A “String of Feature Graphs” Model for Recognition of Complex Activities in Natural Videos

U. Gaur, Y. Zhu, B. Song, A. Roy-Chowdhury *
University of California, Riverside, CA 92521, USA
utkarsh.gaur@yahoo.com yzhu010@ucr.edu bsong@ee.ucr.edu amitrc@ee.ucr.edu

Abstract

Videos usually consist of activities involving interactions between multiple actors, sometimes referred to as complex activities. Recognition of such activities requires modeling the spatio-temporal relationships between the actors and their individual variabilities. In this paper, we consider the problem of recognition of complex activities in a video given a query example. We propose a new feature model based on a string representation of the video which respects the spatio-temporal ordering. This ordered arrangement of local collections of features (e.g., cuboids, STIP), which are the characters in the string, are initially matched using graph-based spectral techniques. Final recognition is obtained by matching the string representations of the query and the test videos in a dynamic programming framework which allows for variability in sampling rates and speed of activity execution. The method does not require tracking or recognition of body parts, is able to identify the region of interest in a cluttered scene, and gives reasonable performance with even a single query example. We test our approach in an example-based video retrieval framework with two publicly available complex activity datasets and provide comparisons against other methods that have studied this problem.

1. Introduction

The dynamical interactions between objects in a scene can be described using the following characterization: kinesics of individual objects (e.g., walking, running), chronemics or temporal aspects (e.g., standing in a line), proximics or spatial relationship between objects (e.g., approaching), and haptics, (e.g., shaking hands, exchanging) [1]. Most work in activity recognition has concentrated on analyzing only one of these aspects (predominantly kinesics) as evidenced by the popular activity datasets like KTH [20] and Weizmann [6]. Many video analysis based applications such as surveillance, sports video analysis, content-based search, etc. require effective approaches for modeling and recognition of far more complex activities than these test datasets.

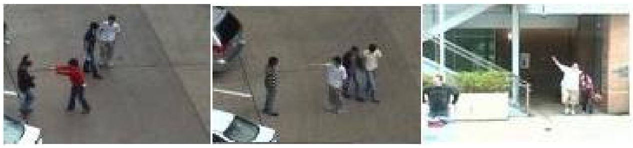

Figure 1. Representative frames of the datasets used in this work. Note that the videos contain multiple actors performing activities simultaneously, sometimes in the presence of irrelevant subjects.

Recognition of complex activities requires understanding of spatio-temporal relationships between different objects, in addition to individual variability, cluttered background, viewpoint changes, and other environment induced conditions. Modeling all these parameters proves to be a challenging task. In this work, we focus primarily on activities that involve multiple interacting objects - people and vehicles - in cluttered scenes (see Fig. 1 for examples). We term these as complex activities. We study the problem of modeling and recognition of such activities in realistic environments, provide detailed performance evaluation of our method, and comparisons against existing approaches.

1.1. Overview and Main Contributions.

The main challenge that needs to be overcome is to develop a representation of the video that respects the spatio-temporal ordering of the features. To achieve this goal, we build upon existing well-known feature descriptors and spatio-temporal representations that when combined together provide a powerful framework to model complex activities in video and efficient computational strategies to estimate similarities between them.

*This work has been partially supported by NSF grant IIS-0712253, the DARPA VIRAT program and a subcontract from Mayachitra Inc., through DARPA STTR award W31P4Q-11-C0042.

Y. Zhu, B. Song and A. Roy-Chowdhury are with the Electrical Engineering Dept at UCR. U. Gaur worked on this paper while a graduate student in Computer Science at UCR. He is currently at UCSB.

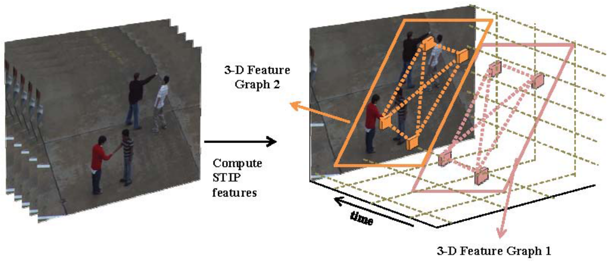

Figure 2. Activity modeling: STIP features are computed from the video and grouped together to form local feature collections. Temporally ordered series of these local feature collections is termed as "string of feature-graphs" (SFGs).

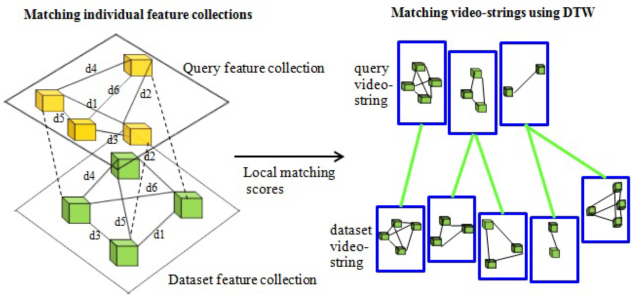

Figure 3. (Left) Local feature-graphs are matched across query and test video using the graph-based spectral technique in Section 3.1. (Right) The local feature-graph match scores thus generated are used in DTW matching of the query video SFG and the test video SFG to account for difference in speed of execution.

A video can be thought of as a spatio-temporal collection of primitive features (e.g. STIP features). We divide the features into small temporal bins and represent the video as a temporally ordered collection of such feature-bins, each bin consisting of a graphical structure representing the spatial arrangement of the low-level features (see Fig. 2). We call this representation of the activity in the video as a "string of feature graphs" (SFGs). Thus the query becomes a string of such graphs, while a test video is also a string of graphs, albeit of a possibly higher complexity.

The problem now is to match these two strings of graphs. This is cast as a combination of sub-graph matching and time sequence alignment (see Fig. 3). The local feature-bins are first matched in a graph-theoretic manner, thereby preserving the spatio-temporal relationships between features. The final match score between the query and test video is a dynamic programming based temporal alignment score between their corresponding feature-bins, thus compensating for differences in speed of execution. Thus, by combining local spatial matching with global temporal alignment, we are able to match videos while respecting their spatio-temporal structure. This gives us the ability to recognize activities that involve interactions between multiple objects like people entering/exiting a facility, following, dispersing, coming in close contact, and so on. Our sub-graph matching scheme supports partial matching, i.e., given query examples, similar actions in a testing video can be retrieved even if the testing video contains other actions happening simultaneously.

Our method does not rely on tracking; it uses primitive video features and proposes a model on top of these features which satisfies the spatio-temporal relationships. We do not need to recognize body parts, unlike [14], or primitive activities [8, 23]. Our method can be thought of as a generalization of the scheme in [15] where the spatio-temporal relationships were modeled using a collection of simple rules; our proposed method allows a more general structure on the video. Additionally, our feature model is not intrinsically tied to any classification mechanism hence enabling its use in scenarios such as query-based retrieval, i.e. recognition with only a single (or very few) example video(s) of the activity in question. This is a highly desired feature since obtaining multiple training examples for increasingly complex activities is often difficult. We show experimental results on two relatively complex datasets, namely the UT-Interaction dataset [16] and UCR VideoWeb activity dataset [3]. Both of these datasets comprise of multiple interactive activities in realistic settings with clutter and changing backgrounds.

1.2. Related Work

Activity recognition has been widely studied, but most of the literature has concentrated on relatively simple activities as evidenced in the KTH or Wiezmann datasets [24]. We focus on the modeling and recognition of more complex activities as explained above.

Complex activities usually involve several humans interacting with each other and other objects like buildings and vehicles. The literature on complex activity modeling and recognition can be classified into three categories: graphical, syntactic, and logical approaches [12, 24]. Dynamic Bayesian networks (DBNs), which encode complex conditional dependencies between a set of random variables, is a representative graphical model used for complex activities [7]. Motivated by grammars in language modeling, syntactic approaches specify how activities can be constructed from action primitives, and use these rules as grammars for visual activity recognition [8, 14]. Logic-based methods form logical rules to express common-sense knowledge to describe activities; for example, [23] represented each logical rule as first-order logic formula. All these approaches rely on either tracking body parts [7, 14], or object detection [7, 23], or atomic action/primitive event recognition [8, 23].

Tracks and precise primitive action recognition may not be easily obtained for complex/interactive activities since such scenes frequently contain occlusions and clutter. Additionally, these approaches may suffer due to poor tracking caused by changes in lighting conditions, actor appearance, video resolution etc. Spatio-temporal feature based approaches, like [4], hold more promise since no tracking is assumed. The statistics of these features are then used in recognition schemes [13]. However, as these approaches are built upon the statistics of extracted local features, spatial and long-term temporal correlations are often ignored.

The work in [2] models the video as a time-series of frame-wide feature histograms. It does bring the temporal aspect into picture; however the spatial structure information gets lost in the histogram representation. In [5], spatio-temporal relationships are considered by modeling activities as "strings of motion words". However, this method is limited to the availability of the tracks of objects involved. A matching kernel using "correlograms" was presented in [19], which looked at the spatio-temporal proximity among features. A recent work [15] proposed a match function to compare spatio-temporal relationships in the features by using temporal and spatial predicates. By considering the statistics of these relationships, the benefits of spatio-temporal modeling were demonstrated. The number of training videos needed to be large enough to represent the dataset.

Often, there are not enough training videos available for learning complex human activities; thus, recognizing activities based on just a single video example is of high interest. An approach for creating a large number of semi-artificial training videos from an original activity video was presented in [17]. A self-similarity descriptor that correlates local patches was proposed in [22]. A generalization of [22] was presented in [21], where spacetime local steering kernels were used. These methods require a sliding window through time and space.

2. Modeling Complex Activities Using String of Feature-Graphs

As local spatial-temporal features, such as STIPs [9], cuboids [4] etc., have shown success in representing interesting events in video, a video depicting a complex activity can be represented as a collection of these feature points spread out in space and time. Formally, let a video V be represented as $V = \{f^t_{x,y} | t \in [1, T]\}$ where $f^t_{x,y}$ is a feature point at spatial location x, y and time index t. Matching two videos would involve matching their corresponding feature points in a spatio-temporal order preserving manner. Consider an alternate representation of $V = \{F_1, F_2, \dots\}$, where each F represents a local collection of feature points, for example $F_1 = \{f^t_{x,y} | t \in [t_0, t_1)\}$, $F_2 = \{f^t_{x,y} | t \in [t_1, t_2)\}$, etc. Now, the spatio-temporal matching of two videos V_1 and V_2 would involve matching their individual feature collections $\{F^{(1)}_i | i = 1 \dots N_1\}$ and $\{F^{(2)}_i | i = 1 \dots N_2\}$ in a temporal order-preserving fashion, wherein the similarity measure between two feature collections would involve feature content matching as well as geometric structure matching. This representation of a video naturally leads us to a string representation, where local feature collections F form the elements of the string. In order to keep the structure information within each feature collection F, a graphical description is used and F is represented as a feature-graph. Therefore the temporally ordered collection of F forms a string of feature-graphs (SFGs). Fig. 2 visually explains the modeling process.

2.1. Feature-Graph Construction

To extract spatial-temporal features, we rely on the spatio-temporal interest point (STIP) detector proposed in [9]. The STIPs are detected by finding the center locations of local spatio-temporal volumes, which have large variations along both the spatial and the temporal directions, using a spatio-temporal extension of 2D Harris operator [9]. Note that other spatial-temporal feature representations can also be used in our framework. For a given video, we divide the detected STIPs into different time windows. In each time window, the collection of STIP features form a feature graph, where the STIP features form the nodes and the pairwise spatio-temporal distances between them are the edge weights. Then, matching two feature collections is equivalent to finding correspondences between two graphs. An efficient spectral solution to this problem was recently proposed in [10], which we use in this work (Sec. 3.1). Note that we use the terms feature collections and feature graphs interchangeably.

3. Spatio-temporal Matching of String of Feature-Graphs

As explained earlier, the match score between two videos is the string alignment score between their corresponding S-FGs. Since string alignment of any form requires a known method of measuring distance between the characters of the strings, we describe in the following subsections how we a) use a spectral technique to compute similarity between two feature-graphs (feature-graphs being the characters in the SFG strings) and b) use the computed feature-graph match scores to find the optimal alignment score between two S-FGs.

3.1. Matching Two Feature-Graphs

Computing the similarity between two feature graphs involves matching individual feature-descriptors (i.e., nodes) as well as pairwise feature neighborhood relationships (i.e., edges).

We represent each feature collection, i.e., each character in the string, as a fully-connected three dimensional graph where feature points form the nodes. Then the feature correspondence problem can be formulated as a graph matching problem by considering the matching between both nodes and edges. Given two such graphs, one being a feature collection from the testing video, P, with n_P nodes, and one being a feature collection from query video, Q, with n_Q nodes, we follow the spectral technique described in [10] to find correspondences between their respective feature points (nodes). This approach avoids the combinatorial explosion inherent to the correspondence problem by formulating it in closed form as a spectral analysis problem on a graph adjacency matrix.

An assignment (i, i') is defined as a correspondence between a pair of nodes from two graphs, where $i \in P$ and $i' \in Q$. For each candidate assignment $a = (i, i')$, there is a distance score between feature i and feature i' associated with it. Let L be a list (with length $n_L = n_P \times n_Q$) of all possible candidate assignments between features of P and Q. Given such a list, let a matrix $\mathbf{M}$ (size $n_L \times n_L$) store the affinities of every possible pair of assignments $(a, b) \in L$. Note that $\mathbf{M}(a, a)$ for $a = (i, i')$ measures how well the feature point i matches the feature point i', and $\mathbf{M}(a, b)$, where $a = (i, i')$ and $b = (j, j')$, describes the relative pair-wise relationships of points (i, j) in P with points (i', j') in Q. We define $d_n(i, i')$ as the distance between the nodes i and i'. It measures the Euclidean distance between the features of nodes i and i'. In order to account for scale, we consider the geometric structure of the graphs based on the angles between the edges in the graph. We define $d_e(\vec{ij}, \vec{i'j'})$ as the distance between edges (i, j) and (i', j') based on the angle difference between them. For candidate assignments $a = (i, i')$ and $b = (j, j')$, the elements $\mathbf{M}(a, a)$ and $\mathbf{M}(a, b)$ of matrix $\mathbf{M}$ are defined as

$$\mathbf{M}(a,a) = \begin{cases} \tau_n - d_n(i,i') & d_n(i,i') \leq \tau_n \\ 0 & d_n(i,i') > \tau_n \end{cases}$$

$$\mathbf{M}(a,b) = \begin{cases} \tau_e - d_e(\vec{ij}, \vec{i'j'}) & d_e(\vec{ij}, \vec{i'j'}) \leq \tau_e \\ 0 & d_e(\vec{ij}, \vec{i'j'}) > \tau_e \end{cases}$$

where τ_n is a pre-defined maximal distance between two features whose relationship should not be ignored and τ_e is a pre-defined threshold for edge difference. d_n and d_e are normalized between [0,1] and thus τ_n and τ_e are also chosen in that range.

Now, let x be an indicator vector of length n_L such that $x(a) = 1$ if candidate assignment $a = (i, i')$ represents a corresponding pair of nodes and 0 otherwise. We aim to find an optimal solution x^* which maximizes the score

$$x^* = \arg\max_x x^T \mathbf{M} x. \tag{1}$$

The solution to the above problem, x^*, gives the optimal correspondence between feature points in P and Q. It can be solved based on the greedy algorithm proposed in [10].

Once we estimate the optimal match, x^*, of two feature collections P and Q, their similarity can be measured by

$$sim(Q,P) = (x^*)^T \mathbf{M} x^*, \tag{2}$$

and the distance between them defined as

$$d(Q,P) = 1 - \frac{sim(Q,P)}{sim(Q,Q)}. \tag{3}$$

3.2. SFG Matching using Dynamic Time Warping

Recall that an SFG of a video is a time-ordered series of its feature-graphs. Matching two SFGs should be flexible, in that it should be robust to the different rates at which an activity might occur and also the actual length of the template video and the test video. This can be achieved by time normalizing the two SFGs. The speech recognition community has successfully used a dynamic programming approach termed dynamic time warping (DTW) [18] for non-linear time normalization. We borrow this idea and apply it to flexibly match two SFGs, hence making them robust to speed differences in different instances of the activity.

The aim of DTW is to minimize the local distortion between two sequences by finding an optimal warping function ϕ. For our case, the local distortion is defined as the sum of local pair-wise distances between their feature collections. Formally, for two SFGs $\mathcal{Q} = \{Q_1 \dots Q_{N_\mathcal{Q}}\}$ and $\mathcal{P} = \{P_1 \dots P_{N_\mathcal{P}}\}$, where $N_\mathcal{Q}$ and $N_\mathcal{P}$ are the number of characters (i.e. feature graphs) in $\mathcal{Q}$ and $\mathcal{P}$ respectively, the sequence distortion is defined as

$$D_\phi(\mathcal{Q}, \mathcal{P}) = \frac{1}{M_\phi} \sum_{k=1}^{K_\phi} d(Q_{\phi(k)}, P_{\phi(k)}) m_k \tag{4}$$

and the distance between the two SFGs can be computed as

$$D(\mathcal{Q}, \mathcal{P}) = \arg\min_\phi D_\phi(\mathcal{Q}, \mathcal{P}). \tag{5}$$

Here m_k are the path-weights, and $M_\phi = \sum_k m_k$ is a normalization factor. The details of the solution to this optimization problem can be found in [18]. The entire matching process is pictorially presented in Fig. 3.

3.2.1 Subsequence DTW for Continuous Video

In real applications, the test video is often a continuous video containing multiple persons performing multiple activities. Given a query video, which often contains only the desired activity, we would want to find a subsequence within the testing video sequence that optimally fits the query sequence, i.e., identify the fragment within the testing video that is most similar to the query. For this purpose, we utilize a variant of DTW – subsequence DTW [11], by releasing the restriction on the boundary condition, as explained below.

Let $\mathcal{Q} = \{Q_1 \dots Q_{N_\mathcal{Q}}\}$ and $\mathcal{P} = \{P_1 \dots P_{N_\mathcal{P}}\}$ be two SFGs of the query and testing videos respectively, where $N_\mathcal{P} >> N_\mathcal{Q}$. The goal is to find a subsequence $\mathcal{P}'(a^*, b^*) = \{P_{a^*} \dots P_{b^*}\}$ with $1 \leq a^* \leq b^* \leq N_\mathcal{P}$ such that

$$(a^*, b^*) = \arg \min_{(a,b):1 \leq a \leq b \leq N_\mathcal{P}} \left(D(\mathcal{Q}, \mathcal{P}'(a^*, b^*)\right). \quad (6)$$

The indices a^* and b^* can be computed by a small modification of the classical DTW algorithm in the generation of the accumulated cost matrix $\mathbf{C}$ used to describe the cost of aligning two sequences [11]. The goal of DTW is to find the minimal cost path through an accumulated cost matrix. By applying subsequence DTW, it can be shown that $b^* = \arg\min_{b \in [1, N_\mathcal{P}]} \mathbf{C}(N_\mathcal{Q}, b)$. $a^* \in [1, N_\mathcal{P}]$ is the maximal index such that path $(a^*, 1)$ belongs to the warping path.

It is usually the case that the database contains multiple instances of the activity that are similar to the query example. It is desirable to retrieve all the subsequences of $\mathcal{P}$ that are close to $\mathcal{Q}$ with respect to the DTW distance. This can be achieved by recursively repeating the above process. We present our implementation of matching continuous video using subsequence DTW in Algorithm 1. More details on subsequence DTW can be found in [11].

4. Experimental Results

In order to evaluate the efficacy of our method to recognize complex activities involving multi-person interactions, we conducted experiments on two state-of-the-art datasets with many challenging characteristics. These datasets were used in the recent activity recognition contest at ICPR 2010 [16]. We provide comparisons against other methods that have provided results on these data. First, in accordance with the motivation of the paper, we work in an example video-based retrieval framework wherein the algorithm is provided with one (or, at most, a few) video(s) depicting an action of interest. The aim is to retrieve videos which have similar activity as the query video(s). For this experiment, we work with the UCR VideoWeb activity dataset [3], which is very challenging due to the wide variation in the activities and the clutter in the scene.

For our second experiment, we test on the UT Interaction dataset [16], which is composed of both segmented and unsegmented videos, and include several pairs of interacting people simultaneously executing activities across different background, scale and illumination. We first evaluate the performance of our method on the segmented videos, and compare with previous systems. Then we show that our method is able to analyze continuous video using a subsequence DTW strategy as described in Section 3.2.1. These two datasets have moderate variations in view points.

Algorithm 1 Matching SFG of continuous video through subsequence DTW

```
Input:   $\mathcal{Q} = \{Q_1 \dots Q_{N_\mathcal{Q}}\}$   SFG of the query video
         $\mathcal{P} = \{P_1 \dots P_{N_\mathcal{P}}\}$   SFG of the testing video
         $\tau \in \mathbb{R}$                               cost threshold
Output:  Ranked list of all subsequences of $\mathcal{P}$ that have a DTW
         distance to $\mathcal{Q}$ below the threshold $\tau$.

1. Initialize the ranked list to be an empty list.
2. Construct accumulated cost matrix $\mathbf{C}$ whose elements are defined
   as
     $\mathbf{C}(n,1) = \sum_{k=1}^{n} d(Q_k, P_1), n \in [1, N_\mathcal{Q}],$
     $\mathbf{C}(1,m) = d(Q_1, P_m), m \in [1, N_\mathcal{P}],$
     $\mathbf{C}(n,m) = \min\{\mathbf{C}(Q_{n-1}, P_{m-1}), \mathbf{C}(Q_{n-1}, P_m),$
         $\mathbf{C}(Q_n, P_{m-1})\} + d(Q_n, P_m).$
3. Define a distance function: $\Delta(b) \triangleq \mathbf{C}(N_\mathcal{Q}, b), b \in [1, N_\mathcal{P}]$.
4. Determine $b^* \in [1, N_\mathcal{P}]$ that gives minimal $\Delta$.
5. If $\Delta(b^*) > \tau$ (which means no additional subsequence of $\mathcal{P}$
   close to $\mathcal{Q}$ exists), then terminate the procedure.
6. Compute the corresponding DTW-minimizing index $a^* \in$
   $[1, N_\mathcal{P}]$ using standard DTW algorithm, which searches optimal
   warping path in $\mathbf{C}$ in reverse order of the indices starting with
   $(N_\mathcal{Q}, b^*)$.
7. Extend the ranked list by the subsequence $\mathcal{P}'(a^*, b^*)$.
8. Set $\Delta(b) \triangleq \infty$ for all $b$ within a suitable neighborhood of $b^*$.
9. Continue with Step 4.
```

4.1. Query-based Retrieval Results on UCR VideoWeb Dataset

The portion of the UCR VideoWeb activity dataset [3] we work on (details can be obtained from the authors) involves up to 10 actors interacting in various ways with each other, vehicles and facilities. The activities were: people meeting, people following, vehicles turning, people dispersing, shaking hands, gesturing, waving, hugging and pointing. We work with video clips from this dataset, report the best matches found by our system and accordingly present and analyze the accuracy/false positive rates.

For this experiment, we proceed by taking a small video clip depicting a complex activity and search the dataset for matches. The STIP features for the query and the dataset videos are computed. The query and dataset videos were uniformly segmented into temporal segments, the feature points in each segment forming a feature graph, and the string of time ordered graphs forming the SFG descriptor. In our implementation, the length of each segment is set to be 20 frames. Next we find the pair-wise correspondences between each of the feature collections from the query video

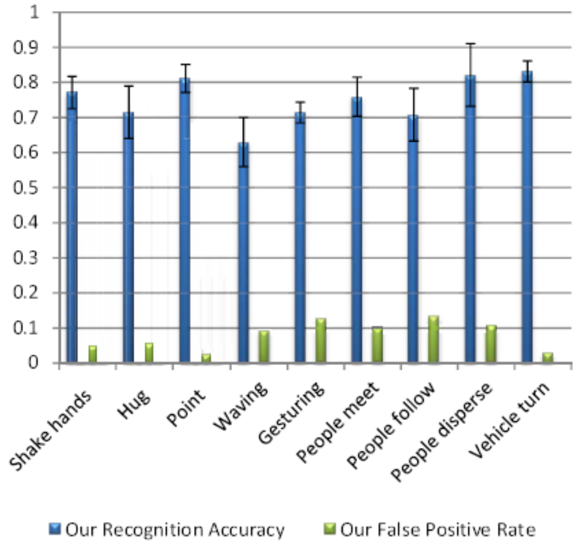

Figure 4. Recognition accuracy and false positives on 9 activities from the UCR VideoWeb dataset in a query-based retrieval framework. Standard deviation in performance (accuracy) for different queries is marked on the bars.

with those of dataset videos using the spectral solution in Section 3.1. We finally perform the DTW match across the entire query and dataset SFGs (composed of time ordered feature-graphs) based on the local match scores calculated.

The results from our first experiment involving query-based activity video retrieval are shown in Fig. 4. For each activity class, we chose 3 random videos from the samples of that class to be the query. The results reported here are obtained by averaging across the 3 test cases. Recognition on activities like vehicle turning and shaking hands performed especially well since they continue for longer time periods and hence generate better feature points. On the other hand, activities such as point happen in a short amount of time and are thus more difficult to recognize. We found that the recognition results obtained based on a single sample video generate higher false positive rates. This is justifiable due the fact that in a single query-based retrieval framework, there is no statistically reliable way to set the acceptance threshold.

We also studied variation in recognition performance of our method with change in query videos. The standard deviation in the scores for different query-videos is marked in Fig. 4. In line with our previous argument, short-duration activities such as "pointing" had higher variability. The activity "hug" was confused with background clutter or actors crossing each other.

Finally, we show some results on activity retrieval using one query video in Fig. 5. The query videos are shown on the left and the other three columns show the top three best matches. The bounding boxes of the sub-graphs that best match the feature graphs of query video are also shown. This demonstrates the capability of our system in locating the activities of interest in the spatial-temporal video volumes.

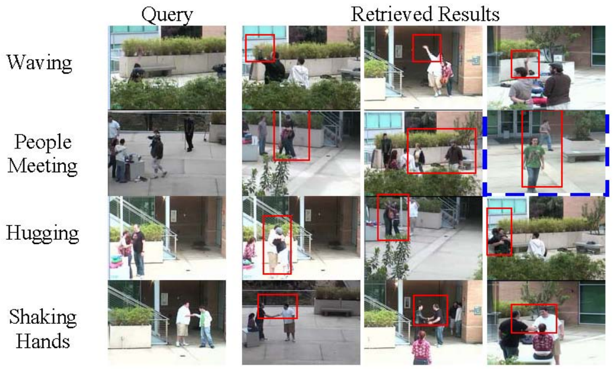

Figure 5. Retrieval results: The left column depicts the query videos and the other three columns are the best matches on UCR VideoWeb dataset. The bounding boxes of the sub-graphs that best match the feature graphs of the query video are shown. A blue dash box represents an incorrect match.

4.2. Performance Comparison on UT-Interaction Dataset

In the UT Interaction dataset [16], the interaction activities which we looked at are shaking hands, hugging, pointing, punching, kicking and pushing. We first test our method on the segmented videos. In order to compare with previous systems, we use an experimental setting similar to [15], which proposed a supervised learning method for the same set of activities on this dataset. We randomly choose two among the ten sets to form the training set and leave out the other sets for testing.

Similar to [15], we use a voting scheme to decide whether a testing video contains the specific action when multiple labeled training examples are available. We compute the DTW aligning cost between the SFGs of the testing video and each query video containing a specific action and count the instances that the DTW distance is less than a threshold. Based on this number (i.e., number of similar training videos), the system makes a decision on the recognized activity.

We first compare our proposed approach with existing methods on 10 atomic activities from segmented videos: stretch arm, withdraw arm, stretch leg, lower leg, and shift forward, repeated for both left and right sides [16]. A binary decision is made for each type of activity, and the performance is averaged. The ROC curves are compared in Figure 6. It can be observed that when training examples are available, our system can achieve an accuracy similar to [15] and

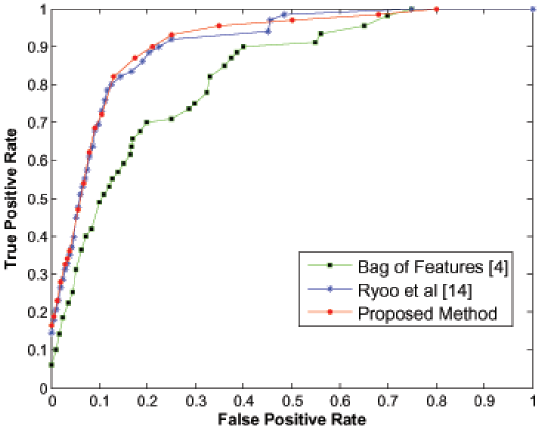

Figure 6. ROC curves of action recognition on UT Interaction dataset. It can be observed that when training examples are available, the performance of our method is significantly better than Bag-of-Features [4], and similar to [15] in accuracy.

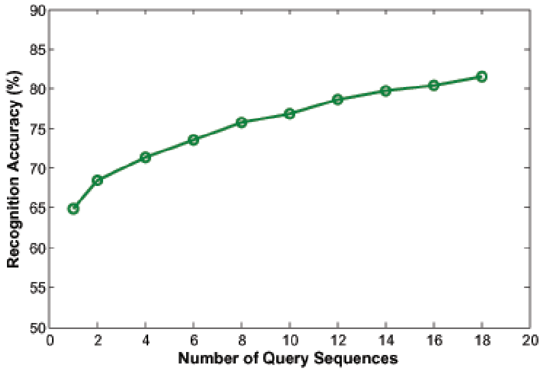

Figure 7. The recognition accuracy of our method with respect to number of query examples on the UT Interaction dataset. It can be seen that when number of query example decreases, the performance of our method does not drop precipitously.

significantly higher accuracy than Bag-of-Feature approach [4]. We also test the performance of our method with varying number of query examples as shown in Figure 7. It can be seen that when number of query example decreases, the performance of our method does not drop precipitously. Even with just one query, our average recognition accuracy is 65%. This demonstrates the ability of our method to work in the situation when only a single query video is present. This is major difference with other methods like [4] or [15].

Next, we verify that our system is able to recognize multiple complex activities from continuous videos on the UT Interaction dataset (note that we dealt with continuous video in the UCR Videoweb dataset). We were able to achieve high recognition scores and lower false positive rates. We compare our results with previous methods in Figure 8. Our overall performance on the UT Interaction dataset is superior to Bag-of-Feature approach. Here the results of Bag-of-Feature approach are reported on segmented video clips, while our results and [15] are reported on continuous video (it is probable that the Bag-of-Features approach will perform even worse on unsegmented video). Our results are similar to that in [15] for some activities and better for others. However, our approach can use only a single query to perform recognition as demonstrated in Figure 7 and hence has a wider generalizability. In [16], recognition results of several approaches are reported on the same dataset; the average recognition accuracy is in the range from 0.49 to 0.88. Our performance is comparable to the best performance in [16]. Note that the experiment settings in [16] are slightly different from ours. Their results are reported by leaving one out among a set of ten for testing and using the other 9 for the training, and the videos are segmented, while we use 2 sets as labeled query videos and test on 8, and we work with continuous videos (a significantly harder problem).

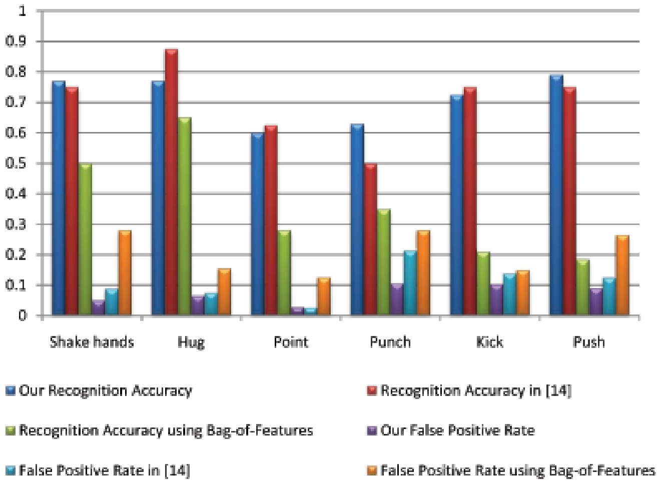

Figure 8. Recognition accuracy on the UT-Interaction dataset by using voting scheme on top of SFG model.

Finally, we test our system on activity retrieval using one query video on UT Interaction dataset. Some results are shown in Fig. 9. The query videos are shown on the left and the other three columns show the top three best matches.

5. Conclusion

In this work, we demonstrated that spatio-temporal relationships are critical to discriminate real-world activities. We proposed a model based on a string representation of the video which respects the spatio-temporal dynamics of the complex activities. We leveraged a graph-based spectral technique to find correspondences between local feature collections. Finally, the string formed by the time-ordered set of local feature collections was matched with

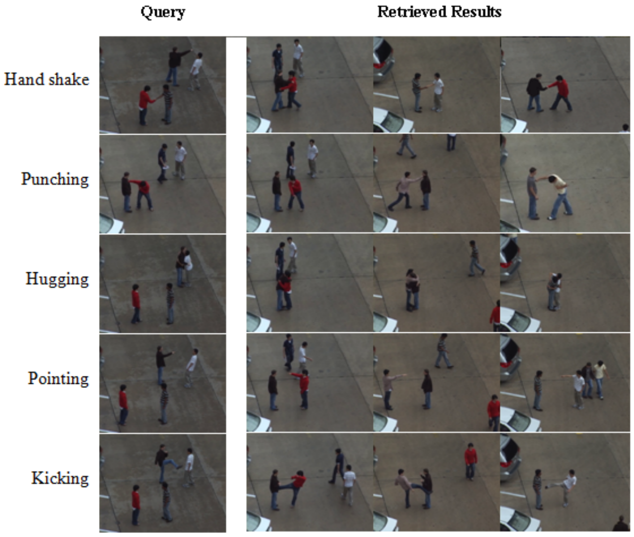

Figure 9. Retrieval results: The left column depicts the query videos and the other three columns are the best matches on UT-Interaction dataset.

other strings in a dynamic programming framework to obtain the match score. This match score was used to classify a test video as being similar or non-similar to the template video. Our experiments demonstrated the effectiveness of our approach to successfully recognize and localize complex activities even with multiple interacting actors.

References

[1] P. A. Anderson. *Nonverbal Communication: Forms and Functions*. Waveland Press, 2nd edition, 2008. 1

[2] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009. 3

[3] G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda. Videoweb dataset for multi-camera activities and non-verbal communication. In *Distributed Video Sensor Networks*. Springer, 2011. 2, 5

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005. 3, 7

[5] U. Gaur, B. Song, and A. K. Roy-Chowdhury. Query-based retrieval of complex activities using "strings of motion-words". In *IEEE Workshop on Motion and Video Computing*, 2009. 3

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Dec. 2007. 1

[7] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007. 2

[8] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000. 2

[9] I. Laptev. On space-time interest points. In *International Journal of Computer Vision*, 2005. 3

[10] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE Intl. Conf. on Computer Vision*, 2005. 3, 4

[11] M. Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007. 4, 5

[12] N. Nayak, R. Sethi, B. Song, and A. Roy-Chowdhury. Motion pattern analysis for modeling and recognition of complex human activities. In *Guide to Video Analysis of Humans: Looking at People*. Springer, 2011. 2

[13] J. C. Niebles, H. Wang, and L. F. Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, Sept. 2008. 3

[14] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 2

[15] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *IEEE Intl. Conf. on Computer Vision*, 2009. 2, 3, 6, 7

[16] M. S. Ryoo, C.-C. Chen, J. K. Aggarwal, and A. Roy-Chowdhury. An overview of contest on semantic description of human activities (SDHA). In *Intl. Conf. on Pattern Recognition*, 2010. 2, 5, 6, 7

[17] M. S. Ryoo and W. Yu. One video is sufficient? human activity recognition using active video composition. In *IEEE Workshop on Motion and Video Computing*, 2011. 3

[18] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, Feb. 1978. 4

[19] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei. Spatial-temporal correlations for unsupervised action classification. In *IEEE Workshop on Motion and Video Computing*, 2008. 3

[20] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Intl. Conf. on Pattern Recognition*, 2004. 1

[21] H. J. Seo and P. Milanfar. Detection of human actions from a single example. In *IEEE Intl. Conf. on Computer Vision*, 2009. 3

[22] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007. 3

[23] S. Tran and L. S. Davis. Visual event modeling and recognition using Markov logic networks. In *Euro. Conference on Computer Vision*, 2008. 2

[24] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. In *IEEE Trans. on Circuits and Systems for Video Technology*, 2008. 2