Spatio-Temporal Phrases for Activity Recognition

Yimeng Zhang[†⋆], Xiaoming Liu[‡], Ming-Ching Chang[‡]
, Weina Ge[‡], and Tsuhan Chen[†]

[†]School of Electrical and Computer Engineering, Cornell University
[‡]GE Global Research Center, 1 Research Circle, Niskayuna, NY
[†]{yz457,tsuhan}@cornell.edu [‡]{liux,changm,gewe}@research.ge.com

Abstract. The local feature based approaches have become popular for activity recognition. A local feature captures the local movement and appearance of a local region in a video, and thus can be ambiguous; e.g., it cannot tell whether a movement is from a person's hand or foot, when the camera is far away from the person. To better distinguish different types of activities, people have proposed using the combination of local features to encode the relationships of local movements. Due to the computation limit, previous work only creates a combination from neighboring features in space and/or time. In this paper, we propose an approach that efficiently identifies both local and long-range motion interactions; taking the "push" activity as an example, our approach can capture the combination of the hand movement of one person and the foot response of another person, the local features of which are both spatially and temporally far away from each other. Our computational complexity is in linear time to the number of local features in a video. The extensive experiments show that our approach is generically effective for recognizing a wide variety of activities and activities spanning a long term, compared to a number of state-of-the-art methods.

Key words: Activity Recognition, Spatio-Temporal Phrases

1 Introduction

Activity recognition in videos has attracted increasing interest recently. An activity can be defined as a certain spatial and temporal pattern involving the movements of a single or multiple actors. The recognition task requires capturing enough spatial and temporal information to distinguish different activity categories, while handling the large intra-category variations. Also, most video analysis applications, such as surveillance, require high computational efficiency.

⋆ This work was done while the first author was visiting GE Global Research as an intern. This work was supported by the National Institute of Justice, US Department of Justice, under the award #2009-SQ-B9-K013. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Justice.

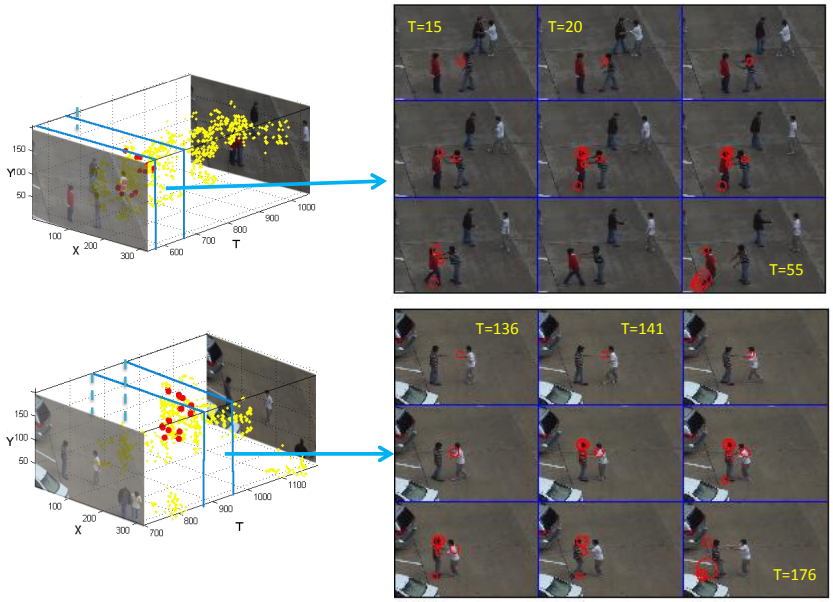


Fig. 1: An example co-occurring spatio-temporal (ST) phrase of two video sequences. The left images show space-time locations of the local features in each video. Red points indicate the features composing the co-occurring phrase. The right images show the frames (sampled at rate 5, frame index is shown) composing the phrase. Red circles indicate the local features composing the phrase. The ST phrase captures the causality relationships among body parts from different individuals of a long time span for the same activity "push".

The bag-of-words (BoW) model based on local features is very popular for activity recognition due to its robustness to real-world environments [1–5]. Despite its computational efficiency and intra-category invariance, the BoW model discards any structural spatial and temporal information among the local features. However, a local word only captures the local movement of a particular region, and can be ambiguous sometimes. For example, using the local features alone, the activities of "push" and "kick" can be quite similar when it is hard to tell whether a movement is from a person's hand or foot due to video resolution.

In order to better distinguish different types of activities, many works have been done to incorporate spatio-temporal relationships among the local features [6–12]. In particular, approaches that model the mutual relationships among the words are spatio-temporally shift invariant [6, 8, 13–15], and thus do not require the knowledge on where and when the activity occurs in a video. However, due to the computational concern, these works have one or more of the following limitations: 1) only create a combination (phrase) with words in a local space or time neighborhood [8,14,15], and thus cannot capture the *long-term* interactions of different body parts, which can be extracted from different individuals; 2) only consider a pair of local words [6, 13], and therefore cannot encode interactions among more than two time stamps or local regions; 3) only encode the distances

between the words, while discarding the temporal ordering and spatial layout among them [6,13], and thus is incapable of modeling the casuality relationships.

Along the direction of modeling the mutual relationships among words, this paper presents a generic activity recognition approach that addresses *all* of the aforementioned limitations. Our approach is formed around a novel concept named the "spatio-temporal phrase (ST phrase)". A ST phrase of length k is defined as a combination of k words in a certain spatial and temporal structure, including their order and relative positions. Hence, it encodes rich temporal ordering and spatial geometry information of local words. Fig. 1 illustrates example ST phrases in two videos, both of which include the "push" activity at different time stamps. The ST phrases capture the patterns, e.g. the hand movement of one person in one frame and the foot movement of another person several frames later. The illustrated phrase consists of local movements that do not occur within the same space-time neighborhood region. Moreover, the same phrase, which characterizes the "push" activity, occurs at different locations and time stamps in the two videos.

The algorithm for identifying ST-phrases extends a recent work [16] proposed for image retrieval, which efficiently detects co-occurring 2D phrases in two static images. We extend the work to the space-time domain, and show that we can identify all ST phrases of any length in a video in time linear to the number of local features. We also provide an algorithm that makes the ST phrases speed invariant to deal with different speeds or durations of the activities. The co-occurring ST phrases are further used to compute a kernel for discriminative learning with SVM, which implicitly determines the importance of different phrases. In addition, we propose an algorithm that can update the kernel values incrementally when a new frame arrives, thereby enabling us to efficiently detect the activities from video streams in an online fashion.

1.1 Related Work

Many techniques have been proposed to incorporate spatial and temporal information into the BoW model. *Space-time pyramid matching* [2] captures absolute spatio-temporal locations of the local features with quantized space-time bins; therefore, it is not spatio-temporal shift invariant. Aligned space-time pyramid matching [10] relaxes the fixed bin matching, and identifies optimal matching between the bins of two videos. The method is shift invariant at the cost of discarding the spatio-temporal ordering among different bins. *Trajectories of the local features* [12,17,18] are created by feature tracking methods and are capable of incorporating temporal information of the same feature in a certain period. However, exploring rich spatio-temporal relationships of different trajectories remains challenging. *Hough Voting* or the *Implicit Shape Model* [7, 19] allow the local features to vote for the action center in both space and time domains, and can encode space-time information of the words relative to the action center; however, the recognition process requires an iterative EM process to predict the center [7] or a preprocess for detecting the bounding box of the person [19]. *Mutual word relationships* have been explored [6, 8, 13–15]. As discussed in Section

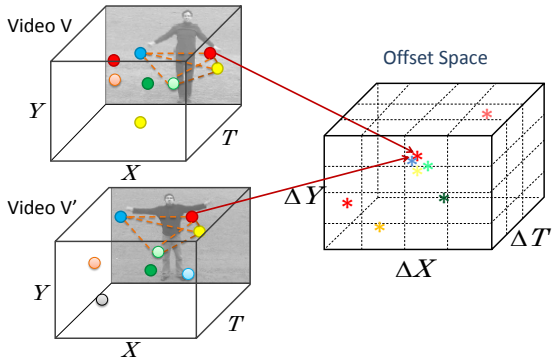


Fig. 2: 3D correspondence transform. Circles represent local features, and colors represent word assignments. A co-occurring ST phrase of length 4 is shown.

1, due to the high computational cost, previous works do not capture higher order and long range dependencies.

2 Approach

In activity recognition, a video can be represented as a collection of visual words in the xyt-space, which are created by clustering the local descriptors detected at local space-time volumes. Various promising local feature detectors, descriptors and clustering methods [1, 3–5] can be adopted. Specifically, a video is denoted as $V = (w_i, x_i, y_i, t_i)$, where w_i is the word entry for feature i, and $(x_i, y_i), t_i$ are the space and time index of the feature respectively.

2.1 Bag of Spatio-Temporal Phrase

A spatio-temporal (ST) phrase of length k is defined as a combination of k local words in a certain spatial and temporal structure, including their order and relative positions. We represent a video as a bag of ST phrases (BoP): $V = \{P_i\}$, which encodes much richer spatio-temporal information than the BoW model. A straightforward implementation of using the BoP is to calculate the histogram of the phrases and then forward to a classifier like SVM. However, the length of the histogram, which is the number of possible ST phrases, is exponential to the phrase length k.

2.2 3D Correspondence Transform

Rather than representing a video with an intractably long histogram of phases, we directly resort to estimating the distance of two videos by finding the co-occurring ST phrases and such distance can be easily utilized by SVM learning via the kernel trick. A co-occurring ST phrase of length k in two videos must have the same k words and the same space-time layout, as shown in Fig. 1. The

algorithm for identifying co-occurring ST phrases is an extension of the correspondence transform algorithm proposed in [16,20] for 2D images. As illustrated in Fig. 2, for each pair of local features (w_i, x_i, y_i, t_i) and (w'_i, x'_i, y'_i, t'_i) that has the same word assignment $(w_i = w'_i)$ in the two videos V and V', we calculate their space-time offset as $(\Delta x_i, \Delta y_i, \Delta t_i) = (x_i - x'_i, y_i - y'_i, t_i - t'_i)$, and create a vote in the XYT offset space $(\Delta X, \Delta Y, \Delta T)$.

In the quantized XYT offset space, if we have k votes at the same location, we have a co-occurring ST phrase of length k in the two videos with the same word entries and the same spatio-temporal structure. Thus, to count the number of co-occurring ST phrases of length k, we can simply use the XYT offset space. Whenever we have n_l $(n_l \geq k)$ votes at the same location $l = (\Delta x, \Delta y, \Delta t)$ in the offset space, we increase the number of co-occurrences by n_l choose k, $\binom{n_l}{k}$, since the same space-time structure of n_l words also indicates the same structure of $\binom{n_l}{k}$ number of k word. Thus, the number of co-occurring length-k ST phrases $K_k(V, V')$ is calculated as:

$$K_k(V, V') = \sum_l \binom{n_l}{k}. \tag{1}$$

Activity Speed Variations: the same activity category can occur with different speeds and time durations in different videos. For example, people may "run" at different speeds, or take different time to form a group in order to "fight". To explicitly deal with this problem, we would like to detect k words from two videos as the same ST phrase if their only structural difference is the temporal rates. To this end, we add another dimension to the offset space, which indicates the temporal scaling Δd, and allow each pair of words vote for multiple Δd. The offset location of a pair of features with the same word assignment is: $(\Delta x_i, \Delta y_i, \Delta t_i, \Delta d) = (x_i - x'_i, y_i - y'_i, t_i - t'_i \times \Delta d, \Delta d)$.

In this 4D space, if we have k votes at the same location, we have a co-occurring ST phrase with a particular temporal scale difference Δd. In the experiments, we set the scale Δd to $1, 1.5, 1.5^2, .., 5$, which tolerates up to 5 times speed difference between activities of two videos.

Scale Variations: people in an activity can appear in different scales for different videos. The scale invariance can be obtained by adding another dimension to the offset space as scale difference, similar as the solution for the speed invariance. We can also utilize the detected scales for local features and vote for only the scale difference that is the same as that of the local features. Therefore, we still create one vote for each pair of same local words when speed is not considered.

2.3 ST Phrase Kernel

Using similar proof as [20] for 2D phrases, we can show that the number of co-occurring ST phrases of length k equals exactly the inner product of the bag of length-k-ST phrase histograms of two videos. Therefore, we can use the number of co-occurrences (Eqn. 1), which can be efficiently computed, as the kernel to

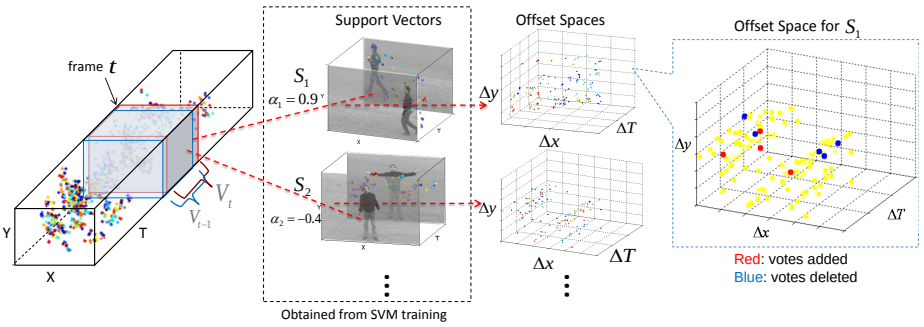


Fig. 3: Illustration of the incremental kernel computing algorithm. The support vectors (selected training videos) S and their coefficients α (second column) are obtained during training, while the offset spaces for each support vector (third column) need to be computed online during testing. The right most image shows the enlarged offset space between the video segment V_t and support vector S_1.

the SVM. The kernel obviously satisfies the Mercer's condition. The final kernel value between two videos V and V' is a weighted summation of the normalized kernel values for the ST phrase of all lengths:

$$K(V, V') = \sum_{k=1}^{\infty} \mu^k \frac{K_k(V, V')}{\sqrt{K_k(V, V) K_k(V', V')}}, \tag{2}$$

where μ is a factor (greater than 1) chosen to ensure that the normalized kernel values for different k are in similar scales, i.e. the values are not too small.

The SVM with the above kernel projects the word space to the higher order ST phrase space for classification. During training, we use both positive and negative examples to compute the kernel matrix, which is used to train a SVM. The SVM will implicitly assign different weights to different ST-phrases by learning the coefficients for different training videos. Let the training videos that have non-zero weights (support vectors) be S_j and the coefficients be α_j, the decision score for a test video V of a particular activity category is computed as:

$$Score(V) = \textstyle\sum_j \alpha_j K(V, S_j). \tag{3}$$

2.4 Incremental Kernel Computing for Activity Localization

Some applications, such as surveillance or security monitoring, require online activity detection from real-time video streams instead of recognizing temporally segmented video clips in a batch mode. One simple yet commonly used approach is to make predictions for every frame based on the video segment composed of the current frame and the previous T frames as the context. If some detection delay is allowed, the following T' frames are also included in the segment.

As illustrated in Fig. 3, to determine the activity category of the current frame, we need to compute the kernel values of the corresponding video segment

Algorithm 1: Compute kernels for frame t and a support vector S_j
1. Initialize $K_k(V_t, S_j) = K_k(V_{t-1}, S_j)$
2. For each same word pair in video S_j and frame t of the test video (a) Compute their offset $l = (\Delta x, \Delta y, \Delta t)$ (b) Increase $K_k(V_t, S_j) += \binom{n_l}{k-1}$, ($n_l$ is the original votes at l) (c) Add one vote to location l Save the list of offset locations contributed by frame t
3. For each saved offset location l of frame $t - T$ (a) Decrease $K_k(V_t, S_j) -= \binom{n_l - 1}{k-1}$ (a) Delete one vote from location l

with the support vectors as in Eqn.(3). The essential for obtaining the kernel value of a video segment V_t and a support vector S_j is to compute the offset space which is created with the votes from the local features. As discussed in Section 2.2, this computational complexity is linear to the number of local features in the segment.

However, note that the video segment of the current frame and that of the previous frame have a significant overlap; we can further accelerate the kernel computation with an incremental updating algorithm. The difference between the video segments of the current and the previous frame only involves two frames, i.e., the current frame t, and the frame at $t - T$ (Fig. 3). If we have stored the offset spaces for the previous frame segment V_{t-1}, to compute the new offset space for the current V_t and a support vector S_j, we only need to add the votes made by the local features of the current frame, and delete the votes that were contributed by the features of the frame $t - T$.

When we add a vote at location l in the offset space that originally had n_l votes, the number of co-occurring length-k ST phrases (Eqn.(1)) would be changed as follows:

$$K_k^{new}(V, V') = K_k^{old}(V, V') - \binom{n_l}{k} + \binom{n_l + 1}{k} = K_k^{old}(V, V') + \binom{n_l}{k-1}. \quad (4)$$

The change for the number of co-occurring length-k ST phrases when we delete a vote can be calculated similarly.

The algorithm for updating the kernel values is listed in Algorithm 1. Consequently, to compute the kernel values for the current frame and a support vector, we only need to perform the operations of Eqn.(4) $2N_t$ times, with N_t being the number of local features at frame t. This acceleration enables us to consider the long-term context (a large number of previous frames) without increasing the recognition time for each frame. This property is especially useful when detecting complex activities that usually last a long period of time.

2.5 Computational Complexity

The computational complexity for calculating the number of co-occurring ST phrases (Section 2.2) is linear to the number of same word pairs in two videos. Let N be the number of local features per video. The computation is $O(N^2)$ for worst case when all the words are the same, but $O(N)$ in practise. For memory

usage, although the offset space has high dimension (3D or 4D), it is actually very sparse with most locations having zero votes. Since we only care about the number of votes that fall to the same location, we can simply store the non-zero locations, which is again linear to the local feature number per video. Thus, we have the same $O(N)$ computational complexity with BoW for classifying a video clip, when BoW uses a non-linear kernel, such as χ^2 or Gaussian kernel.

With the incremental kernel computing, computing the kernel value between a support vector and a video segment defined for the current frame depends only on the number of words in the current and previous frames. Therefore, making prediction for one frame requires $O(SN_t)$ computations in total, where S is the number of support vectors, and N_t is the number of words per frame.

3 Experiments

We first perform experiments on single person activities with the KTH dataset [21], YouTube Action dataset [5], and a hospital surveillance dataset. The KTH dataset is created in a controlled environment, while the hospital dataset is collected from real surveillance cameras in a more complex environment and the YouTube Action dataset includes more pose and scale variations taken with shaky cameras.

Then we evaluate our performance for the multiple-person activity recognition problem, which is an up-coming topic that recent work are addressing [8, 22, 23], with the UT interaction dataset [24] and the MPR dataset [22]. On the MPR dataset, we evaluate our incremental algorithm (2.4) for online activity detection.

Baseline: We aim to verify that the proposed bag-of-ST-phrase (BoP) representation outperforms BoW with exactly the same local features, same visual words, and same classifier (SVM). For BoW, we use the χ^2-kernel, which already captures the co-occurrence statistics of different words in a video. Therefore, the comparison of BoW and BoP verifies whether the spatial-temporal layout of the words helps activity recognition. Moreover, we compare favorably with other state-of-the-art approaches [2, 13, 19, 25, 26] that incorporate spatial and/or temporal information to the BoW representation.

3.1 Single Person Activity: KTH Dataset

The KTH dataset includes 2391 videos performed by 25 different subjects. We use the same local features (HOF) as [2]. Features are computed with the published code by the authors [1].

Vocabulary Size: Table 1(a) compares BoP with BoW using different vocabulary sizes under the same settings as [2]: videos from 16 subjects for training and videos from the other 9 for testing. The performance of BoW decreases when a smaller vocabulary size is used since local words are more ambiguous. Meanwhile, BoP achieves similar accuracies even with a smaller vocabulary, since the

[1] For fair comparison, feature HOF uses version 1.1 code, same as [2].

Method	500	1000	2000	4000
BoW	90.8	91.2	91.3	91.5
BoP	**94.0**	**93.8**	**94.1**	**94.0**

(a)

Setting	BoW	BoP	[2]	[19]	[13]	[26]
16 train / 9 test	91.5	**94.0**	91.8	n/a	n/a	91.1
5-fold	92.9	**94.6**	n/a	92.0	n/a	n/a
leave-one-out	91.9	**95.5**	n/a	n/a	94.2	93.8

(b)

Table 1: (a) Classification accuracies (%) with different vocabulary sizes on the KTH dataset. (b) Accuracies (%) under each train and test setting on the KTH dataset. We compare BoP with other methods that incorporate spatio-temporal relationships to BoW.

ST phrases capture the spatio-temporal relationships among the words, thereby increasing the discrimination.

Comparison: Table 1(b) compares BoP with other methods that encode spatial relationships into BoW. By capturing higher order and more detailed spatio-temporal information with the ST phrases, our approach outperforms other methods: spatio-temporal pyramid matching [2], Hough voting [19], correlogram [13], and predefined spatio-temporal relationship rules [26].

3.2 Single Person Activity: Hospital Surveillance Dataset

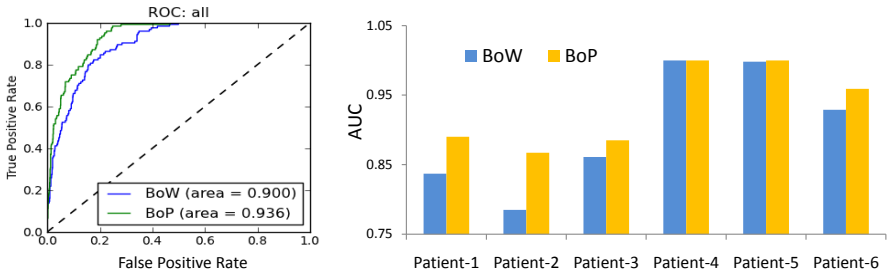
This dataset includes the realistic surveillance videos of 6 patients in the private sickrooms of a hospital[2]. The goal is to detect abnormal behaviors of the patients (*get up from the bed*) from other normal ones, which is also a real application used in the hospital. The environment of this dataset is much more complex than the KTH dataset, since the patients conduct many variations of normal activities and many noisy motion features are detected on non-person objects or other people; eg. the activity that a nurse cleans the bed may lead to a false positive. To collect the ground-truth, we segmented the videos into 10-second clips, and manually labeled each clip. In total, the dataset has 124 positive (abnormal) examples, and 1067 negative examples. We perform a leave-one-patient-out cross-validation experiment. We extract similar features as the KTH dataset, and create a vocabulary of size 500. Figure 4 shows that our approach outperforms the BoW method.

3.3 Single Person Activity: YouTube Action Dataset

The YouTube action dataset [5] consists of 11 categories with 1168 videos. The videos are separated into 25 groups, which are taken in different environments or by different photographers. Following [5], we perform a leave-one-group-out cross-validation experiment. We extract HOG/HOF features with the code [2], and create a vocabulary of size 2000 with K-means.

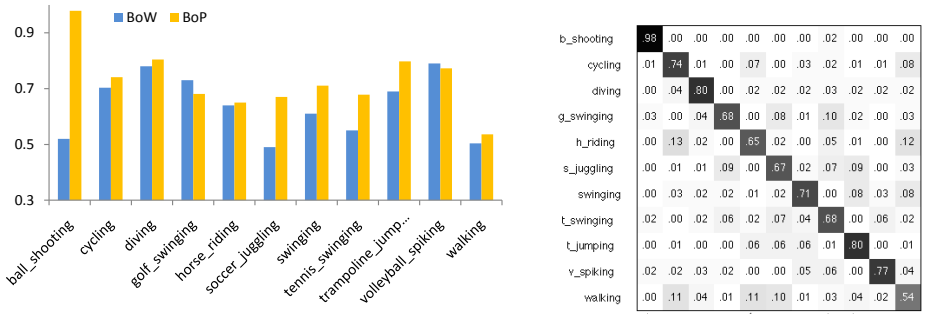
Figure 5 shows the classification performance. The BoW achieves 63.7% accuracy averaged over the 11 categories. Our implementation of BoW achieves

[2] For patient privacy, we cannot show the example images from the dataset.



(a) (b)

Fig. 4: Hospital Surveillance Dataset. (a) ROC curve for all patients. (b) The AUC score for each patient.



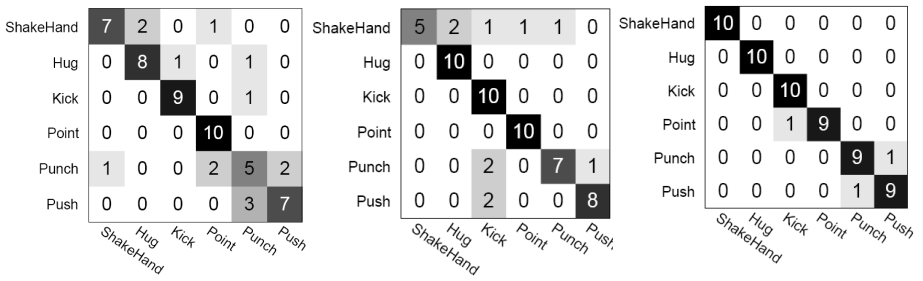
(a) (b) average: 72.9%

Fig. 5: (a) Accuracies for each category of the YouTube action dataset. Average accuracies for BoW and BoP for 63.7% and 72.9%, respectively. (b) Confusion matrix using BoP.

similar performance as the one in [5] (65.4%) when similar (motion) features are used. With BoP, we improve the performance by 9.2%. Our BoP result (72.9%) is also better than the final result of [5] (71.2%), where additional static features, feature pruning techniques, and semantic vocabulary learning are adopted. These techniques that improve the local features are complimentary to our work. Therefore, further improvement can be expected when these techniques are applied. We notice the main improvement is the discrimination of different *swing* activities, and different categories that both involve *jump* actions, such as *basketball shooting*, *trampoline jumping*, and *horse riding*. The main reason is that BoP can capture the ST relationships among the local movements.

3.4 Multiple-Person Activities

We evaluate our approach using the UT interaction dataset [24], which consists of six activities of two-person interactions: *shake hands, hug, kick, point, punch*, and *push*. There are two sets in the dataset. Set 1 was recorded with relatively stable camera, while Set 2 was taken on a lawn in a windy day. Background is moving slightly (e.g. tree leaves move), and the set contains more camera jitters. For each set, every type of activity has 10 video clips. These high-level



(a) BoW (b) Hough Voting [25] (c) BoP

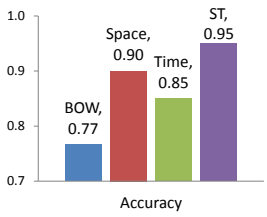
Fig. 6: Confusion Matrices on the UT interaction dataset (Set 1).

activities are more complex, and involve a combination of atomic movements of two people over a period of time. Therefore, the BoW model which only captures the atomic movements may work poorly. This dataset was used in the activity recognition contest at ICPR 2010 [24]. We use the cuboid [1] as local features and a vocabulary of size 500.

Comparison: We compare BoP with BoW and the best results reported in the contest [25]. We adopt the classification settings described for the contest, where a 10-fold leave-one-set-out cross validation is performed. Figure 6 shows the confusion matrix for different methods on Set 1. Our implementation for BoW with a χ^2 kernel for SVM demonstrated 76.7% accuracy, which is similar to rates of the BoW method reported in the contest. Table 2 (b) compares the performance on both Set 1 and Set 2. By modeling the atomic moves to the activity center, the Hough voting based method [25] improves BoW by around 7%. Using BoP, we further outperform the Hough voting method by around 10%.

Analysis: Since the activities in this dataset involve more atomic movement interactions of different body parts and from different persons, the rich spatio-temporal interactions modeled with BoP are beneficial. As shown in Fig. 6, BoW confuses *push* vs. *punch*, and *shake hands* vs. *hug*, since the local movements of each body part are similar for these activities. With the ST phrases, we can capture the spatio-temporal combinations, thereby better discriminating these activities.

Analysis for Spatial and Temporal Information: We show the benefit of simultaneous spatial and temporal modeling with ST-phrases in Table 2(a). To incorporate space alone, we still use the BoP algorithms we proposed in this paper, but discarding the temporal domain when computing the co-occurring ST phrases. In other words, the temporal domain is modeled with BoW. To incorporate time alone, we ignore the space domain. According to the table, both spatial and temporal information improves BoW, and incorporating spatial information helps more than incorporating temporal information. Simultaneous spatial and temporal modeling obtain the best result.



(a)

Dataset	Method	Avg.	Shake	Hug	Kick	Point	Push	Punch
Set 1	BoW	0.77	0.70	0.80	0.90	**1.00**	0.50	0.70
	Hough Voting	0.83	0.50	1.00	1.00	1.00	0.70	0.80
	BoP	**0.95**	**1.00**	**1.00**	**1.00**	0.90	**0.90**	**0.90**
Set2	BoW	0.73	0.70	0.70	0.80	0.80	0.70	0.70
	Hough Voting	0.80	0.70	0.90	1.00	**1.00**	0.80	0.40
	BoP	**0.90**	**0.80**	**1.00**	**1.00**	0.80	**0.90**	**0.90**

(b)

Table 2: Recognition accuracies on UT interact dataset. (a) The performance of incorporating spatial, temporal, and spatio-temporal information into BoW on Set 1. Note that ST-phrase is not a simple combination of spatial and temporal phrases. (b) The accuracies of different methods on both Set 1 and Set 2 of the dataset. For Hough Voting, we cite the reported results in [25].

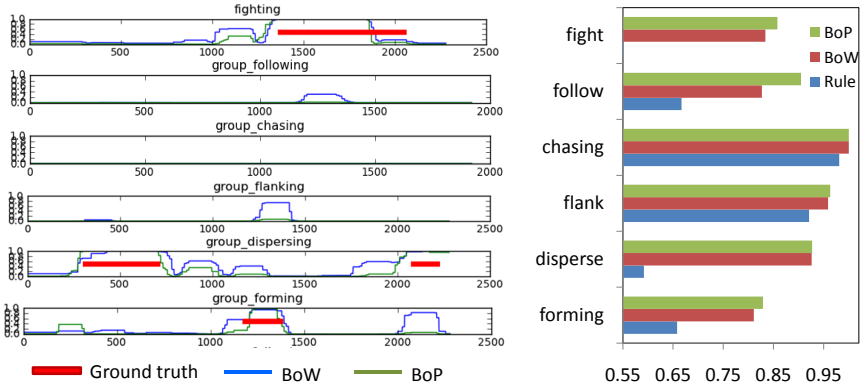
(a) group dispersion (b) group formation (c) approaching (d) group flanking (e) group fighting

Fig. 7: Example scenarios we aim to recognize from videos of the MPR dataset.

3.5 Online Group-Level Activity Detection

Finally, we evaluate on group-level activities with the Mock Prison Riot (MPR) dataset [22]. The dataset has 19 surveillance videos captured in an abandoned prison yard. Several volunteer correctional officers enact typical behaviors of the inmates, including 6 group-level events of interest: *group formation, group dispersion, group following, group chasing, group flanking,* and *group fighting.* The goal is to detect these events of interest from other random behaviors of the group of people. The duration of each event ranges from 2 to 30 seconds. These properties of the dataset require the system to use a long duration context when detecting events at each frame, and thus the speed of the detection algorithm is essential. Figure 7 gives sample snapshots of the dataset. We extract the same features as [27], which use a BoW approach on these features.

Online Detection with Incremental Kernel Computing: We perform experiment for event detection on continuous videos. For this task, we use the incremental algorithm proposed in Section 2.4. We randomly select 60% out of the 19 videos for training and the rest for testing. Prediction for every frame is made using observations from a four-second temporal window ($[t-4s, t]$), i.e., the previous 4 seconds. Figure 8(a) shows the predicted probabilities for one of the test videos. Although BoW can detect the events of interest well, it generates much more false positives than BoP because of the lack of discrimination of the events of interest with normal behaviors. Figure 8(b) compares BoP with previous works using the AUC (area under curve) scores of ROC curves for different categories. BoP improves the previous works, especially for *group forming*, *group following*, and *group fighting* events, where the local group changes are quite confusing with those of random behaviors.



(a) (b)

Fig. 8: (a) The predicted probabilities (vertical) of various events at each frame index (horizontal) for a test video. (b) Comparison of area under curve (AUC) scores for each event category on the whole test dataset. We compare the rule based method [22], BoW [27], and BoP. Note that the rules for *fighting* are not defined in [22].

Running Time: We implement our approach with C++ and Python on an Intel 2.4G dual-core computer. Excluding person tracking and feature extraction (around 0.02s per frame), classifying a 640× 480 frame using a 4-second context takes less than 5 millisecond with the incremental algorithm proposed in Section 2.4. Therefore, we can perform event detection in real time for continuous video streams. In comparison, directly classifying its 4-second context for each frame takes more than 200 millisecond.

4 Conclusion

We proposed the spatio-temporal (ST) phrases to model rich spatial and temporal relationships among the local features, and present an algorithm which can identify all ST phrases in time linear to the number of local features in an video. Experiment results demonstrated that our approach improves the state-of-the-art approaches in activity recognition. Our approach is independent of local feature representation and is widely applicable to a large variety of activities, as illustrated by the diversity of experimental datasets.

References

1. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.J.: Behavior recognition via sparse spatio-temporal features. In: PETS Workshop. (2005)
2. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008)
3. Willems, G., Tuytelaars, T., Gool, L.J.V.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: ECCV. (2008)

4. Wang, X.G., Ma, X.X., Grimson, W.E.L.: Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models. PAMI **31** (2009) 539–555
5. Liu, J.G., Luo, J.B., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR. (2009)
6. Liu, J.G., Shah, M.: Learning human actions via information maximization. In: CVPR. (2008)
7. Wong, S.F., Kim, T.K., Cipolla, R.: Learning motion categories using both semantic and structural information. In: CVPR. (2007)
8. Gaur, U., Zhu, Y., Song, B., Roy-Chowdhury, A.: A "string of feature graphs" model for recognition of complex activities in natural videos. In: ICCV. (2011)
9. Wang, P., Abowd, G.D., Rehg, J.M.: Quasi-periodic event analysis for social game retrieval. In: ICCV. (2009)
10. Duan, L., Xu, D., Tsang, I.W.H., Luo, J.: Visual event recognition in videos by learning from web data. In: CVPR. (2010)
11. Nowozin, S., Bakir, G., Tsuda, K.: Discriminative subsequence mining for action classification. In: ICCV. (2007)
12. Sun, J., Wu, X., Yan, S.C., Cheong, L.F., Chua, T.S., Li, J.T.: Hierarchical spatio-temporal context modeling for action recognition. In: CVPR. (2009)
13. Savarese, S., Pozo, A.D., Niebles, J.C., Li, F.F.: Spatial-temporal correlatons for unsupervised action classification. In: WMVC. (2008)
14. Gilbert, A., Illingworth, J., Bowden, R.: Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In: ECCV. (2008)
15. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR. (2010)
16. Zhang, Y., Jia, Z., Chen, T.: Image retrieval with geometry-preserving visual phrases. In: CVPR. (2011)
17. Wu, S., Moore, B.E., Shah, M.: Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In: CVPR. (2010)
18. Messing, R., Pal, C., Kautz, H.A.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV. (2009)
19. Yao, A., Gall, J., Gool, L.J.V.: A Hough transform-based voting framework for action recognition. In: CVPR. (2010)
20. Zhang, Y., Chen, T.: Efficient kernels for identifying unbounded-order spatial features. In: CVPR. (2009)
21. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: ICPR. (2004)
22. Chang, M.C., Krahnstoever, N., Ge, W.: Probabilistic group-level motion analysis and scenario recognition. In: ICCV. (2011)
23. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV. (2011)
24. Ryoo, M.S., Chen, C.C., Aggarwal, J.K., Roy-Chowdhury, A.: An overview of contest on semantic description of human activities (SDHA) 2010. In: ICPR Contests. (2010)
25. Waltisberg, D., Yao, A., Gall, J., Gool, L.J.V.: Variations of a hough-voting action recognition system. In: ICPR Contests. Lecture Notes in Computer Science, Springer (2010)
26. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV. (2009)
27. Zhang, Y., Ge, W., Chang, M.C., Liu, X.: Group context learning for event recognition. In: WACV. (2012)