EXEMPLAR-BASED HUMAN INTERACTION RECOGNITION: FEATURES AND KEY POSE SEQUENCE MODEL

by

Bo Gao
B.Eng., Xi'an Jiaotong University, China, 2009

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2011

APPROVAL

Name: Bo Gao

Degree: Master of Science

Title of Thesis: Exemplar-Based Human Interaction Recognition: Features and Key Pose Sequence Model

Examining Committee: Dr. Hao Zhang
Chair

Dr. Greg Mori, Senior Supervisor

Dr. Mark Drew, Supervisor

Dr. Ze-Nian Li, SFU Examiner

Date Approved: 16 June 2011

Declaration of Partial Copyright Licence

Abstract

Due to intra-class variation, camera jitter, background clutter, etc, human activity recognition is a challenging task in computer vision. We propose an exemplar-based key pose sequence model for human interaction recognition. In our model, an activity is modelled with a sequence of key poses, important atomic-level actions performed by the actors. We employ a strict temporal ordering of the key poses for each actor, an exemplar representation is used to model the variability in the instantiation of key poses. To utilize interaction information, spatial arrangements between the actors are included in the model. Quantitative results that form a new state-of-the-art on the benchmark UT-Interaction dataset are presented. Results on a subset of the TRECVID dataset are also promising.

Keywords: Human interaction recognition, Exemplar, Key poses

Acknowledgements

This is a great opportunity to express my gratitude to people who helped me during my study and research in Simon Fraser University.

First of all, I want to give thanks to my supervisor Dr. Greg Mori, it would be impossible for me to complete the thesis without his endless help and inspiration. He is knowledgeable, kind and thoughtful. He taught me the essence of research, guided me to convert an idea into an exciting work. His advising helped me go though many obstacles in my research journey which lead to the thesis.

Thanks to my lab-mate Arash Vahdat. The teamwork with him was great, he is always full of new ideas and never lack practical spirit in research. I learnt a lot from him.

I also would like to thank Tian Lan, Mani Ranjbar and Weilong Yang, I appreciate their time and patients for my questions. I benefit a lot from their suggestions. It is a great time working with them.

Finally, I am grateful to my parents for years of encouragement and sacrifice. All your love and support make language pale and powerless.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computer vision based analysis of human movement is a broad, active area of research. In this thesis, we are interested in human activity recognition. Human activity recognition is crucial to understand the visual world. The techniques can be used in surveillance, entertainment, video search, etc. Nowadays, video surveillance systems are widely used in airports, nursing homes and other public places. Thousands of hours of videos are captured everyday across the world. However, most surveillance systems are not equipped with effective algorithms to perform surveillance tasks, thus people are required to perform those laborious and tedious monitoring. Using computers to analyse videos and trigger alarms when abnormal activity detections arise is an intelligent solution, which could frees human labour from heavy burden in video surveillance.

Recognition of each individual's activity using state-of-the-art methods in computer vision is challenging. The reasons are manifold. The first and probably the most important reason is intra-class variations – people perform the same activity in their own way, which introduces variations within each activity. Intra-activity variations are common. For example, when people deliver kicks, they raise legs to different heights before they attack targets, hence there are low kicks, high kicks, etc. Fig. 1.1 is an example of intra-class variations in *kick* and *punch*. In real-world scenario, activity recognition systems also have to face the challenges of different lighting situations, camera jitter, background clutter, occlusion, etc. Different lighting conditions change the appearance of a person therefore cast influence on extracted features. Camera jitter not only blurs images but also confuses classifiers when motion features are included. Cluttered environments and human part occlusions introduce high level of noise. All these issues together increase the difficulty of localizing and recognizing human activities. A successful activity recognition system should be able to solve classification tasks without being affected by the factors.

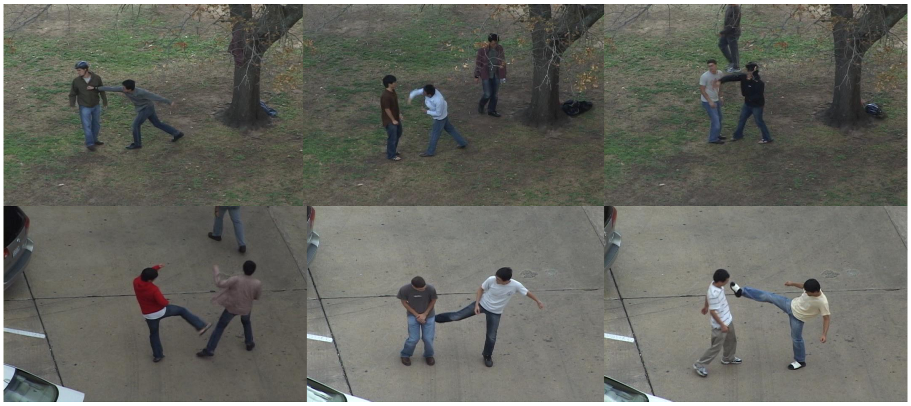


Figure 1.1: Activities are performed with large intra-class variations. The figure illustrates the variations of human poses in two activities–*punch* and *kick*.

Activity recognition starts by extracting features from video sequences. Generally speaking, most of the activity feature extraction methods can be categorized into two lines of work – spatial feature extraction and temporal feature extraction.

For spatial features, the popular approaches include local features and global features. In local features, human figure are decomposed into parts, which are described by separate features. Global features represent human figure holistically and the features are computed densely on a regular grid within a region of interest(ROI). The approach is simpler than local features, thus more computationally efficient. It also preserves structural information of images intrinsically. Hence, we use global features for spatial feature representation in our model.

Template-based approaches and key frames are frequently used to represent temporal features. Templates represent human movement in temporal blocks of video sequences. Templates are usually computed over long sequences of frames. Consequently, they tend to be large in size, and place heavy burden on computation. Characteristic frames of an activity are key frames, such as crucial moments containing salient human poses in each activity. The representation describes a video as a few time independent frames, thus dramatically reduces the computational load. The representation is also robust to temporal variation such as the speed of an activity. However, key frame approach does not model temporal relationships in activities.

Based on the above consideration, we combine templates and key frames approaches together as our temporal feature representation. We use single-frame templates – exemplars to represent human poses in our model. A few exemplars describing human dynamics are selected to represent an activity. Introduction of exemplars is beneficial for our model since a good exemplar usually is representative for a few similar human poses. Nevertheless, exemplar representations have difficulty in adapting to new examples which have not appeared in training. We hope to cover most of pose variations by a set of exemplars extracted from training data. In our case, key frames are actually frames with characteristic poses in an activity. We choose to use key pose sequences to represent activities for three reasons. First of all, it is intuitive to describe an activity using its key frames. Second, it is robust to noisy frames and tracking errors. The representation can tolerant some inaccurate human localization in trajectory since only a few human poses will be included in a key pose sequence. Third, key pose representation shrinks feature space greatly and makes our algorithm computationally efficient.

Human activities are rarely performed in isolation. The way people interact with others also provides us crucial cues. Body-part motion and poses are important for activity recognition. For instance, *push* and *kick* are less likely to be confused given their unique motion and poses. However, simply considering motion and poses can be misleading. Take *push* and *punch* for example, since both of the activities contain outreach arms and defensive actions with similar motion, it is hard to differentiate between them if a classifier only accounts for information contained in poses and motion. At this moment, interaction can serve as another crucial information resource. The fact that *push* ends with relatively larger inter-person distance comparing with that of *punch* may help to classify one activity from the other.

Based on these observations, we develop a discriminative model for recognizing human interaction in video. The intuition is many of the interactions can be summarized by a few key frames performed by actors. For example, a standard scenario for *push* is: one person steps forward, raises his hands, and pushes the other while the other takes a defensive pose, steps backward and falls back in the end. We use an animation term "key pose" to denote human pose in key frames. Observing key poses and their chronological order can be used to recognize an interaction. We propose a model that enables us to find the key poses of an

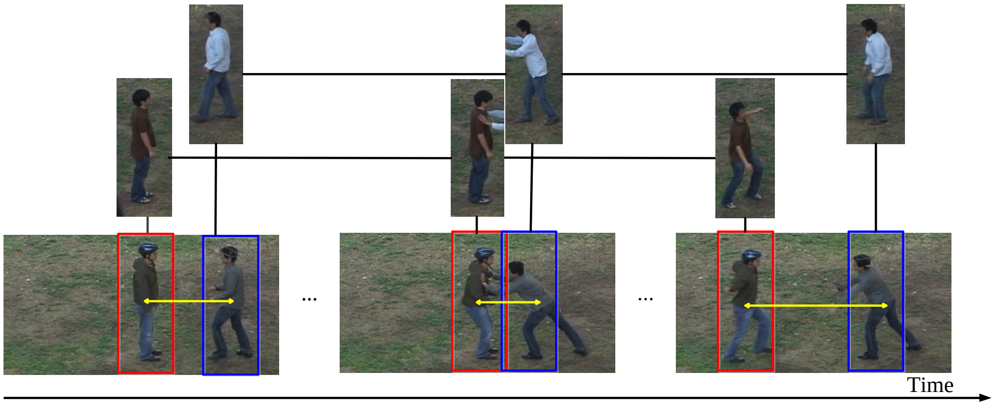


Figure 1.2: High level depiction of our model. Horizontal axis represents time. Localizations of key poses are highlighted in red and blue bounding boxes, exemplars are matched correspondingly. Spatial distances are marked by double-headed arrows in yellow.

activity in a video. As we do not know the key poses in the training data, we treat them as latent variables in a constrained variant of a structured latent variable model. To benefit from the interaction of individuals, we also use inter-person distance in interaction to help recognition. A high-level depiction of the model is shown in Figure 1.2.

At the beginning of the dissertation, we would like to clarify some terminologies frequently used throughout the thesis. We use the term "action" to denote a simple, atomic pose or movement performed by a single actor. If we consider the moment one delivers a punch, this is an action. We use the term "activity" to refer to a more complex scenario that involves a series of human actions. For instance, in *punch*, one person steps forward and delivers a punch while the other one steps backward and takes a defensive action. The series of actions compose an activity. We view human interactions as part of human activities. The interactions we consider include *handshake*, *hug*, *punch* and *push*. In these activities, *handshake* and *hug* are symmetric, one can not distinguish subjects from objects. On the other side, one can tell subjects from objects in activities like *punch* and *push*. We distinguish symmetric interactions from asymmetric ones and make use of activity symmetry in interaction modelling. Activity recognition is the assignment of an output value or a class label to a given input video instance. An example of activity recognition is activity classification, whose goal is to correctly classify video sequences into some predefined categories according to human activities performed in videos. However, activity recognition is a more general problem, it may also include other types of output, such as a valued output or a structured output. Different from activity classification, activity detection requires localizing spatial-temporal activity locations in videos instead of just choosing class labels for the entire video. We design our model for activity recognition. However, it can be adapted to tackle detection tasks.

1.1 Individual Activity Recognition

Individual activity recognition is our first step towards interaction recognition. Given a video consisting of actors performing an activity, we want to find key poses in the sequence and use them to describe the activity. Key poses have large variations in appearances. To handle the variations, we introduce a set of exemplars to match key poses. Key poses occur rarely in a video. Much of each video may consist of highly variable human action that can be misleading when attempting to build an activity model. To cope with the problem and extract representative exemplars, We use heuristics to select representative exemplars from noisy training data. Details about the method to select discriminative exemplars are presented in Chapter 3.

In our model, we assume a reliable human detector and human tracker can detect individuals and extract trajectories of individuals in videos. For this reason, its performance depends on the accuracy of detection and tracking. To be robust to the performance of detectors and trackers, we instantiate every key pose of a video sequence as a three-element tuple, not only includes **what** it looks like, appearance described by the selected exemplar, but also include the other two aspects: **where** it occurs and **when** it appears. We search for the perturbation around each trajectory to find where the best key pose matching locates. Actions occur in order, so it is also intuitive to match key poses with respect to their chronological order. We encode time constraint in time via an efficient algorithm to guarantee that key poses are matched to the input sequence in chronological order. Please refer to Chapter 3 for implementation details.

1.2 Human Interaction Recognition

In this thesis, our goal is to recognize human interactions in videos. There are several ways to extend our single person model to capture interactions. The easiest way is to learn parameters of the model for each individual involved in the interaction and then use them to score each participant separately. The method fails to capture crucial interaction. It also goes against the intuition that the model parameters should be different for each participant especially in asymmetric activities like *push*, *kick* and *punch*. For example, in *push* the key poses for the subject are stepping forward, rising hands in front, and shoving. However, for the object who is pushed the key poses are defending, stepping backward, and falling back. We expect to see a different group of key poses for the subject and object trajectories.

One can capture the interaction by defining a potential function between the latent key poses of the two trajectories. In this case, the model is rich enough to capture the co-occurrence of the key poses in an interaction. A rich model that encodes co-occurrence of key pose will be computationally prohibitive. In the matching of a key pose to the subject, all other possible key poses at all possible frames of the object trajectory should be considered.

Considering both computational speed and model richness, we take an intermediate approach. We assume that we are given the rough trajectories of a potential subject and object in interaction, and then we match key poses to each trajectory. To model the asymmetry in the interaction, we define two different key pose sequence models for subject and object trajectories, and include a new hidden boolean variable to decide whether a participant of interaction is subject or object. In this way, we can match different key poses for subjects and objects, and learn different weights in a joint way.

To utilize spatial arrangements in interaction, we include the distances between individuals when key poses appear in our model. Individual distances are discriminative both within activities and among activities. For example in *hug*, subjects open their arms at certain distances and then embrace at very nearby spatial locations afterwards. Among activities, it is also obvious that the distances between individuals are usually larger in activities without physical contact. For instance, the distances between individuals in *point* should be larger than that in *hug*, which can also be used to classify activities.

We do not limit ourselves to model interactions between two people. We can modify our model for group interaction. Group interaction with more than two people get involved always can be factored into interaction pairs. Our algorithm can be applied on the pairs before we aggregate responses for group interaction recognition. Therefore, our model has an good generality.

1.3 Collaboration

The work is done in collaboration with Arash Vahdat, Mani Ranjbar and Greg Mori. The contribution of the author was centred around the design and implementation of interaction key pose sequence model. The author is also responsible to make the TRECVID Embrace dataset, select suitable human detector and tracker for UT-Interaction dataset and TRECVID Embrace dataset. Besides, the part using pose information to help exemplar matching is proposed and implemented, and experimentally evaluated by the author.

1.4 Outline

The rest of this thesis is organized as follow:

Chapter 2 reviews related works in computer vision. First, we survey popular approaches for human action recognition. Then we expand the chapter around topics most relevant to our work: popular approaches in human action recognition, exemplar representation, temporal models and interaction recognition methods.

Chapter 3 focuses on key pose sequence model for recognizing human interactions. Detailed descriptions of single subject key pose sequence model and interaction key pose sequence model are included. We also propose a way to extract exemplars and a method to encode time constraint in exemplar matching via an efficient algorithm. In the end, inference and problem solving details are given.

Chapter 4 shows the effectiveness of our model in recognizing human interactions. We evaluate our model's performance on the UT-Interaction dataset and a subset of the TRECVID dataset. Experimental results demonstrate our model's strength. We also visualize the learnt weights and explain their meanings.

Chapter 5 concludes this thesis and discusses potential future work.

Chapter 2

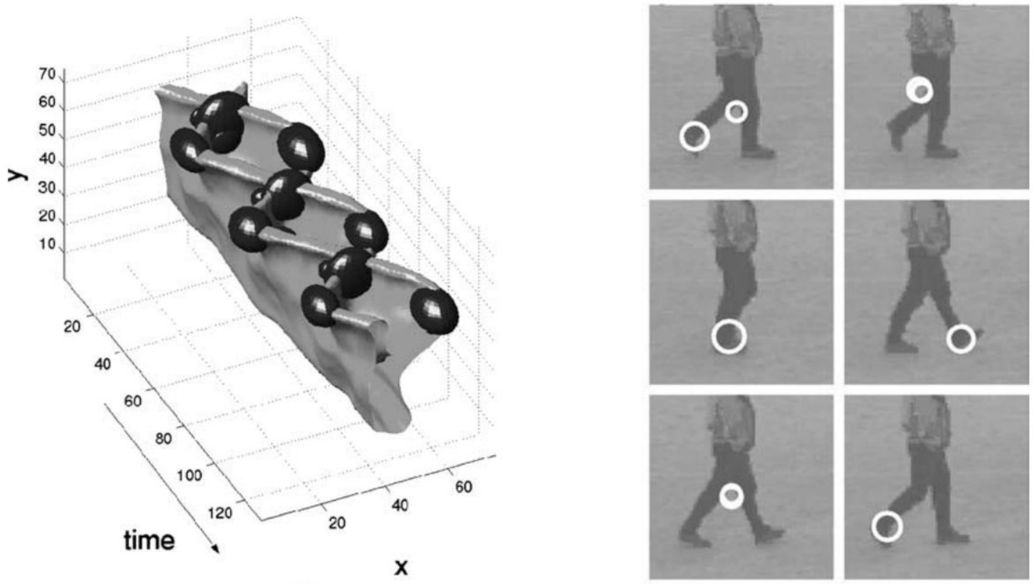
Previous work

In this chapter, we review the works related to our model. Section 2.1 will briefly summarize popular approaches for human activity recognition. Section 2.2 will focus on methods related to exemplar representation. Section 2.3 will give an overview of a variety of temporal models have been developed, ranging from template matching to probabilistic temporal sequence models. Section 2.4 will summarize the models proposed for interaction recognition in video.

2.1 Human Activity Recognition

Human activity recognition is a challenging task in computer vision. Literature in this field is immense [12, 17, 36]. A typical activity recognition system includes two critical components: feature representation and model construction. Generally speaking, proposed features can be divided into local features and global features. Local features represent human figure as sets of local interest points or cuboids with critical information. Global features describe human figure holistically, using appearance, motion, etc.

One of the popular approaches in activity recognition is based on local features. Schuldt et al. [26] propose a local space-time feature descriptor that can be adapted to size, frequency and velocity of moving patterns. They detect local structures in space-time volumes where the pixel values have significant local variations and compute scale-invariant spatio-temporal features at theses locations. Finally, support vector machines (SVM) are trained on the features for classification tasks. Figure 2.1 is a visualization of local space-time features used in their work.



(a) 3D video volume

(b) Spatial-temporal feature point

Figure 2.1: Results of detecting spatio-temporal interest points from the motion of the legs of a walking person. (a) 3-D plot with a thresholded level surface of a leg pattern (here shown upside down to simplify interpretation) and the detected interest points illustrated by ellipsoids; (b) spatio-temporal interest points overlayed on single frames in the original sequence. The figure comes from [26].

Niebles and Fei-Fei [18] use a collection of spatial-temporal words extracted from space-time interest points to represent a video sequence. They first extract local space-time cubes using a space-time interest point detector and then cluster them to form a codebook. Their algorithm learns the probability distributions of the words by using a probabilistic Latent Semantic Analysis model.

Shechtman and Irani [28] define a behaviour-based similarity measurement to correlate small space-time video to entire video sequences in spatial-temporal volume. They exhaustively search space-time intensity patterns of two different video segments for similar underlying motion, and then aggregate responses for behaviour matching.

Appearance and motion features are global features widely used in activity recognition. Dalal and Triggs [4] use Histograms of Oriented Gradient (HOG) as features for pedestrian detection. They study influential parameters in feature computation (e.g. gradient scale, orientation and spatial binning, local contrast normalization, etc) and illustrate their influences on HOG feature quality. Further, they compare with other feature sets to demonstrate

HOG descriptor is promising for human detection tasks.

Efros et al. [7] aggregate blurred optical flow in time to form spatial-temporal motion features. They account for activity variations using temporal smoothing and recognize activities according to frame-level feature similarity between sequences. Similar to their method, we also calculate optical flow and bin flow directions to build histogram of optical flow (HOF) as described in [5]. We concatenate the HOG and HOF together as our features aiming to capture both appearance and motion information.

Given extracted features, a model is required to accomplish classification tasks. Constructed models typically can be divided into two categories: generative models and discriminative models.

In a generative approach, the joint distribution of features and labels $P(x, y)$ are modelled. This can be done by learning the class prior probability $P(y)$ and the class-conditional density $P(x|y)$ separately. Classification can be done by calculating conditional probability of a class given an observation $P(y|x)$ using Bayesian rules. Hidden Markov Model(HMM) is probably the most famous generative model, its fame rises with its great success in speech recognition. HMM are also widely used in activity recognition. Ogale et al. [20] represent human activities as short sequences of body poses. Body poses are stored as sets of silhouettes seen from multiple viewpoints. Activities and their atomic poses are extracted from sets of multi-view video sequences by applying an automatic key frame selection algorithm, and they are used to construct HMMs. Given new single viewpoint sequences, the system can recognize key pose sequences and changes viewpoint. Further, activity classification can be achieved by recognizing pose sequences.

In a discriminative approach, a parametric model for the posterior probabilities is introduced, and the values of the parameters is inferred from a set of labelled training data. Wang and Mori [34] model a human action in a discriminative way. They use a flexible set of parts conditioned on image observations and combine both large-scale global features and local patch features in a max-margin hidden conditional random field framework (MMHCRF) for action recognition. Our work is closely related to the method since we also use MMHCRF framework to build our model, however, we aim for interaction recognition. To cope with problems we encountered in interaction recognition, such as intra-class pose variations and trajectory jitter, we develop an exemplar-based key-pose sequence model.

2.2 Exemplar-based Representation

Traditional approaches model activities as space-time representations which explicitly or implicitly encode human dynamics in time. Bobick and Davis [2] extract temporal template features in video. They use binary cumulative motion images referred to motion-energy images (MEI) and temporal history of motion called motion-history images (MHI) as features. MEI suggest the shape of movement and MHI record how motion is moving. In recognition, a Mahalanobis distance is calculated between the features of the input and that of each known movement. Our model is similar to their approach in using templates to represent video sequences. However, our algorithm learns to select key poses.

In contrast, Weinland et al. [35] represent sequences as a set of exemplars without modelling any temporal ordering. The time-invariant features simplify learning and recognition by removing information in time. In their methods, an activity sequence is matched against a set of exemplars. For each exemplar the minimum matching distance to any of the frames in the sequence is determined. The resulting set of distances form a vector in the embedding space, so that point representations of videos can be used in activity recognition. Their method is similar to ours in using exemplars. On the other hand, our model uses the concept of key pose, and focuses on those informative frames instead of the whole video. Considering the importance of temporal information, we also encode hard temporal order for matched exemplars.

Schindler and van Gool [25] present a system for activity recognition from snippets of 1 – 10 frames. Experimental results show that short snippets can be effective in activity recognition. It strengthens our belief that a small number of key poses can be enough to effectively recognize activities. As an extension of activity recognition from snippets, Satkin and Hebert [24] present a framework for estimating what portions of videos contain the most salient information. They explore the impact of temporal cropping of training videos on the overall accuracy of an activity recognition system. Their work could be integrated into ours as a preprocessing step to filter out noisy data and help us extract exemplars in a more accurate way.

Exemplar matching approaches usually require a large set of training images as exemplar candidates. Sequentially searching for the best match is a slow process in nature. To overcome the problem, it is possible to organize exemplars in a data structure suitable for fast searching. Shakhnarovich et al. [27] use parameter sensitive hashing for approximate exemplar matching. The idea is to build a hash function that is more likely to map similar exemplars in the same bucket. Lin et. al [15] organize exemplars in a prototype tree via hierarchical k-means clustering, and use it to speed up the exemplar matching process. We can adapt the methods in our model for fast exemplar matching, and it will be helpful when the size of exemplar set is huge.

Deformation of highly articulated human figure also contributes to the difficulty of exemplar matching. A matching that takes configuration cues (e.g. pose) into consideration can be more accurate. Yang et al. [38] adapt the concept of "poselet" proposed by Bourdev and Malik [3] into their work. They treat human poses as latent variables and make use of pose information in action recognition. The method may help us to achieve a better exemplar matching. Developing a faster exemplar searching method and utilizing pose for better exemplar matching are two promising directions. We consider them as part of our future work.

2.3 Temporal Model for Activity Recognition

In activity recognition, some methods represent activities as sequences of templates and the activity recognition problem is approached as a template matching process.

Lin et al. [15] build an action prototype tree, which is learned in a shape and motion space via hierarchical k-means clustering, then they match input frames to prototypes based on shape and motion similarity. Dynamic time warping is used to align two activity sequences and measure distances between them. Similar to our method, their work employs shape and motion similarity in prototype matching. However, our matching scheme is built in a patch-based manner and our model only focuses on key pose matching, not the whole video.

Recently, Niebles et al. [19] develop a model representing activities as temporal compositions of motion segments. They extend key frames to short motion segments and exploit the temporal structure of human activities. Although similar to our model, they model without strict temporal ordering and the non-parametric exemplar matching.

Instead of representing activities as templates, some temporal models describe activities as sequences of moments in feature space. A common way for these models to approximate activities is to model similar features and configurations as states and learn transitions between states. The first work using Hidden Markov Models (HMM) for activity recognition dates back to Yamato et al. [37]. In this paper, a discrete HMM is used to represent a specific activity sequences over a set of quantized image features. To recognize an observed sequence, the HMM that best matches the sequence is chosen. A drawback of HMM model is the assumption that a state transition is conditioned on only previous state, not on other states. The assumption about state independence is necessary to make the model computationally tractable. However, it comes with contextual information loss. The generative HMM also has difficulty in modelling independent movements of human parts. Different from their approach, our work uses a discriminative framework focusing on salient poses of an activity. Furthermore, our part-based matching scheme is able to model independent shape and movements of human parts.

Given a video of one person conducting a sequence of continuous actions, Shi et al. [29] define a set of features to capture the characteristics of action segments, boundary frames of segments and the relationship between neighbouring action segments. They combine the features under a discriminative semi-Markov framework for human action segmentation and recognition. Different from HMM, frames in one segment share one label and this label depends on its adjacent segment labels in the semi-Markov model. Their work provides an interesting way to model multiple activity detection in a video, which could be integrated into our model for multiple activity detection in videos.

2.4 Models for Interaction Recognition in Video

A variety of interaction recognition algorithms have been proposed. Ryoo and Aggarwal [22] introduce a spatio-temporal relationship matching kernel, which is designed to measure structural similarity between features extracted from two videos. It considers spatio-temporal patterns among interest points, enabling detection and localization of complex activities.

Yao et al. [39] use a set of interest points to represent a video and approach activity recognition in the Hough transform voting framework. They train random trees to learn a mapping between sampled feature patches and votes in a Hough space. Leaves of trees are learnt to be a discriminative codebook, so they can vote for activity centres with probabilities. In testing, randomly extracted patches from a video are used to pass through the trees in the forest and the leaves that the patches arrive in are used to cast votes.

Yu et al. [40] present a real-time solution which utilizes local appearance and structural information. Semantic texton forests (STFs) are applied to convert local space-time patches to discriminative codewords. To capture the structural information of activities, pyramidal spatio-temporal relationship match (PSRM) is introduced. Different from searching for spatial-temporal structure of interest points in previous methods, we model exemplars and their temporal order to address interaction recognition problems.

Chapter 3

Human interaction recognition

In this chapter, we consider the problem of human interaction recognition. In Section 3.1, a single subject key pose sequence model is proposed to model the activity of a single person involved in interaction. In Section 3.2, we extend it to interaction model after introducing spatial placement between individuals and distinguishing subject from object. A description of the feature we use and how to select exemplars in training is available in Section 3.3. We present our inference for the model and learn the weights using NRBM in Section 3.4.

Our goal in this thesis is to recognize human interactions in videos. We will model these interactions by a sequence of key poses. Observing them and their chronological order can be used to recognize an interaction.

Given an input video and a putative interaction, four things are unknown:

1. **Who** is involved in the interaction? More specifically, which person is taking which role in the interaction – many interactions, such as pushing or kicking, have distinct "subject" and "object" roles.

2. **When** do the key poses occur? We model each interaction by a fixed-length sequence of key poses, but we do not know a priori when these key poses occur in an input video.

3. **How** are the key poses executed? There is variation in appearance for the key poses of an interaction – e.g. is the push with one hand, two hands, a forceful push, or a weak push.

4. **Where** are the people when the key poses occur? The spatial arrangement of these key poses is important – interactions such as pushing or embracing have stereotypical relative distances between the people involved.

These are unknown and, while inferring them is useful, are not our direct goal of recognizing interactions. Hence, we treat them as latent variables in a novel constrained variant of a structured latent variable model.

Following the standard notation in structured latent variable models, we now provide a formulation of our model. Let $x \in \mathcal{X}$ be a video sequence that consists of people performing an interaction $y \in \mathcal{Y}$ where $\mathcal{Y}$ is the finite set of interactions. Given a set of video and interaction label pairs, our task in training is to learn a scoring function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \Re$ over these pairs. Following the usual latent variable formulation, we will assume F maximizes a model G that includes the latent variables $\mathbf{H}$: $F(x, y) = \max_{\mathbf{H}} G(x, y, \mathbf{H})$.

In our work, the latent variables $\mathbf{H}$ answer the four questions above. Namely, $\mathbf{H} = [b, t, e, p]$, where:

1. b specifies who takes which role in the interaction. In this work we assume we are provided roughly correct tracks of the people in a scene, and b denotes which person is the subject and object of the interaction.

2. t specifies when the key poses occur. Our interaction model has a fixed number of key poses (e.g. 5 in experiments). t specifies when in the (much longer) input video x these key poses occur. This key pose sequence will be constrained to be in chronological order.

3. e specifies how the key poses are executed. We use an exemplar-based representation in which e specifies which discrete type of execution of a key pose is present in a video. Essentially, this is similar to an aspect or mixture model to account for key pose variation.

4. p specifies the spatial locations in the video frames for the key poses. As with b, we will rely on a tracker to assist with this information, allowing small shifts in position from tracker output to account for tracker error.

3.1 Single Subject Key Pose Sequence Model

We start from a model of a single person performing an activity. Given a set of videos, our goal is to find a set of key poses in these sequences and use them to describe the activity class. First, large portion of each video in our dataset consist of highly irrelevant, even misleading human activities, which is one of our main obstacle when attempting to build an activity model. Considering *push* for example, there are poses such as standing or walking at the beginning or the end of the video that are variable and not discriminative. We introduce key poses, which are important, infrequent in activities to provide robustness to noisy and ambiguous frames. Second, each of the key poses will have variation in appearance. We would like to use a set of discriminative exemplars automatically extract from training data to enumerate the major pose variation. Finally, our model is built on the assumption that trajectories provided by trackers are reliable, however, even state-of-the-art method in human tracking can not always assure accurate trajectory, for this reason, the spatial arrangement of these key poses is locally perturbed for robustness to trackers performance, so the model also should include where in a video frame key pose may locate.

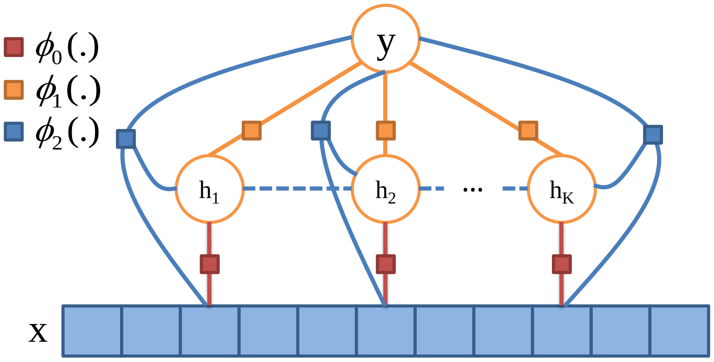


Figure 3.1: The graphical depiction of our model for single subject key pose sequence matching. The lower layer x is the observed sequence of frames, and the middle layer h is the key pose sequence layer and the top layer y is the activity label. Edges with boxes denote factors in our model. Dash lines represent time constraints between key poses.

An instantiation of a single subject key pose model in a video sequence consists three parts: **when** do the key poses occur, **how** is each key pose executed, and **where** in space do they occur. We assume that we are given a rough track of the subject, via human detection and tracking algorithms. We represent each key pose in a sequence by a triple $h = [e, t, p]$. Variables t and p are its spatio-temporal locations, with p restricted to locations near the tracker output. The variable $e \in \mathcal{E}$ denotes which appearance variant of the key pose is taking place at time t and location p. A discrete set of exemplars $\mathcal{E}$ is used as a representation of the appearance of key poses - for instance, the different types of pushes noted above would each be represented by its own element of $\mathcal{E}$. As noted above, a model contains multiple key poses in sequence, and we denote the K key poses of a sequence by $\mathbf{H} = [h_1, h_2, .., h_K]$, where each h_i is a triple $[e_i, t_i, p_i]$. Our model also has a constraint on the temporal component of the key poses $\mathbf{H}$. The key poses should be matched to the input sequence in chronological order, hence $t_i < t_j$ if $i < j$. This hard constraint will be enforced in inference via an efficient algorithm.

We now describe the scoring function $G(x, y, \mathbf{H})$ for a single subject model. A graphical depiction of our model is shown in Fig. 3.1. Factors in this model include terms measuring compatibility between input sequences and instantiations of key poses, between key poses and activity label, and among the three. Based on this model, a sequence of key poses $\mathbf{H}$ is scored for the input x and the label y by $G(x, y, \mathbf{H}) = \boldsymbol{\omega}^T \boldsymbol{\Phi}(x, y, \mathbf{H})$ which is a linear function on $\boldsymbol{\omega}$, the parameters of the model. We formulate the scoring function as:

$$\boldsymbol{\omega}^T \boldsymbol{\Phi}(x, y, \mathbf{H}) = \sum_{i=1}^{K} \boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, h_i) + \sum_{i=1}^{K} \boldsymbol{\beta}_i{}^T \boldsymbol{\phi}_1(y, h_i) + \sum_{i=1}^{K} \boldsymbol{\gamma}^T \boldsymbol{\phi}_2(x, y, h_i) \tag{3.1}$$

where $\boldsymbol{\phi}_0(\cdot)$, $\boldsymbol{\phi}_1(\cdot)$ and $\boldsymbol{\phi}_2(\cdot)$ are the potential functions defined on the links which will be described below. $\boldsymbol{\alpha}$, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_K]$ and $\boldsymbol{\gamma}$ are the parameters of the model which are grouped in $\boldsymbol{\omega} = [\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$.

Exemplar Matching Link: $\boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, h_i)$ measures the compatibility between key pose h_i and the whole image of one track at time t_i and location p_i. It is formulated as:

$$\boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, h_i) = \sum_{e \in \mathcal{E}} \boldsymbol{\alpha}_e{}^T \boldsymbol{D}(f(x, t_i, p_i), g(e_i)) \mathbb{1}_{\{e_i = e\}} \tag{3.2}$$

where $f(x, t, p)$ computes features for sequence x at the location p and time t contained in h. Similar to $f(\cdot)$, $g(\cdot)$ calculates the features for exemplars. The details of these features and distance measure $\boldsymbol{D}$ are described in Sec. 3.3. $\mathbb{1}$ is an indicator function selecting for the weight vector associated with the exemplar used in h.

Activity-Key Pose Link: $\boldsymbol{\beta}_i{}^T \boldsymbol{\phi}_1(y, h_i)$ models the compatibility between activity y and exemplar e_i as the i^{th} key pose ($h_i = (\cdot, \cdot, e_i)$). It reflects our model's exemplar preference at different time in each activity, and high value means that particular type of key pose is strongly associated with i^{th} key pose in activity y:

$$\boldsymbol{\beta}_i{}^T \boldsymbol{\phi}_1(y, h_i) = \sum_{a \in \mathcal{Y}} \sum_{e \in \mathcal{E}} \beta_{iae} \mathbb{1}_{\{y = a\}} \mathbb{1}_{\{e_i = e\}} \tag{3.3}$$

The activity-key pose term is indexed on key poses $\boldsymbol{\beta}_i$, and it means that an exemplar may have different compatibility with an activity at different times. This models the fact that key poses have a particular order in each activity. For example bending starts with a standing pose, continues with bending until the subject reaches ground, and ends with a standing pose. An exemplar record bending should have a high probability to be selected to match key pose in the middle instead of the first or the last key pose.

Direct Root Model: $\boldsymbol{\gamma}^T\boldsymbol{\phi}_2(x,y,h_i)$ measures the compatibility of global features extracted from x at h_i and activity class label y. This directly models the features of the input to the activity class label, without exemplars. It is parametrized as:

$$\boldsymbol{\gamma}^T\boldsymbol{\phi}_2(x,y,h_i)=\sum_{a\in\mathcal{Y}}\boldsymbol{\gamma}_a{}^T\boldsymbol{M}(f(x,t_i,p_i))\mathbb{1}_{\{y=a\}} \tag{3.4}$$

A multi-class SVM is trained on training data. Given input feature, $\boldsymbol{M}$ return a vector containing scores for classifying the input as all the class labels. $\boldsymbol{\gamma}$ is the concatenation of γ_a for all $a\in\mathcal{Y}$.

3.2 Interaction Key Pose Sequence Model

Our goal is to recognize human interactions in a video. There are several ways to extend our model in Sec. 3.1 to capture interactions. The easiest way would be to learn parameters of the model for each individual of the interaction, and use them to score each participant separately. The problem with this method is that it cannot capture any information about interaction. For asymmetric activities such as *kick*, *push*, or *punch* the model parameters should be different for each participant. The participants of these interactions include the subject of activity, the one who does the activity, and the object of the activity, the one to whom activity occurs. The subject and object in an interaction should have different key poses. For example, in *push* the key poses for the subject are stepping forward, putting hands in front, and shoving actions. However, for the object who is pushed the key poses are a defensive pose, stepping backward, and falling back. So, we expect to see a different group of key poses for the subject and object trajectories. Further, as noted above relative spatial position of the subject and object of an interaction is an important cue for recognition.

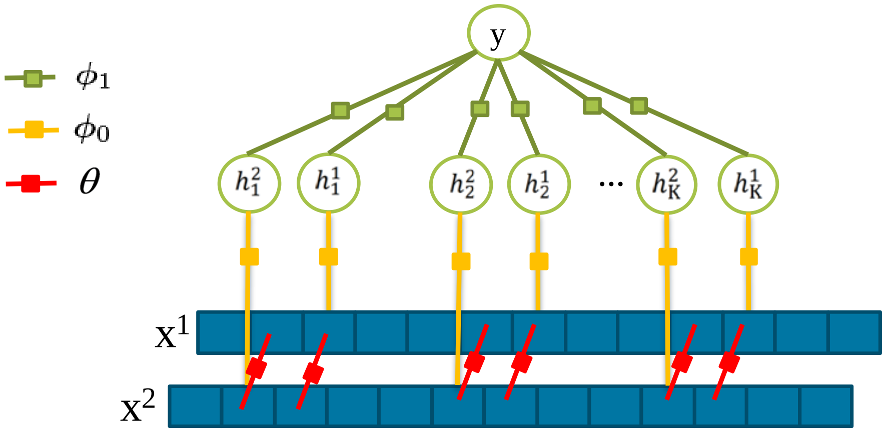


Figure 3.2: Illustration of our interaction model. The lower layer x^1 and x^2 is the observed sequence of frames for two trajectories in interaction, and the middle layer h is the key pose sequence layer and the top layer y is the activity label. Edges with boxes denote factors in our model. We hide ϕ_2 for clearance of the figure.

We modify our single subject model to incorporate this information: **who** is playing which role, and additional cues about **how close** these people are. The model is depicted in Fig. 3.2. We assume we are given the rough trajectories of a potential subject and object of an interaction, and similar to our model in Fig. 3.1 we match key poses to each trajectory. However, we model the asymmetry in the interaction, and we define two different compatibilities between key poses and activity for subject and object tracks. In other words, in Eq. 3.1, we use $\boldsymbol{\beta}^s$ and $\boldsymbol{\beta}^o$ for subject and object trajectories. Further, we model the spatial distance of the key poses by an additional term in the scoring function, denoted by $\boldsymbol{\theta}$. The intuition is that the key poses of an activity occur at common spatial distances from each other. For example in *hug* subjects open their hands at a certain distance and then embrace at very nearby spatial locations afterwards.

Let x be a video that contains two people interacting. In our interaction model the latent variables are $\mathbf{H} = [\mathbf{H^1}, \mathbf{H^2}, b]$. $\mathbf{H^1}$ and $\mathbf{H^2}$ are the key pose configuration for each person. The variable $b = (b^1, b^2)$ selects which person trajectories take the subject and object roles in the interaction. We assume a tracker provides the rough trajectories of the people in the video. We use $l(x, t, b^1)$ to denote the location of subject actor in sequence x at time t (same as $l(x, t, b^2)$ for object trajectory). Given a sequence, a latent variable configuration, and a class label, we calculate the score of each participant, and include the score of the spatial distance link. The scoring function for the interaction model is formulated as:

$$L(x, y, \mathbf{H}; \boldsymbol{\omega}) \quad = \quad G(x, y, \mathbf{H^1}, b^1; \boldsymbol{\omega}^s) + G(x, y, \mathbf{H^2}, b^2; \boldsymbol{\omega}^o) + Q(x, y, \mathbf{H}; \boldsymbol{\mu}) \tag{3.5}$$

where we make explicit the dependence of G on different parameter subsets

$\boldsymbol{\omega}^s = [\boldsymbol{\alpha}, \boldsymbol{\beta}^s, \boldsymbol{\gamma}]$, $\boldsymbol{\omega}^o = [\boldsymbol{\alpha}, \boldsymbol{\beta}^o, \boldsymbol{\gamma}]$ for different trajectories. The parameter b is used to select tracks (not considered in the single-subject model). Note that $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are assumed to be identical for the subject's and object's trajectories, while $\boldsymbol{\beta}$, the compatibility of key poses and activity is different. $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ...\boldsymbol{\mu}_K]$, $\boldsymbol{\mu}_i$ are the parameters that measure the compatibility between activity y and binned distance between tracks at the time of the i^{th} key pose. $Q(x, y, \mathbf{H}; \boldsymbol{\mu})$ measures the relative distance of two tracks at the time of the key poses and is formulated as:

$$Q(x, y, \mathbf{H}; \boldsymbol{\mu}) = \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x, y, h_i^1, b) + \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x, y, h_i^2, b) \tag{3.6}$$

where

$${\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x, y, h_i^j, b) = \sum_{a \in \mathcal{Y}} {\boldsymbol{\mu}_{ia}}^T \boldsymbol{bin}(\|l(x, t_i^j, b^1) - l(x, t_i^j, b^2)\|_2) \mathbb{1}_{\{y=a\}} \tag{3.7}$$

i.e., the distance between the tracks at the time of the i^{th} key pose in j^{th} trajectory. The function $\boldsymbol{bin}(\cdot)$ discretizes this distance. To summarize, the full set of parameters is $\boldsymbol{\omega} = [\boldsymbol{\beta}^s, \boldsymbol{\beta}^o, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\mu}]$. Note that the scoring function L is a linear function of $\boldsymbol{\omega}$. Essentially, the average of the distances at the times of these key poses are considered to model the spatial distance between the people. Figure 3.2 is an illustration of our interaction model.

3.3 Features

Our goal of exemplar matching is to find an exemplar that shares maximum pose similarity with input from an exemplar set. We employ patch-based similarity in appearance and motion feature space to tackle the problem. We also try to parse human poses directly and include them as part of our features. The weights learnt on exemplar patches or pose features should specify their importance in exemplar matching.

3.3.1 Appearance and Motion Features

In order to match key poses to the input sequence with similarity in both appearance and motion, we choose histograms of oriented gradients (HOG) and histograms of optical flow (HOF) as features to capture shape and movement of individuals. We use 240 × 180 bounding box for each individual. To limit feature dimension, we resize each cropped image into half of its origin size before features are extracted. We break each resized image into 8×8 non-overlapping cells, each cell is represented by a histogram of oriented gradient and oriented optical flow. HOG feature extraction starts from gradient computation, followed by creating cell histogram. Pixels within a cell will cast a weighted vote for an orientation-based histogram channel based on gradient magnitudes. To account for illumination and contrast, gradient values are locally normalized. The final descriptor is the concatenation of the histograms of oriented gradients for each cell. We use the code provided by Felzenszwalb et al. [10] to calculate HOG features. Their code bins oriented gradients into 9 orientations for each pixel and aggregates all discretized oriented gradients in a cell to form a histogram. In the end, principle component analysis is employed to further slim feature dimension. We compute optical flow of entire images using the Lucas-Kanade [16] algorithm, then crop out motion in bounding boxes, and we calculate HOF in a similar way. We represent images using a concatenation of HOG and HOF features in 8×8 non-overlapping cells organized on grids inside bounding boxes around the subjects. Figure 3.3.1 is an illustration of HOG and HOF features we used.

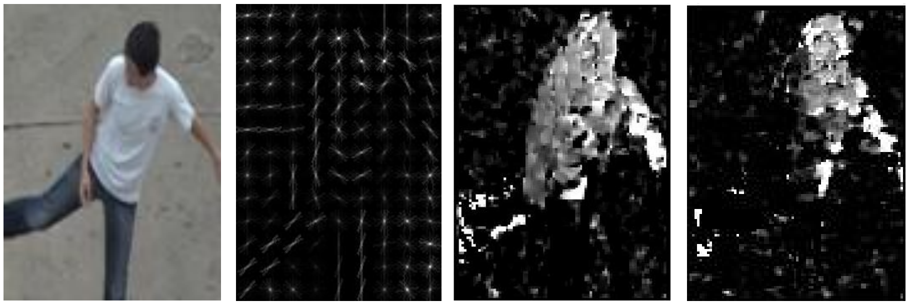


Figure 3.3: Illustration of features we use. The first image is the output of human detection in a video clip of *kick*. We illustrate HOG features to its right, followed by its optical flow magnitude in horizontal and vertical directions. We concatenate their patch-based features to form our appearance and motion descriptor.

In Eq. 3.2 we use a function $\boldsymbol{D}(\cdot, \cdot)$ to measure the distance of two bounding boxes in feature space. The inputs to $\boldsymbol{D}$ are HOG and HOF features of the two bounding boxes and the output is a vector with i^{th} component storing normalized Euclidean distance between HOG and HOF features at the i^{th} cell. In other words, $\boldsymbol{D}$ calculates the Euclidean distance of features at corresponding cells provided by HOG and HOF. We sample feature distances from training data to calculate means and variances on each feature dimension. Then, we apply feature distance normalization using the means and variances. The learnt weights for $\boldsymbol{\alpha}$ explain how important each patch-based similarity between input and exemplar is in measuring exemplar-input similarity, i.e. the weights specify how discriminative the path-based distances are in exemplar matching process.

3.3.2 Pose Features

A more straight forward way is to parse human poses directly and include pose as one part of features. Introduction of pose features should be beneficial to our model, however, pose estimation in images, specifically for articulated human body is hard since large number of degrees of freedom need to be estimated. Estimation results may be not accurate, even noisy. Therefore, it is still unknown whether pose features will improve our algorithm's performance or not.

Ferrari et al. [11] integrate a group of their works in pose estimation together in a 2D articulated human pose estimation software. Their approach is built on top of the pictorial structure given an initial human body detection. They progressively reduce the search space for body parts, and extend the model by adding priori about the orientation of the torso and head to be near-vertical. We use their software to parse human poses. The pose estimation software returns the joint locations of kinematic body parts (upper-limb, lower-limb, torso, head). A pose representation is required given the parsing results. The absolute positions of joints in bounding boxes are an intuitive way to represent poses. However, it is sensitive to trajectory jitter, which is common in our case. Given two endpoint coordinates of the ith human body parts $\{(x_1^i, y_1^i), (x_2^i, y_2^i)\}, i \in [1, 2, ..., 10]$. We can obtain its center coordinate and denote it as (x^i, y^i). Then we concatenate the Euclidean distance between the center of torso (x^1, y^1) and that of the rest body parts as pose representation:

$$\boldsymbol{o} = [d((x^1, y^1), (x^2, y^2)) \; d((x^1, y^1), (x^3, y^3)) \dots d((x^1, y^1), (x^{10}, y^{10}))] \tag{3.8}$$

$d(\cdot)$ is the function compute the Euclidean distance between two coordinates. In the end, we normalize pose features to zero mean and unit variance on each feature dimension.

Given a set of bounding boxes p_i restricted to N locations near tracker output x at time t, we run our parser for each bounding box to obtain human pose features $\boldsymbol{o}_i$ and corresponding confidence score s_i,

$$[s_i \ \boldsymbol{o}_i] = c(x, t, p_i), \ i \in [1, 2, ..., N] \tag{3.9}$$

We select $\boldsymbol{o}_{i^*}$ – pose features corresponding to the maximum score, to describe human pose in track x at time t.

$$i^* = \arg\max_i s_i, \ i \in [1, 2, ..., N] \tag{3.10}$$

We redefine function $f(x, t, p)$ used in Equation 3.2 to concatenate pose features as well as motion and appearance features.

$$f(x, t, p) = [f_1(x, t, p) \ \ f_2(x, t)] \tag{3.11}$$

where

$$f_2(x, t) = l \cdot \boldsymbol{o}_{i^*} \tag{3.12}$$

$f_1(\cdot)$ calculate HOG and HOF features as described in Section 3.3.1. Since the dimension of HOG and HOF features are much larger than that of pose features, we normalize pose feature so that the summation of pose feature is comparable to that of HOG/HOF features. The constant l is the scale for normalization, it is the ratio of HOG/HOF feature dimension and pose feature dimension. Please refer Chapter 4 for experimental results with pose features.

3.3.3 Selecting Exemplars

Our model requires an exemplar set including various discriminative key poses. Given the tracks of subjects in training sequences we have access to thousands of samples of cropped images of human subjects. We define the distance between samples using function $\boldsymbol{D}(\cdot, \cdot)$. A clustering algorithm such as k-means could be used to extract various human poses from cropped bounding boxes. But naive clustering methods focus on common rather than discriminative poses. In order to get varied, discriminative key poses, we trained a multiclass linear SVM classifier using LIBLINEAR [8] on top of all cropped bounding boxes from different activities. This classifier is used to score the training samples as a measure of how discriminative a sample is. Next, we clustered the samples with highest score using k-means. Note that the k-means centres are virtual poses that does not exist in training samples. We use the nearest samples of the training set to the centres provided by k-means as set of key human pose candidates. This heuristic procedure is efficient and effective in our experiments. Figure 3.4 is an illustration of the exemplars selected from the training data. Other supervised clustering techniques could also be used. Lazebnik and Raginsky [14] use a technique for simultaneously discretizing features and the posterior of their class labels through minimizing information loss, so that the quantized representation retains as much information as possible for correctly classifying the feature.

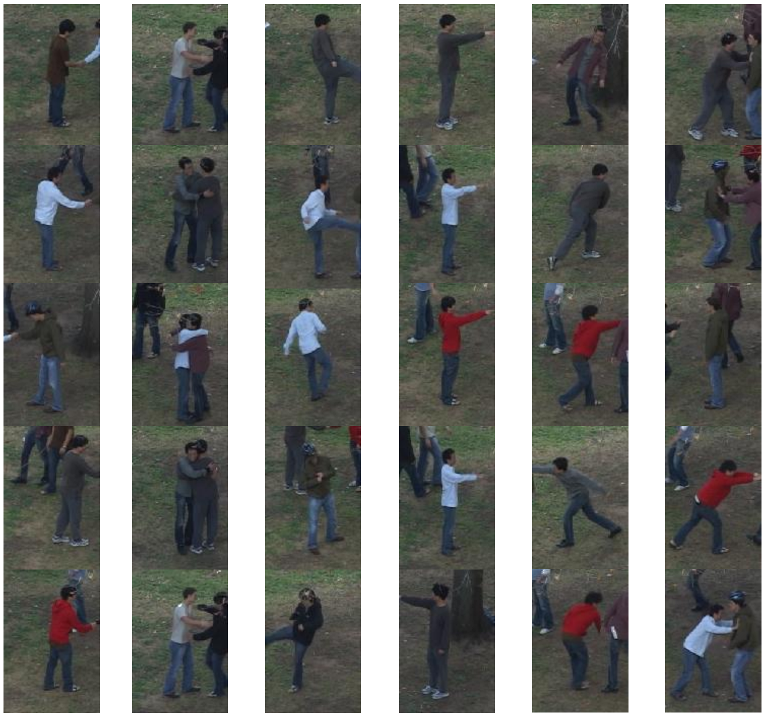


Figure 3.4: Visualization of exemplars selected from UT-Interaction dataset. Each column includes five exemplars for an activity. The activities are *handshake*, *hug*, *kick*, *point*, *punch*, *push*. Large pose variations are presented in selected exemplars for each activity.

3.3.4 Initialization

Parameter initialization is crucial in learning latent variable models. We use the following heuristics to initialize the parameters. In order to initialize $\boldsymbol{\beta}$, which affects the valid key pose sequence, each trajectory in class a is divided into K (number of key poses) equal length, non-overlapping temporal segments. Each frame of a trajectory in the i^{th} segment is matched to its nearest exemplar, and β_{iae} is set to the frequency of matching exemplar e. We initialize $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ as equally distributed values with unit summations.

3.4 Learning and Inference

Given the training set, we need to learn the parameters of the model to be able to find the key poses in a test sequence and recognize its activity class. The learning algorithm we use requires the inference procedure, so we first describe the inference procedure to find the key poses for a sequence, and then explain how we train the parameters of the model.

3.4.1 Inference

Given a video sequence x, model parameters $\boldsymbol{\omega}$, and a hypothesized activity label y, we score the sequence by finding the best sequence of key poses. The activity label for a sequence is the y that maximizes this score. We assume we are given a tracker that produces human trajectories, but we do not know which of these people takes which role in the activity. We define the scoring function $E(x, y)$:

$$E(x, y; \boldsymbol{\omega}) = \max_{b_1, b_2} \max_{\mathbf{H} \in \mathcal{H}_1 \times \mathcal{H}_2} L(x, y, \mathbf{H}; \boldsymbol{\omega}), \tag{3.13}$$

with L being the dual-trajectory scoring function defined in Eq. 3.5. b_1 and b_2 select which person trajectories take the subject and object roles in the interaction. Recall from Sec. 3.1 that key pose sequences are constrained by a chronological ordering. $\mathcal{H}_1$ and $\mathcal{H}_2$ are the sets of chronologically valid keypose sequences for the trajectories corresponding to people b_1 and b_2.

Note that the interaction distance term Q in Eq. 3.6 uses the trajectory of each person, which is provided by the tracker and is fixed. Hence, the maximization in Eq. 3.13 can be decomposed into maximization for each trajectory. So, we can write it as:

$$\begin{aligned}
&\max_{b_1,b_2} \max_{\mathbf{H}\in\mathcal{H}_1\times\mathcal{H}_2} L(x,y,\mathbf{H};\boldsymbol{\omega}) = \\
&\max_{b_1,b_2}\Bigg\{ \underbrace{\max_{\mathbf{H^1}\in\mathcal{H}_1} \left\{G(x,y,\mathbf{H^1};\boldsymbol{\omega}^s) + \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x,y,h_i^1)\right\}}_{\text{subject trajectory}} \\
&\quad \underbrace{+ \max_{\mathbf{H^2}\in\mathcal{H}_2} \left\{G(x,y,\mathbf{H^2};\boldsymbol{\omega}^o) + \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x,y,h_i^2)\right\}}_{\text{object trajectory}} \Bigg\}.
\end{aligned} \tag{3.14}$$

The score maximization for each trajectory consists of finding K key poses, $h_i = (e_i, t_i, p_i), \forall i \in 1, ..., K$ that match to the sequence. However, our model has a chronological ordering constraint on the key poses found in the input sequence, which states $t_1 < t_2 < \cdots < t_K$. The exemplar and spatial perturbation of the key pose are free from this constraint, so we can maximize the score of our model for the i^{th} key pose at frame t over possible exemplars and spatial perturbation:

$$A_i^t = \max_{e_i,p_i} \left\{ \boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x,h_i) + {\boldsymbol{\beta}_i}^T \boldsymbol{\phi}_1(y,h_i) + \boldsymbol{\gamma}^T \boldsymbol{\phi}_2(x,y,h_i) + {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x,y,h_i) \right\}$$

where $t = 1, 2, \cdots, T$, and T is the number of frames in x. Next, considering the constraint, Arash Vahdat proposes an efficient dynamic programming algorithm to solve this maximization. He rewrites the score maximization of a trajectory in Eq. 3.14 as:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{K} A_i^{t_i} \\
\text{s.t.} \quad & t_i < t_{i+1} \;\; \forall i = 1, 2, \cdots, K-1
\end{aligned} \tag{3.15}$$

He defines M_j^τ as the best score using j elements of A until the τ^{th} frame:

$$\begin{aligned}
M_j^\tau &= \max \sum_{i=1}^{j} A_i^{t_i} \\
\text{s.t.} \;\; & 1 \le t_i < t_{i+1} \le \tau \;\; \forall\, i = 1, 2, \cdots, j-1
\end{aligned} \tag{3.16}$$

Writes M_j^τ as a recursive function:

$$\begin{aligned}
M_j^\tau &= \max\{M_{j-1}^{\tau-1} + A_j^\tau, M_j^{\tau-1}\} && 1 < j \le K, j < \tau \le T \\
M_j^j &= M_{j-1}^{j-1} + A_j^j && 1 < j \le K \\
M_1^\tau &= \max\{A_1^1, A_1^2, \cdots, A_1^\tau\} && 1 \le \tau \le T
\end{aligned} \tag{3.17}$$

The optimal solution of Eq. 3.15 is M_K^T, and can be calculated in time $O(KT)$, the number of keyposes multiplied by the number of frames in the video sequence.

3.4.2 Learning

We use $y^* = \arg\max_y E(x, y; \boldsymbol{\omega})$ as the predicted label of x. Given $\{(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)\}$, the set of training data, we aim to find parameters that score x^i and y^i higher than other activity types. Similar to Felzenszwalb et al. [10] and Wang and Mori [34] we formulate the training criteria in the Max-Margin framework. We set $\boldsymbol{\omega}$ by:

$$\begin{aligned} &\min_{\boldsymbol{\omega},\xi^i} \frac{\lambda}{2}\|\boldsymbol{\omega}\|^2 + \sum_i^n \xi^i \\ \text{s.t.} \quad & E(x^i, y^i; \boldsymbol{\omega}) - E(x^i, y; \boldsymbol{\omega}) > \Delta(y^i, y) - \xi^i \quad \forall i, \forall y \in \mathcal{Y} \end{aligned} \tag{3.18}$$

where λ is a tradeoff constant and $\Delta(y^i, y)$ is 0-1 loss.

The constraint in Eq. 3.18 forces the score of the true labeling for each training sequence to be higher than the best score for an incorrect hypothesized label. The optimization problem in Eq. 3.18 is a non-convex optimization problem and we use the non-convex extension of the cutting plane algorithm using NRBM [6] to learn the parameters.

3.5 Summary

In this chapter, we build an exemplar-based key pose model for single subject at the beginning, then enrich the model for human interaction recognition task. We infer key poses in each sequence using an efficient dynamic programming algorithm, and learn weights using NRBM. We describe features used, the ways to select exemplar set and how to initialize model parameters. We would like to demonstrate our model's power by showing recognition accuracy improvement on both benchmark choreographed dataset and surveillance videos, visualize weights learnt by our model and explain their meanings.

Chapter 4

Experiments

We consider two datasets to gauge our model's effectiveness in classifying human interactions. First, we test our model on the UT-Interaction dataset [22], a publicly available benchmark with comparative results. Second, we construct a dataset for recognizing *embrace* interactions by selecting a subset of the TRECVID 2008 Surveillance Event Detection challenge [30] and demonstrate our model on a non-choreographed dataset. See Fig. 4.1 for example frames from the UT-Interaction dataset and TRECVID embrace datasets.

4.1 UT-Interaction Dataset

The UT-Interaction dataset contains videos of 6 classes of human-human interactions: *handshake*, *hug*, *kick*, *point*, *punch*, and *push*. There are 20 video sequences in total. Each video contains at least one execution per interaction, providing 8 executions of human activities per video on average. The dataset is divided into two sets. Set 1 is recorded in a parking lot with a stationary background and set 2 is recorded on a lawn with slight background movement and camera jitter. Ground truth labels for these interactions are provided, including time intervals and bounding boxes. Note that for the *point* activity, the ground truth in the UT-Interaction dataset only contains the person performing the activity without the other one being pointed at. We decide to search horizontally for a person nearest to the one performing the point activity and include him as the other part of the activity. We follow the experimental setting of the classification task described in the High-level Human Interaction Recognition Challenge [22] – bounding boxes are used as input and the performance of our model is evaluated using leave-one-out cross validation on each set. Note that no additional

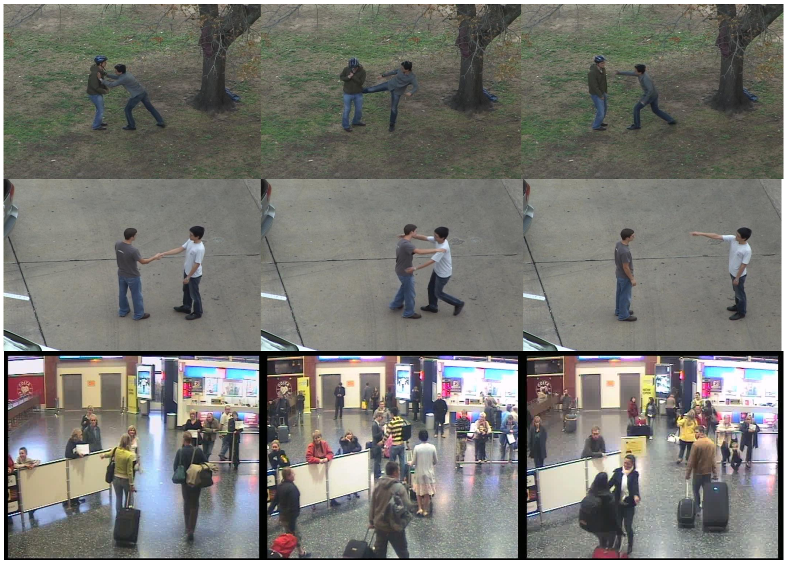


Figure 4.1: Representative frames from UT-Interaction dataset and TRECVID embrace dataset. The first two rows are sample frame of six actions in UT-Interaction dataset and the third row shows sampled embrace event come from our embrace dataset.

information is used – in particular roles in the interaction (b variables) are inferred both in learning and test time.

4.1.1 Implementation Details

The bounding boxes provided as input contain the two humans performing an interaction, not tracks of individuals. We employ a pedestrian detector [4] to obtain initial positions of the people in the first frame of every video clip. Figure 4.2 is some detections given by our human detector. We select a pair of detections with the minimum horizontal distance out of the three highest scoring detections, then run a tracker [1] to find trajectories of two individuals interacting with each other in the subsequent frames. Figure 4.3 is an example of our tracker's output given human detection in the first frame. To handle tracker jitter, we allow key pose positions to have freedom to perturb around the tracker output. We use a 20 pixel step size and allow up to 1 step horizontally, a 15 pixel step size and allow up to 1 step vertically to locate p, the position of key pose in the track, so p has 9 positions to enumerate in our case. In UT-Interaction dataset, actors's movement occur in horizontal direction, so our horizontal perturbation step is larger than vertical one. Considering camera

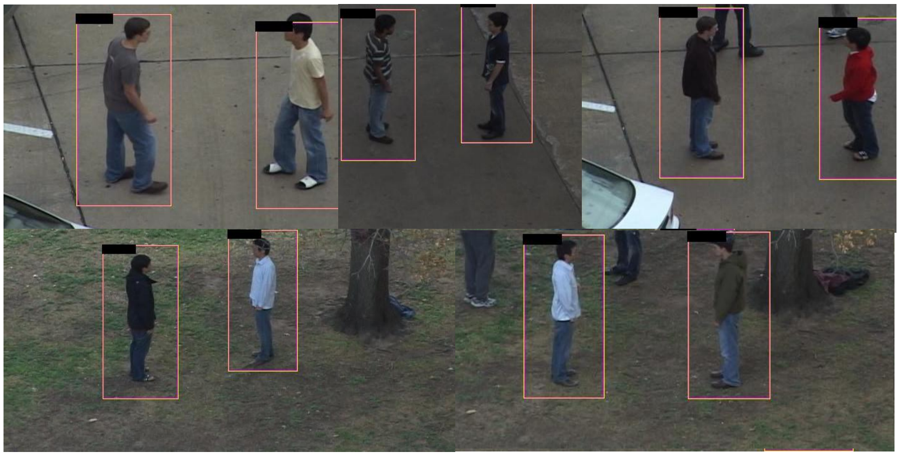


Figure 4.2: Sample of Human detection results of the first frames of video clips in UT-Interaction dataset.

zoom in Set 1, we also perform multi-scale search at 2 scales. In multi-scale searching, we use different bounding box sizes to crop images and resize them to standard size, compute potential scores on different scales for the maximum. Our model can extend for multi-scale search with linear computational time increase.

4.1.2 Results

Confusion matrices of the two sets in the UT-Interaction dataset are shown in Fig. 4.4. The figure shows some confusion between the activities *push* and *punch* on Set 2. This is consistent with the fact that pushing and punching are similar in both appearance and motion. Comparisons with other approaches are summarized in Table 4.1. Two methods for activity recognition on UT-Interaction dataset are available, they are proposed by Yao et al. [39] and Yu et al. [40]. Yao et al. train random trees to learn a mapping between sampled feature patches and votes in a Hough space. Leaves of trees are learnt to be a discriminative codebook and vote activity centres with probability. In testing, randomly extracted patches are used to pass through each of the trees in the forest, the leaves that the patches arrive in are used to cast votes. Yu et al. use semantic texton forests to convert local space-time patches to discriminative codewords. To capture the structural information of actions, pyramidal spatio-temporal relationship match is introduced. A direct comparison is possible, and our methods clearly outperform their methods.

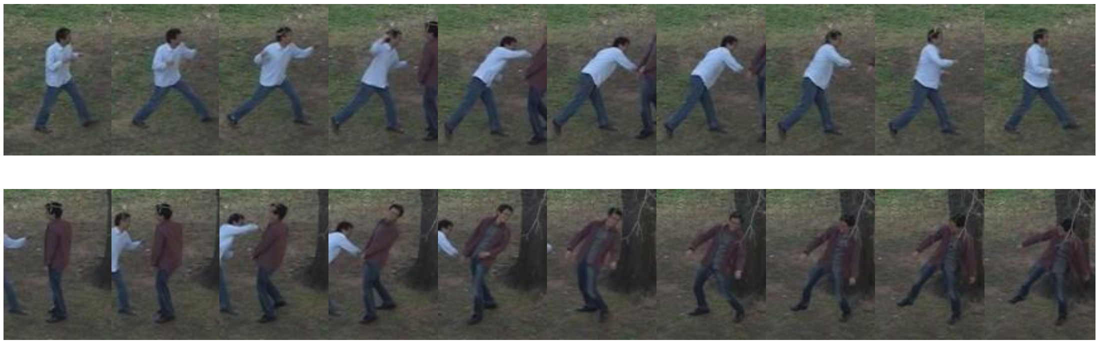
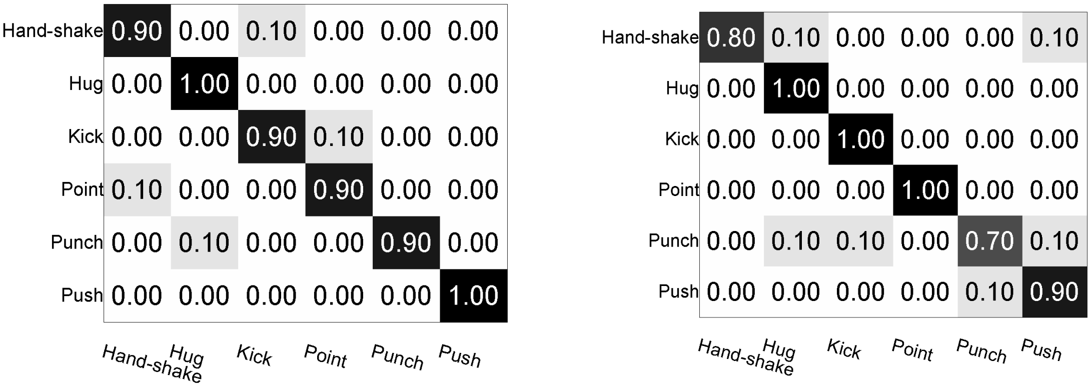


Figure 4.3: Sample trajectories extracted from a 63 frame–long video. The trajectories are the output of our tracker given the human detection in the first frame of the video clip.

Table 4.1: Comparison of per-clip classification accuracy with other approaches on UT-interaction dataset.

Method	Set 1	Set 2	Avg
Our method	**0.93**	**0.90**	**0.92**
Yu et al. [40]	N/A	N/A	0.83
Yao et al. [39]	0.88	0.80	0.84



	Hand-shake	Hug	Kick	Point	Punch	Push
Hand-shake	0.90	0.00	0.10	0.00	0.00	0.00
Hug	0.00	1.00	0.00	0.00	0.00	0.00
Kick	0.00	0.00	0.90	0.10	0.00	0.00
Point	0.10	0.00	0.00	0.90	0.00	0.00
Punch	0.00	0.10	0.00	0.00	0.90	0.00
Push	0.00	0.00	0.00	0.00	0.00	1.00

(a) Set 1

	Hand-shake	Hug	Kick	Point	Punch	Push
Hand-shake	0.80	0.10	0.00	0.00	0.00	0.10
Hug	0.00	1.00	0.00	0.00	0.00	0.00
Kick	0.00	0.00	1.00	0.00	0.00	0.00
Point	0.00	0.00	0.00	1.00	0.00	0.00
Punch	0.00	0.10	0.10	0.00	0.70	0.10
Push	0.00	0.00	0.00	0.00	0.10	0.90

(b) Set 2

Figure 4.4: Confusion matrices of per-clip classification result on UT-Interaction dataset. Horizontal rows are ground truth and vertical columns are predictions.

We also test our algorithm's performance with pose features. First, we visualize some results given by pose parser in Fig. 4.5.

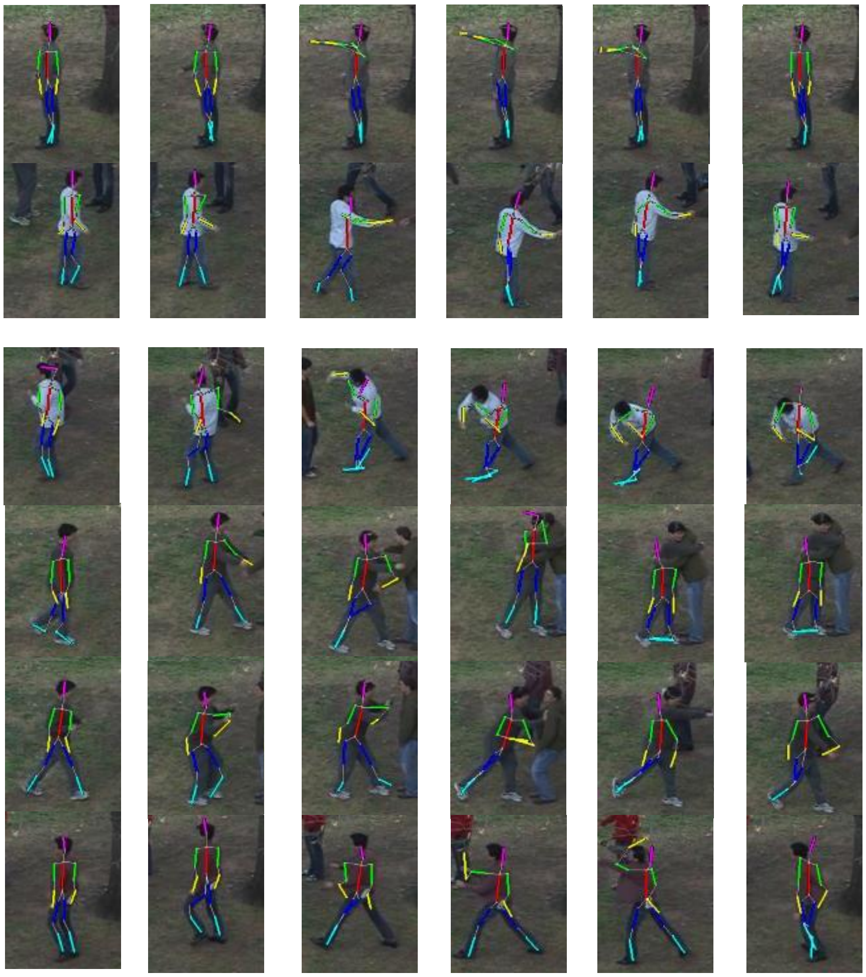


Figure 4.5: Parsing results for different activities in UT-Interaction dataset. The parser performs reasonably for activities like *point* and *handshake*. The upper part of the figure shows some good parsing results on the activities. However, it has difficulty in parsing extreme poses in activities like *hug*, *push*. Those pose estimation results are listed in the lower part of the figure.

Since videos in Set 2 of UT-Interaction dataset have little scale difference, we choose to test our pose features on it as the first step. After adding pose features, we see our algorithm's performance drop slightly, please refer to Table 4.2.

Table 4.2: Experimental result after adding pose features on Set 2 of UT-Interaction dataset.

Method	Set 2
HOG + HOF	**0.90**
HOG + HOF+ Pose	0.87

The main reason that more confusion present can be inaccurate pose estimation. The pose parser performs reasonably at the beginning and the end of video sequences when actors maintain upright poses, but when extreme poses appear, such as *punching* and *pushing*, pose estimation fails. The failure is probably due to the fact that extreme poses are hard to parse in nature, and the situation becomes worse when near-vertical human torso orientation is added as priori. Extreme poses that our estimator fails to parse are those discriminative poses, which should be selected as key poses. The inaccurate pose estimation mingles noisy data with discriminative information, should be responsible for the drop of classification accuracy.

"Poselet" can be another direction to utilize pose information. This notation of poselet is proposed by Bourdev and Malik [3] and used to denote a set of patches with similar pose configurations. It is an exemplar-based pose representation. Given 2D images and their joints labels, each poselet provides examples to train a classifier which can then slide over entire images for detections. Instead of trying to infer poses correctly using a pose estimation algorithm, poses are treated as latent variables in a model for action recognition task in the paper by Yang et al. [38]. They manually label human joints, train poselet classifiers for different body parts. Part configurations are treated as latent variables, they connect different human body parts represented by poselets in a tree structure for action recognition. We can combine patch-based matching with their scheme for better exemplar matching accuracy. Given their good experimental results, we believe it is a promising direction worth exploring.

4.1.3 Visualization of Model Weights

In this section we provide visualization of portions of our model to understand what it has learned. We visualize the exemplar matching model to demonstrate that our model is able to localize key poses in the trajectory and fire on discriminative patches for pose. Figure 4.6 shows our exemplar-matching model. We show weights between exemplars and activity labels to show our model can handle pose variation via the exemplar representation. Figure 4.7 visualizes our learned activity-exemplar weights. We visualize the weights for distances between the localization of key poses in each trajectory to illustrate the contribution of spatial constraints. The first bin (bin 1) is assigned to distance smaller than a threshold, and the last bin (bin 5) is assigned to all distances larger than the maximum step size. Figure 4.8 shows the learned spatial distance weights.

4.2 TRECVID Embrace Dataset

We collected a subset from the development dataset of the TRECVID 2008 Surveillance Event Detection challenge [30] for the embrace event classification task. Our goal is to examine performance on non-choreographed activities. The full TRECVID dataset is very challenging, and state-of-the-art methods perform poorly on it. Considering the fact that human detectors and trackers have difficulty in challenging datasets like TRECVID, we manually select a subset of the dataset on which the detector/tracker perform well. This subset will certainly be easier than the full dataset, but it can be argued that with a better detector/tracker, performance should improve.

We choose five days of video recorded in 2007 from camera view 3. We manually select a positive set of 36 *embrace* clips where our detector and tracker provide reasonable output, from all 343 *embrace* clips. We randomly sample 300 video clips that do not temporally overlap with the *embrace* events using the same human detector and tracker used for positive examples to obtain trajectories. We select from this a negative set of 108 pairs of trajectories that overlap in space.

The TRECVID Event Annotation Guidelines states that embrace starts at the lastest time when subjects do not have physical contact prior to the embrace. However, we believe important and discriminative information is also present in frames before people have physical contact. For example, pairs of people with both arms outstretched strongly indicates the upcoming embrace event. So we decide to label the starting frame of *embrace* 20 frames earlier that the TRECVID ground truth. We also fix the length of *embrace* activity event as 60 frames for both positive and negative samples. Note that the negative class come from videos randomly sampled in time, hence is a fair comparison to non-embrace videos but our dataset lacks the "near"-*embrace* events that would require non-maximum suppression. Our embrace dataset excludes groups hugging and other serious occlusions in which case

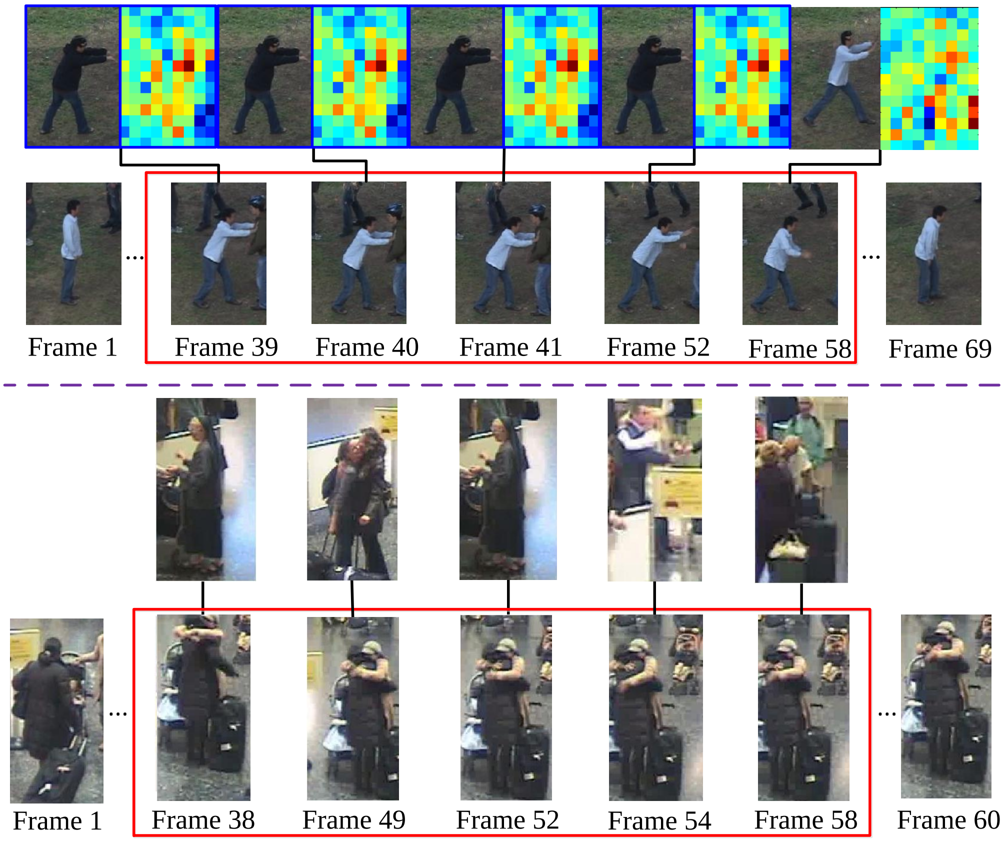


Figure 4.6: Discriminative frames of a trajectory are automatically extracted. Separated by a dashed line, the upper part of the figure comes from the UT-Interaction dataset and the lower part from the TRECVID embrace dataset. The localizations of key poses in trajectories are highlighted by red bounding boxes. In the upper part, our model localizes 5 key poses in a 69-frame long trajectory and selects exemplars for each of them. The frame number under each key pose localization indicates its time in the trajectory. Exemplars are selected based on similarity in appearance and localization of key pose. The similarity is defined as patch-weighted distance. The model learns to give high weights on patches where poses appear to be unique. Patch-based weights are shown beside each exemplar. The weights spread over the contour of each individual and concentrate on outstretched arms for the push activity. Similar visualizations are shown in the lower part for a trajectory from the TRECVID embrace dataset.

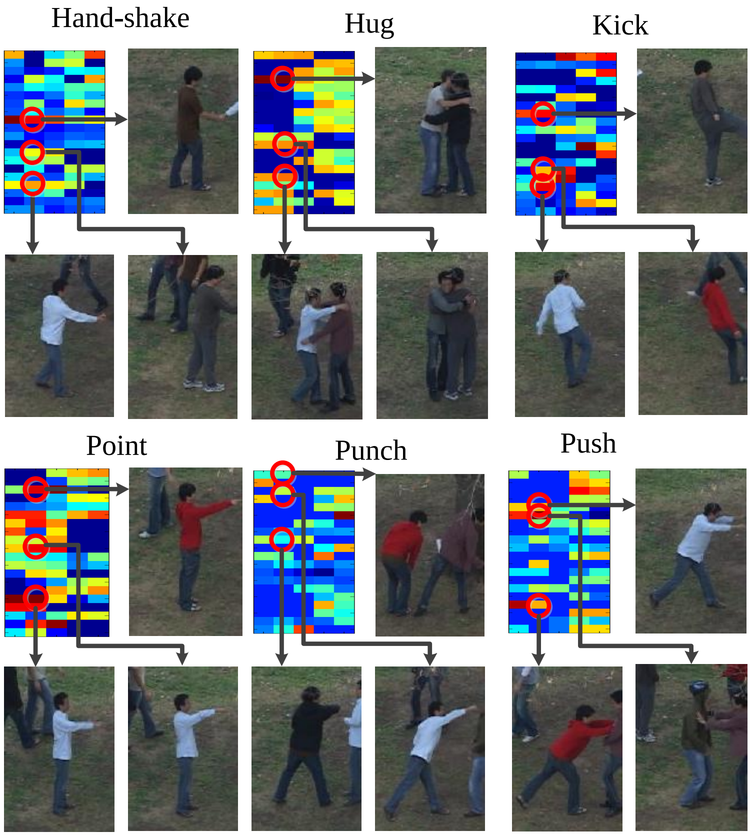


Figure 4.7: Visualization of exemplar indicator model for one trajectory. For the heatmap of each activity, the horizontal axis is the concatenation of the 5 key poses in the activity and the vertical axis specifies 20 exemplars belong to the activity. Each pixel describes the score for an exemplar being matched to a key pose in the activity. The weights represent our model's preference for an exemplar in a key pose. For the second key pose in each activity, we also visualize the exemplars with highest weights. For each activity, selected exemplars have large pose variation.

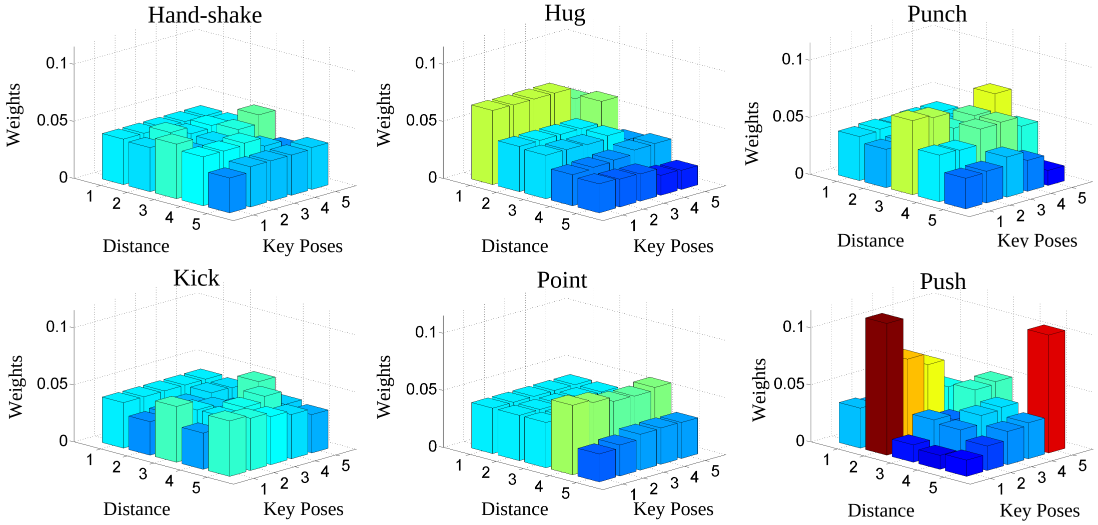


Figure 4.8: Spatial distance model for all six activities in UT-Interaction dataset. Three axes are discrete distance, key poses and weights. For a key pose in each activity, the heights of bars indicate our model's preference among different distances. Bars are also coloured according to height. The spatial distances in the *hug* activity are preferred to be smaller than that in the *point* activity, which illustrates the fact that people are closer to each other in hugging compared with pointing. For the *push* activity, the spatial distance preferred by the last key pose is much greater than previous ones, reflecting the separation of the two individuals at the end of the activity.

one can barely see embrace event. However, the dataset still inherit the characteristics of TRECVID videos, it contains large activity variety on a cluttered background, which make it challenging. The precise dataset will be available for download at our website.

4.2.1 Preprocessing

Our dataset is created by collecting a set of trajectories from the TRECVID dataset. There are 539 *embrace* activities occurs in the TRECVID development dataset. We use those captured by camera 3 for our new dataset, which include around 63% of all the *embrace* activities in the whole dataset. We fix the length of clips to 60 frames for negative examples considering the average time people take to embrace. To avoid potential bias introduced by difference between the length of negative examples and positive examples in classification, we also fix the clip length for positive examples to be 60 frames, filter out those clips whose length is shorter and crop those clips whose length is longer. It left with us 260 clips. Then we run an SVM human detector on the first frames of the clips, the number of clips with both interacting individuals detected reduce to 73. Figure 4.9 is a demo of the output of human detector. We manually select out interacting individuals from the detections given by the human detector. In the end, we track individuals and successfully obtain 36 pairs of trajectories as positive examples. Figure 4.10 is an example of the trajectories extracted from positive examples.

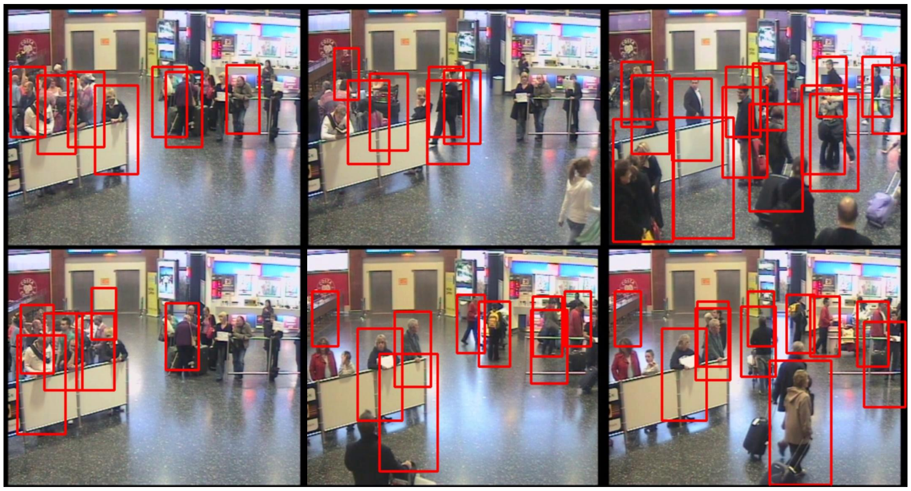


Figure 4.9: Human detection on the first frame of video clips in TRECVID embrace dataset.

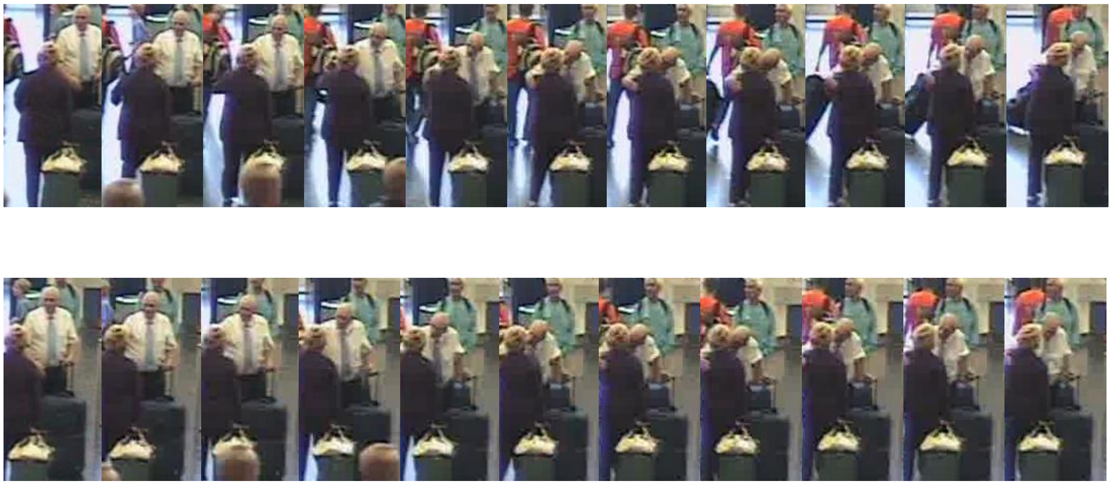


Figure 4.10: Sample trajectories extracted from a TRECVID embrace video. The trajectories are the output of our tracker given the human detection in the first frame of the video clip. Embracing people are manually selected from the output of human detector.

As noted above, a negative set is created by randomly sampling pairs of trajectories that overlap spatio-temporally. We detect people using a combination of motion and HOG human detection. We use the tracker proposed by Babenko et al. [1] to obtain trajectories of individuals. The positive set is acquired using the TRECVID ground-truth, adding an annotation of which two people in a scene are embracing. The task is now a classification task – given a pair of trajectories, is there an *embrace* activity occurring or not.

4.2.2 Results

We evaluated our method using 6-fold cross-validation on the dataset. To evaluate the effectiveness of different parts of our model, we introduce two baseline methods to compare with our full model. The first baseline is our full model without the root model, the direct link between key poses and activity labels. The second baseline is our full model without the spatial distance model, the link between localizations of key poses in one trajectory and poses in the other trajectory simultaneously.

We produce a ROC curve to measure the performance of our algorithm, it is shown in Figure 4.11. Even though we try our algorithm on the *embrace* dataset, we still can interpret performance of our method on the complete TRECVID dataset. Camera view 3 captures

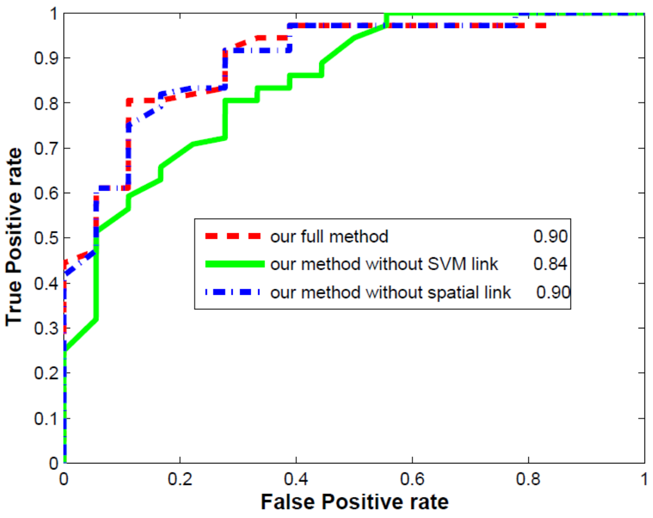


Figure 4.11: ROC curve on TRECVID *embrace* dataset. The 6% increase in Area Under ROC (AUR) from the first baseline to our full model reflects the contribution of the root model to our full model. Since we only select trajectories that overlap spatio-temporally for negative examples, which restrain the benefits of spatial distance model, one can expect the AUR of models with and without spatial distance have little difference.

the majority of *embrace* events. In the worst case, if we misclassify all the rest positive examples, maximum achievable true positive rate (TPR) in ROC drops to 63%. Due to the failure of human detector, tracker and ignorance of short positive samples, our TPR will at most decrease to 10% × 63% of our reported TPR. However, our negative examples are randomly selected pairs of trajectories which overlap in space and time, they are much harder comparing to most of the negative examples in the complete TRECVID dataset. The experimental result on our dataset indicate promising performance on the full TRECVID dataset.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The main contribution of this thesis is developing an exemplar-based key pose sequence model for recognizing human interactions. To tackle the problem of interaction recognition, we propose to use a sequence of key poses to represent an activity. We use heuristics to extract exemplars from training data. We represent key poses by exemplars to handle human pose variations. We define the matching distance between input and exemplar as a weighted sum of patch-based distance in HOG and HOF feature space. Spatial arrangements between interacting people are modelled. We train our model in max-margin hidden conditional random filed (MMHCRF) [34] and use non-convex regularized bundle method (NRBM) [**?**] to learn the weights. Quantitative results that form a new state-of-the-art on the benchmark UT-Interaction dataset [23] are presented. Experiments on a subset of the TRECVID dataset [30] also demonstrate the potency of our model.

5.2 Future Work

The work in this dissertation is illustrated to form state-of-the-art classification performance in the benchmark UT-Interaction dataset, and quantitative results on TRECVID embrace dataset also demonstrate its effectiveness. Seeking for better solutions for subtasks in our model uncovers lots of interesting research directions in the future.

Our model is built based on the assumption that a human tracker is able to provide approximately accurate human trajectories in video clips. Although slide around trajectories and employ key pose representation for robustness to trajectory jitter, our algorithm still heavily relies on the performance of human tracker. However, human tracking is challenging, state-of-the-art methods available in the field are not always reliable, which could be a bottleneck of our model. Examining the addition of tracking as a latent variable could alleviate this direct prerequisite. Inspired by our perturbing operation around humans in trajectory, we can go one step further – use sliding window approach to search entire frames for best matches. We can add human tracking into our model to solve both human tracking and classification task in a joint way. Designing a smart sliding window approach with acceptable computational cost and treating human trajectory as latent variables is an interesting direction for further research.

Input-exemplar matching scheme is heavily used in searching for best exemplars in videos. Nevertheless computing input-exemplar similarity for each exemplar is time consuming. We need to maintain a large number of exemplars for accurate matching since exemplar representation is used to cover human pose variations. Considering both computation cost and matching accuracy, we believe there is a reason to organize exemplars according to pose similarity in a data structure (e.g. tree or hash table) suitable for fast matching. Reducing our parameter space would also be possible via sharing weights via a tree structure on exemplars. Exemplar methods are successful, but require enormous numbers of exemplars. Scaling our method would be interesting.

We use patch-based Euclidean distance in feature space to measure similarity between input and exemplar. The patch-based matching scheme measures global similarity with tolerance to local appearance and motion deformation. However, the similarity measurement does not model discriminative pose information in images. One can directly use available pose estimation algorithms to extract human poses and include pose as part of features for better matching. An alternative way is to treat human poses as latent variables, infer the best poselet configurations, and define input-exemplar distance based on poselet configuration similarity. Both of the methods directly utilize human pose similarity in exemplar matching. Pose information seems to be quite different from our HOG and HOF features, so we expect the new approaches will contribute to a better recognition performance.

Our model is suitable for interaction recognition between two people, and could be easily simplified for single subject activity recognition. Group interaction with more than two people get involved always can be factored into interaction pairs. Therefore our model has an good generality. It is interesting to extend our model for multiple activity detection in a video. Modifying our model for more complex yet real-world scenario (e.g. group interaction recognition, multiple activity detection) holds significant meanings, we believe it would be worth exploring.

Our interaction model goes beyond individual activity recognition. It not only develops an effective way to model individual activity but also utilize human interactions to help recognition. We believe the model opens a door to many interesting research problems and holds promise in improving interaction recognition and related vision applications.

Bibliography

[1] B. Babenko, Ming-Hsuan Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

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

[3] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, sep 2009.

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

[5] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *In European Conference on Computer Vision*. Springer, 2006.

[6] Trinh-Minh-Tri Do and Thierry Artières. Large margin training for hidden markov models with partially observed states. In *ICML*, 2009.

[7] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *J. of Machine Learning Research*, 9:1871–1874, 2008.

[9] A. Fathi and G. Mori. Action recognition using mid-level motion features. In *CVPR*, 2008.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. PAMI*, 32(9), 2009.

[11] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, June 2008.

[12] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.

[13] Thorsten Joachims. A support vector method for multivariate performance measures. In *ICML '05*, pages 377–384, New York, NY, 2005.

[14] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. PAMI*, 31(7):1294–1309, 2009.

[15] Zhe Lin, Zhuolin Jiang, and Larry S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, April 1981.

[17] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268, 2001.

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

[19] Juan Carlos Niebles, Chih-Wei Chen, , and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[20] Abhijit S. Ogale, Alap Karapurkar, Gutemberg Guerra-filho, and Yiannis Aloimonos. View-invariant identification of pose sequences for action recognition. In *In VACE*, 2004.

[21] David Ross, Jongwoo Lim, and Ruei-Sung Lin. Incremental learning for robust visual tracking. *IJCV*, May 2008.

[22] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[23] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[24] Scott Satkin and Martial Hebert. Modeling the temporal extent of actions. In *Proc. 10th Europ. Conf. Comput. Vision*, 2010.

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

[26] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *17th International Conference on Pattern Recognition*, 2004.

[27] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. 9th Int. Conf. Computer Vision*, volume 2, pages 750–757, 2003.

[28] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR*, 2005.

[29] Qinfeng Shi, Li Cheng, Li Wang, and Alex Smola. Human action segmentation and recognition using discriminative semi-markov models. *Int. Journal of Computer Vision*, 2010.

[30] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR*, 2006.

[31] Paul A. Viola, John C. Platt, and Cha Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005.

[32] Daniel Waltisberg, Angela Yao, Juergen Gall, and Luc Van Gool. Variations of a hough-voting action recognition system. In *Proceedings of ICPR 2010 Contests*, 2011.

[33] Yang Wang and Greg Mori. Human action recognition by semi-latent topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1762–1774, 2009.

[34] Yang Wang and Greg Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009.

[35] Daniel Weinland and Edmond Boyer. Action recognition using exemplar-based embedding. In *CVPR*, 2008.

[36] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 2010.

[37] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, 1992.

[38] Weilong Yang, Yang Wang, and Greg Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.

[39] A. Yao, J. Gall, and L.J. Van Gool. A hough transform-based voting framework for action recognition. In *CVPR*, 2010.

[40] Tsz-Ho Yu, Tae-Kyun Kim, and Roberto Cipolla. Real-time action recognition by spatiotemporal semantic and structural forest. In *BMVC*, 2010.