# A Discriminative Key Pose Sequence Model for Recognizing Human Interactions

Arash Vahdat*, Bo Gao*, Mani Ranjbar, and Greg Mori
School of Computing Science
Simon Fraser University
{avahdat, bog, mra33, mori}@sfu.ca

## Abstract

*In this paper we develop a model for recognizing human interactions – activity recognition with multiple actors. An activity is modeled with a sequence of key poses, important atomic-level actions performed by the actors. Spatial arrangements between the actors are included in the model, as is a strict temporal ordering of the key poses. An exemplar representation is used to model the variability in the instantiation of key poses. Quantitative results that form a new state-of-the-art on the benchmark UT-Interaction dataset are presented, along with results on a subset of the TRECVID dataset.*

## 1. Introduction

Computer vision-based analysis of human movement is a broad, active area of research. In this paper we focus on recognizing interactions between individuals. We describe an algorithm for recognizing *activity*-level interactions, those composed of *atomic*-level poses or movements. Examples of such activities are people embracing, shaking hands, and pushing each other.

The desiderata for models of such activities include a number of key points. Similar to any activity recognition problem, robustness to the intra-class variation in the atomic poses or movements comprising the activity is required. Further, tolerance of extraneous atomic poses or movements by the actors performing an activity is required – not every movement or pose by an actor is relevant to the activity to be recognized. Finally, spatial and temporal relations between the atomic actions should be enforced – order matters.

The approach we take in this paper models an activity with a set of key poses of the individuals involved. A high-level depiction of the model is shown in Fig. 1. We use an exemplar-based model of the instantiation of these key poses. Spatial and temporal relations among the locations of these key poses are formulated in the model. In contrast with many exemplar-based methods, our model provides temporal decomposition and a sparse key pose representation that can make it more scalable and robust against intra-class variation.

* indicates equal contribution.

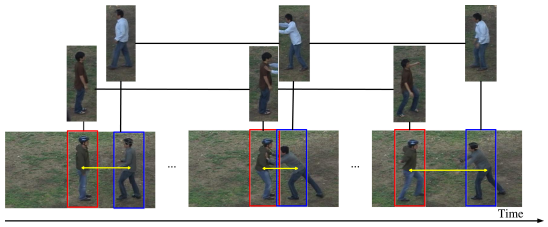



Figure 1. High level depiction of our model. Horizontal axis represents time. Localizations of key poses are highlighted in red bounding boxes, exemplars are matched correspondingly. Spatial distances are marked by double-headed arrows in yellow.

The main contribution of this paper is the development of this model. We argue that it holds aforementioned key properties such as robustness to atomic-level variation and enforces spatial and temporal constraints. We show empirically that it is effective, outperforming state-of-art methods on the UT-Interaction (SDHA) dataset and obtaining promising results on a subset of the TRECVID dataset.

The activity recognition literature in computer vision is immense. Weinland et al. [20] provide a recent survey, we review closely related methods below. In this paper we develop a temporal model consisting of key poses. A variety of temporal models have been developed in the literature, ranging from template matching to probabilistic temporal sequence models. Template matching methods include Efros et al. [4] using features derived from blurred optical flow estimates and Lin et al. [9] matching to prototypes based on shape and motion similarity. Temporal smoothing [4] and dynamic time warping [9] are used to account for variation in speed. These methods seem effective for recognizing atomic actions, but it is unclear that they are flexible enough to handle the temporal variation in longer activities. Methods that ignore explicit temporal modeling also exist, such as a best match to exemplars [19] or frequency of exemplars [17] over a sequence. Again, for activities, especially with interactions, a more explicit temporal model seems required. A variety of probabilistic temporal models has been deployed for this purpose. Generative probabilistic models have been deployed for activity recognition – Hidden Markov Models by Yamato et al. [22]

and Dynamic Bayesian Networks by Xiang and Gong [21]. In contrast, our work uses a discriminative framework focusing on the important parts of an activity. Shi et al. [14] use discriminative semi-Markov models for labeling each frame of an input sequence with an action label. Our work recognizes higher-level activities and only focuses on the relevant atomic actions rather than labeling each frame.

Other key pose approaches exist in the literature. Sullivan and Carlsson [16] perform shape matching, but for retrieving a single pose rather than an activity sequence. Niebles et al. [11] develop a similar model to ours, but for recognizing single person activities, without hard temporal ordering, without the non-parametric exemplar matching.

Much work focuses on frame-level atomic action recognition of individuals. Interactions were considered by Intille and Bobick [7] who used probabilistic techniques for recognizing hand-specified structured activities such as American football plays. Medioni et al. [10] developed a system for recognizing events from aerial video surveillance data, for instance interactions between vehicles and road checkpoints. In our experiments we use the UT-Interaction (SDHA) dataset introduced by Ryoo and Aggarwal [13]. Ryoo and Aggarwal develop a matching kernel that considers spatial and temporal relations between space-time interest points. Yao et al. [23] use a Hough transform voting scheme from an interest point representation. Yu et al. [24] develop an efficient algorithm using semantic texton forests with a pyramidal version of Ryoo and Aggarwal's matching kernel. These methods have less explicit modeling of the presence of individuals than our method, and our method obtains higher accuracy on the UT-Interaction dataset.

## 2. Modeling Human Interactions

Our goal in this paper is to recognize human interactions in videos. The interactions we consider are activities such as pushing, handshaking or hugging that involve two people interacting.

We will model these interactions by a sequence of key poses. For example, as shown in Fig. 1 a common scenario for pushing is the following: one person steps forward, raises his hands, and pushes the other person while he takes a defensive pose, steps backward, and falls back at the end. Similarly, we can decompose other interaction scenarios to sequences 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 $\boldsymbol{H}$: $F(x, y) = \max_H G(x, y, \boldsymbol{H})$.

In our work, the latent variables $\boldsymbol{H}$ answer the four questions above. Namely, $\boldsymbol{H} = [b, t, e, p]$, 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. More generally, one could build $b$ from tracklets, or infer it while tracking.

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.

In the following sections we provide more precise details on these latent variables and the scoring function $G$. For ease of exposition, we start with a single subject key pose sequence model (Sec. 3), followed by a model for interactions (Sec. 4). We do not assume the key poses are provided as training data, and instead aim to automatically discover them. Algorithms for this learning, and associated inference, are in Sec. 5.

## 3. Single Subject Key Pose Sequence Model

We start by describing a model of videos of a single person performing an activity. Given a set of such videos, we

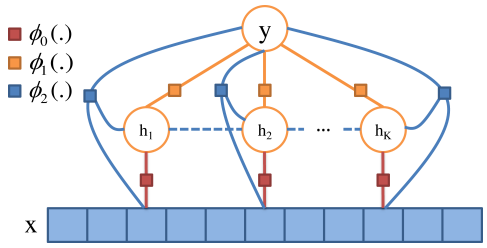



Figure 2. Graphical depiction of our model for single subject key pose sequence matching. The lower layer $x$ is the observed sequence of frames, and the middle layer $h$ is the key pose sequence layer and the top layer $y$ is the activity label. Edges with boxes denote factors in our model. Dashed lines represent temporal constraints between key poses.

want to find a set of key poses in these sequences and use them to model the activity class. Key poses are meant to be important, infrequent actions; much of each video can consist of highly variable human action that can be misleading when attempting to build an activity model. Considering our *pushing* example, there are poses such as standing or walking at the beginning or the end of the video that are variable and not discriminative. Further, each of the key poses will have variation in appearance. Finally, the spatial arrangement of these key poses is important (particularly for interactions), so the model will also include what spatial location in a video frame contains the key pose.

An instantiation of a single subject key pose model in a video sequence will consist of three things: **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 $\boldsymbol{h} = [e, t, p]$. Variables $t$ and $p$ are its spatio-temporal locations, with $p$ restricted to locations near the tracker output. The variable $e = 1, 2, \ldots, |\mathcal{E}|$ denotes which appearance variant of the key pose is taking place at time $t$ and location $p$ where $\mathcal{E}$ is a discrete set of exemplars 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 $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, .., \boldsymbol{h}_K]$, where each $\boldsymbol{h}_i$ is a triple $[e_i, t_i, p_i]$. Our model also has a constraint on the temporal component of the key poses $\boldsymbol{H}$. 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 $G(x, y, \boldsymbol{H})$ for a single subject model. A graphical depiction of our model is shown in Fig. 2. 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 $\boldsymbol{H}$ is scored for the input $x$ and the label $y$ by $G(x, y, \boldsymbol{H}) = \boldsymbol{\omega}^T \boldsymbol{\Phi}(x, y, \boldsymbol{H})$ which is a linear function on $\boldsymbol{\omega}$, the parameters of the model. We formulate the scoring function as:

$$\boldsymbol{\omega}^T \boldsymbol{\Phi}(x, y, \boldsymbol{H}) = \sum_{i=1}^{K} \boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, t_i, e_i, p_i) + \sum_{i=1}^{K} {\boldsymbol{\beta}_i}^T \boldsymbol{\phi}_1(y, e_i) + \sum_{i=1}^{K} \boldsymbol{\gamma}^T \boldsymbol{\phi}_2(x, y, t_i, p_i) \quad (1)$$

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:** $\boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, t_i, e_i, p_i)$ measures the compatibility between key pose with appearance of the ${e_i}^{th}$ exemplar and the image evidence of one track at time $t_i$ and location $p_i$. It is formulated as:

$$\boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, t_i, e_i, p_i) = \sum_{e=1}^{|\mathcal{E}|} {\boldsymbol{\alpha}_e}^T \boldsymbol{D}(f(x, t_i, p_i), g(e_i)) \mathbb{1}_{\{e_i = e\}} \quad (2)$$

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

**Activity-Key Pose Link:** ${\boldsymbol{\beta}_i}^T \boldsymbol{\phi}_1(y, e_i)$ models the compatibility between activity $y$ and exemplar $e_i$ as the $i^{th}$ key pose. It is a scalar for each activity and exemplar, and if it is high it means that particular type of exemplar is strongly associated with the activity label $y$:

$${\boldsymbol{\beta}_i}^T \boldsymbol{\phi}_1(y, e_i) = \sum_{a \in \mathcal{Y}} \sum_{e=1}^{|\mathcal{E}|} \beta_{iae} \mathbb{1}_{\{y=a\}} \mathbb{1}_{\{e_i = e\}} \quad (3)$$

The activity-key pose term is indexed on key poses $\boldsymbol{\beta}_i$, and it means that an exemplar in a sequence of key poses may have different compatibility with the activity at different times. This models the fact that key poses have a particular order for each activity. For example bending starts with a standing pose, continues with bending until the subject reaches ground, and ends with a standing pose.

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

$$\boldsymbol{\gamma}^T \boldsymbol{\phi}_2(x, y, t_i, p_i) = \sum_{a \in \mathcal{Y}} {\boldsymbol{\gamma}_a}^T f(x, t_i, p_i) \mathbb{1}_{\{y=a\}} \quad (4)$$

**Features:** In order to match key poses to the input sequence, we choose the histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features to capture shape and movement of human. Thus, we represent

each frame using a concatenation of HOG and HOF features of $8 \times 8$ non-overlapping cells organized on a grid inside a bounding box around the subject.

In Eq. 2 we use a function $\boldsymbol{D}(\cdot,\cdot)$ to measure the distance of two bounding boxes. 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.

## 4. 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 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 kicking, pushing, or punching the model parameters should be different for each participant. The participants of these interactions are the subject of activity, the one who does the activity, and the object of the activity, the one to whom the activity occurs. The subject and object in an interaction may have different key poses. For example, in pushing 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 because of the push. So, we expect to see a different group of key poses for the subject and object trajectories. Further, as noted above the relative spatial position of the subject and object of an interaction is an important cue for recognition.

We modify our single subject model to incorporate this information: **who** is playing which role, and additional cues about **where** these people are. The model is depicted in Fig. 1. We assume we are given the rough trajectories of a potential subject and object of an interaction, and similar to our model in Fig. 2 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. 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 hugging 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 $\boldsymbol{H} = [\boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{b}]$. $\boldsymbol{H}^1$ and $\boldsymbol{H}^2$ are the key pose configuration for each person. The variable $\boldsymbol{b} = (b^1, b^2)$ 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:

$$L(x,y,\boldsymbol{H};\boldsymbol{\omega}) = G(x,y,\boldsymbol{H}^1,b^1;\boldsymbol{\omega}^s) + G(x,y,\boldsymbol{H}^2,b^2;\boldsymbol{\omega}^o) + Q(x,y,\boldsymbol{H};\boldsymbol{\mu}) \quad (5)$$

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 $\boldsymbol{b}$ 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. $Q(x,y,\boldsymbol{H};\boldsymbol{\mu})$ measures the relative distance of two tracks at the time of the key poses and is formulated as:

$$Q(x,y,\boldsymbol{H};\boldsymbol{\mu}) = \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x,y,t_i^1,\boldsymbol{b}) + \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x,y,t_i^2,\boldsymbol{b}) \quad (6)$$

where

$${\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x,y,t_i^j,\boldsymbol{b}) = \sum_{a \in \mathcal{Y}} {\boldsymbol{\mu}_{ia}}^T \boldsymbol{bin}(\|l(x,t_i^j,b^1) - l(x,t_i^j,b^2)\|_2) \mathbb{1}_{\{y=a\}} \quad (7)$$

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 $\boldsymbol{\omega}$.

## 5. 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.

### 5.1. 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 person trajectories, but we do not know which of these people takes which role in the activity. We define the scoring function $E(x,y)$:

$$E(x,y;\boldsymbol{\omega}) = \max_{(b^1,b^2)\in S} \max_{(\boldsymbol{H}^1,\boldsymbol{H}^2)\in \mathcal{H}_1 \times \mathcal{H}_2} L(x,y,\boldsymbol{H};\boldsymbol{\omega}) \quad (8)$$

with $L$ being the dual-trajectory scoring function defined in Eq. 5. $b^1$ and $b^2$ select which person trajectories take the

subject and object roles in the interaction. $S$ is the set of all ordered pairs of actors in sequence $x$. In the case with many actors in a sequence, i.e. TRECVID experiment we limit $S$ to pairs of temporally overlapping trajectories which are close spatially. Recall from Sec. 3 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. 6 measures the distance of a trajectory from the other one at time point of its key pose for both trajectories which is decomposable. Hence, the maximization in Eq. 8 can be decomposed into maximization for each trajectory. So, we can write $E(x, y; \boldsymbol{\omega}) =$

$$\max_{(b_1,b_2)} \Big\{ \underbrace{\max_{\boldsymbol{H}^1 \in \mathcal{H}_1} \{G(x, y, \boldsymbol{H}^1, b^1; \boldsymbol{\omega}^s) + \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x, y, t_i^1, b)\}}_{\text{subject trajectory}} + \underbrace{\max_{\boldsymbol{H}^2 \in \mathcal{H}_2} \{G(x, y, \boldsymbol{H}^2, b^2; \boldsymbol{\omega}^o) + \sum_{i=1}^{K} {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x, y, t_i^2, b)\}}_{\text{object trajectory}} \Big\} \quad (9)$$

The score maximization for each trajectory consists of finding $K$ key poses, $\boldsymbol{h}_i = (e_i, t_i, p_i), \forall i \in 1, ..., K$ 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:

$$A_i^t = \max_{e_i, p_i} \Big\{ \boldsymbol{\alpha}^T \boldsymbol{\phi}_0(x, t, e_i, p_i) + {\boldsymbol{\beta}_i}^T \boldsymbol{\phi}_1(y, e_i) + {\boldsymbol{\gamma}}^T \boldsymbol{\phi}_2(x, y, t, p_i) + {\boldsymbol{\mu}_i}^T \boldsymbol{\theta}(x, y, t, b) \Big\} \quad (10)$$

where $t = 1, 2, \cdots, T$, and $T$ is the number of frames in $x$. Next, considering the constraint, we can rewrite the score maximization of a trajectory in Eq. 9 as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{K} A_i^{t_i} \\ \text{s.t.} \quad & t_i < t_{i+1} \ \ \forall i = 1, 2, \cdots, K-1 \end{aligned} \quad (11)$$

We use an efficient dynamic programming algorithm to solve this maximization. We define $M_j^\tau$ as the best score using $j$ elements of $A$ until the $\tau^{th}$ frame:

$$\begin{aligned} M_j^\tau &= \max \sum_{i=1}^{j} A_i^{t_i} \\ \text{s.t.} \quad & 1 \le t_i < t_{i+1} \le \tau \ \ \forall\, i = 1, 2, \cdots, j-1 \end{aligned} \quad (12)$$

We can write $M_j^\tau$ as a recursive function:

$$\begin{aligned} M_j^\tau &= \max\{M_{j-1}^{\tau-1} + A_j^\tau, M_j^{\tau-1}\} && 1 < j \le K, j < \tau \le T \\ M_j^j &= M_{j-1}^{j-1} + A_j^j && 1 < j \le K \\ M_1^\tau &= \max\{A_1^1, A_1^2, \cdots, A_1^\tau\} && 1 \le \tau \le T \end{aligned} \quad (13)$$

The optimal solution of Eq. 11 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.

### 5.2. 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. [6] and Wang and Mori [18] we formulate the training criteria in the Max-Margin framework. We set $\boldsymbol{\omega}$ by:

$$\begin{aligned} \min_{\boldsymbol{\omega}, \xi^i} \ & \frac{\lambda}{2} \|\boldsymbol{\omega}\|^2 + \sum_i \xi^i \\ \text{s.t.} \quad & E(x^i, y^i; \boldsymbol{\omega}) - E(x^i, y; \boldsymbol{\omega}) > \Delta(y^i, y) - \xi^i \ \ \forall i, \forall y \in \mathcal{Y} \end{aligned} \quad (14)$$

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

The constraint in Eq. 14 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. 14 is a non-convex optimization problem and we use the non-convex extension of the cutting plane algorithm using NRBM [3] to learn the parameters.

**Selecting Exemplars:** Our model requires an exemplar set consisting of instantiations of 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 use the distance function $\boldsymbol{D}(\cdot, \cdot)$ to compare samples. 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 multi-class linear SVM classifier using LIBLINEAR [5] 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 the highest scores using k-means. Note that the k-means centers are virtual poses that do not exist as training samples. We use the nearest samples of the training set to the centers provided by k-means as a set of key human pose candidates. This heuristic procedure is efficient and effective in our experiments; though other supervised clustering techniques could also be used (e.g. Lazebnik and Raginsky [8]).

**Initialization:** Parameter initialization can be crucial in learning latent variable models. We use the following heuristic 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 $\beta_{iae}$ is set to the frequency of matching exemplar $e$.

## 6. 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 [13], 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 [15] and demonstrate our model on a non-choreographed dataset. Fig. 4 shows sample frames from the UT-Interaction and TRECVID embrace datasets.

### 6.1. UT-Interaction Dataset

The UT-Interaction dataset contains videos of 6 classes of human-human interactions: *shake hands*, *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. We follow the experimental setting of the classification task described in the High-level Human Interaction Recognition Challenge [13] – 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 information is used – in particular roles in the interaction ($b$ variables) are inferred both in learning and test time.[2]

#### 6.1.1 Implementation Details

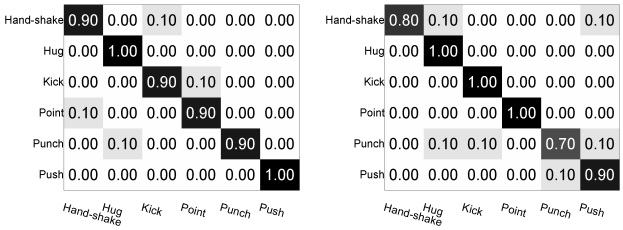
The bounding boxes provided as input contain the two humans performing an interaction, not tracks of individuals. We employ a pedestrian detector [2] to obtain initial positions of the people in the first frame of every video clip. We select a pair of detections with the minimum horizontal distance out of the three highest scoring detections, then run a tracker [12] to find trajectories of two individuals interacting with each other in the subsequent frames. To handle tracker jitter, we allow key pose positions at small spatial perturbations 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. Considering camera zoom in set 1, we also perform multi-scale search at 2 scales.

#### 6.1.2 Results

Confusion matrices of the two sets in the UT-Interaction dataset are shown in Fig. 3. 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 1. A direct comparison is possible to the methods by Yao et al. [23] and Yu et al. [24]. Our method clearly outperforms the other methods.

Table 1. Per-clip classification accuracy on UT-interaction dataset.

| Method | Set 1 | Set 2 | Avg |
|---|---|---|---|
| Our method | **0.93** | **0.90** | **0.92** |
| Yu et al. [24] | N/A | N/A | 0.83 |
| Yao et al. [23] | 0.88 | 0.80 | 0.84 |




(a) set 1 (b) set 2

Figure 3. Confusion matrices of per-clip classification result on UT-Interaction dataset. Horizontal rows are ground truth and vertical columns are predictions.

#### 6.1.3 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, which is patch-based weights, to demonstrate that our model is able to localize key poses in the trajectory and fire on discriminative patches for pose. Figure 4 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 5 visualizes our learned activity-key pose 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 6 shows the learned spatial distance weights.

### 6.2. TRECVID Embrace Dataset

We collected a subset from the development dataset of the TRECVID 2008 Surveillance Event Detection challenge [15] 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 (>95% **miss** rate at 10 FP/hour). Considering the fact that human detectors and trackers have difficulty in challenging datasets like TRECVID, we manually select a subset of the dataset 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 from camera view 3, which contains 343 *embrace* events (63% of those in the whole dataset). We manually select a positive set of 36 *embrace*

[2]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 search horizontally for a person nearest to the one performing the point activity and include him as the other part of the activity.

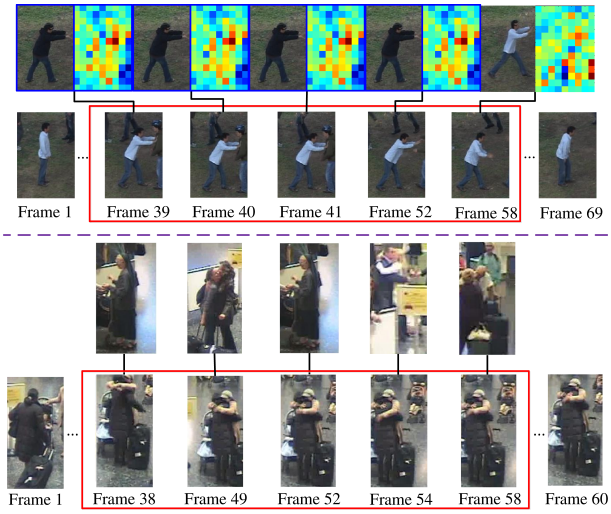



Figure 4. Discriminative frames of a trajectory are automatically extracted. Separated by a dashed line, the upper part of the figure comes from the UT-Interaction dataset and the lower part from the TRECVID embrace dataset. The localizations of key poses in trajectories are highlighted by red bounding 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.

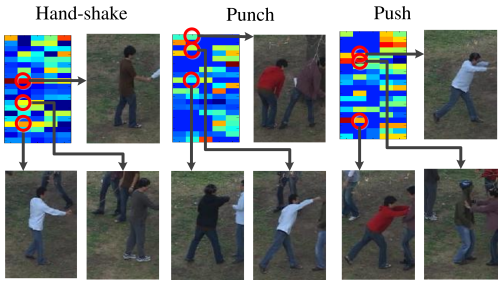



Figure 5. Visualization of activity-key pose model. For the heatmap of each activity, the horizontal axis is the concatenation of the 5 key poses in the activity and the vertical axis specifies 20 exemplars belong to the activity. Each pixel describes the score for an exemplar being matched to a key 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.

clips where our detector and tracker provide reasonable output. We randomly sample 300 video clips that do not temporally overlap with the *embrace* events, and use the same

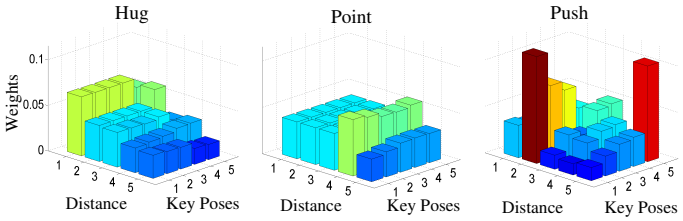



Figure 6. Spatial distance model for three activities in UT-Interaction dataset. Three axes are discrete distance, key poses and weights. For a key pose in each activity, the heights of bars indicate our model's preference among different distances. Bars are also coloured according to height. The spatial 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.

human detector/tracker used for positive examples to obtain trajectories. There are 1074 pairs of trajectories that intersect spatio-temporally, but are not *embrace* events. We sample 108 pairs of trajectories to use.

The TRECVID Event Annotation Guidelines state that embrace starts at the latest 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 than the TRECVID ground truth. We also fix the length of *embrace* samples at 60 frames for both positive and negative samples. Note that the negative samples come from videos randomly sampled in time, hence is a fair comparison to non-embrace videos, though 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 one can barely see embrace event. However, the dataset still inherits the challenging characteristics of TRECVID videos: it contains large intra-class variation with a cluttered background. The precise dataset will be available for download at our website.

#### 6.2.1 Preprocessing

Our dataset is created by collecting a set of trajectories from the TRECVID dataset. We run a HOF/HOG SVM human detector on the first frames of the clips and use a tracker [1] to obtain trajectories of individuals. The task is now a classification task – given a pair of trajectories, is there an *embrace* activity occuring or not.

#### 6.2.2 Results

We evaluated our method using 6-fold cross-validation. To evaluate the effectiveness of different parts of our model, we introduce two baseline methods. The first baseline is our

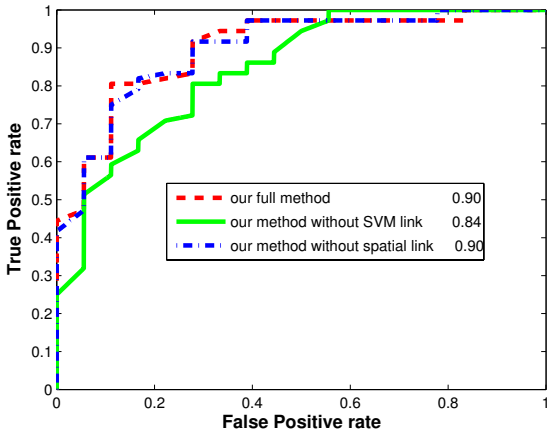



Figure 7. ROC curves on TRECVID *embrace* dataset. Legend shows Area Under ROC (AUR) for methods.

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. The ROC curves in Fig. 7 show the effectiveness of our method relative to these baselines.

The 6% increase in AUR from the first baseline to our full model reflects the contribution of the root model to our full model. We only select trajectories that overlap spatio-temporally for negative examples, so as expected the models with and without spatial distance are similar.

Our experiments are on a subset of the TRECVID *embrace* dataset, but we can extrapolate performance to the complete TRECVID dataset. Camera view 3 captures the majority of the *embrace* events. In the worst case, if we misclassify all other positive examples, the maximum achievable true positive rate (TPR) in ROC will be 63%. Due to failures of the human detector, tracker and ignorance of short positive samples, our TPR will at most decrease to $10\% \times 63\%$ of our reported TPR. However, our negative examples are randomly selected pairs of trajectories which overlap in space and time, they are a very difficult subset of negative examples. The results on our dataset indicate promising performance on the full TRECVID dataset.

## 7. Conclusion

In this paper we presented a discriminative model for human interactions based on a sequence of key poses. Strict temporal ordering, the spatial relation between actors in an interaction, and variability in instantiation of key poses are all enforced in this model. An efficient dynamic programming algorithm for inferring key pose sequences was presented, and parameters were learned using the discriminative max-margin criterion. Experiments on the benchmark UT-Interaction dataset verified the effectiveness of the model. Further, non-choreographed activities were explored using a subset of the TRECVID dataset. The current method uses human trajectories as input. The TRECVID experiments showed the promise of this method in situations where tracking is challenging, though as future work, examining the addition of tracking as an additional latent variable could alleviate this direct prerequisite.

## References


[1] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009. 7

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 6

[3] T.-M.-T. Do and T. Artières. Large margin training for hidden markov models with partially observed states. In *ICML*, 2009. 5

[4] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003. 1

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. of Machine Learning Research*, 9:1871–1874, 2008. 5

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. PAMI*, 32(9), 2009. 5

[7] S. S. Intille and A. Bobick. Recognizing planned, multiperson action. *CVIU*, 81:414–445, 2001. 2

[8] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. PAMI*, 31(7):1294–1309, 2009. 5

[9] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009. 1

[10] G. Médioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Trans. PAMI*, 23(8):873–889, 2001. 2

[11] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 2

[12] D. Ross, J. Lim, and R.-S. Lin. Incremental learning for robust visual tracking. *IJCV*, May 2008. 6

[13] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009. 2, 6

[14] Q. Shi, L. Cheng, L. Wang, and A. Smola. Human action segmentation and recognition using discriminative semi-markov models. *Int. Journal of Computer Vision*, 2010. to appear. 2

[15] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR*, 2006. 6

[16] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *ECCV*, 2002. 2

[17] Y. Wang and G. Mori. Human action recognition by semi-latent topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1762–1774, 2009. 1

[18] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009. 5

[19] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *CVPR*, 2008. 1

[20] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 2010. to appear. 1

[21] T. Xiang and S. Gong. Beyond tracking: Modelling activity and understanding behaviour. *IJCV*, 67(1):21–51, 2006. 2

[22] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, 1992. 1

[23] A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *CVPR*, 2010. 2, 6

[24] T.-H. Yu, T.-K. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forest. In *BMVC*, 2010. 2, 6