# Action Recognition in Video using a Spatial-Temporal Graph-based Feature Representation

Iveel Jargalsaikhan
iveel.jargalsaikhan2@dcu.ie

Suzanne Little
suzanne.little@dcu.ie

Remi Trichet
remi.trichet@gmail.com

Noel E. O'Connor
noel.oconnor@dcu.ie

INSIGHT centre for data analytics, Dublin city university, Glasnevin, Dublin 9, Ireland

## Abstract

*We propose a video graph based human action recognition framework. Given an input video sequence, we extract spatio-temporal local features and construct a video graph to incorporate appearance and motion constraints to reflect the spatio-temporal dependencies among features. them. In particular, we extend a popular dbscan density-based clustering algorithm to form an intuitive video graph. During training, we estimate a linear SVM classifier using the standard Bag-of-words method. During classification, we apply Graph-Cut optimization to find the most frequent action label in the constructed graph and assign this label to the test video sequence. The proposed approach achieves state-of-the-art performance with standard human action recognition benchmarks, namely KTH and UCF-sports datasets and competitive results for the Hollywood (HOHA) dataset.*

## 1. Introduction

An important question in action recognition is how to efficiently and effectively represent a video scene while maintaining the discriminative appearance, motion and contextual cues of the scene. In recent years, Bag-of-words representations have demonstrated excellent results in action recognition. However, as noted by many authors [26] [22][18], such approaches typically ignore the spatiotemporal distribution of the visual words, limiting fine-grained analysis of the video. Many authors noted [20][5] that capturing the spatiotemporal patterns in an action recognition framework can improve system performance. Hence we propose an action recognition framework based on graph-structured local features to explicitly exploit their connections for action recognition. Figure 1 illustrates how a video volume (segment) can be represented as a graph.

This paper presents two contributions. First, we propose

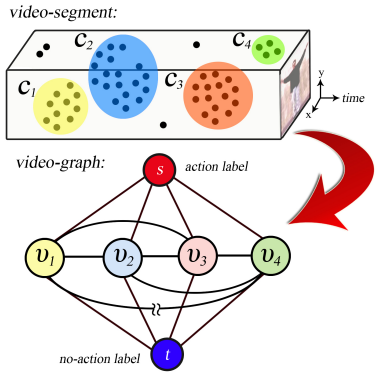


Figure 1: Action recognition is formulated as a graph cut optimization problem over a constructed video graph.

to extend the popular $dbscan$ clustering algorithm [8] towards a graph-based video representation. In this graph, nodes describe a set of clustered local features, $c_i$ and their connectivity (edge) is determined by proximity in space and time (see Figure 1). Each node, $v_i$, is associated with a learned weight indicating the degree of support for the action class of interest based on the local visual descriptors.

Second, the paper explores the application of the graph-cut optimization method from 2D image segmentation to 3D spatio-temporal volume analysis to investigate its effectiveness for action recognition in video. Graph-cut based methods have achieved impressive performance for object segmentation, even on difficult image datasets [6]. It is interesting to study how successful approaches could be extended to action recognition problem. The proposed approach has several important properties. First, the method accommodates a variety of features and classifiers, making it flexible as a general action recognition tool. To illustrate, we have used four descriptors effectively for action classification. Second, as Chen et al [5] highlighted, the graphical representation is equivalent to that of an exhaustive sliding

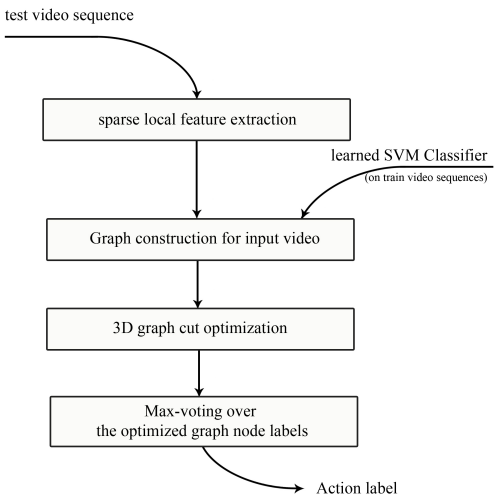


Figure 2: The framework of our action recognition system.

window search, yet requires orders of magnitude less search time. Finally, the graph-based representation is sufficiently generic that one can directly apply any graphical probabilistic inference methods to gain insight about the video data.

We evaluate our approach on three benchmark action datasets. Our results indicate that our model performs competitively in the overall classification task, in particular it achieves state-of-art performance for the KTH and UCF sports datasets.

This paper describes the action recognition framework (section 3), an extension to the $dbscan$ algorithm and graph construction (section 3.2) and action recognition using graph cut (section 3.3) and its evaluation (section 4). The final section concludes and proposes future work direction.

## 2. Related Work

The current approaches for action and activity recognition task can be divided into three categories. The first uses bag-of-words representations. This technique have shown promising result for many benchmarking datasets, however it ignores the spatio-temporal distribution of visual words. The second category uses global spatio-temporal templates, such as motion history [2], spatio-temporal shapes [1], and other templates [11], that retain the spatial structure. This class of approach suffers from sensitivity to nuisance factors such as vantage point, scale, or partial occlusions.

The third class of approaches attempts to decompose an action or activity into parts capturing vague aspects of the local spatial or temporal structure in the data. Sequential data models have been employed to represent the temporal variability [10][19]. For instance, Brendel and Todorovic [4] use a time series of activity codewords, identifying at each frame only one promising region as a part of an activity and modeling the temporal consistency through a Markov chain. More complex part-based models have been proposed [25] explicitly encoding pairwise relationships among predefined image patches. However, the performance of this model relies heavily on the independent detector of salient image patches. Further, Niebles et al. [18] defined the notion of a spatial segment [4] as a set of consecutive video frames. This enables temporal composition, but lacks the ability to spatially localize action parts, because each video segment is represented as a collection of spatio-temporal interest points [25]. Our approach belongs to this latter class. However, we extend these methods [18] and encode spatial and temporal considerations into the volume descriptor to improve recognition, robustness to noise and potentially facilitate action localization.

In an approach similar to ours, Raptis et al [20] proposed an action-part based graphical model and formulated the action recognition task as a Markov random field (MRF) problem. However this method is not generic enough and fine-tuned only with trajectory features to construct nodes in the video graph. On the other hand, Chen at al [5] introduced a sub-graph based model for detection and localization. It uses high-level features, which heavily relies on person and object detection. However, the underlying assumption restricts its applicability where the actor's figure is occluded in the video scene. Our approach aims to overcome these restrictions by focusing on local spatio-temporal regions, as groupings of local features and their pairwise interactions.

## 3. Overview

In our recognition framework (Figure 2), given a test video, the local feature extraction is performed in a sparse manner. Based on their spatio-temporal distribution, we create a video graph representation using an extension of the $dbscan$ algorithm. In this graph, each node indicates a set of local features and its action label likelihood discriminatively assigned by trained classifier (section 3.3.1). Then, graph cut optimization is applied to solve the optimum labelling problem by minimizing an energy function. We assign the most frequent node action label to the test video.

### 3.1. Local features

Our method can be used with any spatio-temporal local features. In the experiment, we adopt the approach of Wang et al. [24]. This approach analyses the 3D volumes along the extracted sparse motion trajectories. The size of the volume is $N \times N$ pixels and $L$ frames, with $N = 32$ and $L = 15$ used in our experiments. For each trajectory, four different types of descriptors are calculated to capture the


(a) The $fixed\text{-}grid\text{-}regions$ approach divides a video volume into a fixed grid of $\delta t \times \delta x \times \delta y$ and the feature points inside the space-time volume constitutes node $u_i$ in the video-graph.

(b) The $adaptive\text{-}regions$ does not suffer from restriction of a pre-defined grid boundary and the resulting graph is sparse and more intuitive compared to $fixed\text{-}grid\text{-}regions$

Figure 3: Construction of the video-graph

different aspects of motion trajectory. We compute HOG and HOF [14] along our trajectories to capture the local appearance, motion around the trajectories. Additionally, MBH [7] and TD [24] are computed in order to represent the relative motion and trajectory shape. The feature vector dimensions of HOG, HOF, MBH and TD are respectively 96, 108, 192 and 30.

### 3.2. Construction of the video-graph

We describe graph $G_V(V, E)$ of a new test video, where $V$ is a set of vertices (nodes) and E is a set of edges. In particular, we present two variants in the construction of the node (Section 3.2.1) and link structures (Section 3.2.2).

#### 3.2.1 Node Structure

Each node in the graph is the abstraction of a set of local features extracted within a spatio-temporal neighborhood. The smallest possible node is a single feature point, and the largest possible one would be the full test sequence, i.e all features from all frames. The factors to be considered for choosing the scale is the granularity of detection and the computational complexity. Note that nodes with a larger number of feature points are favorable not only for computational efficiency, but also their aggregated descriptor statistics have better discriminative power.

We consider two possible node structures: $fixed\text{-}grid\text{-}regions$ and $adaptive\text{-}regions$. The first divides the video into a fixed grid of $\delta t \times \delta x \times \delta y$ space-time volumes as shown in Figure 3(a). In our experiments, we empirically set $\delta t = 24$ $video\text{-}frames$, $\delta x$ and $\delta y$ be $\frac{1}{3}$ of the frame dimensions. The $fixed\text{-}grid\text{-}regions$ will serve as a baseline to measure the effectiveness of the adaptive $adaptive\text{-}regions$ method. For *adaptive-regions*, we propose a feature-point clustering method inspired by the density-based clustering method, particularly the $dbscan$ algorithm. The density-based clustering does not require one to specify the number of clusters in the data as a prior and can find arbitrarily shaped clusters by tuning the only two parameters, a maximum search radius $\epsilon$ and the minimum number of points $minPts$. As shown in Figure 3(b), the algorithm groups only feature points that are densely interlocated. If density of the feature point's spatio-temporal neighbourhood is less than the threshold value, $minPts$, such a feature point is considered as noise and does not contribute towards an action class. Sometimes, the feature sampling technique or the context of video may result in densely distributed local features. The $dbscan$ algorithm can not properly handle dense data points. It merges each data point to produce a single giant cluster, which is not ideal.

Therefore we extended the $dbscan$ algorithm designed for clustering the feature points to take into account not only their location $[x, y, t]$ but also the descriptor characteristic. In addition, the maximum search radius parameter, $\epsilon$ is split into two components: spatial radius $r_{sp}$ and temporal radius $t_{tmp}$. This allows us reduce the pairwise distance calculation space by bounding using $t_{tmp}$ radius and independently treating spatial and temporal dimensions, respectively measured in $pixels$ and $video\text{-}frames$. To account for the trajectory shape, we calculate a lower dimensional *trajectory code*, of size $k = 64$, over randomly sampled trajectory descriptors. The trajectory descriptor is a sequence of displacement vector, for scale invariance, scaled by the sum of the magnitudes. Therefore, the extended $dbscan$ algorithm operates on 4-dimensional data points, $(x, y, t, vCat)$, where $x, y, t$ is a mean coordinate of the extracted trajectory and $vCat$ is the nearest codeword associated with this trajectory. A cluster is formed if its neighborhood contains enough points ($minPts$) with the same *trajectory code* ($vCat \in \{1, .., 64\}$). This ensures similar trajectories between the feature points.

**Algorithm 1** The pseudo code for our extended $dbscan$ algorithm. It clusters local features based on their spatio-temporal location.

**Data**: $D$, $R_{sp}$, $R_{tmp}$, $MinPts$, $vCat$
**Result**: Cluster $C$ for $D$ data points
Initialization

**for** *each unvisited point* $P$ *in dataset* $D$ **do**
  mark $P$ as visited
  $NeighPts$ = REGIONQUERY($P$, $R_{sp}$, $R_{tmp}$, $vCat$)
  **if** *sizeof*($NeighPts$) $< MinPts$ **then**
    mark $P$ as NOISE
  **else**
    $C$ = next cluster
    EXPANDCLUSTER($P$, $NeighPts$, $C$, $R_{sp}$, $R_{tmp}$, $MinPts$, $vCat$)
  **end**
**end**

**function** EXPANDCLUSTER($P$, $NeighPts$, $C$, $R_{sp}$, $R_{tmp}$, $MinPts$, $vCat$)
  mark $P$ as visited
  **for** *each point P' in* $NeighPts$ **do**
    **if** *P' is not visited* **then**
      mark $P$' as visited
      $NeighPts'$ = REGIONQUERY($P$', $R_{sp}$, $R_{tmp}$, $vCat$)
      **if** SIZEOF($NeighPts'$) $> MinPts$ **then**
        $NeighPts = NeighPts \cup NeighPts'$
      **end**
      **if** $P'$ *is not yet member of any cluster* **then**
        add $P'$ to cluster $C$
      **end**
    **end**
**end**

**end function**

**function** REGIONQUERY($P, Rsp, Rtmp, vCat$)
  **return** all points within $P$'s temporal, $R_{tmp}$, and spatial, $R_{sp}$, neighborhood, with the same visual category $vCat(P)$

**end function**

#### 3.2.2 Linking strategies

The connectivity between nodes also affects both the shape of the graph structure and the cost of graph-cut optimization. We explore the different strategies for $fixed\text{-}grid\text{-}regions$ and $adaptive\text{-}regions$ respectively.

For $fixed\text{-}grid\text{-}regions$, we adopt a straightforward strategy, linking only temporally and spatially adjacent nodes, as shown in Figure 3(a). In our experiments, we empirically chose the 26-neighborhood connectivity scheme.

The strategy, for $adaptive\text{-}regions$, is based on the distance between nodes constructed from the extended $dbscan$ clustering algorithm. The connectivity between nodes $v_i$ and $v_j$ is determined by the distance between their corresponding centroids $c_i$ and $c_j$. For example, if the distance between $c_1$ and $c_4$ (See Figure 3(b))is greater than a predefined threshold value, then an edge between node $v_1$ and $v_4$ will not be formed.

### 3.3. 3D graph cut optimization

Given a video sequence represented as a graph of clustered feature nodes ( Figure 3), we now seek to determine regions where there is significant label agreement. The 3D graph cut algorithm solves the labeling problem by minimizing the following energy function defined on the 3D graph G:

$$E(L) = \sum_{r \in V} -E_1(l_r) + \lambda \sum_{(r,s) \in N} -E_2(l_r, l_s) \quad (1)$$

where $l_r$ is the action label of node $r$, and $L = (l_r : \forall r)$. The first term $E_1$ (likelihood) measures the conformity of the local features extracted in the region $r$ to the action class label. The second term $E_2$ measures the agreement between two adjacent nodes.

#### 3.3.1 Likelihood ($E_1$) and Prior ($E_2$)

To measure node likelihood, the discriminative classifier should satisfy two properties. First, it must be able to score an arbitrarily shaped set of feature points. Second, it must be defined such that features computed within local space-time regions can be combined additively to obtain the cumulative classification for a larger region. Suitable additive classifiers include linear support vector machines (SVM), boosted classifiers, or Naive Bayes classifiers. In our experiments, we use a linear SVM with histograms (bags) of quantized space-time descriptors. We consider BoFs computed over several types of local descriptors discussed in Section 3.1.

We compute a vocabulary of K visual words by quantizing a subset of randomly sampled features from the training videos. A training video subvolume with N local features is initially described by the set $S = \{(\boldsymbol{x}_i, v_i)\}_{i=1}^{N}$ , where each $\boldsymbol{x}_i = (x_i, y_i, t_i)$ refers to the 3D feature position in space and time, and $v_i$ is the associated local descriptor. Then the volume is converted to a K-dimensional BoW histogram $h(S)$ by mapping each $v_i$ to its respective visual word $c_i$ ,and tallying the word counts over all $N$ features.

We use the training instances to learn a linear SVM for each action label, which means the resulting scoring function has the form: $f(S) = \beta + \sum_i \alpha_i < h(S), h(S_i) >$ where $i$ indexes the training examples, and $\alpha$, $\beta$ denote the learned weights and bias. This can be rewritten as a sum over the contributions of each feature. Let $h^j(S)$ denote the j-th bin count for histogram $h(S)$. The $j$-th word is associated with a weight $w^j = \sum_i \alpha h^j(S_i)$, for $j = 1, ..., K$. Thus the classifier response for any subvolume $S$ is:

$$f(S) = \beta + \sum_{j=1}^{K} w^j h^j(S) = \beta + \sum_{i=1}^{N} w^{c_i} \qquad (2)$$

where $c_i$ is the index of the visual word that feature $v_i$ maps to, $c_i \in [1, K]$. By writing the score of a subvolume as the sum of its $N$ features word weights, we now have a way to associate each local descriptor occurrence with a single weight based on its contribution to the classifier score.

This same property of linear SVMs is used in [5] to enable efficient subgraph search for action detection.

**Likelihood**, $E_1$, is defined as:

$$E_1(l_r) = \sum_{\boldsymbol{x}_j \in r} w^{c_j} \qquad (3)$$

where $\boldsymbol{x}_j$ is the 3D coordinate of the $j$-th local descriptor falling within node $r \in V$ , and $c_j$ is its quantized feature index. Note that $\boldsymbol{x}_j$ is the feature point position of the low-level descriptors. Intuitively, nodes with high positive weights indicate that the activity covers that space-time region, while nodes with negative weights indicate the absence of the activity.

**Prior energy**, $E_2$, simply measures the label agreement between adjacent nodes, defined as:

$$E_2(l_r, l_s) = \begin{cases} 1, & \text{if } l_r = l_s \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

The objective function of Equation (1) can be globally minimized by an efficient graph cut algorithm [3] and the resulting most frequent labels over graph nodes will determine an action label for the test sequence, that can be found by a simple voting strategy. The default parameter is empirically fixed to $\lambda = 0.85$ in all our experiments.

## 4. Evaluation Dataset and Result

**KTH actions** [21] is to date the most common dataset used in evaluations of action recognition. The dataset is used for comparison and validation of the method. We follow the original experimental setup of the dataset publishers [21]. The average accuracy is a commonly accepted performance measurement for the KTH dataset. The first column of Table 1 shows the comparison of methods applied to the KTH dataset. It is observed that *fixed-grid-region* and *adaptive-region* achieve 96.80% and 98.12 %, respectively, which improve the current state of the art. In particular, the *adaptive-region* based approach has very high accuracy compared to the *fixed-grid* case.

For the **HOHA** dataset contains 430 videos with 8 different actions. This dataset is extremely challenging due to significant camera motion, rapid scene changes and occasionally significant clutter. We followed the experimental setting previously proposed in [13]. As compared with the state-of-art methods in HOHA dataset, our method is less accurate with mean AP of 32.8 % (*fixed-grid-region*) and 35.2 % (*adaptive-region*). This can be attributed to the use of the simple linear SVM classifier in our method, while the latter methods [26][20] use flexible learning techniques such as multi-instance based learning and non-linear kernel method. In addition, our classifier is learned over the training set where only temporal extent of the depicted action is unknown. We hypothesise that the cluttered background, camera movement etc, loosened the classifier discriminative power. For instance, Raptis et al [20] work, that is similar to our approach, performed with mAP of 40.1 % however, authors used the manual annotated spatio-temporal bounding box for each training sequence to learn the model. Finally, the *adaptive-region* based method outperforms the *Fixed-grid-region* approach.

For the **UCF-Sports** dataset, we used the experimental protocol proposed by Lan et al [13]. The dataset is split into 103 training and 47 test samples. The mean-per class accuracies are summarized in Figure 4. As one can see in the third of column of Table 1, our method improves the state-of-art performance by 5% in terms of average accuracy. We associate this good performance with the following points. First, the characteristic of the UCF-Sports dataset is rather simple compared to the HOHA, and the average action duration is relatively short. Thus, this facilitates learning a cleaner classifier (noise free). Secondly, we believe the graph-structure has a significant impact on the system's performance. Because we notice a significant improvement over the baseline performance ( the same classifier applied for the test set using BoW representation) as shown in Figure 4. In all experiments, the $adaptive\text{-}region$ based video-graph consistently outperforms the $fixed\text{-}grid\text{-}region$ structure. It indicates that the graph construction strategy has a strong influence on a video-graph based action recognition system's performance.

## 5. Conclusion and Future Work

We propose a Graph-Cut based approach for action recognition. From an input video, we extract dense trajectories features and construct a spatio-temporal graph to incorporate appearance and motion constraints for the spatio-temporal dependencies among them. Using linear

| KTH (Avg.Acc) | | HOHA (mAP) | | UCF-Sports (Avg.Acc) | |
|---|---|---|---|---|---|
| *Laptev et al.* [14] | 91.80% | *Raptis et al.* [20] | 40.1 % | *Raptis et al.* [28] | 79.4 % |
| *Kovashka et al.* [12] | 94.53% | *Yeffet et al.* [29] | 36.8 % | *Lan at al.* [13] | 73.1 % |
| *Gilbert et al.* [9] | 95.70% | *Laptev et al.* [14] | 38.4 % | SDPM [23] | 75.2 % |
| *Le et al.* [15] | 93.90% | *Matikanien et al.* [17] | 22.8 % | Ma et al.[16] | 81.7 % |
| *Wang et al.* [24] | 94.20% | *Shandong et al.* [26] | 47.6 % | Xu et al.[27] | 78.8 % |
| *Fixed-grid-region* | 96.80% | *Fixed-grid-region* | 32.8 % | *Fixed-grid-region* | 77.1 % |
| *Adaptive-region* | 98.12% | *Adaptive-region* | 35.2 % | *Adaptive-region* | 86.7 % |

Table 1: Comparison of the method with the state-of-the-art methods

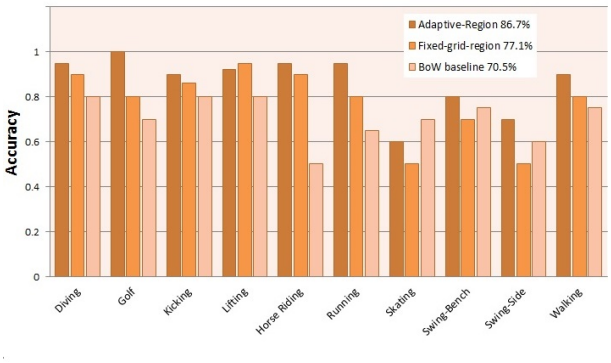


Figure 4: Per-class classification accuracy for UCF-Sports dataset

SVM combined with a BoV approach, we generated spatio-temporal graph with node weight as a likelihood of action class. We evaluate the proposed method in standard benchmark datasets and it achieves the state-of-art and competitive performance.

In future, we will explore an alternative approach to learn a discriminative classifier and further extend the work to not only classify but also localize actions.

## Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant number SFI/12/RC/2289.

## References

[1] Blank. Actions as space-time shapes. In *ICCV 2005*.

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *PAMI 2001*.

[3] Boykov. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV 2001*.

[4] W. Brendel and S. Todorovic. Activities as time series of human postures. In *ECCV 2010*.

[5] C.-Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *CVPR 2012*.

[6] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV 2011*.

[7] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *ECCV 2006*.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd 1996*.

[9] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *IEEE T-PAMI 2011*.

[10] N. Ikizler and D. Forsyth. Searching video for complex activities with finite state models. *Urbana*, 51:61801, 2007.

[11] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV 2007*.

[12] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE CVPR 2010*.

[13] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV 2011*.

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE CVPR 2008*.

[15] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE CVPR*, pages 3361–3368, 2011.

[16] S. Ma. Action recognition and localization by hierarchical space-time segments. In *ICCV 2013*.

[17] P. Matikainen. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Workshops 2009*.

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

[19] Prabhakar. Temporal causality for the analysis of visual events. In *CVPR 2010*.

[20] Raptis. Discovering discriminative action parts from mid-level video representations. In *CVPR 2012*.

[21] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR 2004*.

[22] J. Sun. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR 2009*.

[23] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR 2013*.

[24] H. Wang, A. Klaser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *IEEE CVPR 2011*.

[25] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *T-PAMI 2011*.

[26] S. Wu. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *ICCV 2011*.

[27] R. Xu. Compositional structure learning for action understanding. *arXiv preprint arXiv:1410.5861*, 2014.

[28] X. Yang, C. Yi, L. Cao, and Y. Tian. MediaCCNY at TRECVID 2012: Surveillance Event Detection.

[29] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV 2009*.