Coaction Discovery: Segmentation of Common Actions Across Multiple Videos

Caiming Xiong
Department of Computer Science and Engineering
SUNY at Buffalo
Buffalo, NY, 14260-2500
cxiong@buffalo.edu

Jason J. Corso
Department of Computer Science and Engineering,
SUNY at Buffalo,
Buffalo, NY, 14260-2500
jcorso@buffalo.edu

ABSTRACT

We introduce a new problem called coaction discovery: the task of discovering and segmenting the common actions (coactions) between videos that may contain several actions. This paper presents an approach for coaction discovery; the key idea of our approach is to compute an action proposal map for each video based jointly on dynamic object-motion and static appearance semantics, and unsupervisedly cluster each video into atomic action clips, called actoms. Subsequently, we use a temporally coherent discriminative clustering framework for extracting the coactions. We apply our coaction discovery approach to two datasets and demonstrate convincing and superior performance to three baseline methods.

Categories and Subject Descriptors

H.5.1 [**Information System**]: Information Interfaces and Presentation—*Multimedia Information Systems* ; I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Algorithms, Experimentation

Keywords

coaction discovery, time series clustering, discriminative clustering

1. INTRODUCTION

Human action modeling and understanding in the video is one of the most popular topics in the current computer vision community with many recent papers about action recognition [10, 18] and action detection [6, 20]. In this paper, we introduce a new term for action: *coaction*, which is defined as the common actions (coactions) between multiple long term videos each of which is composed of multiple actions (Figure 1).

MDMKDD'12, August 12, 2012, Beijing, China.

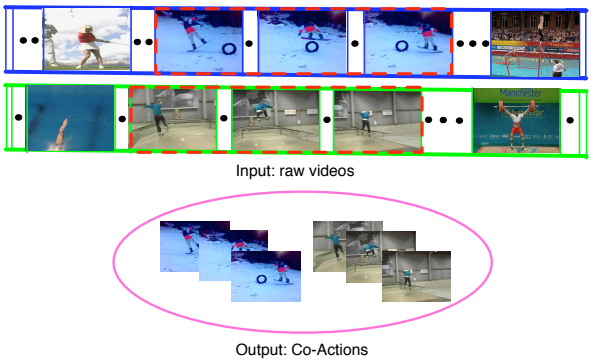

Figure 1: Our approach is to discovery the common actions (coactions) between videos each of which consists of several actions.

Coaction discovery has rich potential in computer vision. First, given two long term videos, we can calculate a more accurate distance of two videos based on the coaction of two videos rather than the whole videos, and improve the video retrieval result. Second, when there are many similar videos from which we need to extract some common action, it is possible to annotate only one video and obtain others automatically through coaction discovery. Third, based on the coaction of two videos, we can do video clustering and understand the structure of a video set, which is increasingly important given the massive amount of videos being produced and uploaded onto the internet in recent years.

Although coaction discovery is a new problem, there are similar works in action detection and image segmentation. In action detection, some papers detect the position of the action in the long term video by using the single action template video [6, 20]. We would consider the extracted subvideos to be the coaction between the multiple action videos and the template video. Whereas these methods are *action-specific*, i.e., they are given the action template and seek its similar segments in other longer videos; in our case, we take a set of videos and seek to discover the coaction naturally from them, without any specific knowledge of which type of action may be present. Also, whereas these methods can be considered to be detection methods (they are given an action template and seek to detect it in the other longer videos), in our case we take the multiple videos, each of which may contain multiple actions, and seek to segment the coactions.

In image segmentation, there is a term called cosegmentation [5, 7, 13] which segments the common parts of an image

pair simultaneously. It is very similar in our high-level problem description, but our problem is on three-dimensional video data. Coactions must be considered at a higher-level than the common parts of an image pair since there are both appearance and motion information. We are able, however, to build upon this existing literature in part of our method, using discriminative coclustering [7].

Coaction discovery is a unique challenge. First, the video background presents an interesting difficulty: many image analysis methods now use background information as context for higher-level tasks (even action recognition from a video [19]) and show performance improvement. However, for coaction in videos, the backgrounds may dominate videos and the coaction may erroneously incorporate background. In the limit case, the discovered coaction is directly the backgrounds. Second, since there are multiple actions in each video and the position and time span of the actions is unknown, we can not compare actions directly between videos to obtain coactions. Conversely, we can not compare individual frames between videos to get the coaction; individual frames carry insufficient motion information to adequately describe realistic actions. The novel nature of coaction discovery and these noted challenges requires an innovative solution to the problem.

We propose an approach that automatically discovers the coactions between multiple videos. Our main idea is to leverage the dynamic and static appearance-semantic (detected humans) cues to generate an action proposal score map for each frame. We extract features based on the score map by an innovative spatially-weighted bag-of-words using this action proposal score map. We then segment the video into meaningful actoms/clips through unsupervised time-series clustering ([4] use a similar notion of actom but theirs is supervised and human-specific), and finally, estimate the coaction between the videos using an extended discriminative segmentation method that incorporates a temporal consistency prior on those actoms. See Figure 3 for an overview.

To implement our method, we first propose an action-like motion and appearance based measure that reflects a position's score of belonging to the action region. To obtain the action-like motion proposal map, we use the method from [15] that computes optical flow and uses properties of point trajectories, such as large difference between surrounding and action region, to distinguish moving objects from moving background. To obtain action-like appearance proposal map, we use a static property of single frame, such as response of human detector [3], since we assume the *host* of the action is human. Our base assumption here is that humans are performing the actions of interest, but this can be relaxed to application-specific semantics. Based on motion and appearance information for action, it can generate the action-like score map using an additive model. Second, we extract the spatially-weighted features of each frame based on the action proposal score map and compute the similarity using the spatial pyramid kernel [9]. Then, we adopt the aligned cluster analysis (ACA) [21], which is a time series clustering method extended from the kernel kmeans through the Dynamic Time Alignment Kernel (DTAK). It jointly segments the video into actoms/clip [4] and clusters the clips in the whole model. Finally, we have the clip-based representation for each video. Considering each clip as a node, the coaction discovery is reduced to cosegmetation on one-dimensional time series data, with one for each video. We innovatively combine temporal local consistancy within discriminative clustering [1, 7] to discovery the coactions between videos.

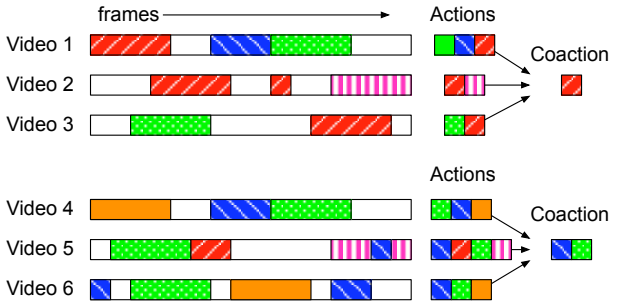

Figure 2: Two illustrative examples of a coaction set over three videos each. Each row depicts the actions of frames in a video and each distinct action is depicted as a color; white indicates no action is present. In the top case, the coaction is the red/slash action (it uniquely occurs in videos one, two and three) whereas in the bottom case, the coaction is the blue/backslash and green/dots set. Even though the orange action occurs in videos four and six, it is not in the coaction because video five lacks it.

In this paper, our main contribution is an automatic approach for coaction discovery between videos. To the best of our knowledge, this is the first paper working on coaction discovery for multiple actions in multiple videos. The important novel parts of our approach include: (1) a new motion and appearance based measure of action proposal score map in the video; (2) a spatially weighted bag of words for each frame based on this score map, (3) unsupervised clustering of the meaningful actoms/clips for videos, and (4) a discriminative clustering with temporal local consistency that accommodates the actoms/clips. In the experiments, we build a coaction dataset based on the UCF sport dataset [11] (CA-UCF), by concatenating multiple single action videos, and we use a small challenging data set collected from Youtube. We apply our approach and other three baseline approachs, our approach shows better performance than other baseline approaches on CA-UCF.

2. APPROACH

A *coaction* is a common action that is occurring in two or more videos. For example, in two videos of a baseball game, the action of the pitcher pitching is common to both of the videos. Figure 1 gives a high-level introduction to coactions.

Concretely, let $\mathcal{V} = \{V_1, V_2, \ldots, V_N\}$ be a set of N videos with each video V_i represented as a sequence of n_i frames $\{v_i^1, \ldots, v_i^{n_i}\}$, or with a shorthand set-notation $v_i^{1:n_i}$, which we will use throughout the paper to denote subsequences of frames, or *clips*, in a video and we will drop the subscript i whenever it is irrelevant.

Let A be some oracle that maps a frame in a video to one action among a set $\mathcal{A}$ of possible actions. An action is then a subsequence of video-frames $v^{a:b}$ such that each frame has the same action: $|\bigcup_{z=a}^{b} A(v_i^z)| = 1$. We can hence extend the action oracle to operate on clips.

Finally, a coaction over $\mathcal{V}$ is then a set of m clips $C =$

$\{v_i^{a_z:b_z} : i \in (1,N)$ and $z = \{1,\ldots,m\})\}$ such that the union of actions over each video's clips in the coaction is the same for all videos in the coaction, which we write via a pairwise equivalence:

$$\bigcup_z A\left(v_k^{a_z:b_z}\right) \equiv \bigcup_z A\left(v_l^{a_z:b_z}\right) \quad \forall \text{ pairs } k,l \in (1,N) \ . \quad (1)$$

See Figure 2 for a pictorial explanation. Our definition of coaction relies upon the common assumption that a single action may occur at any given frame in a video (e.g., [12,14]), but it is not difficult to relax that assumption for a more general definition of coaction.

The basic problem we pose and solve in this paper is to automatically discover and temporally segment the coactions in a given set of videos. Our coaction discovery algorithm has four key steps.

1. We measure the probability that any action is occurring over each frame in each video, which will allow us to discard problematic background/action-free regions.
2. We then extract a bag-of-words feature histogram over the videos using an innovative weighted histogram approach to focus the feature extraction on the actions within the video;
3. The next step is to extract unsupervised *actoms*, clips with atomic actions in them, using Aligned Clustering Analysis (ACA) [21].
4. For the final coaction segmentation, we adapt the discriminative cosegmentation algorithm [7] to our one dimensional actom cosegmentation.

The remainder of this section describes each of these steps in more detail and Figure 3 presents a visual explanation.

2.1 Action Proposals

The first step in our method is to generate action proposals: measurements over each video frame that represent the probability the pixels within the frame are in some action. We assume no classifier based on location, duration, or appearance is available to generate action proposals and, rather, our proposals based on two assumptions about actions in video are: first, there must be motion present around an action, which we call the *attribute* of the action and second, the key object in the action is human, which we call the *host* of the action. The former is a bottom-up notion and the latter is high-level semantic information; these assumptions may be altered depending on the particular coaction application, but in our paper, they will suffice (we ultimately work with human sports videos, such as UCF Sports [12]). We derive a proposal map for each of these two assumptions (described next) and then combine them to yield an overall probability of action at a per-pixel level in the video frames. Figure 3(a) shows the action proposals for example video frames.

Moving-Object Score Map. For static and *mostly static* backgrounds, finding the moving object is straightforward (e.g., frame-differencing or standard background subtraction [17]) and we will not discuss it in any detail. But, when the background is dynamic or the camera is moving freely, computing an action proposal for object motion is quite more difficult. In this situation, we dynamically build a model of the background by sparsely sampling trajectories of salient points and robustly estimating a compact trajectory basis from them based on the Sheikh et al. method [15].

For the zth frame v_i^z of video V_i, we use optical flow to obtain dense trajectories at every fifth pixel p, the length of trajectory is 15 frames in our paper. Optical flow is computed with the method from [2]. Following [15], we assume the background is comprised of three trajectory bases and use RANSAC to estimate them at each frame. Given the three computed bases b_k, $k = 1,\cdots,3$, for the pixel p of v_i^z, the moving object score $M(p;b_k)$ is estimated based on the reconstruction error of the trajectory of position p through the computed three bases b_i:

$$M(p;b_k) = \min_{\alpha} \left\| t(p) - \sum_{k=1}^{3} \alpha_i b_k \right\| \ . \quad (2)$$

where $t(p)$ is the trajectory of position p. The larger the minimal reconstruction error, the more likely the pixel is not part of the dynamic background and hence the more likely it is a moving object.

Human Score Map. We use the state of the art latent SVM-based human detector [3] to compute the human score map, $H(p)$. For each pixel p in the frame v_i^z, we extract its score $H(p)$ directly from the detector.

Combined Action Proposal Score Map. For each frame, we extract the two maps using the low-level moving object extraction $M(\cdot;b_k)$ and high-level human detection methods $H(\cdot)$, and we normalize them such that lies within the range of $[0,1]$ denoted $\overline{M}$ and $\overline{H}$ because each map's score has a different range. Our fusion rationale is that we want the action proposals to focus on regions that have foreground-like moving objects (an inter-frame quantity) and human-like appearance (and intra-frame quantity).

$$\text{score}(p) = \lambda\overline{M}(p) + (1-\lambda)\overline{H}(p) \ . \quad (3)$$

To that end, the final action proposal map is a simple convex combination of the two maps for data fusion and we set $\lambda = 0.4$ in all experiments.

2.2 Feature Extraction from the Score Map

Given the action proposal map score each frame, we next extract features that represent each frame in the video. Our feature representation is based on the common bag of words approach, but to emphasize the focus on parts of the video actually containing human-like object-motion, we compute a novel weighted bag of words using the action proposal map from (3).

First, we use standard k-means clustering to learn our codebook over a typical action dataset [12]. Our raw feature is dense HOG3D [8]. Then, given the codebook and a new video, for each frame, we densely take the points from the frame and extract the HOG3D feature for each pixel. Each HOG3D point is assigned to one of words in the codebook. However, whereas in standard bag of words approach, each HOG3D point is equally weighted in the histogram, our innovation is to weight each HOG3D point based on the action proposal score map score(p) from (3). For the kth bin of histogram hist,

$$\text{hist}(i) = \sum_{f \in F} \mathbb{1}_{c(f)==i} \cdot \text{score}(p(f)) \ , \quad (4)$$

where F is the feature set of all HOG3D point of the frame, $c(f)$ is the assignment of the feature f in the codebook, and $p(f)$ is the spatial position of feature point f. This histogram is referred to as a weighted bag of words histogram.

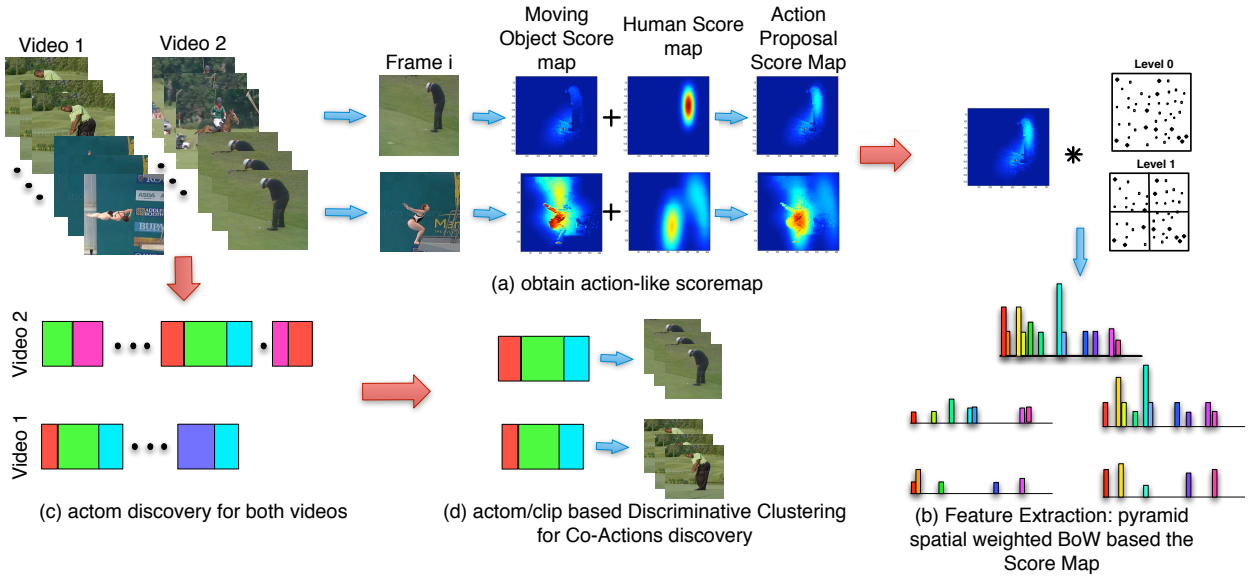

Figure 3: Algorithm overview where steps (a) to (d) correspond to Section 2.1 to Section 2.4, respectively.

Finally, we build the two level pyramid structure (i.e. 1x1, 2x2) [9]. Thus there are one weighted histogram $hist_0$ in the level 0, and four histograms $\{hist_1, hist_2, hist_3, hist_4\}$ in the leve 1.

To measure the similarity between two frames v_i^k and v_i^l, we calculate the summation of the intersection kernel of each histogram for two frames:

$$K(v_i^k, v_i^l) = \sum_{i=0}^{4} \mathcal{I}^i(v_i^k, v_i^l) \tag{5}$$

where $\mathcal{I}^i(v_i^k, v_i^l) = \sum_j \min(hist_i^{(v_i^k)}(j), hist_i^{(v_i^l)}(j))$. Some equivalent actions can be oriented in different directions, such as walking right-to-left or left-to-right. To account for this, we compute two similarities between each pair of frames: flipped and unflipped. The unflipped similarity is computed between the two original frames, while the flipped similarity is produced by flipping one of the two frames left-to-right. We then record the maximum of these two similarities. We then segment the video based on these recorded frame similarities using time series clustering.

2.3 Unsupervised Actoms Discovery

In some sense, a single action can be considered as combination of sub-actions. These sub-actions have been called "actoms" by [4] and are small clips in the video that contain a single action. In [4], the actoms is annotated manually for the training data. But, in our coaction discovery problem, there is neither prior knowledge about what actions will be present in the video nor about what kind of actoms should be obtained for actions. So, we can not get the actom as in [4]. In this section, we show how we obtain the actoms in an unsupervised way via time series clustering.

What is an actom? There are two aspects: firstly, an actom is a *meaningful* clip (semantically) in the video; secondly, this kind of clip should happen with some frequency. The simplest way to unsupervisedly compute the actoms, is to run kernel kmeans on the video frames. The subsequent frames within each cluster could be seen as the actom. However, because kernel kmeans adopts the kernel for two single frames, the generated actoms are collections of similar frames (in appearance and local flow), which are quite different from the actom we seek because they can not capture similar actions. We hence need a time series clustering method that adds the actom generation (temporal actom segmentation) into the objective function of the clustering algorithm. To reach the goal, we adopt an extension of kernel kmeans called Aligned Cluster Analysis (ACA) [21], which combines kernel kmeans with Dynamic Time Alignment Kernel (DTAK).

Given video set $\mathcal{V} = \{V_1, V_2, \cdots, V_k\}$ with k videos, each video $V_i = \{v_i^1, \cdots, v_i^{n_i}\}$ consists of n_i frames. To make the notation simpler, we concatenate the videos into one long video and reindex them $X = \{v_1^1, \cdots, v_1^{n_1}, \cdots, v_k^1, \cdots, v_k^{n_k}\} = \{x_1, x_2, \cdots x_n\}$ with $n = \sum_{i=1}^{k} n_i$.

We set the number of clusters to p based on the assumption that there are p different kinds of actoms. The Aligned Cluster Analysis (ACA) method segments X into m disjoint actoms each of which belongs to one of p clusters. The i^{th} actom, $Y_i = [x_{s_i}, \cdots, x_{s_{i+1}-1}] = x^{s_i:s_{i+1}}$ is composed of frames that begin at frame s_i and end at $s_{i+1} - 1$ frame. We constrain the length of each actom to be less than $n_{\max}$. ACA combines kernel kmeans with DTAK to pursue temporal clustering by minimizing

$$J_{ACA}(G, M, s) = ||\psi(x^{s_i:s_{i+1}}) - MG||_F^2. \tag{6}$$

The objective function of ACA is very similar to kernel kmeans' except for the variable s which determines the start and end of each actom. In the objective function, $G \in \{0,1\}^{p \times m}$ is an indicator matrix that assign each actom to a cluster; $g_{ci} = 1$ if Y_i belongs to cluster c. The columns of M represent the cluster centroids, but in the kernel-based ACA, typical M can not be computed explicitly, $\psi(\cdot)$ is a non-linear

mapping for the actom that, $\tau_{ij} = \tau(Y_i, Y_j) = \psi(Y_i)^T\psi(Y_j)$ is the Dynamic Time Alignment Kernel (DTAK). The DTAK is defined as:

$$\tau(Y_1, Y_2) = \max_{Q} \sum_{c=1}^{l} \frac{1}{n_{Y_1} + n_{Y_2}} (q_{1c} - q_{1c-1} + q_{2c} - q_{2c-1}) K_{q_{1c}q_{2c}}^{Y_1Y_2} \ , \quad (7)$$

where $K_{ij}^{Y_1Y_2} = \phi(Y_{1i})^T\phi(Y_{2j})$ represents the kernel similarity between frame Y_{1i} and Y_{2j}. $Q \in R^{2\times l}$ is an integer matrix that contains indexes to the alignment path between two actom. l is the number of steps needed to align both actoms. DTAK finds the path that maximizes the weighted sum of the similarity between actoms. For convenience, we denote the matrix $W^{Y_1Y_2} \in R^{n_{Y_1}\times n_{Y_2}}$, that $w_{ij}^{Y_1Y_2} = \frac{1}{n_{Y_1}+n_{Y_2}}(q_{1c} - q_{1c-1} + q_{2c} - q_{2c-1})$ if there is $q_{1c} = i$ and $q_{2c} = j$ for some c in the Q, otherwise 0. Then the DTAK can be rewritten as

$$\tau(Y_1, Y_2) = \psi(Y_1)^T\psi(Y_2) = tr(K^{Y_1Y_2T}W^{Y_1Y_2}) \ . \quad (8)$$

There is now an actom-based DTAK kernel matrix $T \in R^{m\times m}$ that can be expressed by rearranging the $m \times m$ blocks of $W_{ij} \in R^{n_i\times n_j}$ into a global correspondence matrix $W \in R^{n\times n}$:

$$T = [\tau_{ij}]_{m\times m} = [tr(K_{ij}^T W_{ij}]_{m\times m} = H(K \circ W)H^T \ , \quad (9)$$

where $H \in \{0,1\}^{m\times m}$ is the actom-frame indicator matrix: $h_{ij} = 1$ if j^{th} frame belongs to i^{th} actom. Then, after replacing the optimal value of M, the J_{ACA} is

$$J_{ACA}(G, s) = tr((L \circ W)K) \ , \quad (10)$$

where $L = I_n - H^TG^T(GG^T)^{-1}GH$. Obviously, the objective function of ACA is a non-convex function. We alternative optimizing for G and s. Given the s, obtaining G is by kernel kmeans; given G, with the video V of length n, the possible number of s is exponential, which is impossible by brute force; here, we adopt dynamic programming to solve the problem efficiently. More details are in [21].

2.4 Discriminative Clustering for Coactions

The unsupervised actom discovery will reduce a video of, say, 500 frames, to a small set of actom clips. First, this will decrease the time complexity for coaction discovery; second, it is more natural to make the discovered coaction composed of actoms than frames. We consider the actoms as a middle layer between the video frames and longer term actions, and, for this final step of the algorithm, we only consider the sequence of actoms for each video, and are, hence, above to leverage the discriminative clustering ideas from the cosegmentation literature [7], to which we add a notion of temporal coherence.

Given the action-based representation of videos, each video V_i is reduced to a few number of c_i actom clips. Assume there are total $c = \sum_i c_i$ actom clips. Our goal is to partition all of the clips from all videos into only two classes, the coaction class and the background class. We denote by y the c-dimensional vector that:

$$|y_i| = \begin{cases} 1 & \text{if the } i^{th} \text{ clip is the coaction;} \\ -1 & \text{otherwise.} \end{cases} \quad (11)$$

We aim to find $y \in \{-1, 1\}^c$, given the actoms of videos.

Discriminative Clustering. Our discriminative clustering framework is based on the Dynamic Time Alignment Kernel (Eq.7). However, because the DTAK can not promise the property of positive semidefiniteness (PSD), we make $\tau = \tau + \sigma I$, σ is the smallest negative eigenvalue of kernel matrix. The PSD kernel τ can be considered as mapping each actom clip c_i into a high dimensional Hilbert space $\mathcal{F}$ through a feature space ψ. The method then aims to learn a classifier through minimizing the objective function with respect to $w \in \mathcal{F}$ and $b \in R$ such that:

$$\lambda|||w|||^2 + \frac{1}{c}\sum_{i=1}^{c} loss(y_i, w^T\psi(x_i) + b) \ , \quad (12)$$

where $y_i \in \{-1, 1\}$ is the associated label of i^{th} clip and $loss(\cdot)$ is a loss function; we consider the squared loss function such that $loss(a, b) = (a - b)^2$. Given the kernel matrix τ and unknown y, we can rewrite the Equ.12 into a function $g(y)$ with y as its variable and the optimal value $g(y)$ is a measure of the separability of the two classes $\{-1, 1\}$. According to [1], we can compute $g(y) = y^TAy$, where $A = \lambda(I_c - \frac{1}{c}1_c1_c^T)(c\lambda I_c + \tau)^{-1}(I_c - \frac{1}{c}1_c1_c^T)$. In order to adapt the discriminative clustering to the coaction discovery task, we add the local temporal consistency by incorporating a term based on the normalized Laplacian matrix, which is regularly used in spectral clustering.

Local temporal consistecy. Temporal consistency within an video V_i is enforced by the similarity matrix W_i based on clip position c_j and DTAK kernel which lead to high similarity for nearby clips with high similarity. Thus for any pair (k, l) of clips that belongs to the i-th video, $W_i(l, k) = \exp(-\chi^2(h(l), h(k))^2)$, where $h(l)$ is the histogram of spatially weighted bag of words for lth clip, we calculate the similarity for temporal consistency based on the χ^2 distance of the histogram of two clips.

We can compute separate similarity matrices W_i, and put them on the diagonal of larger matrix. Then we will get a block-diagonal matrix $W \in R^{c\times c}$. Now we consider the normalized Laplacian matrix L of the block-diagonal matrix W, that $L = I_c - D^{-1/2}WD^{-1/2}$, where D is the diagonal matrix consisting of the row sums of W, I_c is the c-dimensional identity matrix. Based on normalized cut [16], we only need consider the second smallest eigenvector of L that minimize y^TLy. And, since the L is also block-diagonal matrix, the solution of minimize objective function equals to cluster each video independently into two class. Thus we add the term y^TLy into the objective function of discriminative clustering, that enforce local consistency.

Discriminative cluster for Coaction Discovery. Therefore, combining the discriminative cost through the matrix A and the local temporal consistency through the matrix L, we can obtain that:

$$\min_{y\in\{-1,1\}^c} y^T(A + \alpha L)y \quad \text{s.t.} \quad \mu_0 1_c \le (yy^T + 1_c1_c^T)\gamma_i \le \mu_1 1_c \ . \quad (13)$$

The constraint is about the lower bound μ_0 and upper bound μ_1 of number of each class in each video. The $\gamma_i \in R^c$ is an indicator vector, that $\gamma_i^j = 1$ if the j_{th} clip belongs to the video i, otherwise 0. Equation 13 is NP-hard, but we can relax it into a convex optimization problem similar to the original formulation [7].

3. EXPERIMENTS

In order to show the advantage of our approach, we construct a dataset for coaction discovery based on the UCF Sports dataset [12] and compare with the results of some baseline methods. We also test our methods on novel data downloaded from YouTube and show qualitative results on it, with promising findings.

3.1 Dataset

Since coaction discovery is a new problem and there is no available test dataset, we have built a dataset of 100 long videos. Each video consists of 5 different videos from the UCF Sports data set [12] that are concatenated together to allow for quantitative evaluation. Then, based on these long-term multiple action videos, we randomly choose 48 pairs of videos each of which at least has one common action between the pair of videos. We run our approach and baseline methods on these 48 pairs of videos. This dataset of 48 pairs of long videos is named as CA-UCF Sports video dataset.

3.2 Baseline methods and Evaluation metric

For comparison, we propose the three baseline methods:

- *Frame-based* (Frame) that considers each frame as an actom (and does not use ACA to obtain actoms) and discover coactions directly based on frame. Hence, the time-series clustering part is not used.
- *Kmeans-based* (KKM) that replaces the ACA (Section 2.3) method by kernel kmeans.
- *Without Temporal coherence* (Tempo) that adopts the discriminative clustering for coaction discovery but does not use the proposed temporal coherence term.

To estimate the quantitative accuracy of the output of the methods, we define the following accuracy term:

$$Accuracy(C, \overline{C}) = \frac{|C \bigcap \overline{C}|}{|C \bigcup \overline{C}|} \ , \qquad (14)$$

where C is the groundtruth of coactions for videos, and $\overline{C}$ is the output of approaches.

3.3 Results

We first present and discuss our results on the CA-UCF Sports dataset. We use two cases for the static appearance proposal score map: (1) testing on videos generated from the extracted bounding box of the human in the frame, given a manually annotated human bounding box and action proposal map beforehand; and (2) testing on original video without any supervised information, automatically obtaining the action proposal map.

The coaction discovery results over two variations of coaction datasets are shown in Table 1. Our approach outperforms all of the other baseline methods each of which have only one different component from ours in both cases (when the human bounding boxes are given and are not given).

In Figure 5, we show some qualitative results. We see the deficiency of the frame-based method highlighted here because it is grouping the frames based only on appearance (here, red clothing and red image parts). Kernel kmeans is the second best performing method; the only difference between KKM and our approach is that the ACA method we uses extends KKM to time-series data. The result hence indicates the usefulness of emphasizing the temporal nature of video in this problem.

	Frame	KKM	Tempo	Ours
Manual	66.81%	69.37%	65.93%	71.12%
Auto	59.15%	60.72%	59.87%	62.51%

Table 1: Accuracy of our proposed method against the three baselines with and without manual human annotations on the CA-UCF sports dataset.

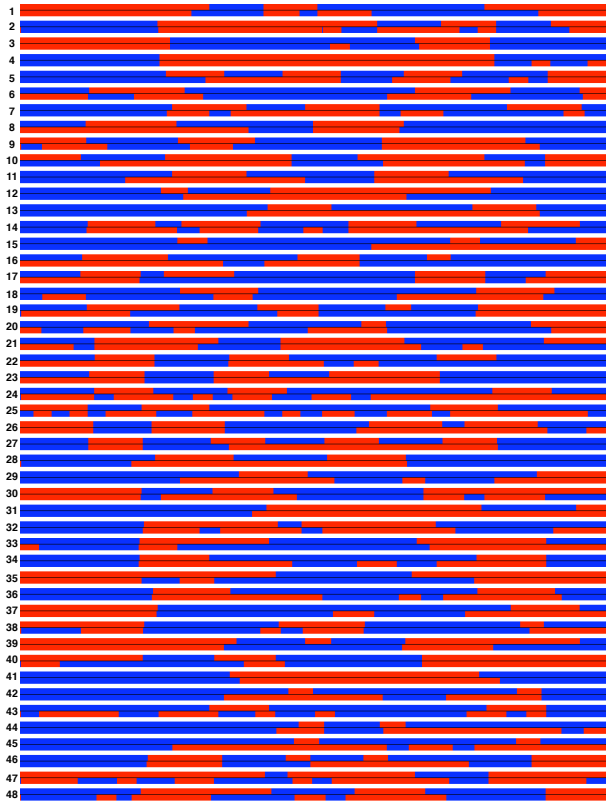

Figure 4: Result of coaction discovery on the full CA-UCF dataset using our approach. We generated 48 pairs of videos, and for each pair generated labels indicating what sections of the videos represented a coaction. The upper bar for each pair displays the ground truth for the two videos (as if they were concatenated together), with red indicating the coaction segments and blue indicating background. The bottom bar shows the results of our coaction discovery in the same format.

Figure 5: Example of qualitative comparison for CA-UCF Sports dataset.

Figure 6: Example results on novel data taken for board-sports in varying settings from Youtube.

In Figure 4, we show the coaction discovery results for the whole CA-UCF Sports dataset. The image in the figures shows each of the 48 clips row wise, the coaction segments rendered as red and the background segments rendered as blue. A perfect coaction result is when the red and blue segments are aligned for the two rows of each video (e.g., 3 is near-perfect). The result is quite good and demonstrates a robustness to different types of actions in the coaction set (UCF Sports is comprised of 10 raw actions).

We have also experimented with the generalizability of our method to more challenging data in which the action across the videos is similar but the actual setting in the video is quite different. For these experiments, we selected videos from board-sports on YouTube. In Figure 6, we show two example results from our experiment. On the left side, we see the method correctly joins a sequence of jumping off a ramp (and losing the skateboard) and jumping down a flight of stairs on a skateboard. The full sequence of the jump has correctly been extracted as a coaction. In the right side, we find a similar outcome where a skateboarder doing a rail-slide is in the same coaction as a snowboarder doing one. These results demonstrate the adaptability of our coaction method to cases of varying video appearance and motion dynamics.

4. CONCLUSION AND FUTURE WORK

In this paper, we introduced a new idea called coaction discovery and developed an approach that automatically discovers the coactions between videos. We proposed an action proposal score map, and extracted a spatially-weighted bag of words to represent the frames of video. We then extended time series and discriminative clustering to compute the coaction segmentation for video pairs. We built a coaction dataset based on the UCF sports dataset and showed the comparative quantitative results against three baseline methods as well as visual results on novel challenging data. In the future, we will test the potential applications of coaction discovery, such as the video distance for retrieval, video editing and video clustering.

5. ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation CAREER grant (IIS-0845282), the Army Research Office (W911NF-11-1-0090), DARPA CSSG (HR0011-09-1-0022 and D11AP00245). Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, ARO, NSF or the U.S. Government.

6. REFERENCES

[1] F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. *Advances in Neural Information Processing Systems*, 20:49–56, 2008.

[2] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[3] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2009.

[4] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. *CVPR*, 2011.

[5] D. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 269–276. IEEE, 2009.

[6] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 128–135. IEEE, 2009.

[7] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1943–1950. IEEE, 2010.

[8] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proceedings of British Machine Vision Conference*, 2008.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. Ieee, 2006.

[10] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[11] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[12] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[13] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. 2006.

[14] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proceedings of International Conference on Pattern Recognition*, 2004.

[15] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1219–1225. Ieee, 2009.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[17] C. Stauffer and W. Grimson. Adaptive Background Mixture Modeling for Real-Time Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

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

[19] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[20] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2442–2449. IEEE, 2009.

[21] F. Zhou, F. De la Torre, and J. Cohn. Unsupervised discovery of facial events. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2574–2581. IEEE, 2010.