 4.2.1). A bottom-up boundary extraction based on the level-set formulation is employed to automatically refine and propagate the extracted contours to all frames (Section 4.2.2). The final human contours are obtained at all frames simultaneously, by jointly optimizing the level-set functions at all frames (Section 4.2.3)

- We propose a novel top-down modeling technique to obtain a probabilistic human body contour.
- A global optimization procedure based on belief propagation is proposed to improve the quality of results.

4.1 Algorithm Overview

Given an input video sequence, the goal of our algorithm is to carve out a spatio-temporal volume for each person in the video. The key strategy of our approach is the sparse introduction of top-down constraints, which are propagated in time in a bottom-up fashion. In this section, we describe the overall architecture of our method, which is also depicted in Figure 4.1.

Person Detection and Clustering We first use an upright human detector [53] to generate potential human regions. The appearance similarity and the spatio-temporal

coherence of the detections are employed to cluster detections in a similar fashion to the method presented in Section 2.2. Each resulting cluster is then associated to a unique individual, for which the spatio-temporal volume will be carved out. In practice, each cluster contains the bounding boxes for a person but there are many missing frames due to detection errors and pose variations.

Top-down Pose Estimation For each identified person, our algorithm performs a top-down extraction of the human region for a small subset of the sequence frames. At each of these frames, a level-set function that captures the contour of the detected person is initialized based on a probabilistic integration of upright human pose templates [97]. Such top-down driven extraction is utilized as an important constraint for the later bottom-up process. In this setting, the top-down information is delivered by the use of a set of fixed templates instead of more expensive part-based articulated models such as pictorial structures [29, 34, 73, 67]. However, we can only apply this top-down process to frames where pedestrian detections are available. At those frames, we have relatively high certainty that the person is in an upright pose, and thus we are more likely to succeed in estimating the contour of the person based on the pedestrian template database. Unfortunately, obtaining an accurate estimation of the initial level-set function is still challenging due to the limited variety in the upright human body template database and the lack of discriminative features.

Bottom-up Contour Extraction and Propagation In the previous step, we obtained the level-set functions for a small subset of the detected pedestrians. The level-set functions for the rest of the frames are initialized by propagating existing ones to adjacent frames bidirectionally using low-level feature observations, with a procedure based on an extension of [5]. The bottom-up level-set approach can handle the arbitrary shape of an object efficiently, but is inherently susceptible to fall in local optima. The combination of the top-down and the bottom-up approaches reduces

the drawbacks of both methods significantly. In addition, our algorithm jointly optimizes the level-set functions at all frames simultaneously, which provides accurate and temporally coherent boundaries of the human body.

4.2 Efficient Extraction of Human Motion Volumes

In this section, we describe three main components of our algorithm in detail, which include a top-down model-based probabilistic level-set initialization, a bottom-up feature-based propagation of level-set functions, and a global optimization process for contour extraction.

4.2.1 Top-down estimation of human body segmentation

For each identified person in the detection and clustering stage, a number of frames with pedestrian detections are available. At the detection windows, we are relatively certain that the humans are in an upright pose. It is therefore natural to consider top-down shape priors in the form of pedestrian silhouettes, which we apply only to a subset of those detection windows.

Suppose that $B = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$ is the set of upright human body templates in the database from [97]. At the j-th pedestrian detection window selected for top-down processing, we generate a set of multiple segmentations, each of which is driven by a different template in the database. In practice, we obtain each segmentation $\mathbf{h}_i$ within a level-set framework [18, 71], where the level-set function is initialized by the template $\mathbf{t}_i$. Each segmentation corresponds to an estimate of the human region. We integrate resulting segmentations probabilistically to obtain the probabilistic template

$\mathbf{p}_j$ as

$$\mathbf{p}_j = \sum_{i=1}^{n} \omega_i \mathbf{h}_i, \tag{4.1}$$

where the weight ω_i is estimated by the matching quality of the original template and the resulting segmentation. We measure the matching quality using multiple features—shape, color and edge—as follows:

Shape We measure the shape distance r_s between the original contour of a template in the database and the level-set segmentation induced in the detection window by

$$r_s = D_{\chi^2}(S(\mathbf{t}_i), S(\mathbf{h}_i)), \tag{4.2}$$

where $S(\cdot)$ are shape descriptors and D_{χ^2} is the χ^2 distance operator. In practice, we describe shapes using a histogram computed from the shape of the estimated region, which is a simplified version of the descriptor in [90].

Color We favor contours that induce the most distinct foreground and background color distributions. We estimate both distributions using the pixel assignments based on the estimated region $\mathbf{h}_i$. The distance between the foreground and the background color model—M_f^{color} and M_b^{color}, respectively—is defined by

$$r_c = D_{\chi^2}(M_f^{color}, M_b^{color}). \tag{4.3}$$

In practice, each color model is a multinomial distribution over the quantized color space.

Edges The dissimilarity between the edge map in the pedestrian window and the estimated region is measured by

$$r_e = \frac{1}{N} \sum_{c_i \in C} \min_{e_j \in \mathrm{E}} D(c_i, e_j), \tag{4.4}$$

where C is the set of N points in the contour of $\mathbf{h}_i$ and E is the set of edge pixels in the edge map. D measures the Euclidean distance between two pixels. This is equivalent to the average distance from the points in the contour to the edge points in the image. The computation is done via Distance Transform [10].

We combine the multiple cues to obtain the weight for each template by

$$\omega_i = \frac{\exp\left(\kappa_0 + \sum_{j \in \{s,c,e\}} \kappa_j r_j\right)}{1 + \exp\left(\kappa_0 + \sum_{j \in \{s,c,e\}} \kappa_j r_j\right)}, \tag{4.5}$$

where parameters κ_j $(j = 0, s, c, e)$ are mixing weights that indicate the relative importance of each cue. We learn this parameters from a small set of segmentation and template pairs using logistic regression. Finally, note that we normalize the template weights to ensure that $\sum_i \omega_i = 1$.

In our algorithm, the top-down constraints are applied to a small number of frames since top-down processing is more computationally expensive than bottom-up processing. Because, we would not gain much benefit from top-down processing many consecutive frames, it is applied to some frames that are temporally far apart. We first select a frame with a pedestrian detection randomly and add more frames with detections that are most distant from the current set of selected frames. A trade-off is observed here; the more frames selected for top-down processing, the more accurate the constraints for bottom-up processing will be, but at the same time the computational cost is increased.

Figure 4.2: Top-down estimation of human segmentation using upright pedestrian templates. (Top) Example templates in the pedestrian database from [97]. (Bottom) Template-driven segmentation results

The resulting estimated set of human body contours from this top-down processing step are used as constraints for the following bottom-up propagation. Some examples of the templates and the extracted contours by the top-down process are presented in Figure 4.2.

4.2.2 Bottom-up propagation of the human motion volume

After obtaining a sparse set of contours with the top-down process, we propagate the contours efficiently to other frames. The problem is formulated within a level-set contour tracking framework that is based on bottom-up cues. It is known that level-set based segmentation methods frequently converge to local optima. In our algorithm, we alleviate the problem by accurate initialization of the level-set functions at multiple frames using the top-down constraints.

In the level-set framework [18, 71], a region of interest R in image I is implicitly represented by a non-parametric level-set function $\mathbf{\Phi}$:

$$R = \{\mathbf{x} \in I | \mathbf{\Phi}(\mathbf{x}) > 0\}, \tag{4.6}$$

Figure 4.3: Bottom-up extraction of human motion volume boundaries. An efficient level-set method is proposed to extract the human region boundary. Initial boundaries from the top-down procedure (a) are propagated across time (b) and refined by evolving the implicit level-set function (c). The final boundary is generally more accurate after a few iterations. This figure is best viewed in color

where $\mathbf{x}$ is a pixel in the image, and the boundary is defined by the set of points such that $\mathbf{\Phi}(\mathbf{x}) = 0$. A foreground segment R is obtained by an iterative procedure based on low-level features from an initial level-set function.

In our formulation, we propagate the level-set functions induced by the top-down templates in both time directions, forward and backward. Let $\mathbf{\Phi}^{t-1}$ be the initialized level-set function at frame $t-1$, and T the length of the sequence of interest. We propagate $\mathbf{\Phi}^{t-1}$ to the temporally adjacent frames, by employing an image registration technique [3] that finds a rigid warping of $\mathbf{\Phi}^{t-1}$ to the new frame, e.g., frame t. We then apply a fixed small number of level-set iterations (typically, 5) to partially optimize the level-set function based on the observation in the new frame. The same process is done to propagate the contour to frame $t-2$. Such bidirectional propagation terminates when the level-set functions in all frames are initialized.

Let us present how the level-set function $\mathbf{\Phi}^t$ is optimized in each frame iteratively based on low-level image features. [1] Figure 4.3 shows an illustration of the within-frame level-set optimization. The goal is to evolve the initial level-set function by

[1] Our level-set evolution is based on the algorithm in [5], where a more detailed presentation is available.

maximizing the conditional probability given by

$$p(\mathbf{\Phi}^t|\mathbf{x}^t,\mathbf{y}^t) = \prod_{i=1}^{N^t} p(\mathbf{\Phi}_i^t|\mathbf{x}_i^t,\mathbf{y}_i^t), \tag{4.7}$$

where N^t is the number of pixels, $\mathbf{y}^t$ is the observed image feature, and the pixel-wise level-set likelihood is given by

$$\begin{aligned} p(\mathbf{\Phi}_i^t|\mathbf{x}_i^t,\mathbf{y}_i^t) &\propto p(\mathbf{x}_i^t|\mathbf{\Phi}_i^t,\mathbf{y}_i^t)p(\mathbf{\Phi}_i^t) \\ &= p(\mathbf{\Phi}_i^t)\sum_M p(\mathbf{x}_i^t|\mathbf{\Phi}_i^t,M)p(M|\mathbf{y}_i^t), \end{aligned} \tag{4.8}$$

where M is the model parameter for foreground (M_f) or background (M_b).

In [5], only color distribution is utilized to model the foreground and background regions. Here, we also introduce motion information based on optical flow. Both cues, motion and color, are used to compute foreground and background probabilities at each pixel, $p(M|\mathbf{y}_i^t)$. In order to use the motion information, we compute the foreground probability of a pixel in the new frame by transforming the foreground probability map in the previous frame using the motion vector given by optical flow. When more than one location in the old frame is transformed to the same location in the new frame, the average probability is assigned to the corresponding position in the new frame. If no pixel is transformed to a location in the new frame, we assign the median of its spatial neighborhood. We combine color and motion likelihoods for a final measurement map as the product of the two factors, which is given by

$$p(M|\mathbf{y}_i^t) = p(M^{color}|\mathbf{y}_i^t) \cdot p(M^{motion}|\mathbf{y}_i^t). \tag{4.9}$$

The integration of motion for the measurement process is particularly helpful to avoid distractions toward background objects visually similar to the target. In practice we use a simple optical flow estimation algorithm based on the Lucas-Kanade method [3].

We also introduce a new geometric prior $p(\mathbf{\Phi}_i^t)$. The new prior favors level-set functions which are close to a signed distance function. In addition to the standard constraint in the size of the gradient [5, 60], we also constrain its direction. The geometric prior $p(\mathbf{\Phi}_i)$ is defined as

$$p(\mathbf{\Phi}_i) \equiv p_m(\mathbf{\Phi}_i) p_d(\mathbf{\Phi}_i), \tag{4.10}$$

where p_m and p_d are the magnitude and direction term, respectively. Each of these terms is given by

$$p_m(\mathbf{\Phi}_i) = \frac{1}{\sigma_{m,i}\sqrt{2\pi}} \exp\left(-\frac{(|\nabla \mathbf{\Phi}_i| - 1)^2}{2\sigma_{m,i}^2}\right) \tag{4.11}$$

$$p_d(\mathbf{\Phi}_i) = \frac{1}{\sigma_{d,i}\sqrt{2\pi}} \exp\left(-\frac{\left(\alpha_i^\top \nabla \tilde{\mathbf{\Phi}}_i - 1\right)^2}{2\sigma_{d,i}^2}\right), \tag{4.12}$$

where α_i is the direction of local center of mass around $\mathbf{x}_i$, $\nabla\tilde{\mathbf{\Phi}}_i$ is the normalized gradient of $\mathbf{\Phi}_i$ and $\sigma_{m,i}$ and $\sigma_{d,i}$ describe uncertainty of each pixel. We favor gradient directions of the level-set function that coincide with the inward direction to the human body. Such prior tends to yield smoother level-set functions and human boundaries.

We can now proceed to optimize the objective function with respect to the level-set function $\mathbf{\Phi}$. The optimization problem is equivalent to maximizing the log-likelihood:

$$\log\left(p(\mathbf{\Phi}|\mathbf{x}, \mathbf{y})\right) \propto \sum_{i=1}^{N} \log\left(p(\mathbf{x}_i|\mathbf{\Phi}_i, \mathbf{y}_i)\right) - \log p(\mathbf{\Phi}_i). \tag{4.13}$$

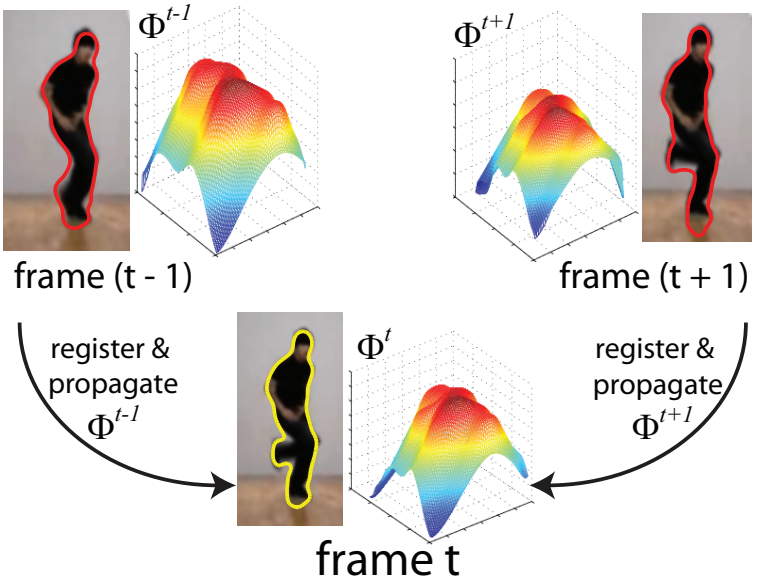


Figure 4.4: Temporally coherent contour extraction. Our formulation globally optimizes the level-set functions at all frames simultaneously. The level-set function that represents the human boundary is propagated in both directions, forward and backward, which yields a temporally coherent and accurate boundary

The optimization is performed iteratively using a gradient ascent algorithm with the following update equation:

$$\frac{\partial \log\left(p(\mathbf{\Phi}_i|\mathbf{x}_i,\mathbf{y}_i)\right)}{\partial \mathbf{\Phi}_i} = \frac{\delta_\epsilon(P_f - P_b)}{p(\mathbf{x}_i|\mathbf{\Phi}_i,\mathbf{y}_i)} - \left(\frac{\partial \log p_m(\mathbf{\Phi}_i)}{\partial \mathbf{\Phi}_i} + \frac{\partial \log p_d(\mathbf{\Phi}_i)}{\partial \mathbf{\Phi}_i}\right). \tag{4.14}$$

4.2.3 Temporally coherent global optimization

We have described the process of obtaining top-down human region estimates and its efficient propagation using bottom-up cues. However, the estimated human volumes in the previous step may not be reliable due to abrupt changes of the target object, falling in local optima or weak image features. To overcome these issues, we employ a global optimization that integrates temporal information more tightly. This is achieved by introducing explicit dependencies between temporally adjacent frames and jointly optimizing the level-set functions at all frames simultaneously. Such dependencies favor the extraction of contours that are more accurate and tem-

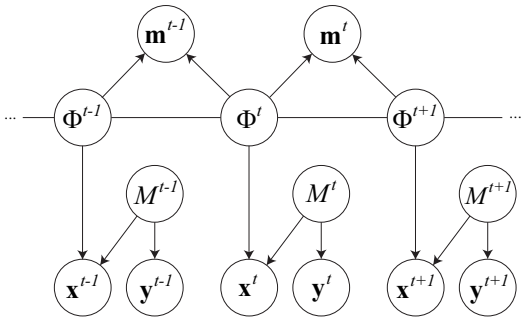


Figure 4.5: Graphical model for the temporally coherent global optimization. Circles indicate random variables, arrows indicate conditional dependencies and undirected links express mutual dependencies. Image locations are represented by $\mathbf{x}$, image observations by $\mathbf{y}$ and appearance models by M. The level-set function that implicitly defines the human contour is represented by $\mathbf{\Phi}$. The motion information that registers the level-set function across frames is indicated by $\mathbf{m}$. Superscripts indicate time step. By introducing dependencies between the level-set functions from adjacent frames, we can jointly estimate the optimal $\mathbf{\Phi}^{1:T}$ that exhibits temporal consistency and better extracts the the human boundary

porally coherent. As illustrated in Figure 4.4, the additional dependencies lead to a bidirectional propagation of the extracted contours among adjacent frames.

The set of $\mathbf{\Phi}^{1:T}$ resulting from the process described in previous sections provides initial level set estimates for the following optimization procedure. We introduce a graphical model to encode the dependencies between $\mathbf{\Phi}^{t}$ and ($\mathbf{\Phi}^{t+1}$, $\mathbf{\Phi}^{t-1}$), which is shown in Figure 4.5. We obtain the globally optimal $\mathbf{\Phi}^{1:T}$ by temporal belief propagation in an iterative message passing procedure.

The optimization problem is thus defined by a new objective function, which is given by

$$
\begin{aligned}
p(\mathbf{\Phi}_i^{1:T}|\mathbf{x}^{1:T},\mathbf{y}^{1:T}) &= \left[\prod_{t=1}^{T}\frac{1}{p(\mathbf{x}_t)}\sum_{M^t}p(\mathbf{x}_i^t|\mathbf{\Phi}_i^t,M^t)p(M^t|\mathbf{y}_i^t)\right]p(\mathbf{\Phi}_i^{1:T}) \\
&= \underbrace{\left[\prod_{t=1}^{T}\frac{1}{p(\mathbf{x}_t)}\sum_{M^t}p(\mathbf{x}_i^t|\mathbf{\Phi}_i^t,M^t)p(M^t|\mathbf{y}_i^t)\right]}_{\text{Pixel-wise likelihood}}\underbrace{\left[\prod_{t=1}^{T-1}\Psi(\mathbf{\Phi}_i^t,\mathbf{\Phi}_i^{t+1})\right]}_{\text{Temporal consistency}}\underbrace{\left[\prod_{t=1}^{T}p(\mathbf{\Phi}_i^t)\right]}_{\text{Geometric prior}}.
\end{aligned}
\tag{4.15}
$$

The first factor in Equation (4.15) specifies the estimations from individual frames, and the second factor defines the relationship between adjacent frames. This relationship is key to the temporal coherence of the extracted volume.

The message for temporal propagation between the frame t and $t+1$ is defined as

$$\Psi(\mathbf{\Phi}_i^t, \mathbf{\Phi}_i^{t+1}) = \exp\left(-\frac{(\mathbf{\Phi}_i^t - \mathbf{\Phi}_i^{t+1})^2}{\sigma_m^2}\right), \tag{4.16}$$

and the update message is

$$\frac{\partial \log\left(\Psi(\mathbf{\Phi}_i^t, \mathbf{\Phi}_i^{t+1})\right)}{\partial \mathbf{\Phi}_i^t} = -\frac{2\left(\mathbf{\Phi}_i^t - \mathbf{\Phi}_i^{t+1}\right)}{\sigma_m^2}, \tag{4.17}$$

which favors the temporal consistency of the human motion boundaries. Note that messages are received at frame t from both directions, from frame $t-1$ and $t+1$. The gradient ascent update including the messages for temporal consistency is obtained by the sum of the terms in Equation (4.14) and the messages for forward and backward update related to Equation (4.17). After the iterative procedure described in Section 4.2.2 converges for each frame, we update the messages in Equation (4.16) and synchronously pass it to neighboring frames, which is repeated until global convergence. Note that, in the above formulation, the two neighboring level-set functions propagated to the current frame are properly registered by a rigid transformation. Such registration accounts for global rigid motion across frames, gives a better prior to the new frame, and reduces the number of level-set iterations. When all the level-set functions $\mathbf{\Phi}^{1:T}$ converge, the human volume is finally given by the set of points such that $\mathbf{\Phi}_i^t > 0$.

Chapter 5

Experimental Results

We evaluate our method in terms of its segmentation accuracy on annotated frames. We compare our algorithm to the method presented in Chapter 2 using the *YouTube* dataset from [67]. It is a very challenging dataset that has 50 sequences containing unknown and arbitrary camera motion, cluttered background, motion blur, compression artifacts, etc.

Precision and recall are computed based on this dataset for three different algorithms—our full system, our method without global optimization and the method in Section 2 [67]. Table 5.1 summarizes the experimental results of the three systems, and shows that our methods even without global optimization improve both precision and recall significantly. We attribute this improvement to the ability of our tracker to leverage salient bottom-up cues for human/background separation that are constrained by effective top-down template-driven segmentation. Our temporally coherent optimization process further improves the precision of the system by integrating information across time. Some comparative results are provided in Figure 5.2.

We also collected a larger video dataset, also composed of sequences downloaded from *YouTube*. Several examples of human body extraction in this dataset are presented in Figure 5.3.

Table 5.1: Experimental results for our efficient extraction of human motion volumes. This evaluation is performed on the *YouTube* dataset from [67]. The segmentation of humans in videos is evaluated as a retrieval problem. Ground truth consists of a set of over 180 masks that correspond to the human regions in selected frames from the dataset. For each retrieved mask, a precision is computed as the area of the intersection of the retrieved and ground-truth mask over the area of the retrieved mask; whereas recall is the area of the intersection over the area of the ground-truth mask

Method	Prec	Rec	F-score
Full model	0.74	0.75	0.74
Full model without global opt.	0.62	0.76	0.68
Niebles et al. [67]	0.57	0.44	0.50

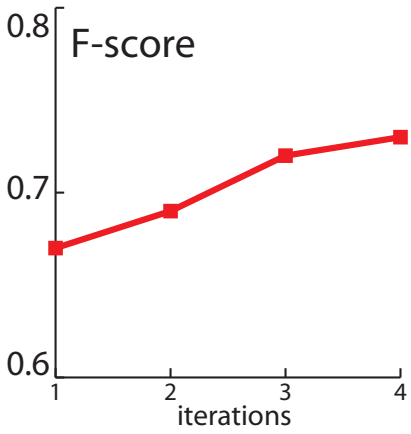


Figure 5.1: Segmentation accuracy vs. number of global iterations. Iteratively propagating the contours across multiple frames helps reduce sporadic artifacts and produces a more temporally smooth human motion boundary

Our model is not only more accurate in moving human extraction, but also computationally much more efficient. In our implementation with Matlab, one boundary is obtained in less than 50 ms per bottom-up propagation. Similarly, a top-down step with template-driven segmentation takes about the same time. In practice, we apply the top-down process to 20% of the pedestrian detections per person in the video.

Figure 5.2: Example extraction results on *YouTube* sequences. For each video, we randomly sample three frames, and compare the extraction results of our method (row 1), with a simplified version of our method (without global optimization and top-down component, in rows 2) and our replication of the method in [67] (rows 3). The outlines of the humans are drawn in color curves. We observe that, in general, our full algorithm performs better than the other two methods

We use a set of 100 templates in the pedestrian silhouette database. When running 4 global iterations and with the top-down process applied to 10% of the total number of frames in the sequence, our algorithm runs in less than a half a second per frame per person on average. Most of the time is spent on those few frames where the top-down process is applied, which takes about 5 seconds per frame per person. The method in [67] runs in more than 20 seconds per frame per person on similar hardware.

Figure 5.3: Example extraction results on eight gymnastics sequences from an assortment of about 500 *YouTube* videos. The players in the video exhibit a rich variety of challenging motions. Nevertheless, our algorithm is able to retrieve the contour of the person. The colored number at the corner of each image indicates the frame number in the original sequence. The outlines of the humans are drawn in color curves

5.1 Discussion

We have demonstrated a technique to efficiently extract moving humans from challenging sequences, where top-down modeling provides important constraints to a bottom-up propagation scheme, and the global optimization refines the contour around each moving person in the sequence. As shown empirically, our method outperforms state-of-the-art techniques at a fraction of the computational cost. This speed allows us to collect a larger set of annotated natural videos containing human motions from *YouTube*. The set contains about 500 sequences with over 70k frames.

While the results are promising, there are still many research oportunities within the algorithm proposed here. For instance, our algorithm does not have any explicit occlusion handling, which is critical especially in videos with crowded scenes. It might be also interesting to incorporate other cues in the appearance model of each target person such as clothing texture or skin color.

Part III

Categorizing Simple Human Actions With Local Features and Statistical Models

Chapter 6

Introduction

Imagine a video taken on a sunny beach, where some people are playing beach volleyball, some are surfing, and others are taking a walk along the beach. Can a computer automatically tell what is happening in the scene? Can it identify different human actions? We explore the problem of human action categorization in video sequences. Our interest is to design an algorithm that permits the computer to learn models for human actions. Then, given a novel video, the algorithm should be able to decide which human action is present in the sequence. Furthermore, we look for means to provide a rough indication of where (in space and time) the action is being performed.

The task of automatic categorization and localization of human actions in video sequences is highly interesting for a variety of applications: detecting relevant activities in surveillance video, summarizing and indexing video sequences, organizing a digital video library according to relevant actions, etc. It remains, however, a challenging problem for computers to achieve robust action recognition due to cluttered background, camera motion, occlusion, view point changes, and geometric and photometric variances of objects.

These challenges are common to a broad range of computer vision tasks. A cluttered background introduces information that is not relevant to the signal of interest,

making the latter harder to isolate. Camera motion creates ambiguities in the motion patterns that are observed in the image plane: it could make an object appear static when it is moving with the same speed and direction as the camera. In addition, human actions can also be observed only partially due to occlusions; thus, the actual signal of interest can be dramatically reduced. Finally, viewpoint changes as well as geometric and photometric variance produce very different appearances and shapes for the same category examples, resulting in high intra-class variances.

Consider for example, a live video of a figure skating competition, the skater moves rapidly across the rink and the camera also moves to follow the skater. With moving cameras, cluttered background, and moving target, few vision algorithms could identify, categorize and localize such motions well. In addition, the challenge is even greater when there are multiple activities in a complex video sequence. In this work, we will present an algorithm that aims to account for these scenarios.

We explore two approaches to action recognition using *statistical models* and a representation based on *local features*. At the feature level, our first approach will focus on the use of spatiotemporal patches, extracted from detected interest points. At the model level, we propose the use of *latent topic models* as the learning and classification tool. These models need to incorporate the "bag of words" assumption, and thus ignore the spatial and temporal arrangement of the local features. On the other hand, our second approach proposes the *hybrid usage of static shape features as well as spatiotemporal features*. At the model level, we propose a novel hierarchical framework which can be characterized as a *constellation of bag of features*. This model allows us to incorporate information about the spatial arrangement of the local features. Thus it permits us to model the mutual geometric relationship among parts. This is specially important for structured objects such as the human body.

The rest of Part III is organized in the following way. We review previous related work in Chapter 6.1. In Chapter 7, we describe our approach to action recognition us-

ing latent topic models, including the spatio-temporal feature representation, a brief overview of the *probabilistic latent semantic analysis* (pLSA) and *latent Dirichlet allocation* (LDA) models in our context, and the specifics of the learning and recognition procedures. In Chapter 8, we present a novel hierarchical model for action recognition. In Chapter 9, we present the experimental results on human action recognition using real datasets, and also compare our performance with other methods.

6.1 Related Work

A large set of previous work has addressed the question of human action categorization and motion analysis. One line of work is based on the computation of correlation between volumes of video data. Efros et al. [25] perform action recognition by correlating optical flow measurements from low resolution videos. Their method requires first segmenting and stabilizing each human figure in the sequence, as well as further human intervention to annotate the actions in each resulting spatio-temporal volume. Shechtman and Irani [81] propose a behavior-based correlation to compute the similarity between space-time volumes in order to find similiar dynamic behaviors and actions. Their method requires to specify a query action template, which will be correlated to videos in database. At each pixel, the space-time gradients of the corresponding video patch must be computed and summarized in a matrix. The eigenvalues of the resulting matrices are used to compute similarity between two spatio-temporal patches. Therefore, this method requires significant computation due to the correlation procedure between every patch of the testing sequence and the video database.

Another popular approach is to first track body parts and then use the obtained motion trajectories to perform action recognition. This is done with much human supervision and the robustness of the algorithm is highly dependent on the tracking

system. Ramanan and Forsyth [74] approach action recognition by first tracking the humans in the sequences using a pictorial structure procedure. Then 3D body configurations are estimated and compared to a highly annotated 3D motion library. The algorithm permits assigning composed labels to the testing sequences; however, it relies heavily on the result of the tracker, and the estimation of the 3D pose may introduce significant errors due to hard-to-solve ambiguities. In Yilmaz and Shah [108], human labeling of landmark points in the human body is first done at each frame in sequences from multiple moving cameras. Then actions are compared using their corresponding 4D (x, y, z, t) trajectories. Thus, their approach can be applied to action recognition and retrieval, with the cost of a significant amount of human annotation. In the work by Song et al. [89] and Fanti et al. [26], feature points are first detected and tracked in a frame-by-frame manner. Multiple cues such as position, velocities and appearance are obtained from these tracks. Then human actions are modeled utilizing graphical models based on triangulated graphs. These models can be learnt in an unsupervised fashion, but cannot deal with dynamic backgrounds or moving cameras.

Alternatively, researchers have considered the analysis of human actions by looking at video sequences as space-time intensity volumes. Bobick and Davis [8] use motion history images that capture motion and shape to represent actions. They introduced the global descriptors *motion energy image* and *motion history image*, which were used as templates that could be matched to stored models of known actions. Their method depends on background subtraction and thus cannot tolerate moving cameras and dynamic backgrounds. Blank et al. [6] represent actions as space-time shapes and extract space-time features for action recognition, such as local space-time saliency, action dynamics, shape structures and orientation. Similarly, this approach relies on the restriction of static backgrounds which allows them to segment the foreground using background subtraction.

Other lines of work have been proposed for video analysis. Boiman and Irani [9] recently proposed composing the new observations as an ensemble of local video patches from previous examples in order to localize irregular action behavior in videos. Dense sampling of the patches is necessary in their approach, and therefore, the algorithm is very time-consuming. It is not suitable for action recognition purpose due to the large amount of video data commonly presented in these settings. Another work regarded as video epitomes is proposed by Cheung et al. [14]. They model the space-time cubes from a specific video by a generative model. The learnt model is a compact representation of the original video, therefore this approach is suitable for video super-resolution and video interpolation, but not for recognition.

Another approach uses a video representation based on spatio-temporal interest points. In spite of the existence of a fairly large variety of methods to extract interest points from static images [79], less work has been done on space-time interest point detection in videos. Laptev [52] presents a space-time interest point detector based on the idea of the Harris and Förstner interest point operators [43]. They detect local structures in space-time where the image values have significant local variations in both dimensions. However, this method produces a small number of stable interest-points and which are often non sufficient to characterize complex sequences. In addition, Dollár et al. [23] proposes a detector based on a set of separable linear filters, which generally produces a high number of detections. This method responds to local regions which exhibit complex motion patterns, including space-time corners. Also, a number of descriptors are proposed for the resulting video patches around each interest point. Ke et al. [47] apply spatio-temporal volumetric features that efficiently scan video sequences in space and time. Their method builds on the rectangle features used by Viola and Jones [94]. Their approach detects interest points over the motion vectors, and requires dense estimation of the optical flow. Additionally, the method requires the calculation of a significant number of features, in the order

of a million, even after discretizing and sampling the feature space. The features are then employed to perform human action categorization with a discriminative cascade classifier, which requires annotated positive and negative examples. Finally, a recent approach by Oikonomopoulos et al. [70] extends the idea of saliency regions in spatial images to the spatiotemporal case. The work is based on the spatial interest points of Kadir and Brady [46], which is extended to the space-time case.

Interest points extracted with such methods have been used as features for human action classification. In [23, 47, 70, 80], the space-time interest points were combined with discriminative classifiers to learn and recognize human actions. Therefore, local space-time patches have been proven useful to provide semantic meaning of video events by providing a compact and abstract representation of patterns. While these representations show good potential, the modeling and learning frameworks based on discriminative classifiers [23, 80] do not have clear applicability in more challenging situations such as multiple action recognition.

Finally, we note the success of generative approaches based on latent topics models for object and scene recognition. In Sivic et al. [85], unsupervised learning and recognition of object classes is performed by applying a pLSA model with the "bag of visual words" representation. The approach permits learning object classes from images with no label and background clutter. Also, Fei-Fei and Perona [27] studied the application of latent topic models to the task of scene categorization. The models are inspired by the LDA model [7], and can learn intermediate topic distributions in an unsupervised manner.

All the previous work suggest that improvement can be made by relaxing assumptions of annotated data, stationary cameras and backgrounds and discriminative approaches. Thus, we are interested in exploring the use of a generative approach where unsupervised learning methods can be applied, in conjunction with a representation

based on local features. We will present our proposed algorithms in the following chapter.

Chapter 7

Latent Topic Models For Human Action Categorization

We propose a generative graphical model approach to learn and recognize human actions in video, taking advantage of the robust representation of sparse spatio-temporal interest points and an unsupervised learning approach. In the context of our problem, unsupervised learning is achieved by obtaining action model parameters from unsegmented and unlabeled video sequences, which contain a known number of human action classes. Thus, we claim that an unsupervised learning setting is desirable because the amount of unlabeled video data is increasing daily and human annotation is expensive. We would like to take advantage of these digital resources without the cost of human supervision. Second, as opposed to discriminative models, a generative approach provides a mean to learn models in an unsupervised fashion.

Our method is motivated by the recent success of object detection/classification or scene categorization from unlabeled static images, using latent topic models [85, 27]. One key consideration in these works is regarded as the "bag of keypoints" representation [21], where the geometric arrangement between visual features is ignored. This is commonly implemented as a histogram of the number of occurrences of particular

visual patterns in a given image. Here, we refer to this assumption as the "bag of words" representation, similar to other approaches utilizing latent topic models. It is worth noting that latent topic models were initially applied to text analysis tasks, a domain from where the "bag of words" assumption has been inherited. In spite of their simplicity, the latent topic models have been successfully applied to challenging computer vision tasks, which motivates us to explore their applicability in the human action categorization domain.

Two related latent topic models are generally used: probabilistic latent semantic analysis (pLSA) by Hofmann [44] and latent Dirichlet allocation (LDA) by Blei et al. [7]. Here, we investigate the suitability of both models for video analysis by exploring the advantages of the powerful representation and the great flexibility of these generative graphical models.

The contributions of this work are twofold. First, we propose an unsupervised learning approach for human actions using a bag of keypoints representation. We apply two latent topic models, pLSA and LDA, to the problem of learning and recognizing human action categories, while adopting a "bag of spatio-temporal words" representation for video sequences.

Second, our method is also able to localize and categorize multiple actions in a single video. In addition to the categorization task, our approach can also localize different actions simultaneously in a novel and complex video sequence. This includes the cases where multiple people are performing distinct actions at the same time, and also situations where a single person is performing distinct actions through time.

In order to gather experimental evidence that supports our proposed approach, we train and recognize action models on three different datasets [6, 80, 99]. Also, we used those models to perform recognition in videos from a different dataset [89], as well as test sequences taken by ourselves. Such results are presented on Chapter 9.

A preliminary version of this work appeared in BMVC 2006 [68] and IJCV 2008 [69].

7.1 Approach Overview

Given a collection of unlabeled videos, our goal is to automatically learn different classes of actions present in the data and to apply the learned model to perform action categorization and localization in the new video sequences. Our approach is illustrated in Figure 7.1. We assume that the videos can contain some camera motion, for instance, the one observed in videos taken with a hand held camera. Also, we expect the videos to contain a dynamic background that might generate some motion clutter. In the training stage, we assume that there is a single person performing only one action per video. However, we relax this assumption at the testing stage, where our method can handle observations containing more than one person performing different actions.

We are given a set of unlabeled video sequences, and we would like to discover a set of classes from them. Each of these classes would correspond to an action category, such that we can build models for each class. Additionally, we would like to be able to understand videos that are composed by a mixture of action categories, in order to handle the case of multiple motions. This resembles the problem of automatic topic discovery in text analysis [7, 44]. Thus, we find a similar interpretation as that initially proposed by the use of latent topic models for object and scene classification [85, 27]. In our case, we would like to analyze video sequences instead of text documents; video sequences are summarized as a set of spatio-temporal words instead of text words; we seek to discover action categories instead of text topics; and we expect to explain videos as a mixture of actions instead of text documents as a mixture of topics. In this work, we investigate two models that were proposed in the text analysis literature to

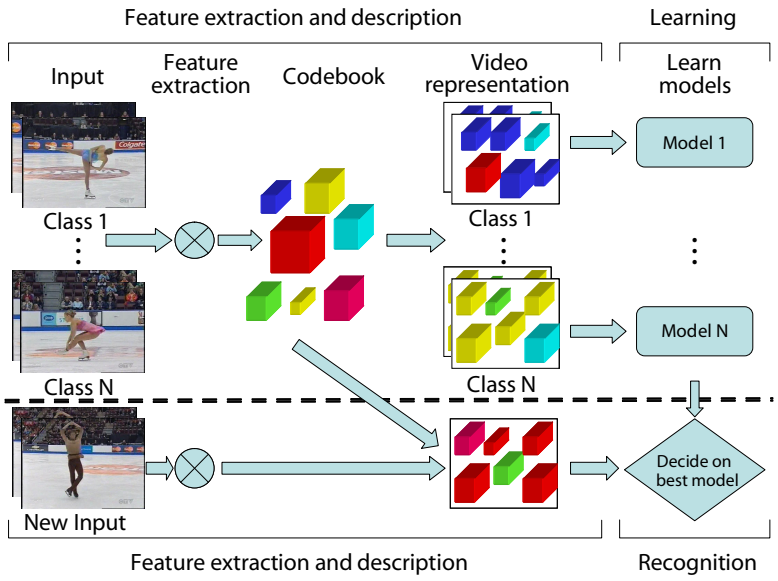


Figure 7.1: Flowchart of our approach to simple action recognition with latent topic models. To represent motion patterns we first extract local space-time regions using the space-time interest points detector [23]. These local regions are then clustered into a set of spatio-temporal words, called *codebook*. Probability distributions and intermediate topics are learned automatically using one of the two models: pLSA or LDA. The learned models can then be used to recognize and localize human action classes in novel video sequences

address the latent topic discovery problem. First, we employ the simpler pLSA model proposed by Hoffman [44]. Second, we consider the LDA model proposed by Blei et al. [7], which provides a rigorous generative setting, permits the inclusion of priors in a bayesian manner, and addresses the overfitting issues presented in the pLSA model. Both models learn their parameters in an unsupervised fashion.

An important characteristic of the pLSA and LDA models is that they are based on the "bag-of-words" assumption, that is, the order of words in a text document can be neglected. This is equivalent to regarding the words in a document as *exchangeable*. In addition, the particular ordering of the documents in the document collection can also be neglected, yielding a further exchangeability assumption at the document level. In the context of human action classification, the "bag-of-words" assumption

translates into a video representation that ignores the positional arrangement, in space and time, of the spatio-temporal interest points.

7.2 Feature Representation from Space-Time Interest Points

There are several choices in the selection of good features to describe pose and motion. In general, there are three popular types of features: static features based on edges and limb shapes [20, 31], dynamic features based on optical flow measurements [20, 83], and spatio-temporal features obtained from local video patches [6, 14, 52, 23, 47, 70]. In particular, features from spatio-temporal interest points have shown to be useful in the human action categorization task, providing a rich description and powerful representation [23, 47, 70, 80].

As Figure 7.1 illustrates, we represent each video sequence as a collection of spatio-temporal words by extracting space-time interest points. Among the available interest point detectors for video data, the interest points obtained using the generalized space-time corner detector [52] are too sparse to characterize many complex videos. This was noted first in [23] and confirmed in our experience with complex sequences such as the figure skating videos (Figure 9.3, p. 85). We choose to use the separable linear filter method in [23], since it generally produces a high number of detections. Note, however, that our method does not rely on a specific interest point detector algorithm, as long as the detector produces a sufficiently large number of interest points. In the following, we provide a brief review of the detector proposed in [23].

Assuming a stationary camera or a process that can account for camera motion, separable linear filters are applied to the video to obtain the response function as follows:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{7.1}$$

where $g(x, y; \sigma)$ is a 2D Gaussian smoothing kernel, applied only along the spatial dimensions (x, y), and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, which are defined as $h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$. The two parameters σ and τ correspond to the spatial and temporal scales of the detector, respectively. In all cases we use $\omega = 4/\tau$, thus reducing to two the number of parameters in the response function R. To handle multiple scales, one must run the detector over a set of spatial and temporal scales. For simplicity, we run the detector using only one scale and rely on the codebook to encode the few changes in scale that are observed in the dataset.

It was noted in [23] that any region with spatially distinguishing characteristics undergoing a complex motion can induce a strong response. However, regions undergoing pure translational motion, or without spatially distinguishing features will not induce a strong response. The space-time interest points are extracted around the local maxima of the response function. Each patch contains the volume that contributed to the response function, i.e., its size is approximately six times the scales along each dimension.

Figure 7.2 shows an example of interest point detection in a hand waving video sequence. Each colored box corresponds to a detected interest point, that is associated with a video patch. The neighborhood size is determined by the scale parameters σ and τ of the detector. Interest points are correctly localized where significant motion occurs.

To obtain a descriptor for each spatio-temporal cube, we calculate its brightness gradients on x, y, and t directions. The spatio-temporal cube is then smoothed at different scales before computing the image gradients. The computed gradients are concatenated to form a vector. The size of the vector is equal to the number of pixels in the cube times the number of smoothing scales times the number of gradients directions. This descriptor is then projected to a lower dimensional space

Figure 7.2: Spatio-temporal interest point detection in an action video sequence using the method of separable linear filters by Dollár et al. [23]. Each red box corresponds to a video patch that is associated with a detected interest point. The neighborhood size is determined by the scale parameters σ and τ of the detector. The interest points are localized where significant motion occurs and can be used to generate a sparse representation of the video sequence. For a visualization of all the frames in particular spatio-temporal patches, please refer to Figure 9.5 (p. 88)

using the principal component analysis (PCA) dimensionality reduction technique. In [23], different descriptors have been used, such as normalized pixel values, brightness gradient and windowed optical flow. We find that both the gradient descriptor and the optical flow descriptor are equally effective in describing the motion information. In the following, we will describe results obtained with gradient descriptors.

7.3 Codebook Formation

The latent topic models pLSA and LDA rely on the existence of a finite vocabulary of (spatio-temporal) words of size V. In order to learn the vocabulary of spatio-temporal words, we consider the set of descriptors corresponding to all detected spatio-temporal interest points in the training data. This vocabulary (or codebook) is constructed by clustering using the k-means algorithm and Euclidean distance as the clustering metric. The center of each resulting cluster is defined to be a spatio-temporal word (or codeword). Thus, each detected interest point can be assigned a unique cluster membership, i.e., a spatio-temporal word, such that a video can be represented as a collection of spatio-temporal words from the codebook. The effect of the codebook

size was explored in our experiments, and is shown in Figures 9.4 (p. 87) and 9.8 (p. 91).

7.4 Learning the Action Models: Latent Topic Discovery

In the following, we will describe the pLSA and LDA models in the context of human action categories analysis, adapting the notation and terminology as needed from the ones introduced by [85, 7].

7.4.1 Learning and recognizing the action models by pLSA

Suppose we have a set of $M(j = 1, \ldots, M)$ video sequences containing spatio-temporal words from a vocabulary of size $V(i = 1, \ldots, V)$. The corpus of videos is summarized in an $V \times M$ co-occurrence table $\bar{M}$, where $m(w_i, d_j)$ stores the number of occurrences of a spatio-temporal word w_i in video d_j. In addition, there is a latent topic variable z_k associated with each occurrence of a spatio-temporal word w_i in a video d_j. Each topic corresponds to an action category, such as walking, running, etc.

The joint probability $P(w_i, d_j, z_k)$ is assumed to have the form of the graphical model shown in Figure 7.3:

$$P(d_j, w_i) = P(d_j)P(w_i|d_j) \tag{7.2}$$

Given that the observation pairs (d_j, w_i) are assumed to be generated independently, we can marginalize over topics z_k to obtain the conditional probability $P(w_i|d_j)$:

$$P(w_i|d_j) = \sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k) \tag{7.3}$$

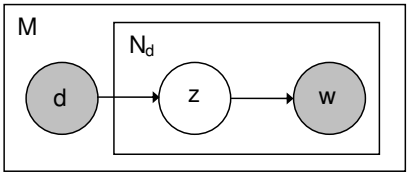


Figure 7.3: The pLSA graphical model. Nodes are random variables. Shaded ones are observed and unshaded ones are unobserved. The plates indicate repetitions. In the context of human action categorization, d represents video sequences, z are action categories, and w are spatio-temporal words. The parameters of this model are learnt in an unsupervised manner using an EM procedure. This figure is reproduced from [7]

where $P(z_k|d_j)$ is the probability of topic z_k occurring in video d_j, and $P(w_i|z_k)$ is the probability of spatio-temporal word w_i occurring in a particular action category z_k. K is the total number of latent topics, hence the number of action categories in our case.

Intuitively, this model expresses each video sequence as a convex combination of K action category vectors. In other words, the video-specific word distributions $P(w_i|d_j)$ are obtained by a convex combination of the aspects or action category vectors $P(w_i|z_k)$. Videos are characterized by a specific mixture of factors with weights $P(z_k|d_j)$. This amounts to a matrix decomposition with the constraint that both the vectors and mixture coefficients are normalized to make them probability distributions. Essentially, each video is modeled as a mixture of action categories: the histogram for a particular video being composed from a mixture of the histograms corresponding to each action category.

We then fit the model by determining the action category histograms $P(w_i|z_k)$ (which are common to all videos) and the mixture coefficients $P(z_k|d_j)$ (which are specific to each video). In order to determine the model that gives the high probability to the spatio-temporal words that appear in the corpus, a maximum likelihood estimation of the parameters is obtained by maximizing the following objective function

using the expectation-maximization (EM) algorithm:

$$\prod_{i=1}^{V}\prod_{j=1}^{M} P(w_i|d_j)^{m(w_i,d_j)} \tag{7.4}$$

where $P(w_i|d_j)$ is given by Equation (7.3).

Given that our algorithm has learnt the action category models, our goal is to categorize new video sequences. We have obtained the action category specific video word distributions $P(w|z)$ from a different set of training sequences. When given a new video, the unseen video is 'projected' on the simplex spanned by the learnt $P(w|z)$. We need to find the mixing coefficients $P(z_k|d_{test})$ such that the KL divergence between the measured empirical distribution $\tilde{P}(w|d_{test})$ and $P(w|d_{test}) = \sum_{k=1}^{K} P(z_k|d_{test})P(w|z_k)$ is minimized [85, 44]. Similarly to the learning scenario, we apply the EM algorithm to find the solution. Thus, a categorization decision is made by selecting the action category that best explains the observation, that is:

$$\text{Action Category} = \arg\max_k P(z_k|d_{test}) \tag{7.5}$$

Furthermore, we are also interested in localizing multiple actions in a single video sequence. Though our "bag of spatio-temporal words" model itself does not explicitly represent the spatial relationship of local video regions, it is sufficiently discriminative to localize different motions within each video. This is similar to the approximate object segmentation case in [85]. The pLSA model models the posteriors by:

$$P(z_k|w_i, d_j) = \frac{P(w_i|z_k)P(z_k|d_j)}{\sum_{l=1}^{K} P(w_i|z_l)P(z_l|d_j)} \tag{7.6}$$

For the spatio-temporal word corresponding to each interest point, we can label the topics for each word by finding the maximum posteriors $P(z_k|w_i, d_j)$. Then we can localize multiple actions corresponding to different action categories.

7.4.2 Learning and recognizing the action models by LDA

As noted in [7], pLSA is not a well-defined generative model of documents, since there is no natural way to use it to assign probability to a new testing observation. In addition, the number of parameters to be estimated in pLSA grows linearly with the number of training examples, which suggest that this model is prone to overfitting. LDA [7] is presented to remedy for these weaknesses.

Suppose we have a set of $M(j = 1, \ldots, M)$ video sequences containing spatio-temporal words from a vocabulary of size $V(i = 1, \ldots, V)$. Each video d_j is represented as a sequence of N_j spatio-temporal words $\mathbf{w} = (w_1, w_2, \ldots, w_{N_j})$. Then the process that generates each video d_j in the corpus is:

1. Choose the number of spatio-temporal words: $N_j \sim \text{Poisson}(\xi)$
2. Choose the mixing proportions of the action categories: $\theta \sim \text{Dir}(\alpha)$
3. For each of the N_j words w_n:
   - Choose an action category (topic): $z_n \sim \text{Mult}(\theta)$
   - Choose a spatiotemporal word w_n from the multinomial distribution $p(w_n|z_n, \boldsymbol{\beta})$

Here we fixed the number of latent topics K to be equal to the number of action categories to be learned. Also, α is the parameter of a K-dimensional Dirichlet distribution, which generates the multinomial distribution θ that determines how the action categories (latent topics) are mixed in the current video. In addition, a matrix $\boldsymbol{\beta}$ of size $K \times V$ parameterizes the distribution of spatio-temporal words conditioned on each action category; each element of $\boldsymbol{\beta}$ corresponds to the probability $p(w_i|z_k)$.

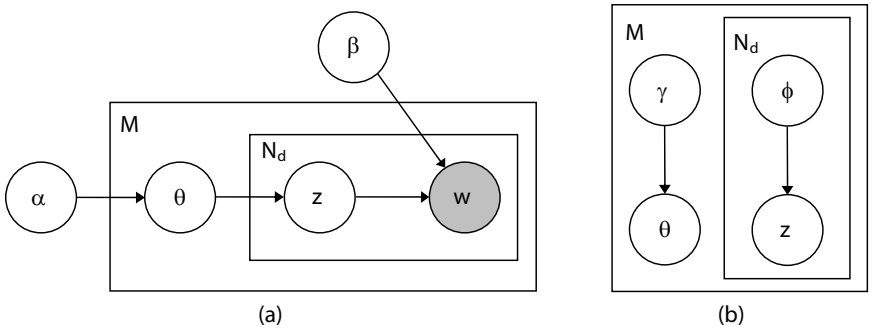


Figure 7.4: (a) LDA graphical model [7]. Nodes are random variables. Shaded ones are observed and unshaded ones are unobserved. The plates indicate repetitions. In the context of human action categorization, θ represents video sequences, z are action categories and w are spatio-temporal words. α is the hyperparameter of a Dirichlet distribution. (b) Graphical model that represents the variational distributions proposed in [7] to approximate the posterior probability in LDA. This figure is reproduced from [7]

The joint distribution of a topic mixture θ, the set of words $\mathbf{w}$ observed in the current video, and their corresponding topic (action category) $\mathbf{z}$ can be written as:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \boldsymbol{\beta}) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \boldsymbol{\beta}) \tag{7.7}$$

The probabilistic graphical model in Figure 7.4 represents the LDA model.

In order to perform video classification with LDA, one must compute the posterior distribution of the hidden variables given a new input:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \boldsymbol{\beta}) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \boldsymbol{\beta})}{p(\mathbf{w}|\alpha, \boldsymbol{\beta})} \tag{7.8}$$

where θ is specific to each input and represents its latent topics distribution. Once θ is inferred, a classification decision can be made by selecting the most likely topic in the current testing video.

Although it is computationally intractable to perform inference and parameter estimation for the LDA model in general, several approximation algorithms have been investigated. A variational inference approach has been proposed in [7]. The family

of variational distributions that are considered can be represented by the model in Figure 7.4(b), and is characterized by:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{7.9}$$

where γ and θ are the free variational parameters. The corresponding optimization procedure produces the parameters (γ^*, ϕ^*) which are a function of $\mathbf{w}$.

Analogously to the pLSA case, the posterior Dirichlet parameters $\gamma^*(\mathbf{w})$ represent the projection of the new observed video into the simplex spanned by the latent topics. Thus, classification is performed by selecting the action category that corresponds to the maximum element in $\gamma^*(\mathbf{w})$.

Furthermore, the localization procedure can also be implemented using LDA. In this case, we can label each interest point with an action category, by selecting the topic that generates its corresponding spatio-temporal word with highest probability. That means, for a fixed i, we select k such that $p(w_i|z_k)$ in $\boldsymbol{\beta}$ is maximum.

7.5 Motivations and Limitations of the Proposed Approach

In the following section, we discuss the suitability of our approach as well as its limitations at three different levels: the representation using local information only, the "bag of words" assumption and the use of latent topic models as the underlying statistical structure of the data.

7.5.1 Local features

Our approach relies on the detection of spatio-temporal interest points, which produce a sparse representation of the video sequences. Small video patches are extracted

from each interest point and constitute the local information that is used to learn and recognize human action categories. By employing local features, we intend to emphasize the importance and distinctiveness of the short-range spatio-temporal patterns. We argue that the observed local patterns are sufficiently discriminative across human action classes (refer to Figure 9.5, p. 88), and provide a reasonable feature space that allows building good human action models. Additionally, this approach relaxes the need for previously common preprocessing steps in global approaches such as background substraction in [8, 6], or figure tracking and stabilization in [25].

In terms of the local descriptor, a number of video patch descriptors have been proposed previously [23, 55]. In our implementation, we have chosen a very simple descriptor based on image gradients [23], noting that such descriptor does not provide scale invariance in the space and time domains. It does not capture relative camera motion. However, more complex descriptors that include small invariances to spatial scale and speed, as well as invariances to small camera motions, are available with the cost of more computational complexity (for instance, local position dependent histograms in [55]). In our implementation, we rely on the codebook to handle scale changes and camera motions. As long as the newly observed local features do not contain patterns of scale change and camera motion that are extremely different from those observed in the data used to form the codebook, we expect that similar local features will be assigned to consistent memberships on the codebook.

7.5.2 Bag of words

We have adopted the "bag of words" assumption in our data representation. This implies ignoring the spatial and temporal arrangement of the detected local interest points. The lack of spatial information provides little cues about the human body configurations, while the lack of longer-term temporal information does not permit to model more complex actions that are not constituted by simple repetitive patterns.

Alternative approaches might include structural information by encoding information of the human body using a pictorial structure model [29], by observing co-ocurrances of local patterns such as those in [78], or by modeling the geometrical arrangement of local features [65].

7.5.3 Latent topic models

We propose the use of latent topic models for human action recognition motivated by the sucess of these approaches in other computer vision tasks [85, 27]. First, these models provide an unsupervised learning framework that permits automatically discovering semantic clusters in the training data. Second, as opposed to discriminative methods such as support vector machines (SVM), pLSA and LDA permit performing meaningful reasoning on the data beyond classification, for example, topic localization. Furthermore, such localization can be realized without the need of scanning thousands or millions of windows per image. These models, however, do not provide spatial nor temporal scale invariances. Thus, they can only work within a small margin of the scales that have been observed in training. Alternative approaches that include such invariances might be based upon models such as those in [32].

Chapter 8

A Hierarchical Model For Human Action Classification

Based on the recent work in human motion categorization [74, 26, 6, 68], we make two key observations that will in turn influence the design of our model. The first observation is based on the usage of different feature descriptors to represent human body and/or human motion. The second observation deals with the choice of the category model that uses such features for corresponding classification.

Using good features to describe pose and motion has been widely researched in the past few years. Generally speaking, there are three popular types of features: static features based on edges and limb shapes [20, 31, 73]; dynamic features based on optical flows [25, 20, 83], and spatio-temporal features that characterizes a space-time volume of the data [6, 14, 23, 54]. Spatial-temporal features have shown particular promise in motion understanding due to its rich descriptive power [9, 80, 68]. On the other hand, to rely on only such features means that one could only characterize motions in videos. Our daily life experiences tell us, however, humans are very good at recognizing motion based on a single gesture. Fanti et al. [26] proposed that it is fruitful to utilize a mixture of both static and dynamic features. In their work,

the dynamic features are limited to simple velocity description. We therefore propose the *hybrid usage of static shape features as well as spatio-temporal features* in our framework.

Model representation and learning are critical for the ultimate success of any recognition framework. In human motion recognition, most models are divided into either discriminative models or generative models. For example, based on the spatio-temporal cuboids, Dollar et al. [23] applied an SVM classifier to learn the differences among videos containing different human motions. Ramanan et al. [73] recently proposed a conditional random field (CRF) model to estimate human poses. While discriminative frameworks are often very successful in the classification results, they suffer from either the laborious training problem or a lack of higher level semantic interpretation of the images beyond the classification task. In the CRF framework, one needs to train the model by labeling by hand each part of the human body. And in the SVM framework, the model is not able to "describe" the actual motion of the person. Some researchers, therefore, have proposed several algorithms based on probabilistic graphical model frameworks in action categorization/recognition. Song et al. [89] and Fanti et al. [26] represent the human action model as a triangulated graph. Boiman and Irani [9] recently propose to extract an ensemble of local video patches to localize irregular action behavior in videos. Dense sampling of the patches is necessary in their approach and therefore the algorithm is very time-consuming. It is not suitable for action recognition purpose due to the large amount of video data commonly presented in these settings.

For structured objects such as human bodies, it is important to model the mutual geometric relationship among different parts. Constellation models offer such a solution [26, 101]. Unfortunately due to the computational complexity of the model, previous work only used a very small number of features (typically 4 to 6) or approximated the connections by triangulation [89, 26]. Another approach is to lose all the

geometric information and consider "bag of words" models. They have proven highly efficient and effective in classifying objects [85, 37] and human motion [23, 68]. We propose here a method to exploit both the geometric power of the constellation model as well as the richness of the "bag of words" model. We recognize the computational limit of having a very small number of fully connected parts in the constellation model. But instead of applying it directly onto the image level features, we attach a "bag of words" model to each part of the constellation model. The overall representation embodies a hierarchical model that combines a constellation model of few parts with bag of words models of a large and flexible number of features (see Figure 8.1). Our model is partly inspired by a hierarchical model proposed by Bouchard and Triggs [11]. In their framework, they also use the idea of attaching a large number of features at the image level to a handful of intermediate level parts. The key difference between our model and theirs is that our intermediate level parts are fully connected, whereas theirs are not, offering a much richer constraint. In addition, we use a mixture of models for our motion classes whereas it is not immediately clear whether their framework could be easily extended to a mixture model.

In summary, we show in this chapter a hierarchical model that learns different categories of human motion using a hybrid of spatio-temporal and static features. Our model can be characterized as a constellation of bag of words. Our results show that compared to previous work, our model offers superior classification performance on a number of large human motion datasets. In addition, it can do so either on a video sequence or in individual frames.

A preliminary version of this work has been published in CVPR 2007 [65].

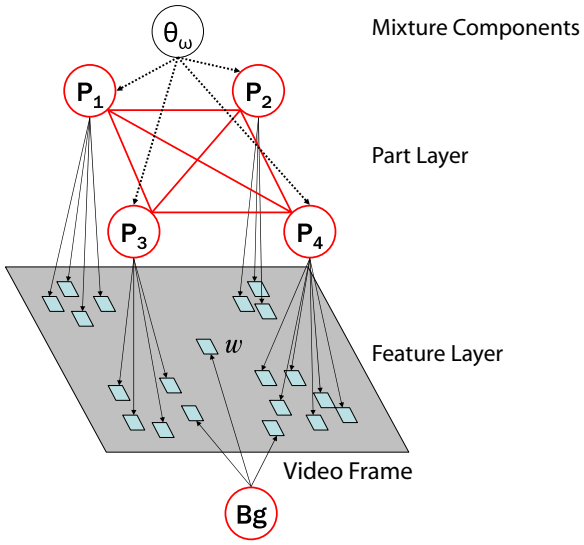


Figure 8.1: Hierarchical model for human actions: The proposed model combines, in a hierarchical way, the geometric strength of the constellation model with the larger number of features utilized in the bag of keypoints models. The higher layer is a constellation of P parts, each associated with a bag of features in the lower layer. The parts are interrelated by a distribution of their relative positions. Additionally, each part defines a distribution of appearance and position of features assigned to it

8.1 Theoretical Framework

In the simplest version, our model is a two layered hierarchical model. The higher layer is close in spirit to the shape term of the constellation model. It is composed of a set of P parts. Our model selects one part as the reference part and represents the relative location of all other parts with respect to the reference by a Gaussian distribution. Each of the P_p parts ($p = 1 \ldots P$) is connected to N_p image features in the lower level, and is associated to distributions of appearance and relative location of the features assigned to it. In other words, the higher layer is a constellation of parts, and each of these parts is associated to a "bag of features" in the lower layer. Due to its geometric constraints, this model is suitable for capturing similar body configurations or poses.

Following the observation that human actions are results of sequences of poses, which arise from a few sets of similar body configurations, we believe that a single

action is better represented as a multimodal distribution of shape and appearance. To account for this multimodality, we use a mixture of hierarchical models, where each component corresponds to a set of poses clustered together according to their similarity.

8.1.1 The hierarchical model

Given a video frame $\mathbf{I}$, we find a set of N observed features $\mathbf{w} = \{\mathbf{x}, \mathbf{a}\}$, where $w_i = \{x_i, a_i\}$ denotes position x_i and appearance a_i information. We also suppose that there is a known finite set $\mathbf{Y}$ of possible positions for the P parts in the image. One can think of $\mathbf{Y}$ as pixel locations or any arbitrary choice. We can compute the likelihood of the observed data given an action model θ as the following:

$$p(\mathbf{w}, \mathbf{Y}|\theta) = \sum_{\omega=1}^{\Omega} \sum_{\mathbf{h} \in H} p(\mathbf{w}, \mathbf{Y}, \mathbf{h}, \omega|\theta) \tag{8.1}$$

$$= \sum_{\omega=1}^{\Omega} \sum_{\mathbf{h} \in H} p(\mathbf{w}, \mathbf{Y}, \mathbf{h}|\omega, \theta) p(\omega|\theta) \tag{8.2}$$

$$= \sum_{\omega=1}^{\Omega} \left[\pi_\omega \sum_{\mathbf{h} \in H} p(\mathbf{w}, \mathbf{Y}, \mathbf{h}|\omega, \theta) \right] \tag{8.3}$$

where ω indicates the mixture component, we define $\pi_\omega = p(\omega|\theta)$ such that $\sum_\omega \pi_\omega = 1$ and $\mathbf{h}$ is an indexing variable which we call a *hypothesis* (similar to the constellation model). If $|\mathbf{Y}|$ is the number of possible locations for the P parts, then $\mathbf{h}$ is a vector of length P, where each element is between 1 and $|\mathbf{Y}|$. Additionally, we introduce the variable $\mathbf{m}$, which indicates an assignment of features to parts. In particular, each $\mathbf{m}$ is a vector of N elements which can take integer values in the interval $[0, P]$. That means each feature can be assigned to the background (0) or to one of the P parts

$(1 \dots P)$. Marginalizing over $\mathbf{m}$, we rewrite the observed data likelihood as

$$p(\mathbf{w}, \mathbf{Y}|\theta) = \sum_{\omega=1}^{\Omega} \left[\pi_\omega \sum_{\mathbf{h} \in H} \sum_{\mathbf{m} \in M} p(\mathbf{w}, \mathbf{Y}, \mathbf{h}, \mathbf{m}|\theta_\omega) \right] \tag{8.4}$$

$$p(\mathbf{w}, \mathbf{Y}|\theta) = \sum_{\omega=1}^{\Omega} \left[\pi_\omega \sum_{\mathbf{h} \in H} \left(p(\mathbf{h}|\theta_\omega) p(\mathbf{Y}|\mathbf{h}, \theta_\omega) \sum_{\mathbf{m} \in M} p(\mathbf{w}|\mathbf{Y}, \mathbf{m}, \mathbf{h}, \theta_\omega) p(\mathbf{m}|\mathbf{Y}, \mathbf{h}, \theta_\omega) \right) \right] \tag{8.5}$$

Calculating the likelihood in Equation (8.5) requires computing $O((P+1)^N)$ different assignments for each $\mathbf{h}$. Considering that $|H| = |\mathbf{Y}|^P$, we need to compute the probabilities of $O\left((P+1)^N |\mathbf{Y}|^P\right)$ different combinations of hypothesis-assignment. In order to make the model more computationally tractable, we propose the following approximation:

$$\sum_{\mathbf{m} \in M} p(\mathbf{w}|\mathbf{Y}, \mathbf{m}, \mathbf{h}, \theta_\omega) p(\mathbf{m}|\mathbf{Y}, \mathbf{h}, \theta_\omega) \approx p(\mathbf{w}|\mathbf{Y}, \mathbf{h}, \mathbf{m}^*, \theta_\omega)$$

That is, we compute only one assignment per hypothesis. If we assume that $p(\mathbf{m}|\mathbf{Y}, \mathbf{h}, \theta)$ is uniform, then $\mathbf{m}^*$ is selected such that:

$$\mathbf{m}^* = \arg\max_{\mathbf{m}} p(\mathbf{w}|\mathbf{Y}, \mathbf{h}, \mathbf{m}, \theta) \tag{8.6}$$

Applying this to (8.5), the approximated observed data likelihood is:

$$p(\mathbf{w}, \mathbf{Y}|\theta) \approx \sum_{\omega=1}^{\Omega} \left[\pi_\omega \sum_{\mathbf{h} \in H} p(\mathbf{h}|\theta_\omega) \underbrace{p(\mathbf{Y}|\mathbf{h}, \theta_\omega) p(\mathbf{w}|\mathbf{Y}, \mathbf{m}^*, \mathbf{h}, \theta_\omega)}_{\textit{Part layer Local feature layer}} \right] \tag{8.7}$$

Part layer term We represent the joint probability of the position of the P parts in the model as a multivariate Gaussian distribution:

$$p(\mathbf{Y}|\mathbf{h}, \theta) = \mathcal{N}(\mathbf{Y_T}(\mathbf{h})|\boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L)$$

In order to obtain translation invariance, we map $\mathbf{Y}$ into a translation invariance space, by constructing $\mathbf{Y_T}(\mathbf{h})$, a $2(P-1)$ dimensional vector that contains the relative positions of $(P-1)$ parts with respect to the topmost part.

Local feature layer term Given a part-to-feature assignment, each part P is instantiated as a set of image features that carry appearance and location information. Thus, each part is associated with an appearance distribution as well as a relative position distribution of image features. We adopt the bag-of-features assumption, where the observations $\mathbf{w}_n \in \mathbf{I}$ are conditionally independent given their parent assignments in $\mathbf{m}$. This assumption allows us to write the likelihood of a set of observations $\mathbf{w}$, given the possible part locations $\mathbf{Y}$, a hypothesis $\mathbf{h}$, an assignment $\mathbf{m}$, and the model parameters θ, as

$$\begin{aligned} p(\mathbf{w}|\mathbf{Y},\mathbf{h},\mathbf{m},\theta) &= \prod_{w_n \in \mathbf{I}} p(\mathbf{w}_n|\mathbf{Y},\mathbf{h},\mathbf{m}_n,\theta) \\ &= \prod_{\mathbf{w}_j \in bg} p(\mathbf{w}_j|\theta_0) \prod_{p=1}^{P} \prod_{\mathbf{w}_i \in P_p} p(\mathbf{w}_i|\mathbf{Y},h_p,\theta_p) \\ &= \prod_{\mathbf{w}_j \in bg} p(x_j^r|\theta_0^X) p(a_j|\theta_0^A) \prod_{p=1}^{P} \prod_{\mathbf{w}_i \in P_p} p(x_i^r|\mathbf{Y},h_p,\theta_p^X) p(a_i|\theta_p^A) \end{aligned} \tag{8.8}$$

where we define $\theta_p^X = \left\{\boldsymbol{\mu}_p^X = 0, \boldsymbol{\Sigma}_p^X\right\}$ to be the parameters of a Gaussian distribution that determines the relative position of the features that belong to the pth parent. Note that given a particular $\mathbf{m}$, the position information x_i of the ith image feature can be transformed to the relative location x_i^r of the feature to its assigned parent. Similarly, θ_p^A are the parameters of a multinomial distribution that describe the appearance of the features assigned to the pth parent. In the same manner, we define θ_0^X and θ_0^A as parameters for the appearance and position distribution of features assigned to the background. Note that the notations $\mathbf{w}_i \in P_p$ and $\mathbf{w}_j \in bg$ indicate assignments that depend on both $\mathbf{h}$ and $\mathbf{m}$.

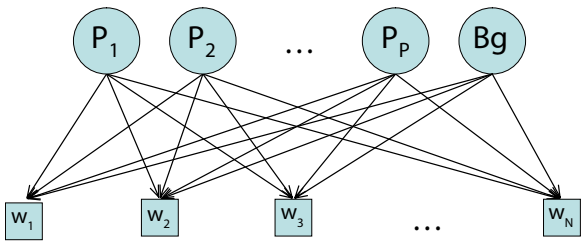


Figure 8.2: Matching features (children) to parts (parents): The weight of each link between a parent node P_p and a child node $\mathbf{w}_n$ is the probability of generating $\mathbf{w}_n$ from the appearance and position distributions assigned to P_p: $p(\mathbf{w}_i|\mathbf{Y}, h_p, \theta_p) = p(x_i|\mathbf{Y}, h_p, \theta_p^X)p(a_i|\theta_p^A)$. We also include a background node at the parent level, to allow features to be assigned to the background

Additionally, the assumption of conditional independence allows us to maximize the probability $p(\mathbf{w}_n|\mathbf{Y}, \mathbf{h}, \mathbf{m}_n, \theta_\omega)$ with respect to $\mathbf{m}$ for each $\mathbf{w}_n$ independently. In other words, our task reduces to find the best parent for each child node $\mathbf{w}_n$ in the graph of Figure 8.2.

Note that from this procedure, it is possible for $\mathbf{m}^*$ to be a feature to part assignment such that a part has no features assigned to it. This allows the model to handle naturally missing or occluded parts. An alternative to be explored is to assign features to parts softly, instead of using the single best parent for each child.

Approximated data likelihood Assuming that the prior probability of selecting a particular hypothesis $\mathbf{h}$ is uniform, i.e., $p(\mathbf{h}|\theta) = |H|^{-1}$, we can finally rewrite our likelihood equation as:

$$p(\mathbf{w}, \mathbf{Y}|\theta) \approx \frac{1}{|H|} \sum_{\omega=1}^{\Omega} \left[\pi_\omega \sum_{\mathbf{h} \in H} \mathcal{N}(\mathbf{Y_T}(\mathbf{h})|\boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L) p(\mathbf{w}|\mathbf{Y}, \mathbf{m}^*, \mathbf{h}, \theta_\omega) \right] \tag{8.9}$$

8.1.2 Learning

Learning consists of estimating the model parameters for each action category. In the case of the mixture of models, each action class is parametrized by

$$\theta_\omega = \left\{ \boldsymbol{\mu}_{L,\omega}, \boldsymbol{\Sigma}_{L,\omega}, \boldsymbol{\Sigma}^X_{p,\omega}, \theta^A_{p,\omega}, \theta^X_0, \theta^A_0 \right\}$$

for $p = 1 \dots P$ and $\omega = 1 \dots \Omega$. To accomplish this purpose, we adopt an EM algorithm.

Initialization The convergence of the EM algorithm to a sensible minimum depends greatly on the starting point. In order to select a good initial point, we cluster video frames from the training data into a number of clusters equal to the number of mixture components. The clustering procedure is done by representing each video frame with a histogram of features. Then we select a small number of frames from each resulting cluster and fit a 1-component model to them. The output of this procedure is a set of initial parameters θ^{old}.

E-step Evaluate the responsibilities using the current parameter values θ^{old}:

$$p(\mathbf{h}, \omega | \mathbf{w}, \mathbf{Y}, \theta^{old}) \approx \frac{\pi_\omega p(\mathbf{Y}|\mathbf{h}, \theta^{old}_\omega) p(\mathbf{h}|\theta^{old}_\omega) p(\mathbf{w}|\mathbf{Y}, \mathbf{h}, \mathbf{m}^*, \theta^{old}_\omega)}{p(\mathbf{w}, \mathbf{Y}|\theta^{old})} \tag{8.10}$$

M-step Calculate updated parameters θ^{new} using the current responsibilities:

$$\theta^{new} = \arg\max_\theta \sum_{\mathbf{h}} p(\mathbf{h}, \omega | \mathbf{w}, \mathbf{Y}, \theta^{old}) \ln p(\mathbf{w}, \mathbf{Y}, \mathbf{h}, \omega | \theta) \tag{8.11}$$

8.1.3 Recognition

Given a new video frame and the learnt models for each action class, the task is to classify the new image as belonging to one of the action models. Suppose that we have learned models for C action classes. We calculate the likelihood of observing

the image data given that it has been generated from each of the C action models. This produces a C-dimensional feature vector of the input in the model space. We calculate these feature vectors for each example in a *validation set*, and use them to train a discriminative classifier. Therefore, a classification decision is made by first calculating the likelihood of the input according to each of the C action models, and then categorizing this C-dimensional feature vector using the discriminative classifier.

Additionally, decisions can be made over a range of video frames by adopting a bag-of-frames strategy. First, each frame is categorized independently, and votes in favor of the detected action class. The complete video sequence is classified to be from the category with the majority of the votes.

8.2 The System

8.2.1 Image features

We represent each video frame as a set of detected patches $\mathbf{w} = \{\mathbf{x}, \mathbf{a}\}$, where $w_i = \{x_i, a_i\}$, $i = 1 \ldots N$. The appearance information $\mathbf{a}$ is obtained by assigning each patch a membership to a large dictionary of codewords. We show now how these patches are obtained and memberships assigned.

We adopt a rich representation by detecting static and motion features. This allows the model to characterize a larger number of human actions than when using motion alone. Specifically, certain actions, such as hand waving in [6], produce a small number of motion features since most body parts remain static.

Static features are obtained by first computing an edge map using a Canny edge detector. A set of edge points is sampled from the edge map, and a descriptor is obtained for an image patch around each selected point by calculating its shape context [4].

Motion features are obtained using the separable linear filter method in [23]. Small video patches are extracted and described by concatenating their gradients on space and time directions.

An example of the extracted static and dynamic features in frames of a particular video sequence is presented in Figure 8.3.

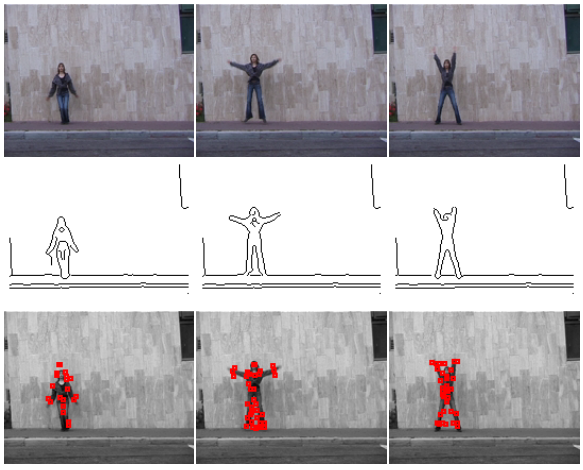


Figure 8.3: Detection of spatio-temporal local features. The first row contains example frames from a training sequence. The edge maps shown in the second row are obtained using the Canny edge detector. The third row illustrates the spatio-temporal interest point detection. The motion features are obtained using the method in [23]. The figure is best viewed in color

Given the collection of detected static features from the training images of all categories, we learn a codebook by the employment of a k-means algorithm. Codewords are then defined as the centers of the learnt clusters, and each static patch is assigned to the closest codeword. A similar procedure is performed to obtain a codebook of motion features, and the corresponding memberships.

The employment of two different types of features requires adopting two different distributions of feature appearance for each part. In particular, $p(a_i|\theta_p^a)$ is actually modeled as two multinomial distributions, one for static features and other for motion features. Thus, given a particular feature, we use the appropriate appearance

distribution when calculating Equation (8.8). Note that the proper distribution to use can be determined unambiguously since the type of the feature is always known.

8.2.2 Implementation details

In our implementation, we detect spatial features at each frame by sampling edge points from the output of the Canny edge detector. The number of samples is fixed at 100. Each sampled edge point is described using shape context with 3 spatial and 8 angular bins. The dimensionality of both descriptor types (static and dynamic) is reduced using PCA. Consequently, we cluster static and motion descriptors into codebooks of size 100. The discriminative classifier described in Section 8.1.3 is instantiated by an SVM. For this purpose, we use a linear SVM trained with the software package *LIBSVM* [13].

Chapter 9

Experimental Results

In this section, we present the human action datasets used in our experiments. We also present the experimental results obtained using the models discussed in the previous sections.

9.1 Datasets

In order to evaluate experimentally our algorithms, we use three different datasets. Here we include a small description for each of them.

9.1.1 KTH human action dataset

The human motion dataset from The Royal Institute of Technology (KTH) is the largest available video sequence dataset of human actions [80]. Each video has only one action. The dataset contains six types of human actions (walking, jogging, running, boxing, hand waving, and hand clapping) performed several times by 25 subjects in different scenarios of outdoor and indoor environment with scale change. It contains 598 short sequences. Some sample images are shown in Figure 9.1.

Figure 9.1: Example images from video sequences in the KTH dataset [80]. The dataset contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. These are performed several times by 25 subjects in different scenarios of outdoor and indoor environment. The camera is not static and the videos contain scale changes. This figure is reproduced from http://www.nada.kth.se/cvap/actions/

9.1.2 Weizmann Institute human action dataset

We employ the human action dataset from [6]. It contains 10 action categories performed by 9 people, to provide a total of 90 videos. Example frames of the action categories are shown in Figure 9.2. This dataset contains videos with static camera and simple background, however, it provides a good testing bed to investigate the performance of the algorithm when the number of categories is increased.

9.1.3 SFU figure skating dataset

We use the SFU (Simon Fraser University) figure skating dataset from [99]. [1] We adapt 32 video sequences which contain seven people executing three actions: stand-spin, camel-spin, and sit-spin, as shown in Figure 9.3. The dataset contains sequences with camera motion, background clutter, and aggressive view point changes.

[1] The work in [99] addresses the problem of motion recognition from still images. There is much other work to model motion in still images, which is out of the scope of this manuscript.

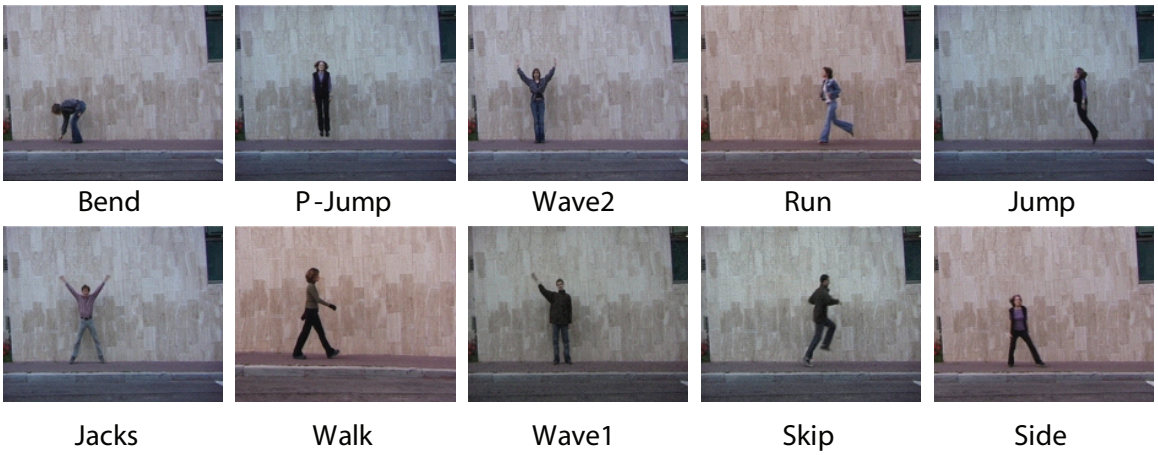


Figure 9.2: Example images from video sequences in the Weizmann Institute human action dataset [6]. The dataset contains 10 action categories, performed by 9 subjects. The videos are taken with static camera and static background

Figure 9.3: Example frames from video sequences in the figure skating dataset [99]. We adapt 32 video sequences from the original dataset, to produce a subset which contains seven people executing three actions: camel-spin (first row), sit-spin (second row) and stand-spin (third row). The videos are taken with a moving camera and dynamic background

9.2 Experiments Using the Latent Topic Models

We test our algorithm using three datasets: the KTH human motion dataset [80], a figure skating dataset [99], and the human action dataset from [6]. These datasets contain videos of cluttered background, moving cameras, and multiple actions; as well as videos exhibiting a single action, with static camera and simple background. We can handle the noisy feature points arisen from dynamic background and moving

cameras by utilizing the latent topic models pLSA and LDA, as long as the background does not amount to an overwhelming number of feature points. In addition, we demonstrate multiple actions categorization and localization in a set of new videos collected by the authors. We present the datasets and experimental results below.

9.2.1 Recognition and localization of single actions

Human action recognition and localization using KTH data

We extract interest points and describe the corresponding spatio-temporal patches with the procedure described in Section 7.2. The detector parameters are set to $\sigma = 2$ and $\tau = 2.5$. Each spatio-temporal patch is described with the concatenated vector of its space-time gradients. Then, the descriptors are projected to a lower dimensional space of 100 dimensions. Examples of the detections for sequences in each category are shown in Figure 9.6 (on p. 90).

In order to build the codebook, we need to cluster the feature descriptors of all training video sequences. However, since the total number of features from all training examples is very large, we use only a subset of sequences to learn the codebook, in order to accommodate the requirements of memory. Thus, we build spatio-temporal codewords using only two videos of each action from three subjects. We keep these sequences out of the training and testing sets, to avoid contamination in the data.

In order to test the efficiency of our approach for the recognition task, we adopt the leave-one-out testing paradigm (LOO). Each video is labeled with the index of the subject performing the action but not with the action class label, so that the algorithm does not have information about the action class contained in the sequences. Thus, for each LOO run, we learn a model from the videos of 24 subjects (except those videos used to build codewords) in an unsupervised fashion, test the videos of the remaining subject, and compute a confusion table for evaluation. The results are reported as the average confusion table of the 25 runs.

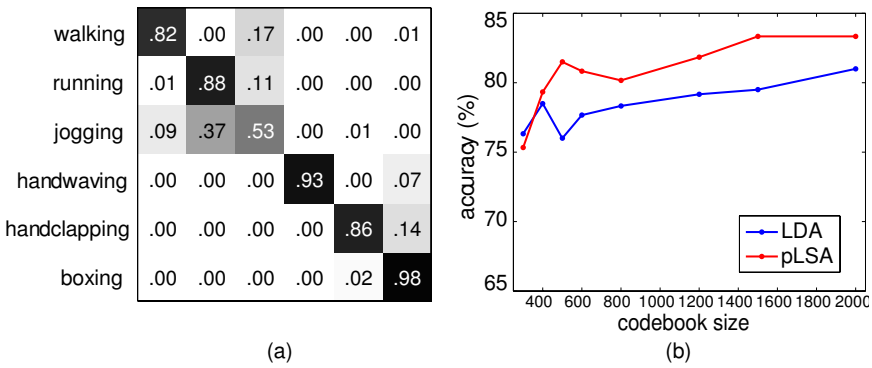


Figure 9.4: Experimental evaluation of action classification with the latent topic models in the KTH dataset. (a) Confusion matrix for pLSA using 1500 codewords (performance average = 83.33%); rows are ground truth, and columns are model predictions; (b) Classification accuracy vs. codebook size for the KTH dataset. Experiments show that the results for the recognition task are consistently better when the pLSA model is adopted. The figure is best viewed in color

Under these settings, we learn and recognize human action categories using the pLSA and LDA models. The confusion matrix for a six-class pLSA model for the KTH dataset is given in Figure 9.4(a) using 1500 codewords. It shows the largest confusion between "jogging" and "running," "walking" and "jogging," and between "hand clapping" and "boxing." This is consistent with our intuition that similar actions are more easily confused with each other, such as those involving hand motions or leg motions. Additionally, at the feature level, we note that the similarity across local patterns from different classes is highest between those categories where our method finds the largest confusion (please refer to Figure 9.5).

We test the effect of the number of video codewords on recognition accuracy on both models, as illustrated in Figure 9.4(b). It shows some dependency of the recognition accuracy on the size of the codebook. Additionally, we can see that pLSA is slightly better than LDA in recognition performance with the same number of codewords. This is an interesting result. Our hypothesis for this outcome is that it is due to large variations and relatively small number of training samples in each action class, which may alleviate the advantages of LDA.

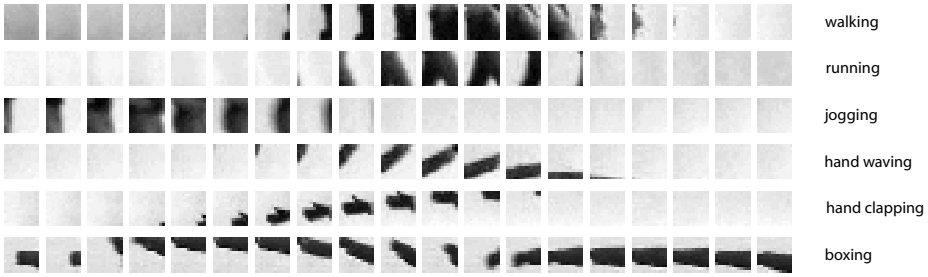


Figure 9.5: The latent topic models provide means to rank the spatio-temporal words given an action class. Here, we illustrate the top word from each category, in the KTH dataset, using a spatio-temporal patch. Each row contains the frames from the neighborhood of a single spatio-temporal interest point, which was assigned to a top word within the category on the right. The spatio-temporal patches clearly characterize each action class; for instance, the top interest point for hand-waving shows its signature of up-down arm motion

We also compare our results with the best results from [23] (performance average = 81.17%), which were obtained using an SVM with the same experimental settings. Our results by unsupervised learning are on par with the current state-of-the-art results obtained by fully supervised training. Furthermore, our generative method provides better interpretability of the learned action models. For example, we can assign individual spatio-temporal features to their most likely action class, as shown in Equation (7.6) (p. 65) and in Figure 9.6 (p. 90). We can also recognize multiple actions that occur simultaneously within the video, see Figure 9.12 (p. 96). Such analysis is not possible in the SVM discriminative approach. Additional comparison of recognition rates from different methods in the KTH dataset is given in Table 9.1.

Table 9.1: Action recognition accuracy with pLSA compared to other methods in the KTH dataset

methods	**recognition accuracy (%)**	**learning**	**multiple actions**
Our method	83.33	unlabeled	Yes
Dollár et al. [23]	81.17	labeled	No
Schuldt et al. [80]	71.72	labeled	No
Ke et al. [47]	62.96	labeled	No

In order to obtain further insight into the model provided by the latent topic approach, we use the distribution of spatio-temporal words given a latent topic. In the

pLSA case these distributions correspond to $p(w|z)$, and in the LDA case the distributions are given in β. These parameters provide means to rank the spatio-temporal words according to their probability of occurrence within each action category. As a first exercise, it is interesting to observe which words are assigned the highest likelihood given an action category. Figure 9.5 shows example spatio-temporal patches that represent the top ranked word within each action category. These spatio-temporal patches clearly correspond to the correct human action class. Second, given a testing sequence, we can assign each of the observed interest points to a corresponding spatio-temporal word. This word in turn, can be assigned to the action class that generate it with highest probability, for example using Equation (7.6) in the pLSA case. We show the result of this procedure in Figure 9.6, using the distributions obtained with the pLSA model. Each interest point has been colored with the corresponding human action category. It is also clear how the model permits the mixture of action classes within a single sequence. Also, note that the dominant color corresponds to the correct action category color.

Finally, we would like to use the models we have learned using the KTH dataset, to detect human actions in sequences from the Caltech human motion dataset [89]. We provide some examples frames from two of these video sequences in Figure 9.7. There, the models learnt with a pLSA approach are used to detect the correct human action class. Most of the action sequences from this dataset can be correctly recognized. To provide further illustration, we have colored each spatio-temporal interest point according to its most likely action category. In the figure, we only draw the space-time features that were assigned to the action class that was detected by our model.

Action recognition and localization using the human action dataset from [6]

In our second experiment, we detect and describe spatio-temporal interest points using the procedure detailed in previous sections. The detector parameters are man-

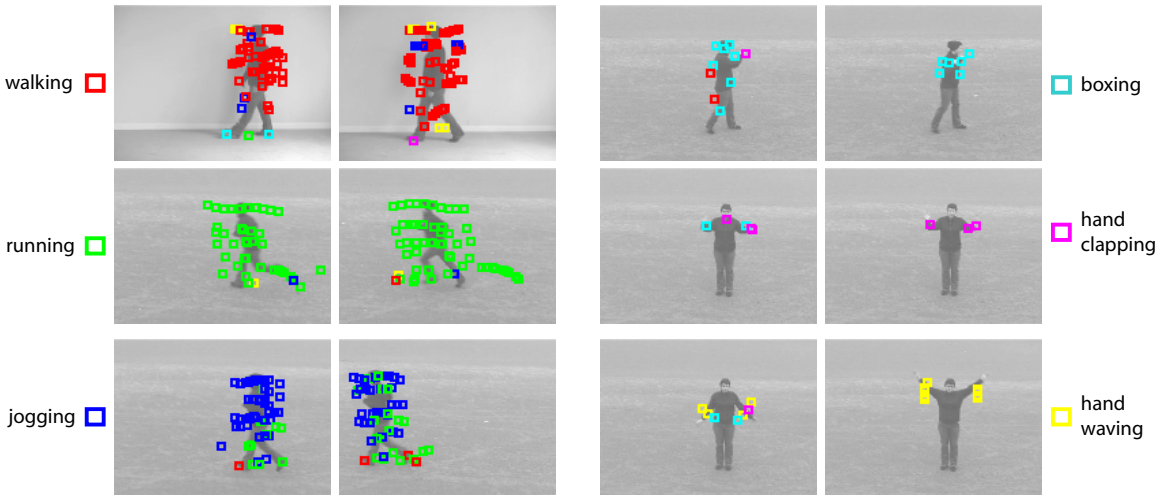


Figure 9.6: Example frames from testing sequences in the KTH dataset. The spatio-temporal patches in each sequence are automatically colored according to action class that most likely generated its corresponding spatio-temporal word. Although some of the words are assigned to the wrong topic, most interest points are assigned to the correct action for each video. Consistently, the predicted action class corresponds to the actual ground truth. In addition, we usually observe that the second best ranked action class corresponds to a similar action: in the "jogging" example of the figure, the second best label is "running." The figure is best viewed in color

ually set to $\sigma = 1.2$ and $\tau = 1.2$, and the dimensionality of the corresponding descriptors is reduced to 100. The codebook is learnt using all the feature descriptors obtained from all the training video sequences.

We again perform leave-one-out cross-validation to test the efficacy of our approach in recognition; i.e., for each run we learn a model from the videos of eight subjects and test those of the remaining subject. The result is reported as the average of nine runs. The confusion matrix for a ten-class model is presented in Figure 9.8(a) for a pLSA model learned using a codebook of size 1200. The average performance of the pLSA model with this codebook size is 90%. Note that the confusion matrix shows how our model is mostly confused by similar action classes, such as "skip" with "jump" and "run," or "run" with "walk."

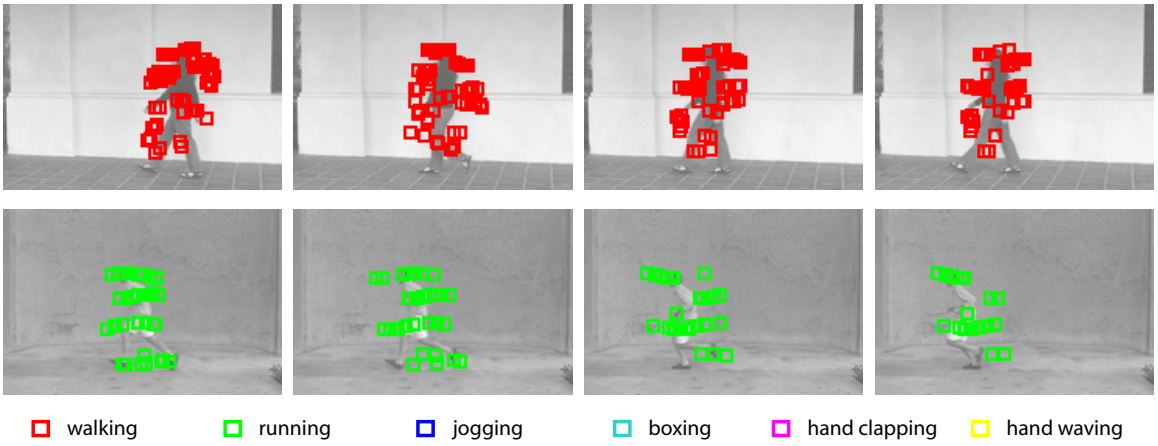


Figure 9.7: Examples frames from sequences in the Caltech dataset. Action category models were learnt using the KTH dataset, and tested again sequences in Caltech dataset. Each interest point is assigned to an action class, and only spatio-temporal interest points from the detected action category are shown. The figure is best viewed in color

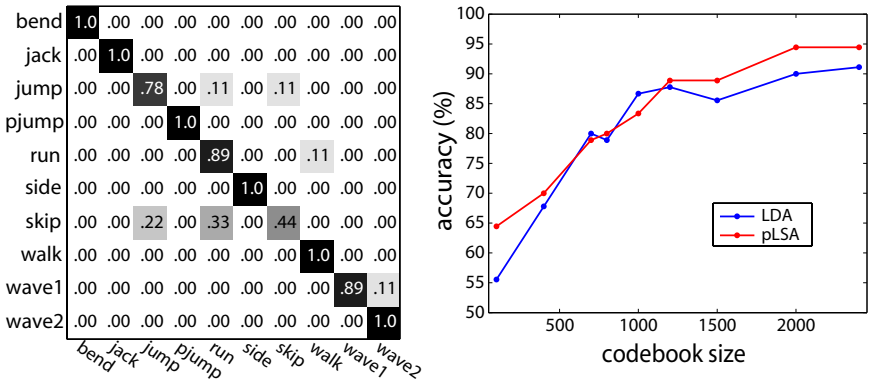


Figure 9.8: Experimental evaluation of action classification with the latent topic models in the Weizmann Institute action dataset. (a) Confusion matrix for the pLSA model; rows are ground truth, and columns are model results. The action models learnt with pLSA and using 1200 codewords show an average performance of 90%. (b) Classification accuracy obtained using pLSA and LDA models vs. codebook size. Our results show that pLSA performs slightly better than LDA in the video categorization task

We test the effect of the number of video codewords on recognition accuracy on the pLSA and LDA models, as illustrated in Figure 9.8(b). It shows some dependency of the recognition accuracy on the size of the codebook.

Similar to the previous experiment, we look for insight on what the latent topic model provides. Figure 9.9 illustrates sample frames from test sequences in each

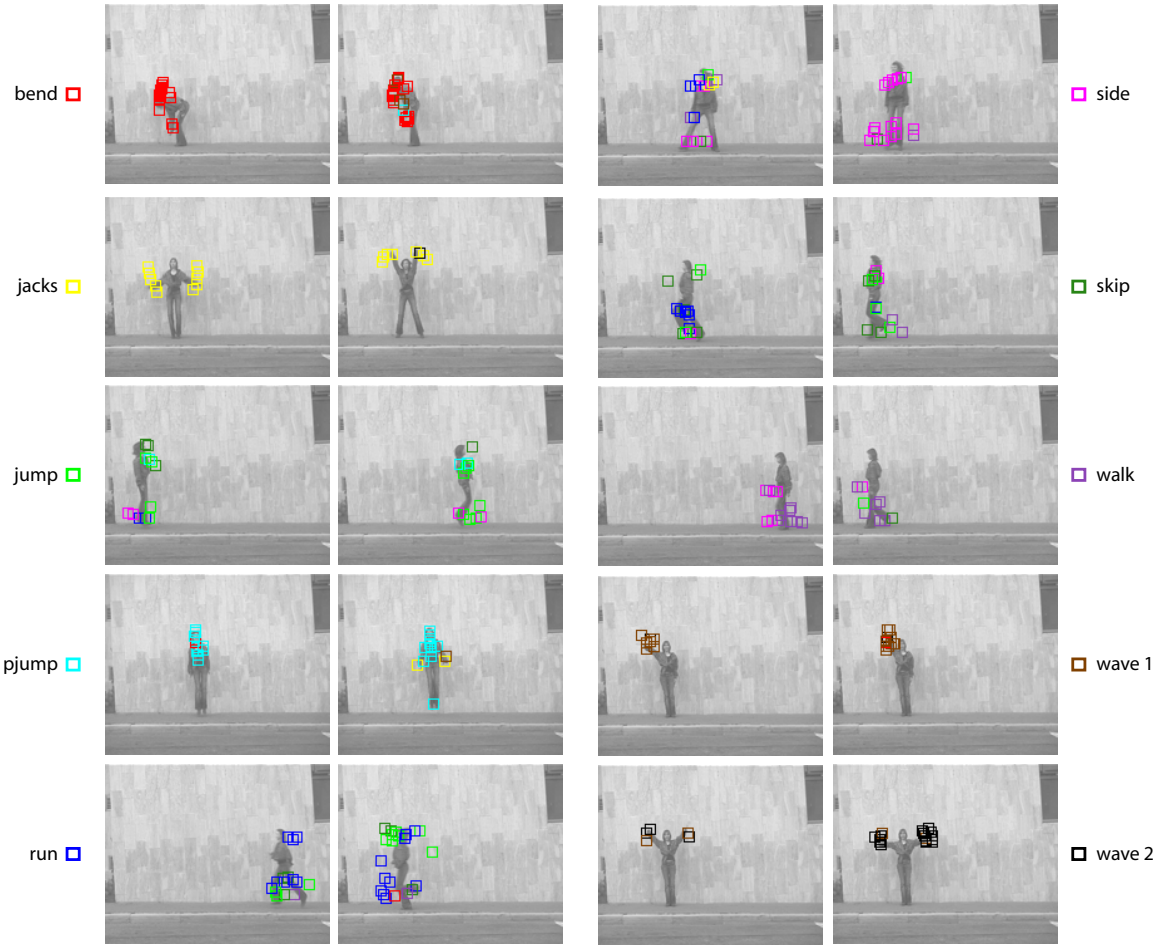


Figure 9.9: Example frames from testing sequences in the Weizmann Institute human action dataset [6]. The spatio-temporal patches in each sequence are automatically colored according to action class that most likely generated its corresponding spatio-temporal word. Although some of the words are assigned to the wrong topic, most interest points are assigned to the correct action for each video. Consistently, the predicted action class corresponds to the actual ground truth. The figure is best viewed in color

action class. We have colored each detected interest-point with its most likely action category. We observe how the model permits the mixture of action classes in each video; however, the actual action category dominates the coloring in all these cases. Additionally, it is also interesting to observe that those interest points that are not colored with the right action, are however assigned to a similar action. For instance, in the frames corresponding to the "jacks" category, there are some interest points assigned to "wave," and it is clear that both actions contain similar arm motion.

Finally we note that in [6], experimental results were reported using 9 of the 10 action categories available in the dataset. Their classification task consisted on determining the action category of a set of space-time cubes, instead of classifying entire video sequences. Also, results on a clustering experiment were presented. These experiments differ from our task, which consists of categorizing complete video sequences. In addition, unlike our video sequence representation using local spatio-temporal words, their approach using space-time shape is sensitive to camera motion and dynamic background.

Recognition and localization of figure skating actions

In our third experiment, we detect and describe interest points using the procedure detailed in previous sections. The detector parameters are manually set to $\sigma = 2$ and $\tau = 1.2$, and the dimensionality of the corresponding descriptors is reduced to 100. We use all the videos available in training to build the codebook, using k-means.

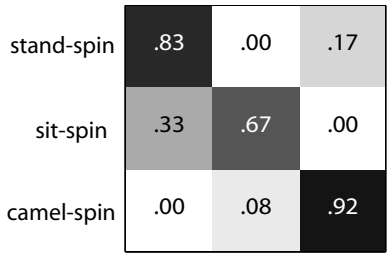


Figure 9.10: Confusion matrix for the figure skating dataset using 1200 codewords (performance average = 80.67%). Our algorithm can successfully categorize the figure skating actions in the presence of camera motion and cluttered background

We use the LOO procedure to test the efficacy of our approach in recognition; i.e., for each run we learn a model from the videos of six subjects and test those of the remaining subject. The result is reported as the average of seven runs. The confusion matrix for a three-class pLSA model for the figure skating dataset is shown in Figure 9.10 using 1200 codewords. The average performance of our algorithm is 80.67%. Note that in spite of the simple representation, our method can perform well in a very challenging dataset with camera motion, scale changes and severe occlusions.

Additionally, the learned three-class pLSA model can be used for action localization as shown in Figure 9.11.

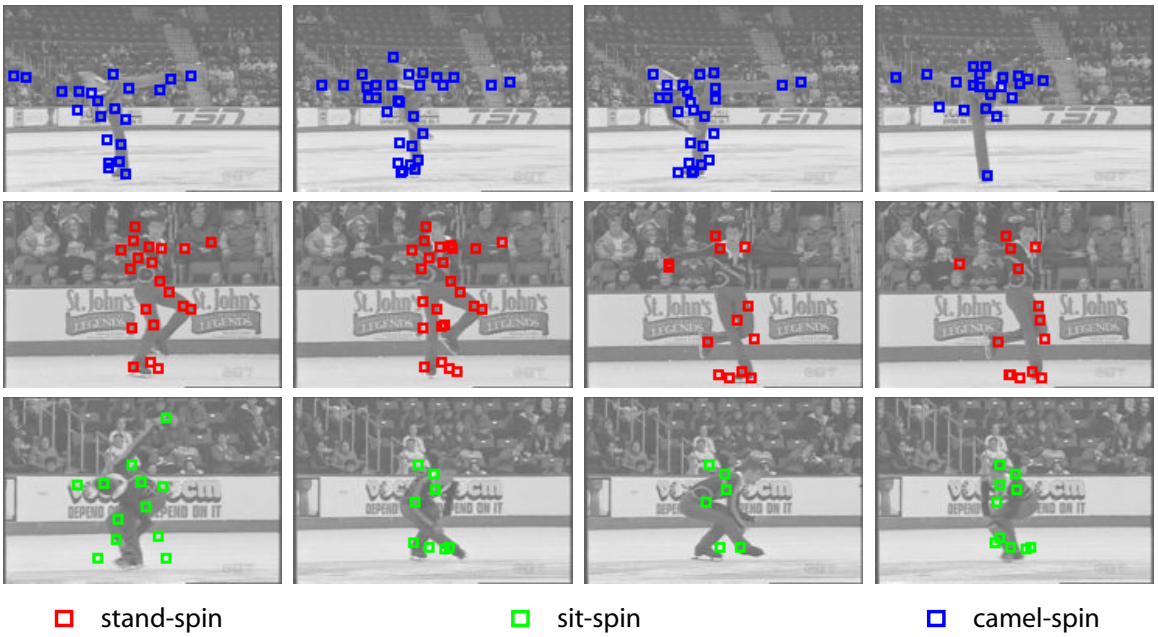


Figure 9.11: Example frames from testing sequences in the figure skating dataset. The interest points in each sequence are automatically colored according to the action class that most likely generated its corresponding spatio-temporal word. Note that only spatio-temporal interest points from the detected action category are shown. The figure is best viewed in color

9.2.2 Recognition and localization of multiple actions in a long video sequence

One of the main goals of our work is to test how well our algorithm could identify multiple actions within a video sequence. For this purpose, we test several long figure skating sequences as well as our own complex video sequences.

When the testing sequence is significantly long, we divide it into subsequences using a sliding temporal window. We process such subsequences independently and obtain classification decisions for each of them. This is necesary due to the nature of our representation: the lack of relative temporal ordering of features in our "bag

of words" representation does not provide means to assign labels at different time instances within a video; instead, the analysis is made for the complete sequence. Thus, by dividing the original long video into subsequences, our method can assign labels to each subsequence within the long sequence.

First, suppose we encounter a testing video that contains multiple simultaneous human action categories. For multiple actions in a single sequence, and assuming we have learnt models employing the pLSA framework, we first identify how many action categories are significantly induced by $P(z_k|w_i, d_j)$. This is possible since $P(z_k|w_i, d_j)$ provides a measurement of the content of each action in the testing sequence. Thus, we allow the algorithm to select more than one action class if $P(z_k|w_i, d_j)$ is bigger than some threshold for more than one k. However, we need to assume that the number of actions present in the sequence is much less than the number of learnt actions categories K; in the extreme case that all action classes are present in the sequence, the distribution $P(z_k|w_i, d_j)$ should be very close to the uniform distribution and we cannot find salient action classes. Once the action categories of interest have been identified, the algorithm can select only the spatio-temporal interest points that are assigned to those classes, and apply k-means to the spatial position of these space-time patches. The number of clusters is set equal to the number of significant action categories. In order to label the resulting clusters with an action class, each word votes for its assigned action within its cluster. Finally a bounding box is plotted according to the principal axis and eigen-values induced by the spatial distribution of video words in each cluster. A further assumption that has to be made in order to use this procedure is that the actions must be performed in spatially distinct positions. Figure 9.12 illustrates examples of multiple actions recognition and localization in one

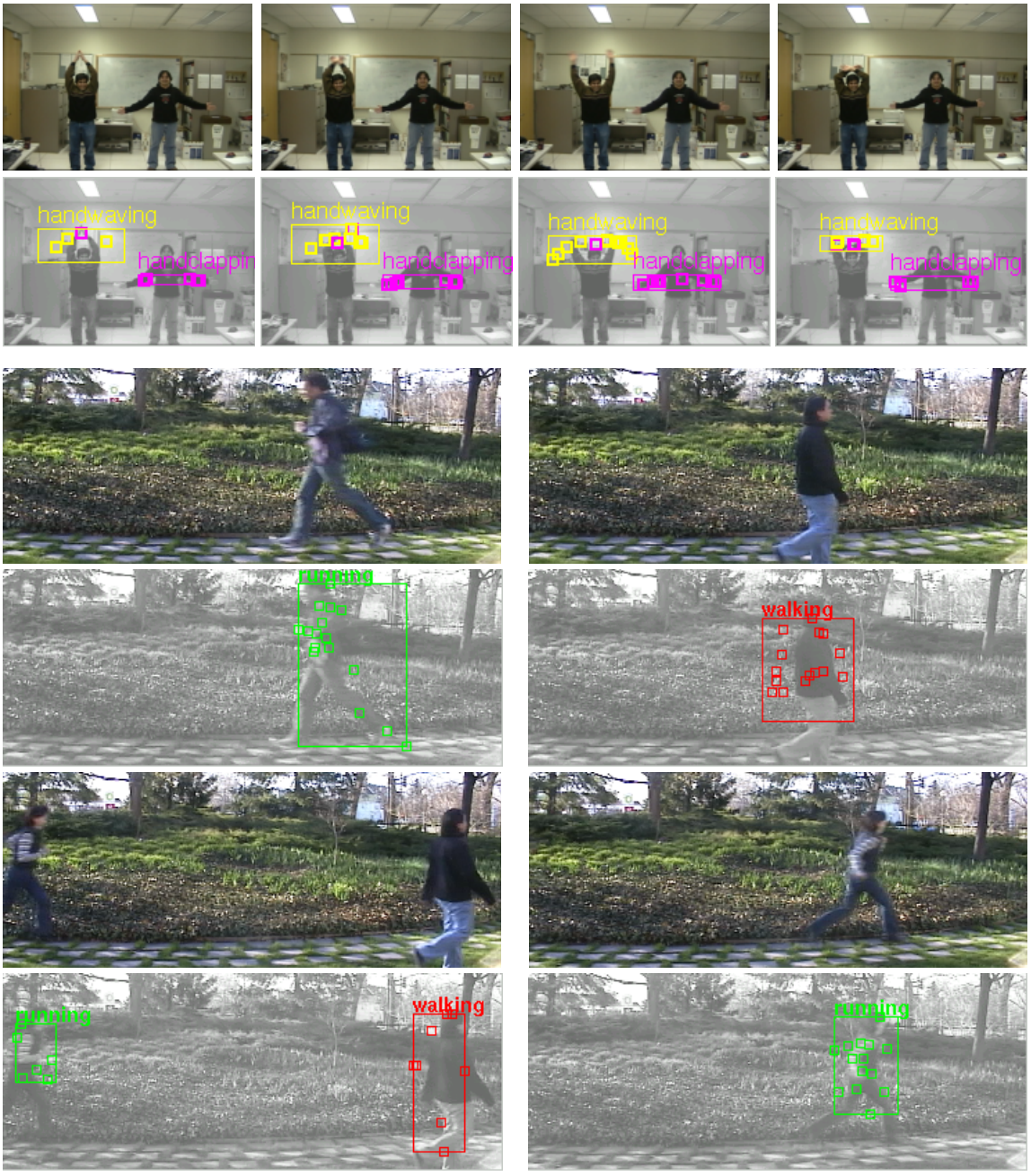


Figure 9.12: Multiple action recognition and localization in long and complex video sequences. The algorithm automatically detects the number of significant actions in a windowed subsequence around each frame. Then a clustering technique is used to group the interest points according to their spatial position. A bounding box is placed around each cluster with the automatically detected action label. The figure is best viewed in color

video sequence using a six-class pLSA model learnt from the KTH dataset (Section 9.2.1).

The second scenario we want to explore consists of a long testing video sequence that contains one subject performing different actions through time. Consider for

example the long skating video sequences in Figure 9.13. Assuming we have learnt models with pLSA, we perform recognition by extracting a windowed sequence around each frame, and identifying which actions receive a high weight according to $P(z_k|w_i, d_j)$. Thus the middle frame in the windowed sequence is labeled with the identified action category. Figure 9.13 shows examples of action recognition in a long figure skating sequence. Here we employed the three-class model learnt from figure skating sequences containing a single action (Section 9.2.1). The three actions (stand-spin, camel-spin and sit-spin), were correctly recognized and labeled using different colors.

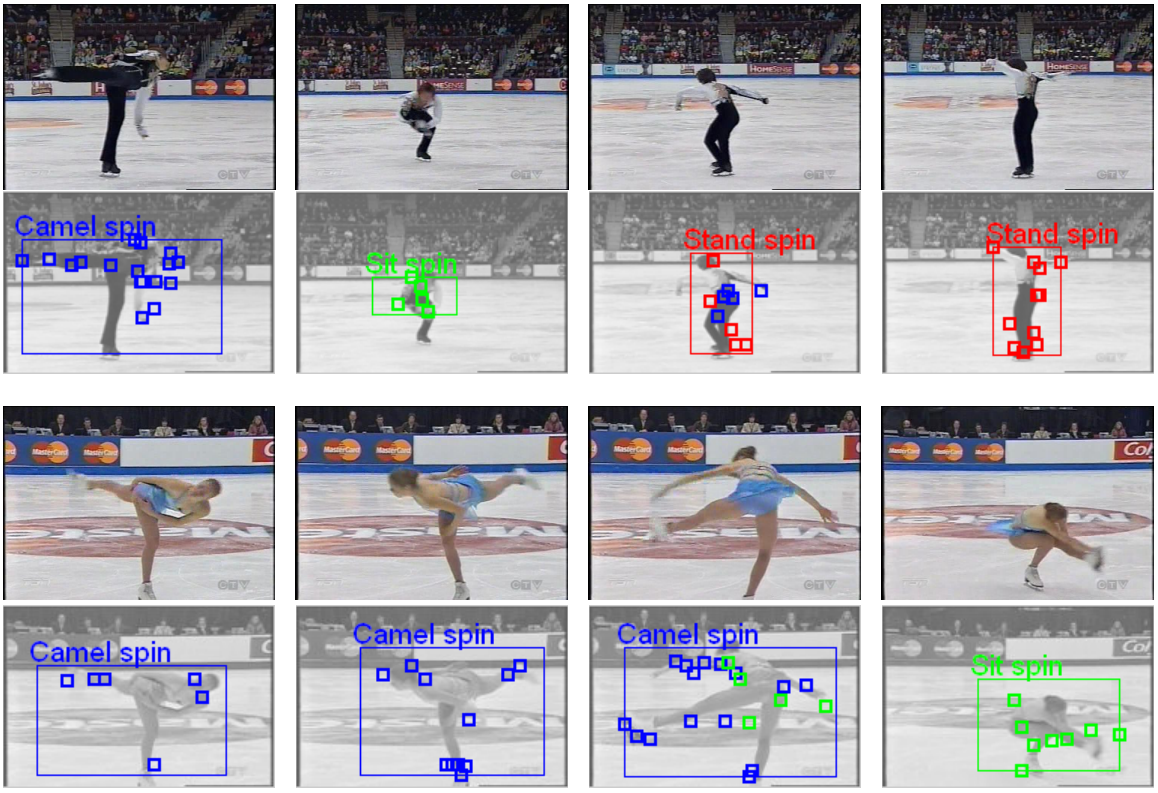


Figure 9.13: Multiple action recognition and localization in long and complex figure skating sequences. The algorithm automatically detects the number of significant actions in a windowed subsequence around each frame. Then a clustering technique is used to group the interest points according to their spatial position. A bounding box is placed around each cluster with the automatically detected action label. The figure is best viewed in color

9.3 Experiments Using the Hierarchical Model

We test our model using the Weizmann Institute human action dataset from [6]. Please note that this is a subset of the dataset presented in Figure 9.2, and it corresponds to the sequences used for the experiments in [6]. The subset contains 9 action classes performed by 9 different subjects, some example frames are shown in figure 9.14. There are 83 sequences in total, since each class contains 9 or 10 videos.

We adopt a LOO scheme for evaluation, by taking videos of one subject as testing data, and randomly splitting the sequences from the remaining subjects into training and validation sets. The training set is always composed by sequences of 5 subjects, while the sequences of the remaining 3 subjects are used for validation.

We train a four-part model with three mixture components for each action class. In order to illustrate the learnt models, Figure 9.15 shows an example frame from a jack sequence with the corresponding action model component over imposed. Parts

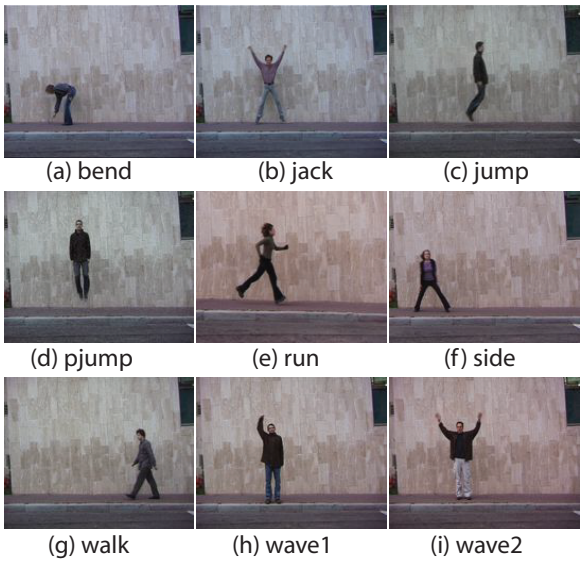


Figure 9.14: Weizmann Institute human actions dataset. Example frames from video sequences in the dataset from [6]. The dataset contains 83 videos from 9 different human action classes. Please note that this is a subset of the dataset presented in Figure 9.2, and it corresponds to the sequences used in [6]

are colored in blue, red, green and cyan, and represented as ellipses which illustrate the Gaussian distribution of the feature relative positions. Static features are represented by crosses and motion features by diamonds. Each feature has been colored with the color of its corresponding parent. Features in yellow and magenta were assigned to the background. Further examples from all classes are shown in Figure 9.16.

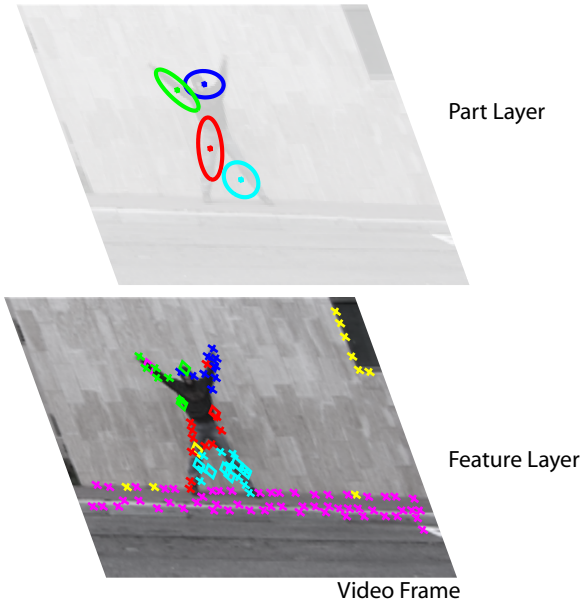


Figure 9.15: Hierarchical human action model overlaid on a testing frame. Parts are represented as ellipses, which illustrate the distribution of the relative position of their children features. Static features are drawn as crosses, while motion features as diamonds. The color of the feature indicates the feature parent. Features in magenta and yellow belong to the background. The figure is best viewed in color

We investigated the performance of our method in frame-by-frame classification, as well as video classification using the voting scheme presented above. The confusion tables are shown in Figures 9.17(a) and 9.17(b). When classifying entire sequences, our system can correctly categorize 72.8% of the testing videos. Note that the confusions are reasonable in the sense that most of the time missclassification occurs between very similar motions, for instance there is confusion between wave1, wave2, and jacks, as well as confusion between run, walk, side, and jump (please refer to Figure 9.14).

Figure 9.16: Learned hierarchical models for human actions. Each row illustrate a different action category: bend, jack, jump, p-jump, run, side, walk, wave1, wave2. Column (a) shows example frames from the original sequence. (b)-(d) show the three mixture components for each action model. Static features are represented by crosses and motion features by diamonds. Each image feature is colored according to its part membership. Ellipses illustrate the variance of the position distributions for each part. The figure is best viewed in color

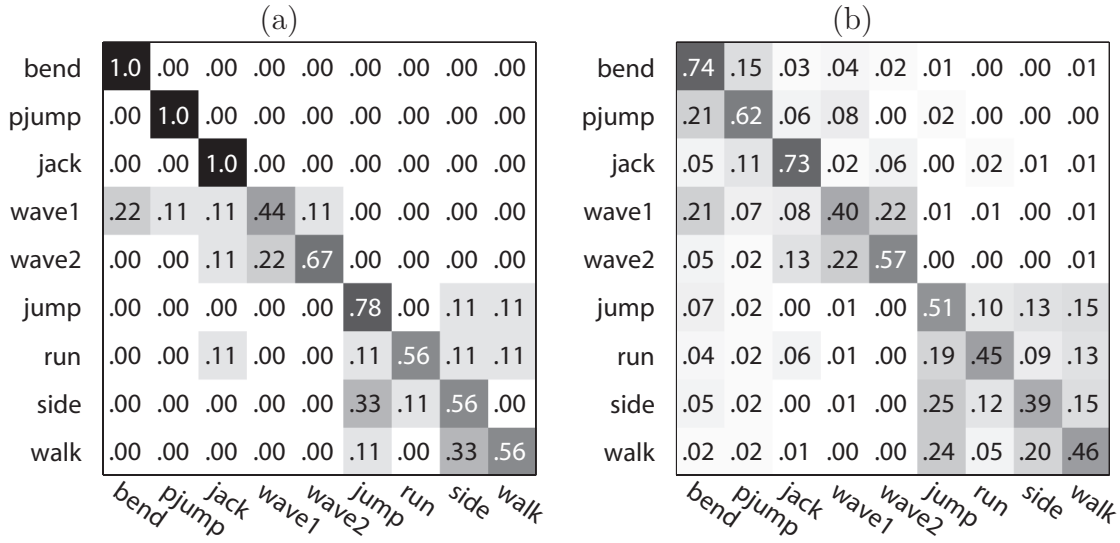


Figure 9.17: Experimental results with our hierarchical model for human actions. (a) Video Classification: Rows are ground truth and columns are predicted labels. The table summarizes the result of 9 runs in a leave-one-out procedure. The system correctly classifies 72.8% of the testing sequences. (b) Frame-by-frame classification: rows are ground truth and columns are predicted labels. The tables are the average over nine runs in a leave-one-out procedure. In average, the algorithm assigns the correct label to 55.0% of the testing frames

In order to evaluate the contribution of the hierarchical model, as well as the use of dynamic and static features, we perform several control experiments. For this purpose, we randomly select one subject and use the corresponding sequences as the testing set. The videos from the remaining subjects are randomly split into training and validation sets.

We evaluate the contribution of the mixture of hierarchical models by comparing it to a one component hierarchical model and a bag of keypoints model. We believe that a class of human action (for example, walking) can be represented by a small number of distinctive (static or dynamic) poses. We have therefore chosen a mixture of models to represent each action. In order to show that this representation is more powerful than a single component, we have trained 1-component models for each action class. Additionally, to demonstrate that including geometric information is useful, we train bag of keypoints models for each action class. For this purpose, each

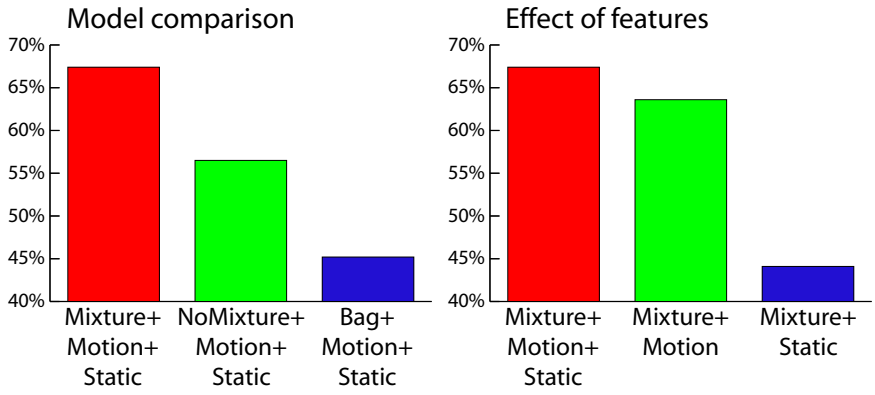


Figure 9.18: Effect of model structure and video features on the action classification accuracy: The plot shows a performance comparison of recognition accuracy under different settings. *Mixture* indicates the use of a three-component mixture model; *NoMixture* indicates the use of a 1-component model; and *Bag* denotes a pure bag of keypoints model. On the *left*, the plot shows that the employment of our new hierarchical model improves the performance over the use of pure bag-of-features model. Also, using a mixture of models helps to account for the multimodality nature of the action models, therefore better recognition is obtained when compared to the 1-component model. On the *right* plot, the results show that using a combination of static and motion features provides the best description of the human actions, which translates into the best recognition accuracy

sequence is represented as a histogram of static and dynamic features. The training examples are kept in a database and new video frames are classified using a nearest neighbor procedure. The bar plot on the left in Figure 9.18 shows the comparison of the performance of each model under the described settings. The outcome of this experiment supports the intuition that human actions contain certain multimodality which can be better represented by a mixture of hierarchical models. The inclusion of the constellation layer and the geometric constraints that it encodes is also useful, since ignoring the geometric arrangement of features and adopting a bag of keypoints model produces poorer classification results.

Finally, we also explore the contribution of each feature type into the classification performance. We trained our mixture of hierarchical models using static features only, dynamic features only and also using both types of features. The bar plot on the right in Figure 9.18 shows the comparison of the performance of the model when

using different types of features. These results empirically support the intuition that a combination of both static and motion features provide the best representation for human actions. Additionally, the experiment shows that if one is to choose a single feature type, motion features are preferable; which is also intuitive in the sense that motion features provides a richer representation of dynamic events than static features.

The first reported classification results on this dataset appeared on [6]. Their method achieved a classification error rate of 0.39%. It is however, difficult to make a fair comparison. Their method requires a background subtraction procedure, global motion compensation, and it cannot take classification decisions frame by frame. Please also note, that our model is general in the sense that it aims to offer a generic framework for human motion and pose categorization.

9.4 Discussion

In this part, we have explored the use of statistical models and local features in the domain of human action categorization.

First, we have presented an unsupervised learning approach, i.e., a "bag of spatio-temporal words" model combined with a space-time interest points detector, for human action categorization and localization. Using three challenging datasets, our experiments show that the classification performance using our unsupervised learning approach is on par with the current state-of-the-art results obtained by fully supervised training. Our algorithm can also localize multiple actions in complex motion sequences containing multiple actions. The results are promising, though we acknowledge the lack of large and challenging video datasets to thoroughly test our algorithm, which poses an interesting topic for future investigation.

Second, we presented a hierarchical model of shape and appearance for human action categorization. The model combines the strong shape representation of the constellation model with the large number of features that utilizes the bag-of-words model. Our constellation-of-bags-of-features model is able to combine static and motion image features in a principled way, as well as perform categorization in a frame-by-frame basis. Future directions include adopting robust features that help to account for more general camera motion and unconstrained environments. We believe this model has the potential to be able to characterize more complex motions and configurations of the highly articulated human body.

Other interesting future endeavours involve the possibilities of using a unified framework by combining generative and discriminative models for human action recognition. For similar actions (e.g., "running" and "walking"), the classification may benefit from a discriminative model. It would also be interesting to explore other models that can incorporate more detailed geometric information, for example, by using explicit models for the human body.

Part IV

Recognizing Complex Actions by Modeling Temporal Structure of Simple Motion Segments

Chapter 10

Introduction

Much recent research in human activity recognition has focused on the problem of recognizing simple repetitive (walking, running, waving) and punctual actions (sitting up, opening a door, hugging). However, many interesting human activities are characterized by a complex temporal composition of simple actions. Automatic recognition of such complex actions can benefit from a good understanding of temporal structures as a contextual cue. We propose a framework for modeling motion by exploiting the temporal structure of human activities. In our framework, we represent activities as temporal compositions of motion segments. We train a discriminative model that encodes a temporal decomposition of video sequences, and appearance models for each motion segment. In recognition, a query video is matched to the model according to the learned appearances and motion segment decomposition. Classification is made based on the quality of matching between the motion segment classifiers and the temporal segments in the query sequence. To validate our approach, we introduce a new dataset of complex Olympic Sports activities. We show that our algorithm performs better than other state of the art methods.

We argue that to understand motion, it is critical to incorporate temporal context information, particularly the temporal ordering of the movements. We propose

a simple discriminative framework for classifying human activities by aggregating information from motion segments that are considered both for their visual features as well as their temporal composition. An input video is automatically decomposed temporally into motion segments of variable lengths. The classifier selects a discriminative decomposition and combination of the segments for matching. Though simple in its form, we highlight a couple of advantages of our framework compared to the previous work.

First, depending on the time scale of the movement, actions have been traditionally grouped into: short but punctual actions (e.g. drink, hug), simple but periodic actions (e.g. walking, boxing), and more complex activities that are considered as a composition of shorter or simpler actions (e.g. a long jump, cooking). Very different algorithms have been proposed for these different types of motion, most of them take advantage of the special properties within its domain, hence perform rather poorly on other types. Our framework is a general one. No matter how simple or complex the motion is, our classifier relies on a temporal composition of various motion segments. Our basic philosophy is clear: temporal information helps action recognition at all time scales.

On the other hand, we note that some work has taken the approach of decomposing actions into "hidden states" that correspond to meaningful motion segments (i.e. HMM's, HCRF's, etc.). In contrast, we let the model automatically discover a robust combination of motion segments that improve the discriminative power of the classifier. The result is a much simpler model that does not unnecessarily suffer from the difficult intermediate recognition step.

In order to test the efficacy of our method, we introduce a new dataset that focuses on complex motions in Olympic Sports, which can be difficult to discriminate without modeling the temporal structures. Our algorithm shows very promising results.

The rest of Part IV is organized as follows. Section 10.1 overviews some of the related work. Chapter 11 presents our framework by first introducing the video representations that can be employed in conjunction with our model (Section 11.1) and then describing the details of the model (Section 11.2). We present experimental validation in Chapter 12.

10.1 Related Work

A considerable amount of work has studied the recognition of human actions in video. Here we overview related work but refer the reader to [91, 35] for a more complete survey.

A number of approaches have adopted the bag of spatio-temporal interest points [52] representation for human action recognition. This representation can be combined with either discriminative [63, 56] classifiers, semi-latent topic models [100] or unsupervised generative [69, 103] models. Such holistic representation of video sequences ignores temporal ordering and arrangement of features in the sequence.

Some researchers have studied the use of temporal structures for recognizing human activities. Methods based on dynamical Bayesian networks and Markov models have shown promise but either require manual design by experts [57] or detailed training data that can be expensive to collect [45]. Other work has aimed at constructing plausible temporal structures [38] in the actions of different agents but does not consider the temporal composition within the movements of a single subject, in part due to their holistic representation. On the other hand, discriminative models of temporal context have also being applied for classification of simple motions in rather simplified environments [87, 98, 72, 77].

In addition to temporal structures, other contextual information can benefit activity recognition, such as background scene context [63] and object interactions [38,

105]. Our proposed framework focuses on incorporating temporal context, but does not exclude future work for combining more contextual information.

Our approach to capturing temporal structures is related to part-based models for object recognition. Both generative [11, 29, 33, 65] and discriminative [28, 48] models have shown promise in leveraging the spatial structures among parts for object recognition.

In the following chapters, we present a new representation for human activities in video. The key observation is that many activities can be described as a temporal composition of simple motion segments. At the global temporal level, we model the distinctive overall statistics of the activity. At shorter temporal ranges, we model the patterns in motion segments of shorter duration that are arranged temporally to compose the overall activity. Moreover, such temporal arrangement considered by our model is not rigid, instead it accounts for the uncertainty in the exact temporal location of each motion segment.

A preliminary version of this work appeared in ECCV 2010 [64].

Chapter 11

A Discriminative Model of Temporal Structure of Simple Motion Segments for Complex Action Classification

11.1 Video Representation

Our model of human actions can be applied over a variety of video descriptors. The key requirement is that a descriptor can be computed over multiple temporal scales. The requirement arises because our motion segment classifiers can operate on video segments of varying length. Frame-based representations and representations based on histograms are particular examples of descriptors that fit well to our framework. Here, we adopt a representation based on histograms of spatio-temporal interest points.

Interest point based descriptors are attractive specially when tracking the subject performing the activity is difficult or not available. Several methods have been

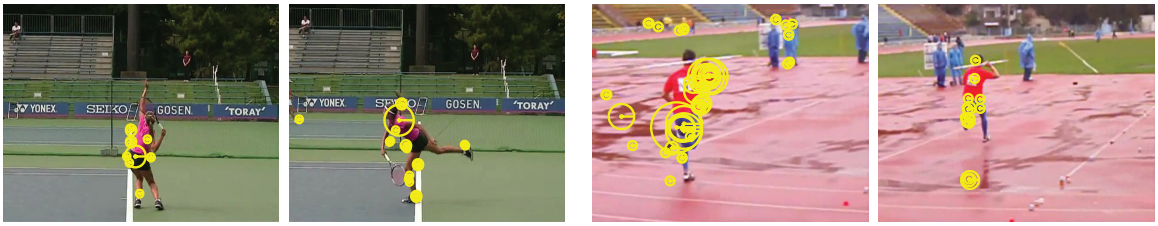


Figure 11.1: Video representation in our discriminative model for complex actions. Our framework can be applied over a variety of video data representations. Here we adopt a representation based on spatio-temporal interest points. This figure shows example spatio-temporal interest points detected with the 3D Harris corner method from [52]. Video patches are extracted around each point, and described by their local shape and motion patterns

proposed for detecting spatio-temporal interest points in sequences [52, 23]. In our approach, we use the 3-D Harris corner detector [52]. Each interest point is described by HoG (Histogram of Gradients) and HoF (Histogram of Flow) descriptors [56]. Furthermore, we vector quantize the descriptors by computing memberships with respect to a descriptor codebook, which is obtained by k-means clustering of the descriptors in the training set. During model learning and matching, we compute histograms of codebook memberships over particular temporal ranges of a given video, which are denoted by ψ_i in the following.

11.2 Our Discriminative Model

In this section we present our framework for recognizing complex human activities in video. We propose a temporal model for recognizing human actions that incorporates simple motion segment classifiers of multiple temporal scales. Figure 11.2 shows a schematic illustration of our human action model. The basic philosophy is very simple: a video sequence is first decomposed into many temporal segments of variable length (including the degenerate case of the full sequence itself). Each video segment is matched against one of the motion segment classifiers by measuring image-based similarities as well as the temporal location of the segment with respect to the full

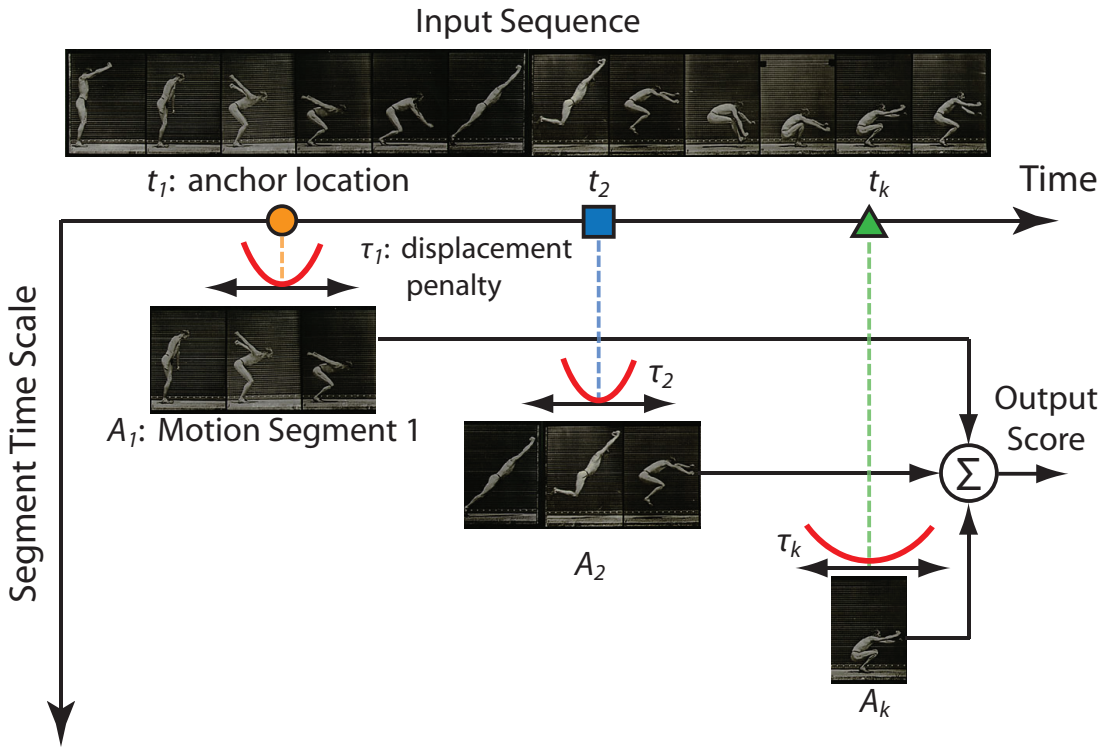


Figure 11.2: Structure of our discriminative model for complex action recognition. The input video V is described by histograms of vector quantized interest points, which are computed over multiple temporal ranges. Each motion segment classifier A_i has a particular temporal scale, and it is matched to the features $\psi_i(V, h_i)$ from temporal segments of the input sequence of that temporal extent. The optimal location of each motion segment classifier is determined by the appearance similarity $(A_i \cdot \psi_i(V, h_i))$ and penalty of temporal displacement from the anchor point t_i $(\tau_i \cdot \psi_(h_i - t_i))$. The overall matching score combines scores of individual components. A classification decision is made by thresholding the resulting matching score. See Section 11.2 for more details

sequence. The best matching scores from each motion segment classifier are accumulated to obtain a measure of the matching quality between the full action model and the query video. As Figure 11.2 illustrates, an action model encodes motion information at multiple temporal scales. It also encodes the ordering in which the motion segments typically appear in the sequence. In the following, we discuss the details of the model, the recognition process and learning algorithm.

11.2.1 Model description

Here we introduce the model of human actions, which is illustrated in Figure 11.2. Our full action model is composed by a set of K motion segment classifiers $A_1, ..., A_K$, each of them operating at a particular temporal scale. Each motion segment classifier A_i operates over a histogram of quantized interest points extracted from a temporal segment whose length is defined by the temporal scale of the classifier s_i. In addition to the temporal scale, each motion segment classifier also specifies a temporal location centered at its preferred anchor point t_i. Lastly, the motion segment classifier is enriched with a flexible displacement model τ_i that captures the variability in the exact placement of the motion segment A_i within the sequence.

We summarize the parameters of our model with the parameter vector $\mathbf{w}$ as the concatenation of the motion segment classifiers and the temporal displacement parameters,

$$\mathbf{w} = (A_1, ..., A_K, \tau_1, ..., \tau_K). \tag{11.1}$$

11.2.2 Model properties

Our model addresses the need to consider temporal structure in the task of human activity classification. In the following, we discuss some important properties of our framework.

Coarse-to-fine motion segment classifiers Our model contains multiple classifiers at different time scales, enabling it to capture characteristic motions of various temporal granularity. On one end, holistic bag-of-features operate at the coarsest scale, while frame-based methods operate at the finest scale. Our framework has the flexibility to operate between these two ends of the temporal spectrum, and it closes the gap by allowing multiple classifiers to reside in a continuum of temporal scales.

Temporal Context While discriminative appearance is captured by our multiple classifiers at different time scales, the location and order in which the motion segments occur in the overall activity also offer rich information about the activity itself. Our framework is able to capture such temporal context: the anchor points of the motion segment classifiers encode the temporal structure of the activity. In particular, these canonical positions prohibit the classifiers from matching time segments that are distant from them. This implicitly carries ordering constraints that are useful for discriminating human activities.

Flexible Model Equipped with classifiers of multiple time scales and the temporal structure embedded in their anchor points, our model is capable of searching for a best match in a query sequence and score it accordingly. However, the temporal structure in videos of the same class might not be perfectly aligned. To handle intra-class variance, our model incorporates a temporal displacement penalty that allows the optimal placement of the each motion segment to deviate from its anchor point.

11.3 Recognition

Given a trained model, the task in recognition is to find the best matching of the model to an input sequence. This requires finding the best scoring placement for each of the K motion segment classifiers. We denote a particular placement of the motion segment classifiers within a sequence V by a hypothesis $H = (h_1, ..., h_k)$. Each h_i defines the temporal position for the i-th motion segment classifier. We measure the matching quality of motion segment classifier A_i at location h_i by favoring good appearance similarity between the motion segment classifier and the video features, and penalizing for the temporal misplacement of the motion segment classifier when h_i is far from the anchor point t_i. That is, the matching score for the i-th motion

segment classifier is

$$A_i \cdot \psi_i(V, h_i) - \tau_i \cdot \psi_{di}(h_i - t_i). \tag{11.2}$$

In the first term of Equation (11.2), which captures the appearance similarity, $\psi_i(V, h_i)$ is the appearance feature vector (i.e. histogram of quantized interest points) extracted at location h_i with scale s_i. In our experiments, we implement the classifier A_i with a χ^2 support vector machine. The kernel function for A_i is given by

$$K(x_k, x_j) = \exp\left(-\frac{1}{2S}\sum_{r=1}^{D}\frac{(x_{kr} - x_{jr})^2}{x_{kr} + x_{jr}}\right), \tag{11.3}$$

where S denotes the mean distance among training examples, $\{x_{ki}\}_{i=1\ldots D}$ are the elements of the histogram x_k and D is the histogram dimensionality. In practice, D is equal to the size of the codebook. In the second term of Equation (11.2), which captures the temporal misplacement penalty, $\psi_{di}(h_i - t_i)$ denotes the displacement feature. The penalty, parametrized by $\tau_i = \{\alpha_i, \beta_i\}$, is a quadratic function of the motion segment displacement and given by

$$\tau_i \cdot \psi_{di}(h_i - t_i) = \alpha_i \cdot (h_i - t_i)^2 + \beta_i \cdot (h_i - t_i). \tag{11.4}$$

We obtain an overall matching score for hypothesis H by accumulating the scores from all motion segment classifiers in the model:

$$\sum_{i=1}^{K} A_i \cdot \psi_i(V, h_i) - \tau_i \cdot \psi_{di}(h_i - t_i). \tag{11.5}$$

Let $f_{\mathbf{w}}(V)$ be a scoring function that evaluates sequence V. In recognition, we consider all possible hypotheses and choose the one with the best matching score:

$$f_{\mathbf{w}}(V) = \max_{H} \sum_{i=1}^{K} A_i \cdot \psi_i(V, h_i) - \tau_i \cdot \psi_{di}(h_i - t_i). \tag{11.6}$$

A binary classification decision for input video V is done by thresholding the matching score $f_{\mathbf{w}}(V)$.

There is a large number of hypotheses for a given input video sequence. However, note that once the appearance similarities between the video sequence and each motion segment classifier are computed, selecting the hypothesis with the best matching score can be done efficiently using dynamic programming and distance transform techniques [29] in a similar fashion to [28, 30].

11.4 Learning

Suppose we are given a set of example sequences $\{V^1, \ldots, V^N\}$ and their corresponding class labels $y_{1:N}$, with $y_i \in \{1, -1\}$. Our goal is to use the training examples to learn the model parameters $\mathbf{w}$. This can be formulated as the minimization of a discriminative cost function. In particular, we consider the following minimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f_{\mathbf{w}}(V^i)), \tag{11.7}$$

where C controls the relative weight of the hinge loss term. This is the formulation of a Latent Support Vector Machine (LSVM) [28]. In the LSVM framework, the scoring function maximizes over the hidden variables. In our method, the hidden variables correspond to the best locations of the motion segment classifiers on each training video. Note that it is not necessary to supervise the locations of the motion segment

classifiers during training, instead this is a weakly supervised setting, where only a class label is provided for each example.

The optimization problem described above is, in general, non-convex. However, it has been shown in [28] that the objective function is convex for the negative examples, and also convex for the positive examples when the hidden variables are fixed.

This leads to an iterative learning algorithm that alternates between estimating model parameters and estimating the hidden variables for the positive training examples. In summary, the procedure is as follows. In the first step, the model parameters $\mathbf{w}$ are fixed. The best scoring locations $H_p^\star$ of the motion segment classifiers are selected for each positive example p. This is achieved by running the matching process described in Section 11.3 on the positive videos. In the second step, by fixing the hidden variables of the positive examples to the locations given by $H_p^\star$, the optimization problem in Equation (11.7) becomes convex. We select negative examples by running the matching process in all negative training videos and retrieving all hypotheses with large matching score. We train the parameters $\mathbf{w}$ using LIBSVM [13] on the resulting positive and negative examples. This process is repeated for a fixed small number of iterations.

In most cases, the iterative algorithm described above requires careful initialization. We choose a simple initialization heuristic. First, we train a classifier with a single motion segment classifier that covers the entire sequence. This is equivalent to training a χ^2-SVM on a holistic bag of features representation. We then augment the model with the remaining $K-1$ motion segment classifiers. The location and scale of each additional motion segment classifier is selected so that it covers a temporal range that correlates well with the global motion segment classifier. This favors temporal segments that exhibit features important for overall discrimination.

Chapter 12

Experimental results

In order to test our framework, we consider three experimental scenarios. First, we test the ability of our approach to discriminate simple actions on a benchmark dataset. Second, we test the effectiveness of our model at leveraging the temporal structure in human actions on a set of synthesized complex actions. Last, we present a new challenging Olympic Sports Dataset and show promising classification results with our method.

12.1 Simple Actions

We use the KTH Human actions dataset [80] to test the ability of our method to classify simple motions. The dataset contains 6 actions performed by 25 actors, for a total of 2396 sequences. We follow the experimental settings described in [80]. In all experiments, we adopt a representation based on spatio-temporal interest points described by concatenated HoG/HoF descriptors. We construct a codebook of local spatio-temporal patches from feature descriptors in the training set. We set the number of codewords to be 1000. Experimental results are shown in Table 12.1. A direct comparison is possible to the methods that follow the same experimental

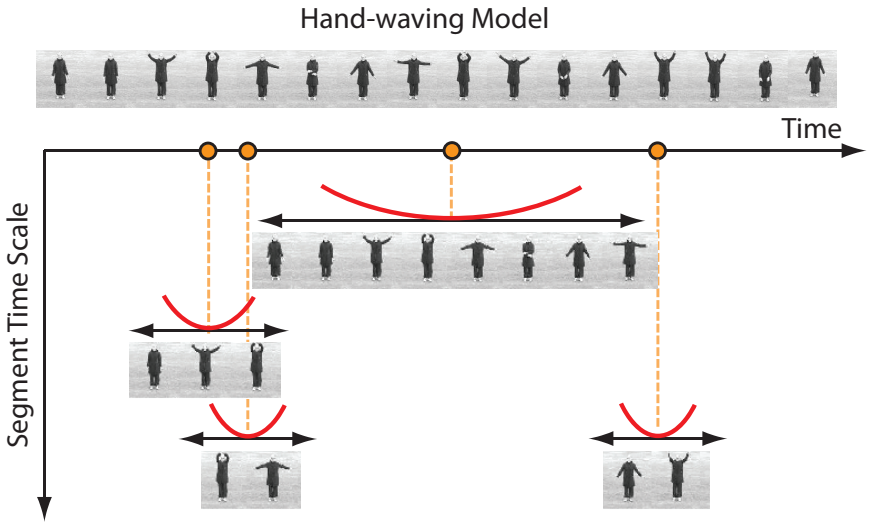


Figure 12.1: Example of our learned discriminative model. In this illustration, the horizontal axis represents time. Each row corresponds to a motion segment classifier learned by our model whose temporal extent is indicated by its vertical location. The appearance of the motion segment is illustrated by a few example frames. The associated dot indicates the anchor position t_i of the motion segment relative to the full sequence. The parameters of the temporal misplacement penalty τ_i are represented by the parabola centered at the anchor point. Notice that the vertical arrangement of the motion segments shows the distinct temporal scales at which each classifier operates

Table 12.1: Left: Action classification accuracy in the KTH dataset. Right: Comparison of our model to current state of the art methods

Action Class	Our Model
walking	94.4%
running	79.5%
jogging	78.2%
hand-waving	99.9%
hand-clapping	96.5%
boxing	99.2%

Algorithm	Perf.
Ours	91.3%
Wang et al. [96]	92.1%
Laptev et al. [56]	91.8%
Wong et al. [103]	86.7%
Schuldt et al. [80]	71.5%
Kim et al. [49]	95%

setup [56, 103, 80, 96]. We note that our method shows competitive results, but its classification accuracy is slightly lower than the best result reported in [96].

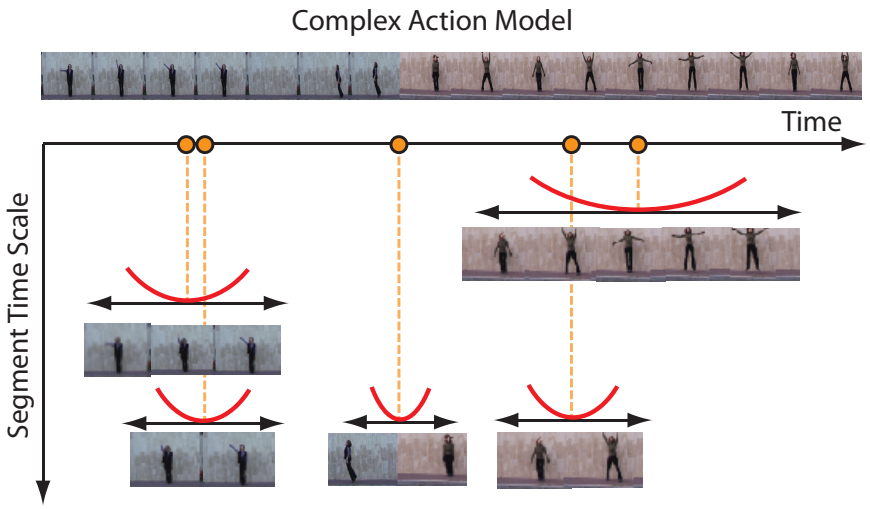


Figure 12.2: A learned model for the synthesized complex action 'wave'-'jump'-'jack'. See Figure 12.1 for a description of the illustration

12.2 Synthesized Complex Actions

In this experiment, we aim to test the ability of our model to leverage the temporal structure of human actions. In order to test this property in a controlled setting, we construct a synthesized set of complex actions by concatenating 3 simple motions from the Weizmann action database: 'jump', 'wave' and 'jack'. In total, we synthesize 6 complex actions classes by concatenating one video of each simple motion into a long sequence.

In this test, a baseline model that uses a single motion segment classifier covering the entire video sequence performs at random chance or $\approx 17\%$. The simple holistic bag-of-features has trouble differentiating actions in this set since the overall statistics are nearly identical. On the other hand, our model which takes advantage of temporal structure and orderings, can easily discriminate the 6 classes and achieve perfect classification performance at 100%. In Figure 12.2 we show a learned model for the complex action composed by 'wave'-'jump'-'jack'. Notice that our model nicely captures discriminative motion segments such as the transitions between 'jump' and 'jack'.

Table 12.2: Average Precision (AP) values for the complex action classification task in our Olympic Sports Dataset

Sport class	Our Method	Laptev et al. [56]
high-jump	**68.9%**	52.4%
long-jump	**74.8%**	66.8%
triple-jump	**52.3%**	36.1%
pole-vault	**82.0%**	47.8%
gymnastics-vault	86.1%	**88.6%**
shot-put	**62.1%**	56.2%
snatch	**69.2%**	41.8%
clean-jerk	**84.1%**	83.2%
javelin-throw	**74.6%**	61.1%
hammer-throw	**77.5%**	65.1%
discus-throw	**58.5%**	37.4%
diving-platform	87.2%	**91.5%**
diving-springboard	77.2%	**80.7%**
basketball-layup	**77.9%**	75.8%
bowling	**72.7%**	66.7%
tennis-serve	**49.1%**	39.6%
Average (AAP)	**72.1%**	62.0%

12.3 Complex Actions: Olympic Sports Dataset

We have collected a dataset of Olympic Sports activities from *YouTube* sequences. Our dataset contains 16 sport classes, with 50 sequences per class. See Figure 12.3 for example frames from the dataset. The sport activities depicted in the dataset contain complex motions that go beyond simple punctual or repetitive actions [1]. For instance, sequences from the long-jump action class, show an athlete first standing still, in preparation for his/her jump, followed by running, jumping, landing and finally standing up.

We split the videos from each class in the dataset into 40 sequences for training and 10 for testing. We illustrate two of the learned models in Figure 12.4. Table 12.2 shows the classification results of our algorithm. We compare the performance of our

[1] In contrast to other sport datasets such as [77], which contains simple periodic or punctual actions such as walking, running, golf-swing, ball-kicking.

Figure 12.3: Olympic Sports Dataset. Our dataset contains 50 videos from each of 16 classes: high jump, long jump, triple jump, pole vault, discus throw, hammer throw, javelin throw, shot put, basketball layup, bowling, tennis serve, platform (diving), springboard (diving), snatch (weightlifting), clean and jerk (weightlifting) and vault (gymnastics). The sequences, obtained from *YouTube*, contain severe occlusions, camera movements, compression artifacts, etc.

model to the multi-channel method of [56], which incorporates rigid spatio-temporal binnings and captures a rough temporal ordering of features.

Finally, Figure 12.5 shows three learned models of actions in the Olympic Sports dataset, along with matchings to some testing sequences. In the long jump example, the first motion segment classifier covers the running motion at the beginning of the sequence. This motion segment has a low displacement penalty over a large temporal range as indicated by its wide parabola. It suggests that the model has learned to tolerate large displacements in the running stage of this activity. On the other hand, in the vault example, the middle motion segment classifier has a low matching score

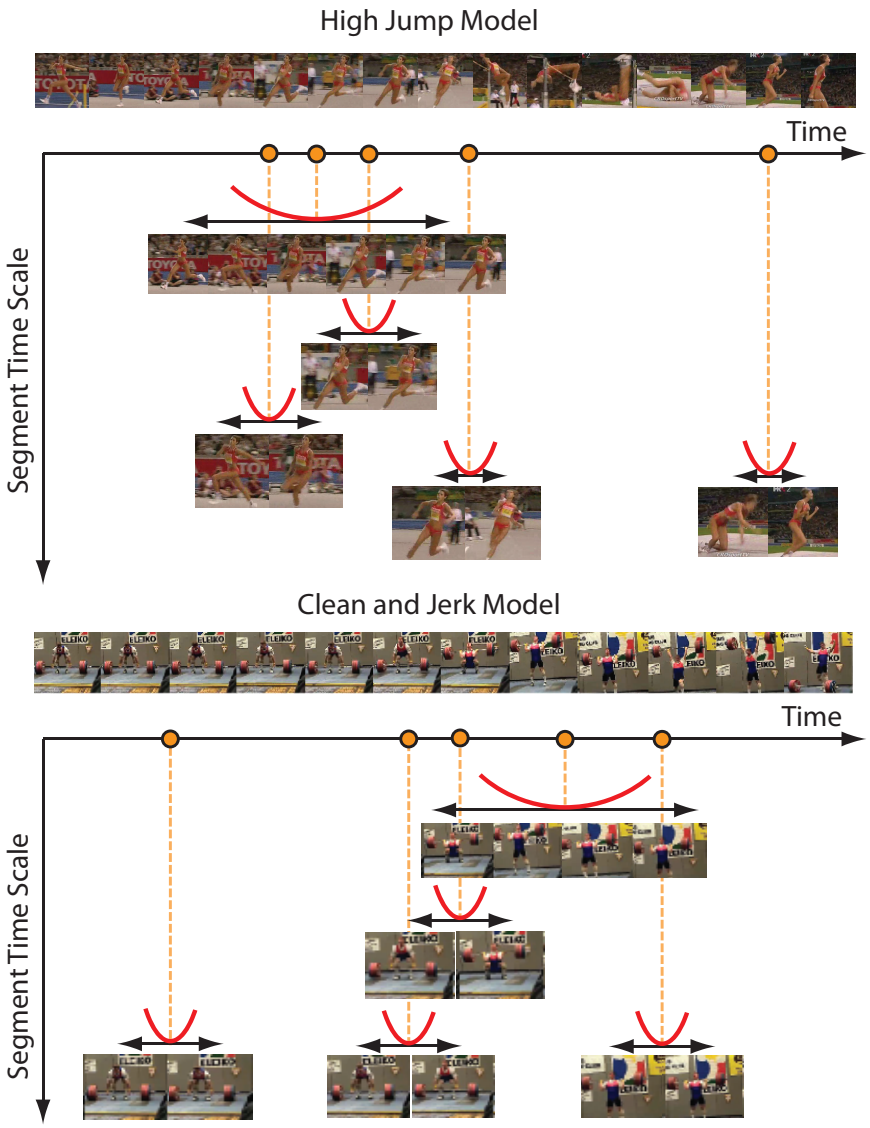


Figure 12.4: Learned model for two complex actions in the Olympic Sports Dataset: high-jump and clean-and-jerk. See Figure 12.1 for a description of the illustration

to the top testing sequence. However, the matching scores in other temporal segments are high, which provides enough evidence to the full action model for classifying this sequence correctly. Similarly, the bottom clean and jerk sequence in the snatch model obtains a high matching score for the last motion segment, but the evidence from the motion segments is rather low. We also observe that our learned motion segment classifiers display a wide range of temporal scales, indicating that our model is able to capture characteristic motion patterns at multiple scales. For example, the longer

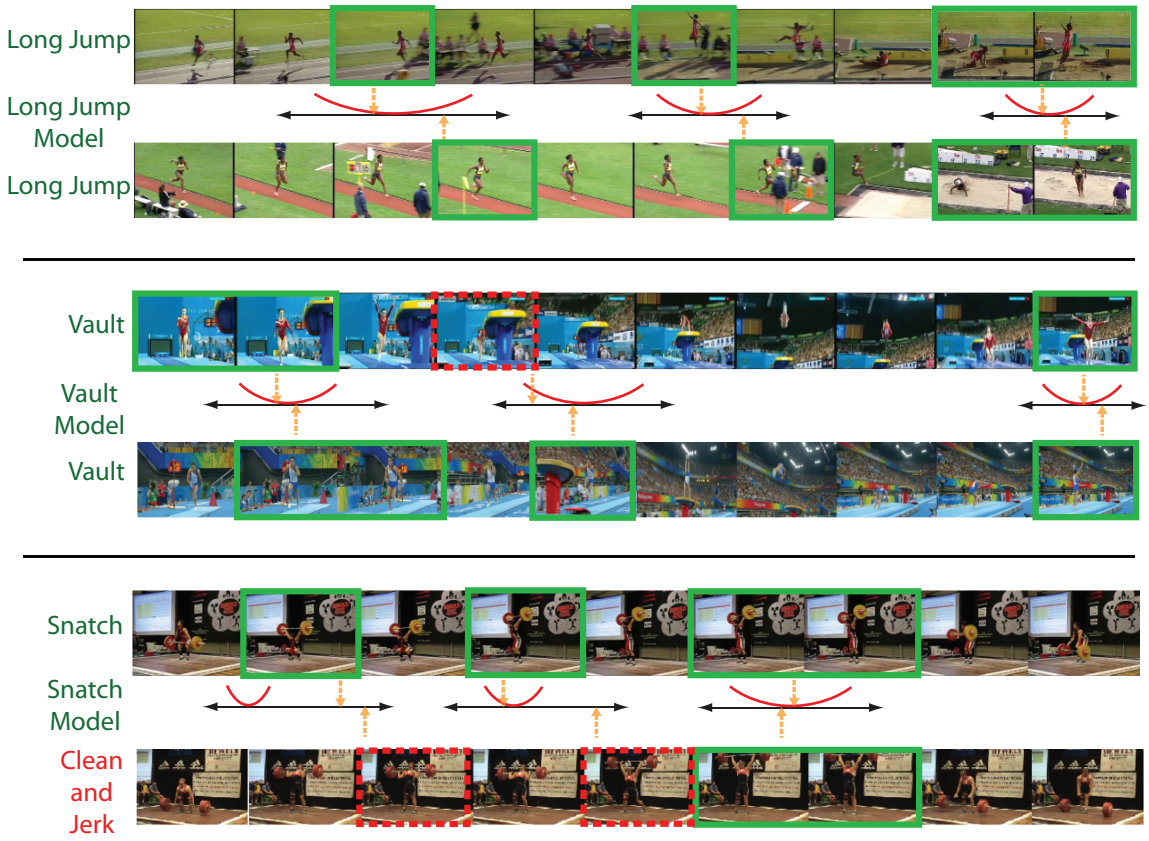


Figure 12.5: Illustration of matching between learned action models for long jump, vault and snatch and some testing sequences. Each group depicts two testing sequences (top and bottom), as well as an illustration of the temporal displacement penalty parameters (middle). Green boxes surround matched temporal segments that are most compatible with the corresponding motion segment classifiers. Red boxes indicate temporal segments that are matched to the motion segment model with a low matching score. The arrows indicate the automatically selected best placement for each motion segment

segments that contain the athlete holding the weights in the snatch model, and the shorter segments that enclose a jumping person in the long jump model.

12.4 Discussion

We have empirically shown that incorporating temporal structures is beneficial for recognizing both complex human activities as well as simple actions. Our video representation is based on vector-quantized spatio-temporal features, but instead of

adopting a global bag-of-words representation, we aim to maintain some of the temporal ordering among features. This enables our model to use temporal structures to discriminate among activities. We presented a discriminative model that decomposes complex actions into a set of simple motion segments and is able to capture the temporal structure among such motion segments. We show promising results in a new dataset of Olympic Sports.

We think there are many potential research directions that can build upon this work. First, our motion segments are represented by the histogram of local features within the segment, so no information about the spatial arrangement of features is encoded. It would be interesting to incorporate a spatial component to this model. Second, while our model does not require tracking of the actor, it might be useful to add tracking of the subject specially in applications with crowded scenarios. This could be achieved by the spatial model or by using tracking as a pre-processing step. Also, while we focus on exploiting the temporal context cue, other types of contextual information are very useful. As shown by others, surrounding objects and the scene in which the activity is performed, can significantly improve recognition rates.

Part V

Future Directions

Computer vision research in automatic understanding of human behavior continues to be a very active topic. There are numerous applications that could benefit tremendously from robust human motion understanding. This is a broad field of research and there are still many unexplored areas.

In this work, we approached the problems of segmenting moving humans from video sequences and categorizing simple and complex actions. While our results are very promising, these problems remain challenging in their most general settings.

In the area of automatic extraction and segmentation of moving people from video, we envision algorithms that are extremely fast and highly accurate at this task. Consider the current state of face detection technology, which has reached mass consumption through point and shoot cameras and webcam software. Robustness, speed and accuracy are key in the success of this technology. It is not difficult to imagine many useful tools that could be built if the technology for segmenting full moving human bodies reaches a similar state. Particular future directions for our work in this topic include formulating our framework to allow processing in an online fashion. As presented here, our algorithms require observing the entire sequence before processing starts. Applications where the extraction is needed as soon as the video is taken will benefit from an online formulation. Other interesting directions include incorporating the ability to track multiple people simultaneously, which can help improving occlusion handling. Additional bottom-up cues such as spatio-temporal video segmentation can also help to improve the segmentation accuracy.

In this work we also explored methods for recognizing simple and complex actions from local spatio-temporal interest points. In general, state of the art algorithms in action categorization are considered less mature than algorithms for object detection and recognition. We see future research in this area taking into account higher level motion semantics and ultimately achieving understanding of the intentions and goals of the actor. As future extensions to the work presented here, further research

will investigate combining spatial structures and temporal structures simultaneously. This will be important especially when the number of actions increases, since subtle variations in space and time structure could be critical for discrimination. A particular approach in this direction would integrate spatial cues from human body configurations given by a pose estimation algorithm. These fine grain spatial cues would also allow to incorporate temporal structure cues at the level of limb and torso movements. On the other hand, future directions might also incorporate cues regarding the objects involved the actions of interest. These cues will provide crucial information for disambiguating goals and intentions between motions that would be identical otherwise. Further contextual cues related to the surrounding scene, camera motion, object trajectories have the potential to provide valuable information. From a broader perspective, it will be interesting to push the activity understanding research to be able to handle motions at a larger range of temporal scales. In our work, we have focused on simple repetitive actions (*e.g.* , walking, hand waving) and complex sport-related actions (*e.g.* , long jump, weight lifting). An important future task is to develop activity recognition systems that are capable of analyzing simple atomic movements (*e.g.* , raising one's hand, standing up) as well as very complex activities (*e.g.* , cooking a meal, shopping for groceries). We have provided some initial foundations on leveraging the temporal and spatial structures for recognizing actions at the middle of this spectrum. Further research is necessary to broaden its applicability to a larger range of temporal scales.

Finally, it is natural to extend our work in the areas of automatic segmentation of moving people from video and human action classification by integrating both tasks. One direction would consider using the two stages as independent components in the processing pipeline. In this scenario, one would first automatically extract the spatio-temporal volume that encloses each actor in the video sequences of interest. A second stage would then focus on learning action models and recognizing actions

from the extracted video sub-volumes. A drawback of such approach is that there is little interaction between the action learning/recognition and extraction task. Our current research is exploring a tighter integration between a human tracking module and an action recognition modules. We believe that closely performing simultaneous action recognition and tracking will create positive feedback, potentially improving the performance of both tasks. In particular, we are investigating a new discriminative formulation for simultaneous tracking and recognition of human actions in video. Our current formulation aims at learning to track and recognize actions from weakly labeled data. This formulation is able to perform tracking and recognition simultaneously, which enables accurate spatio-temporal localization of the actions of interest and provides positive feedback between both tasks.

Bibliography

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021. IEEE, 2009.

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

[3] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, 56(3):221–255, 2004.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[5] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, pages 831–844. Springer, 2008.

[6] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. In *ICCV*, volume 2, pages 1395–1402. IEEE, 2005.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[8] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267, 2001.

[9] O. Boiman and M. Irani. Detecting Irregularities in Images and in Video. In *ICCV*, volume 1, pages 462–469. IEEE, 2005.

[10] G. Borgefors. Distance transformations in digital images. *Computer vision, graphics, and image processing*, 34(3):344–371, 1986.

[11] G. Bouchard and B. Triggs. Hierarchical Part-Based Visual Object Categorization. In *CVPR*, pages 710–715. IEEE, 2005.

[12] T. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, volume II, pages 239–219, 1999.

[13] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[14] V. Cheung, B. J. Frey, and N. Jojic. Video epitomes. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 42–49, 2005.

[15] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1631–1643, 2005.

[16] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, volume II, pages 142–149, June 2000.

[17] D. Cremers. Dynamical statistical shape priors for level set-based tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(8):1262–1273, 2006.

[18] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *IJCV*, 72(2):195–215, 2007.

[19] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[20] N. Dalal, B. Triggs, and C. Schmid. Human Detection Using Oriented Histograms of Flow and Appearance. In *ECCV*, pages 428–441, 2006.

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

[22] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, volume 2, pages 126–133. IEEE, 2000.

[23] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[24] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[25] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, volume 2, pages 726–733 vol.2. IEEE, 2003.

[26] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid Models for Human Motion Recognition. In *CVPR*, volume 1, pages 1166–1173. IEEE, 2005.

[27] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE TPAMI*, pages 1–20, 2009.

[29] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial Structures for Object Recognition. *IJCV*, 61(1):55–79, Jan. 2005.

[30] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8. IEEE, June 2008.

[31] X. Feng and P. Perona. Human action recognition by sequence of movelet codewords. In *3DPVT*, volume 16, pages 717–721. IEEE, 2002.

[32] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proceedings of the Tenth International Conference on Computer Vision*, volume 2, pages 1816–1823, 2005.

[33] R. Fergus, P. Perona, and A. Zisserman. Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition. *IJCV*, 71(3):273–303, 2007.

[34] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, pages 1–8. IEEE, 2008.

[35] D. A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1:77–254, 2005.

[36] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*, chapter 19, pages 234–247. Springer Berlin Heidelberg, 2008.

[37] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 19–25, 2006.

[38] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, pages 2012–2019. IEEE, June 2009.

[39] B. Han, D. Comaniciu, Y. Zhu, and L. Davis. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(7):1186–1197, 2008.

[40] B. Han, Y. Zhu, D. Comaniciu, and L. Davis. Kernel-based bayesian filtering for object tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume I, pages 227–234, Washington, DC, USA, 2005. IEEE Computer Society.

[41] T. X. Han, H. Ning, and T. S. Huang. Efficient nonparametric belief propagation with application to articulated body tracking. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 214–221, Washington, DC, USA, 2006. IEEE Computer Society.

[42] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who? When? Where? What? - A real time system for detecting and tracking people. In *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition,* Nara, Japan, pages 222–227, 1998.

[43] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–152, 1988.

[44] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[45] N. Ikizler and D. A. Forsyth. Searching for Complex Human Activities with No Visual Examples. *IJCV*, 80(3):337–357, 2008.

[46] T. Kadir and M. Brady. Scale saliency: A novel approach to salient feature and scale selection. In *International Conference on Visual Information Engineering*, pages 25–28, 2003.

[47] Y. Ke, R. Sukthankar, and M. Hebert. Efficient Visual Event Detection Using Volumetric Features. In *ICCV*, volume 1, pages 166–173. IEEE, 2005.

[48] Y. Ke, R. Sukthankar, and M. Hebert. Event Detection in Crowded Videos. In *ICCV*, pages 1–8. IEEE, Oct. 2007.

[49] T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor Canonical Correlation Analysis for Action Classification. In *CVPR*, pages 1–8. IEEE, 2007.

[50] D. Klein, S. D. Kamvar, and C. D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML*, pages 307–314, 2002.

[51] X. Lan and D. P. Huttenlocher. A unified spatio-temporal articulated model for tracking. In *CVPR*, pages 722–729. IEEE, 2004.

[52] I. Laptev. On Space-Time Interest Points. *IJCV*, 64(2-3):107–123, 2005.

[53] I. Laptev. Improvements of object detection using boosted histograms. In *BMVC'06, Edinburgh, UK*, volume III, pages 949–958, 2006.

[54] I. Laptev and T. Lindeberg. Velocity adaptation of space-time interest points. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 52–56, 2004.

[55] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *Spatial Coherence for Visual Motion Analysis*, volume 3667 of *Lecture Notes in Computer Science*, pages 91–103, Berlin, 2006. Springer.

[56] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8. IEEE, June 2008.

[57] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *CVPR*. IEEE, June 2007.

[58] C.-S. Lee and A. Elgammal. Modeling view and posture manifolds for tracking. In *ICCV*, 2007.

[59] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume I, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society.

[60] C. Li, C. Xu, C. Gui, and M. D. Fox. Level Set Evolution without Re-Initialization: A New Variational Formulation. In *CVPR*, pages 430–436. IEEE, 2005.

[61] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, 1(3):469–486, 2001.

[62] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *In: Proc. Intl. Joint Conf. on Artificial Intelligence*, pages 674—-679, 1981.

[63] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, pages 2929–2936. IEEE, 2009.

[64] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the 12th European Conference of Computer Vision (ECCV)*, Crete, Greece, September 2010.

[65] J. C. Niebles and L. Fei-Fei. A Hierarchical Model of Shape and Appearance for Human Action Classification. In *CVPR*, pages 1–8. IEEE, June 2007.

[66] J. C. Niebles, B. Han, and L. Fei-Fei. Efficient Extraction of Human Motion Volumes by Tracking. In *CVPR*, 2010.

[67] J. C. Niebles, B. Han, A. Ferencz, and L. Fei-Fei. Extracting Moving People from Internet Videos. In *ECCV*, pages 1–14, 2008.

[68] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. In *BMVC*, Mar. 2006.

[69] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *IJCV*, 79(3):299–318, 2008.

[70] A. Oikonomopoulos, I. Patras, and M. Pantic. Human action recognition with spatiotemporal salient points. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(3):710–719, June 2006.

[71] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282, 2005.

[72] A. Quattoni, S. B. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE TPAMI*, 29(10):1848–53, Oct. 2007.

[73] D. Ramanan. Learning to parse images of articulated objects. In *NIPS*, 2006.

[74] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2003.

[75] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

[76] X. Ren and J. Malik. Tracking as Repeated Figure/Ground Segmentation. In *CVPR*, pages 1–8. IEEE, June 2007.

[77] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In *CVPR*. IEEE, June 2008.

[78] S. Savarese, J. M. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2033–2040, 2006.

[79] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, June 2000.

[80] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, pages 32–36. IEEE, 2004.

[81] E. Shechtman and M. Irani. Space-Time Behavior Based Correlation. In *CVPR*, volume 1, pages 405–412. IEEE, 2005.

[82] H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV*, 2002.

[83] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1-3):183–209, August 2003.

[84] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *CVPR*, 2004.

[85] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *ICCV*, volume 1, pages 370–377. IEEE, 2005.

[86] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *CVPR*, 2005.

[87] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *CVIU*, 104(2-3):210–220, Nov. 2006.

[88] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking. In *CVPR*, 2001.

[89] Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *TPAMI*, 25(7):814–827, 2003.

[90] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, page 561. Springer-Verlag, 2008.

[91] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, Nov. 2008.

[92] O. Tuzel, F. Porikli, and P. Meer. Human Detection via Classification on Riemannian Manifolds. In *CVPR*, pages 1–8. IEEE, 2007.

[93] C. J. Van Rijsbergen. *Information Retreival*, 1979.

[94] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 511, 2001.

[95] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. 9th Intl. Conf. on Computer Vision,* Nice, France, page 734, Washington, DC, USA, 2003. IEEE Computer Society.

[96] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[97] L. Wang, J. Shi, G. Song, and I. Shen. Object detection combining recognition and segmentation. In *ACCV*, volume I, pages 189–199, 2007.

[98] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden Conditional Random Fields for Gesture Recognition. In *CVPR*, volume 2, pages 1521–1527. IEEE, 2006.

[99] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1654–1661, 2006.

[100] Y. Wang and G. Mori. Human action recognition by semilatent topic models. *IEEE TPAMI*, 31(10):1762–74, Oct. 2009.

[101] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 18–32, 2000.

[102] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *CVPR*, pages 794–801. IEEE, June 2009.

[103] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning Motion Categories using both Semantic and Structural Information. In *CVPR*, pages 1–6. IEEE, 2007.

[104] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, volume I, pages 90–97, Washington, DC, USA, 2005. IEEE Computer Society.

[105] B. Yao and L. Fei-Fei. Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities. In *CVPR*. IEEE, 2010.

[106] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

[107] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(11):1531–1536, 2004.

[108] A. Yilmaz and M. Shah. Recognizing Human Actions in Videos Acquired by Uncalibrated Moving Cameras. In *ICCV*, volume 1, pages 150–157. IEEE, 2005.