Int J Comput Vis
DOI 10.1007/s11263-017-1023-9

Tubelets: Unsupervised Action Proposals from Spatiotemporal Super-Voxels

Mihir Jain[1] · Jan van Gemert[2] · Hervé Jégou[3] · Patrick Bouthemy[3] · Cees G. M. Snoek[1]

Received: 25 June 2016 / Accepted: 18 May 2017

Abstract This paper considers the problem of localizing actions in videos as sequences of bounding boxes. The objective is to generate action proposals that are likely to include the action of interest, ideally achieving high recall with few proposals. Our contributions are threefold. First, inspired by selective search for object proposals, we introduce an approach to generate action proposals from spatiotemporal super-voxels in an unsupervised manner, we call them *Tubelets*. Second, along with the static features from individual frames our approach advantageously exploits motion. We introduce independent motion evidence as a feature to characterize how the action deviates from the background and explicitly incorporate such motion information in various stages of the proposal generation. Finally, we introduce spatiotemporal refinement of Tubelets, for more precise localization of actions, and pruning to keep the number of Tubelets limited. We demonstrate the suitability of our approach by extensive experiments for action proposal quality and action localization on three public datasets: UCF Sports, MSR-II and UCF101. For action proposal quality, our unsupervised proposals beat all other existing approaches on the three datasets. For action localization, we show top performance on both the trimmed videos of UCF Sports and UCF101 as well as the untrimmed videos of MSR-II.

Keywords Action localization · Video representation · Action classification

1 Introduction

The goal of this paper is to localize and recognize actions such as ‘kicking’, ‘hand waving’ and ‘salsa spin’ in video content. The recognition of actions has witnessed tremendous progress in recent years thanks to advanced video representations based on motion and appearance e.g. (Laptev 2005; Dollar et al. 2005; Wang et al. 2013, 2015a; Simonyan and Zisserman 2014). However, determining the spatiotemporal extent of an action has appeared considerably more challenging. Early success came from an exhaustive evaluation of possible action locations e.g. (Ke et al. 2005; Lan et al. 2011; Tian et al. 2013). Such a sliding cuboid is tempting, but owing to large number of possible locations demands a relatively simple video representation, e.g. (Dalal and Triggs 2005; Kläser et al. 2008). Moreover, the rigid cuboid shape does not necessarily capture the versatile nature of actions

Mihir Jain currently works for Qualcomm Research, Amsterdam, The Netherlands and Hervé Jégou for Facebook AI Research, Paris, France. The work for the paper was done when they were at Inria and University of Amsterdam.

Communicated by Ivan Laptev.

Electronic supplementary material The online version of this article (doi:10.1007/s11263-017-1023-9) contains supplementary material, which is available to authorized users.

✉ Mihir Jain
mijain@qti.qualcomm.com

Jan van Gemert
J.C.vanGemert@tudelft.nl

Hervé Jégou
rvj@fb.com

Patrick Bouthemy
patrick.bouthemy@inria.fr

Cees G. M. Snoek
cgmsnoek@uva.nl

[1] Universiteit van Amsterdam, Amsterdam, The Netherlands

[2] Technische Universiteit Delft, Delft, The Netherlands

[3] Inria, Rennes, France

Published online: 08 June 2017

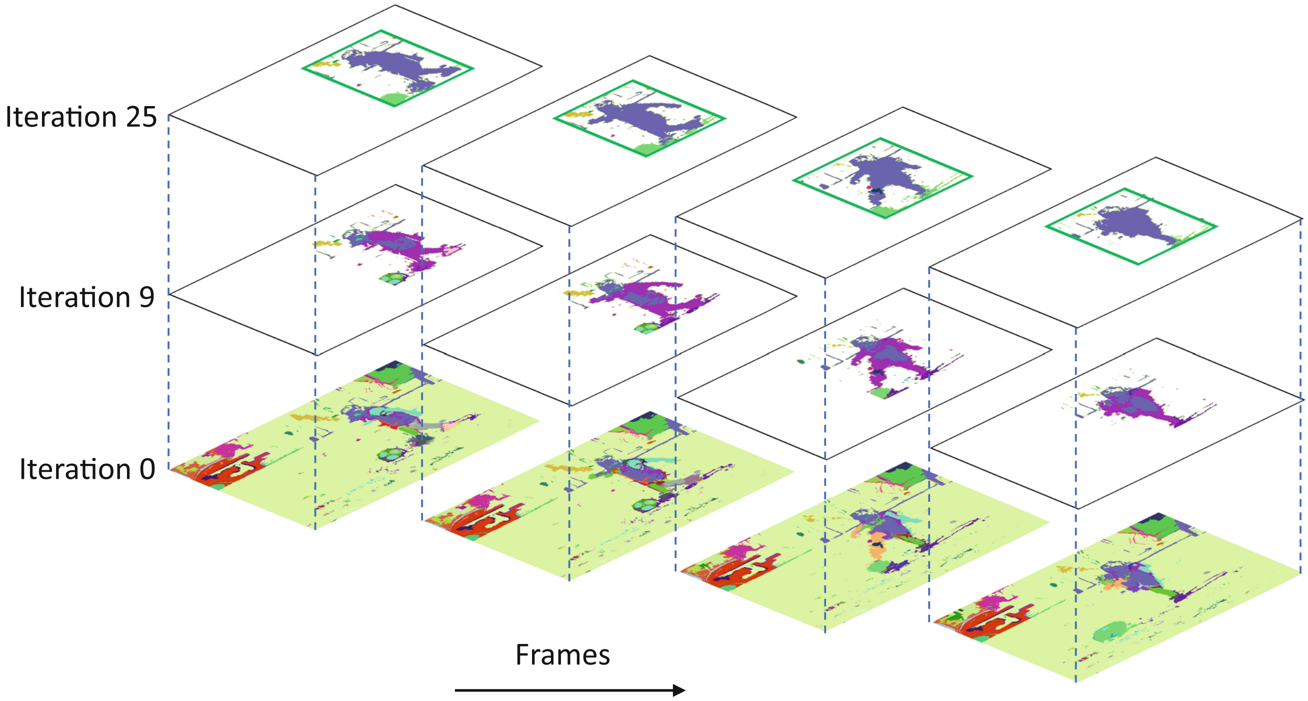


Fig. 1 Overview of unsupervised action proposal from super-voxels: an initial super-voxel segmentation of a video example is shown as a frame sequence in the bottom layer. In each iteration the two most similar neighboring super-voxels are merged into one. The proposed grouping is shown for the super-voxels that eventually lead to a merge that encompasses the action of interest in *blue* (bounded by a *green box*). We refer to the sequence of such *bounding boxes* in a video as a Tubelet (Color figure online)

well. We propose an approach for action localization enabling flexible spatiotemporal subvolumes, while still allowing for modern video representations.

Tran and Yuan pioneered the prediction of flexible spatiotemporal boxes around actions (Tran and Yuan 2011, 2012). They first obtain for each individual frame the most likely spatial locations containing the action, before determining the best temporal path or *action proposal* through the box search space (Tran and Yuan 2011, 2012). Surprisingly, the initial spatial classification is frame-based and ignores motion characteristics for action recognition. More recently both Gkioxari and Malik (2015) and Weinzaepfel et al. (2015) overcome this limitation by relying on a two-stream convolutional neural network based on appearance and two-frame motion flow. While proven effective, these works need to determine the locations in each frame with supervision, and for each action class separately, making them less suited for action localization challenges requiring hundreds of actions. Rather than separating the spatial from the temporal analysis and relying on region-level class-specific supervision, we prefer to analyze both spatial and temporal dimensions jointly to obtain action proposals in an unsupervised manner and avoid supervision until classification. Such an approach is easier to scale to hundreds of classes. Moreover, the same set of proposals can be used for applications requiring different encodings or classification schemes.

We are inspired by a method for object detection in static images called selective search (Uijlings et al. 2013). The algorithm generates box proposals for possible object locations by hierarchically merging adjacent super-pixels from (Felzenszwalb and Huttenlocher 2004), based on similarity criteria for color, texture, size and fill. The approach does not require any supervision, making it suited to evaluate many object classes with the same set of proposals. The small set of object proposals is known to result in both high recall and overlap with the ground-truth (Hosang et al. 2015). Moreover, by separating the localization from the recognition, selective search facilitates modern encodings, such as Fisher vectors of (Sánchez et al. 2013) in (van de Sande et al. 2014) and convolutional neural network features in (Girshick et al. 2016). Following the example set by selective search for object detection, we introduce unsupervised spatiotemporal proposals for action localization.

Our first out of three contributions is to generalize the selective search strategy for unsupervised action proposals in videos. We adopt the general principle designed for static images and repurpose it for video. We consider super-voxels instead of super-pixels to produce spatiotemporal shapes. This directly gives us 2D + t sequences of bounding boxes, without the need to address the problem of linking boxes from one frame to another, as required in other approaches (Tran and Yuan 2011, 2012; Gkioxari and Malik 2015; Weinzaepfel et al. 2015). We refer to our action proposal as *Tubelets* in this paper, and summarize their generation in Fig. 1.

Our second contribution is explicitly incorporating motion information in various stages of the analysis. We introduce *independent motion evidence* as a feature to characterize how the action motion deviates from the background motion. By analogy to image descriptors such as the Fisher vector (Sánchez et al. 2013), we encode the singularity of the motion in a feature vector associated with each super-voxel. We use the motion as an independent cue to produce supervoxels segmenting the video. In addition, motion is used as a merging criterion in the agglomerative grouping of supervoxels leading to better Tubelets.

A preliminary conference version of this article appeared as Jain et al. (2014). This paper adds as third contribution, the spatiotemporal refinement and pruning of Tubelets. The spatiotemporal refinement includes temporal sampling and smoothing the irregular shaped Tubelets. Brox and Malik (2010) realized earlier that temporally consistent segmentations of moving objects in a video can be obtained without supervision. They propose to cluster long term point trajectories and show that these lead to better segmentations than two-frame motion fields. Both Chen and Corso (2015) and van Gemert et al. (2015) build on the work of Brox and Malik (2010) and propose action proposals by clever clustering the improved dense trajectories of Wang and Schmid (2013). Their approaches are known to be very effective for untrimmed videos where temporal localization is essential. We adopt the use of long term trajectories for temporal refinement and pruning of our action proposals, but we do not restrict ourselves exclusively to improved dense trajectories as representation for action classification. Our post-processing methods are heuristic but intuitive for the problem of action localization and considerably improve the performance while keeping the number of proposals manageable.

In addition to technical novelty, the current paper adds: (i) detailed experimental evaluation of motion-based segmentation for better proposals, leading to large gains in both proposal quality and action localization, (ii) apart from UCF Sports and MSR-II we also consider the much larger UCF101 dataset, (iii) revised experiments for all three datasets considering both the quality of the proposal as well as their suitability for action localization using modern video representations (Sánchez et al. 2013; Szegedy et al. 2015), and iv) a new related work section, which will be discussed next.

2 Related Work

Localizing actions in video is similar in spirit to detecting objects in video (Prest et al. 2012; Kwak et al. 2015; Kang et al. 2016). The key difference is that objects are typically captured by appearance whereas actions inherently rely on motion. Our paper is on action localization and motion plays a key role in our approach.

We discuss action localization and action recognition. Action recognition focuses on classifying the action (i.e. what action is it). Action localization adds a spatio-temporal location (i.e. where and when is the action). In Table 1 we link action recognition representations with action localization methods and use it to structure our discussion of related work.

2.1 Action Recognition

Cube Local video features are typically represented by a 3D cube. The seminal work of Laptev (2005) on Spatio-Temporal Interest Points (STIPs) detects points that are salient in appearance and motion and then uses a cube of Gaussian derivative filter responses to represent the interest points. An alternative representation is HOG3D by Kläser et al. (2008) which extends the 2D Histogram of Oriented Gradients (HOG) of Dalal and Triggs (2005) to 3D. Instead of using sparse salient points, the work of Dollar et al. (2005) shows that using denser sampling improves results. Replacing dense points with dense trajectories (Wang et al. 2015a) and flexible track-aligned feature cubes with motion boundary features yields excellent performance. The improved trajectories take into account the camera motion compensation, which is shown to be critical in action recognition (Jain et al. 2016; Piriou et al. 2006; Wang and Schmid 2013). In our work, we build on these dense trajectories as well.

Aggregation (BoW + Fisher) To arrive at a global representation over all local descriptors, BoW represents a cube descriptor by a prototype. The frequency of the prototypes aggregated in a histogram is a global video representation. The BoW representation is simple and offers good results (Everts et al. 2014; Wang et al. 2011). We consider BoW as one of our representations for action localization as well. Where BoW records prototype frequency counts, the Fisher vector (Sánchez et al. 2013) and the VLAD (Jégou et al. 2012) model the relation between local descriptors and prototypes in the feature space of the descriptor. This more sophisticated variant of BoW outperforms BoW (Jain et al. 2013; Oneata et al. 2013, 2014b). Because of the good performance we also consider the Fisher vector as a representation.

Part-Based Action recognition by parts typically exploits the human actor. Correctly recognizing the human pose improves performance Jhuang et al. (2013). A detailed pose model can make fine-grained distinctions between nearly similar actions (Cheron et al. 2015). Pose can be modeled with poselets (Maji et al. 2011) or as a flexible constella-

Table 1 Related work linking the action representation with approaches in action localization

Representation	Approach				
	2D Detect and track		3D spatio-temporal volume		
	Human detector	Generic detector	Cuboid	Trajectory	Voxels
–		Puscas et al. (2015)	Chen et al. (2014)		Oneata et al. (2014a)
Cube	Kläser et al. (2012)	Tran and Yuan (2012)	Ke et al. (2005) Yuan et al. (2009) Cao et al. (2010) Derpanis et al. (2013)		
BoW	Ma et al. (2013)	Tran and Yuan (2011)		Mosabbeb et al. (2014) Chen and Corso (2015)	Jain et al. (2014) Soomro et al. (2015) This paper
Fisher	Yu and Yuan (2015)			van Gemert et al. (2015)	This paper
Part-based	Lan et al. (2011) Wang et al. (2014)		Tian et al. (2013)	Raptis et al. (2012)	
CNN		Gkioxari and Malik (2015)			Jain et al. (2015a) This paper
CNN + Cube		Weinzaepfel et al. (2015)			
CNN + BoW					Jain et al. (2015b)
CNN + Fisher					This paper

Our work does not treat a video as a collection of 2D frames. Instead, we take a holistic spatiotemporal approach by aggregating 3D voxels. From these voxels we build Tubelets on which we evaluate several state-of-the-art action representations tion of parts in a CRF (Wang and Mori 2011). For action recognition in still images where motion is not available the human pose can play a role (Delaitre et al. 2010) as modeled in a part-based latent SVM (Felzenszwalb et al. 2010). In our work, we make no explicit assumptions on the pose, and use generic local video features.

CNNs Deep learning on visual data with CNNs (Convolutional Neural Networks) has revolutionized static image recognition (Krizhevsky et al. 2012). For action recognition in videos, the work of Simonyan and Zisserman (2014) separate video in two channels: a network on static RGB and a network on hand-crafted optical flow. In Wang et al. (2015b) CNN features are used as a local feature in dense trajectories using a Fisher vector. Long term motion can be modeled by recurrent networks (Ngk et al. 2015). The distinction between motion and static objects is analyzed in Jain et al. (2015b) and extended by Jain et al. (2015a) for action recognition without using any video training data. Instead of separating static and motion, 3D convolutional networks combine both (Tran et al. 2015). Due to excellent performance we also adopt CNN features as a representation for action localization.

2.2 Action Localization

2D Human Detector Spatiotemporal action localization can be realized by running a human detector on each frame and tracking the detections. In Kläser et al. (2012) a sliding window upper-body HOG detector per frame is tracked by optical flow feature points for spatial localization. Temporal localization is achieved with a sliding window on track-aligned HOG3D features. HOG3D features are also used in Lan et al. (2011) albeit in BoW, where the 2D person detector is treated as a latent variable and an undirected relational graph inspired by a latent SVM is used for classification. Similarly, the human pose is used by Wang et al. (2014) in a relational dynamic poselet model using cuboids to model a mixture of parts. In Ma et al. (2013) dynamic action parts are extended by incorporating static parts using 2D segments. Segments are grouped to tracks and represented in a hierarchical variant of BoW. In our work, we do not make the assumption that an action has to be performed by a human and do not depend on human detection. Further, Tubelets can be found even if the actor is mostly occluded whereas a generic detector would probably fail.

2D Generic Detector By replacing the human detector with a generic detector the types of actions can be extended beyond a human actor. This can be done by finding the best path trough fixed positions in a frame using HOG/HOF directly (Tran and Yuan 2012) or through BoW (Tran and Yuan 2011). Instead of fixed positions, the work of Gkioxari and Malik (2015) classify object proposals with a two-stream CNN and track overlapping proposals with a high classification score. The work of Weinzaepfel et al. (2015) uses a similar two-stream CNN approach, adding a HOG/HOF/MBH-like cube descriptor at the track level and add temporal localization with a sliding window. The need for strong supervision is removed by Puscas et al. (2015) where generic CNN features are linked through dense trajectory tracks to yield action proposals that could be used for action localization. Similarly, our work requires no supervision for obtaining action proposals, and we experimentally show that these proposals give good results. In addition, we do not first treat a video as a collection of static frames where temporal relations are added as a separate second step. Instead, we respect the 3D spatiotemporal nature of video from the very beginning.

3D Trajectory The strength of 3D dense trajectories by Wang et al. (2015a) for action recognition spilled over to action localization. In Raptis et al. (2012) mid-level clusters of trajectories are grouped and matched with a graphical model. The work of Mosabbeb et al. (2014) groups trajectories to parts which are used in a BoW in an unsupervised manner using low-rank matrix completion and subspace clustering. Similarly, BoW on space-time graph clusters is used by Chen and Corso (2015) and a Fisher vector on trajectories is used on hierarchical clusters in van Gemert et al. (2015) for action localization. These methods specifically target the strength of dense trajectories. Instead, our approach does not commit itself to a single representation.

3D Cuboid The 3D nature of video is respected by building on space-time cuboids for action localization. Such cuboids are a natural extension of 2D patches to 3D. Ke et al. (2005) offer a 3D extension of the seminal face detector of Viola and Jones (2004) using 3D cuboids with optical flow features. The work of Yuan et al. (2009) and Cao et al. (2010) exploit the efficient branch and bound method (Lampert et al. 2008) in 3D. In Tian et al. (2013) the deformable part-based model (Felzenszwalb et al. 2010) is generalized to 3D, an efficient sliding window approach in 3D is proposed by Derpanis et al. (2013) and ordinal regression (Kim et al. 2010) is extended by Chen et al. (2014). Instead of using cuboids, which are rigid in time and space, we choose a more flexible approach using 3D voxels.

3D Voxels As a 3D generalization of 2D image segmentation the voxels from video segmentation methods (Xu and Corso 2012) offer flexible and fine-grained tools for action proposals. In extension of Manen et al. (2013), the work of Oneata et al. (2014a) groups voxels together for action proposals using minimal training. Such action proposals could be used for action localization. This is done by Soomro et al. (2015) who use a supervised CRF to model foreground-background relationships for proposals and action localization. Instead, our proposal method is unsupervised and thus class agnostic. This is beneficial as this makes our algorithm independent on the number of action classes. This paper is an extension of Jain et al. (2014), where 3D voxels are grouped to proposals based on features such as color, texture and motion. The proposals have successfully been used for action localization using objects (Jain et al. 2015b) and in a zero-shot setting (Jain et al. 2015a). We will discuss the mechanics of our unsupervised action proposals next.

3 Unsupervised Action Proposals: Tubelets

In this section we present our approach to obtain action proposals from video in an unsupervised manner, we call the spatiotemporal proposals *Tubelets*. The three stages of the Tubelet generation process are shown in Fig. 2. We first introduce in Sect. 3.1 our motion model based on evidence of independent motion. This motion cue is used in the first two stages of the process. In Sect. 3.2, we discuss the first stage, *super-voxel segmentation*, to generate an initial set of super-voxels from video. For this we rely on an off-the-shelf video segmentation as well as our proposed independent motion evidence. In Sect. 3.3 we detail the second stage of *super-voxel grouping*, where we iteratively group the two most similar super-voxels into a new one. The similarity score is computed using multiple *grouping functions*, each leading to a set of super-voxels. A super-voxel is tightly bounded by a rectangle in each frame it appears. The temporal sequence of bounding boxes forms our action proposal, a Tubelet. In Sect. 3.4, we introduce spatiotemporal refinement and pruning of Tubelets. This enhances the proposal quality, especially for temporal localization, while at the same time keeping the number of proposals feasible to use computationally expensive features and memory demanding encodings for action localization.

3.1 Evidence of Independent Motion

Since we are concerned with action localization, we need to aggregate super-voxels corresponding to the action of interest. Most of the points in such super-voxels would deviate from the background motion caused by moving camera and usually assumed to be dominant motion. In other words, the regions corresponding to independently moving objects do not, usually, conform with the dominant motion in the frame. The dominant frame motion can be represented by a 2D parametric motion model. Typically, an affine motion model of parameters $\theta = (a_i), i = 1, \ldots, 6$, or a quadratic (perspective) model with 8 parameters can be used, depending on the type of camera motion and the scene layout likely to occur:

$$\begin{aligned} & w_\theta(p) = (a_1 + a_2x + a_3y, a_4 + a_5x + a_6y) \\ \text{or } & w_\theta(p) = (a_1 + a_2x + a_3y + a_7x^2 + a_8xy, \\ & \qquad a_4 + a_5x + a_6y + a_7xy + a_8y^2), \end{aligned}$$

where $w_\theta(p)$ is the velocity vector supplied by the motion model at point $p = (x, y)$ in the image domain Ω.

We formulate the evidence that a point $p \in \Omega$ undergoes an independent motion (i.e., an action related motion) at time step t. Let us introduce the displaced frame difference at point p and at time step t for the motion model of parameter θ_t: $r_{\theta_t}(p, t) = I(p + w_{\theta_t}(p), t + 1) - I(p, t)$. Here, $r_{\theta_t}(p, t)$ will be close to 0 if point p only undergoes the background motion due to camera motion. At every time step t, the global parametric motion model can be estimated

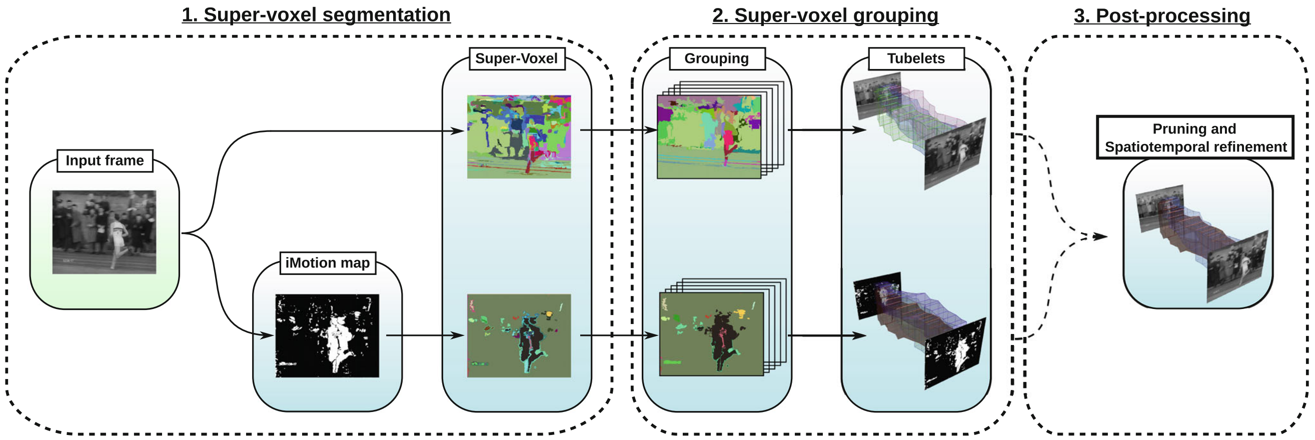


Fig. 2 Tubelet generation: in the first stage a video is segmented into super-voxels. In addition to segmenting video frames, we also segment their *iMotion* maps to also include motion information in the *super-voxel segmentation* stage. In the second stage of *super-voxel grouping*, super-voxels are iteratively merged using several *grouping functions* each of them leading to a set of action proposals. These sets are again grouped by union into a set of Tubelets. The final stage is post-processing that includes pruning and spatiotemporal-refinement of action proposals

with a robust penalty function as

$$\hat{\theta}_t = \arg\min_{\theta_t} \sum_{p \in \Omega} \rho(r_{\theta_t}(p, t)), \tag{1}$$

where ρ is the robust function. To solve (1), we use the publicly available Motion2D software by Odobez and Bouthemy (1995), where $\rho(.)$ is defined as the Tukey function. $\rho(r_{\theta_t})$ produces a maximum likelihood type estimate: the so-called M-estimate (Huber 1981). Indeed, if we write $\rho(r_{\theta_t}) = -\log f(r_{\theta_t})$ for a given function f, $\rho(r_{\theta_t})$ supplies the usual maximum likelihood estimate. Since we are looking for action related moving points in the image, we want to measure *the deviation* to the global (background) motion. This is in spirit of the Fisher vectors by Perronnin and Dance (2007), where the deviation of local descriptors from a background Gaussian mixture model is encoded to produce an image representation.

Let us consider the derivative of the robust function $\rho(.)$. It is usually denoted as $\psi(.)$ and corresponds to the influence function (Huber 1981). More precisely, the ratio $\psi(r_{\theta_t})/r_{\theta_t}$ accounts for the influence of the residual r_{θ_t} in the robust estimation of the model parameters. The higher the influence, the more likely the point conforms to the global motion. Conversely, the lower the influence, the less likely the point approves to the global motion. This leads us to define the *independent motion evidence* as:

$$\xi(p, t) = 1 - \varpi(p, t), \tag{2}$$

where $\varpi(p, t)$ is the ratio $\frac{\psi(r_{\hat{\theta}_t}(p,t))}{r_{\hat{\theta}_t}(p,t)}$ normalized within [0, 1].

In this paper, we use the affine motion model for all the experiments. We chose the affine motion model because it provides a good trade-off between accuracy and efficiency. Moreover, it is safer to use an affine model over a perspective model in videos containing close-ups of moving actors, as suggested by Jain et al. (2016). This is because the affine model cannot completely account for the actor's complex motion, still keeping $\varpi(p, t)$ low at the pixels where close-up actor motion is present. As a consequence, there is no major depletion of the independent motion evidence by Eq. 2.

3.2 Super-Voxel Segmentation

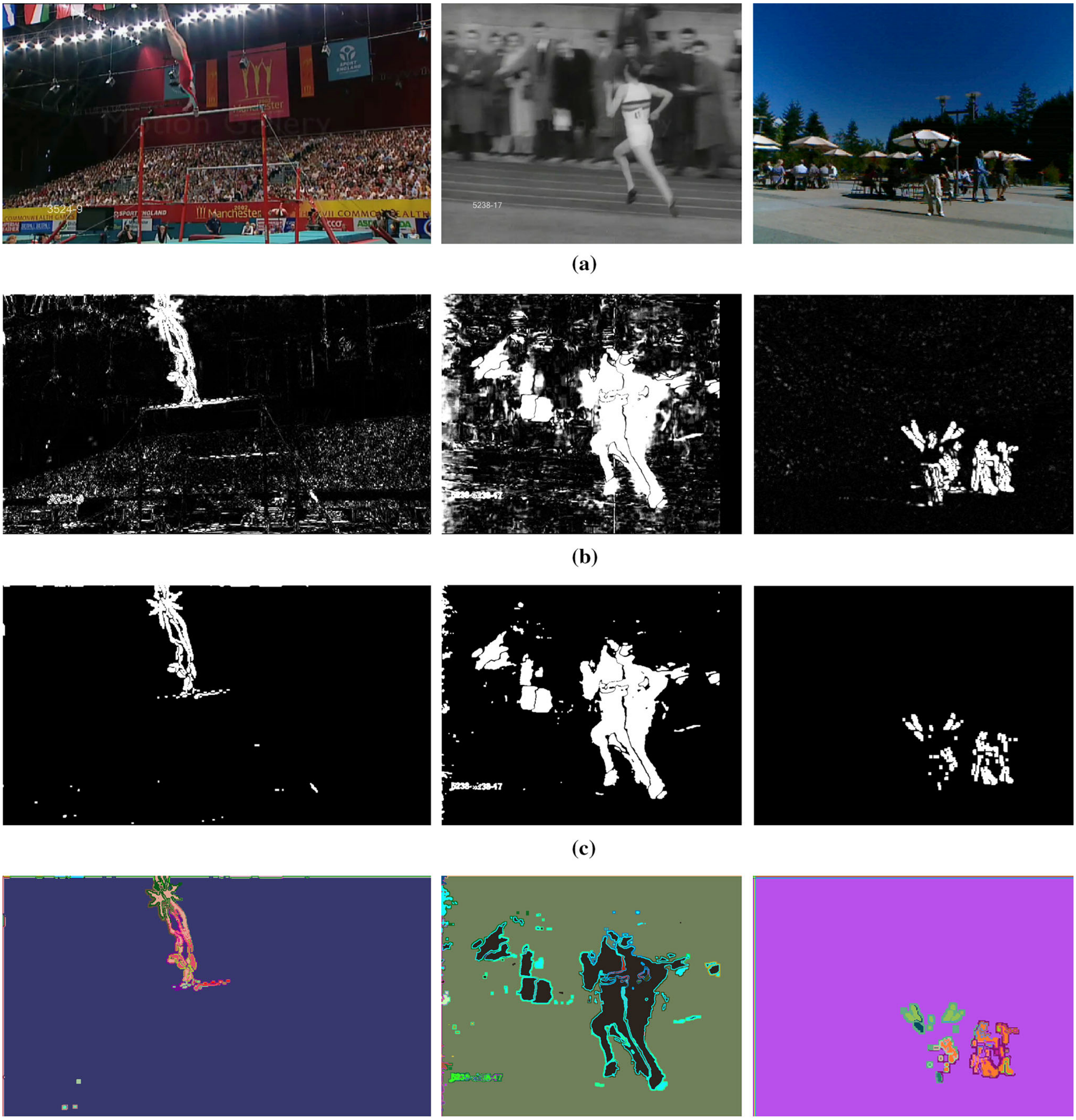
To generate an initial set of super-voxels, we rely on a third-party graph-based video segmentation by Xu and Corso (2012). We choose their graph-based segmentation over other methods in (Xu and Corso 2012) because it is more efficient w.r.t. time and memory. The graph-based segmentation is about 13 times faster than the slightly more accurate hierarchical version (Xu and Corso 2012).

Independent Motion As an alternative to the off-the-shelf video segmentations, each video frame is represented with the corresponding map, $\xi(t)$, of independent motion of pixels. This encodes motion information in the segmentation. We show video frames and their $\xi(t)$ maps in Fig. 3a, b. We post-process the independent motion or $\xi(t)$ maps by applying morphological operations to obtain denoised maps, which we refer to as $iMotion$ maps, displayed in Fig. 3c. More precisely, one iteration of morphological closing operation (dilation followed by erosion) is applied on $\varpi(p, t)$ (Eq. 2), which is then inverted to get cleaner $iMotion$ map. Applying the graph-based video segmentation of (Xu and Corso 2012) on sequences of these denoised maps partitions the video into super-voxels with independent motion. Three examples of results obtained this way are shown in Fig. 3d. The first column shows a frame from action 'Swing-Bench', where the action of interest is highlighted by $iMotion$ map itself and then clearly delineated by segmenting the maps. Second column shows an example from action 'Running'. Here the segmentation does not give an ideal set of initial super-voxels but the $iMotion$ map has useful information to be exploited by our motion feature based merging criterion (described in Sect. 3.3). An example of 'Hand Waving' is shown in the last column. The resulting super-voxels are more adapted and aligned to the action sequences. This alternative for initial segmentation is also more efficient, about 4 times faster than graph-based segmentation on the original video and produces 8 times fewer super-voxels. Unlike graph-based video segmentation on original frames this alternate set of initial super-voxels exploits motion information. The two are complementary and together lead to much better proposal quality as shown later in our experiments.

3.3 Super-Voxel Grouping

Having defined our ways to segment a video sequence into super-voxels, we are now ready to present our method for grouping super-voxels into Tubelets. The grouping is done in two steps. In the first step, initial super-voxels are grouped iteratively to create new super-voxels. A grouping function computes the similarity between any two neighboring super-voxels and the successive groupings of the most similar pairs lead to a new set of super-voxels. Each grouping function leads to a hierarchy of super-voxels. In the second step, the super-voxel hierarchies produced by multiple grouping functions are again grouped by union. This united set of super-voxels is then enclosed by boxes in each frame to yield the Tubelets.

Iterative Grouping We iteratively group super-voxels in an agglomerative manner. Starting from the initial set of super-voxels, we hierarchically group them until the video becomes a single super-voxel. At each iteration, a new super-voxel is produced from two super-voxels, which are then not consid-



(a)

(b)

(c)

(d)

Fig. 3 *iMotion* maps for segmentation: *top two rows* show the original frames and their independent motion. The *iMotion* maps obtained after applying morphological operations are shown in the *third row*. The *bottom row* shows the result of applying graph-based video segmentation on *iMotion* maps. The process is illustrated for three example video clips for actions ‘Swing-Bench’, ‘Running’ and ‘Hand Waving’ respectively. In spite of clutter and illumination variations the *iMotion* map successfully highlights the action. **a** Video frames. **b** Independent motion in frames. **c** *iMotion* maps. **d** Graph-based segmentation of *iMotion* maps (*each color* represents a super-voxel) (Color figure online)

ered any more in subsequent iterations. This iterative merging algorithm is inspired by the selective search method proposed for localization in images by Uijlings et al. (2013).

Formally, we produce a hierarchy of super-voxels that are represented as a tree: the leaves correspond to the n initial super-voxels while the internal nodes are produced by the merge operations. The root node is the whole video and the corresponding super-voxel is produced in the last iteration. Since this hierarchy of super-voxels is organized as a binary tree, it is straightforward to show that $n-1$ additional super-voxels are produced by the algorithm. Out of these $n-1$ super-voxels, those which are very small or contain no motion at all are discarded at this point. This usually leaves much fewer number of super-voxels depending upon the grouping function used.

Grouping Function During grouping, at every iteration a new super-voxel is generated (referred as active), while two that are grouped become inactive. For selection of the two super-voxels to be grouped, we rely on similarities computed between all the neighboring super-voxels that are still active. We employ five complementary similarity measures in our grouping functions to compare super-voxels, in order to decide which should be merged. They are fast to compute. Four of these measures are adapted from selective search in image: the measures based on Color, Texture, Size and Fill were computed for super-pixels (Uijlings et al. 2013). We revise them for super-voxels. As our objective is not to segment the objects but to delineate the actions or actors, we additionally employ a motion-based similarity measure based on our independent motion evidence to characterize a super-voxel. The grouping function is defined as any one of the similarity measures or an equally weighted sum of multiple of them. Next, we present the five similarity measures for super-voxels: *motion*, *color*, *texture*, *size* and *fill*.

Similarity by Motion (s_{M}) We define a motion representation of super-voxels from $iMotion$ maps capturing the relevant motion information. This motion representation is also efficient to compute. We consider the binarized version of $iMotion$ maps obtained by setting all non-zero values to 1. At every pixel p, we count the number of pixels q (including p) in its $3D$ neighborhood that are set to 1 (i.e. pixels likely to be related to actions). In a subvolume of $5 \times 5 \times 3$ pixels, this count value ranges from 0 to 75. A motion histogram of these values, denoted by $h_{\mathrm{M}i}$, is computed over the super-voxel r_i. Intuitively, this histogram captures both the density and the compactness of a given region with respect to the number of points belonging to independently moving objects.

Now, two super-voxels, r_i and r_j, represented by motion histograms are compared as follows. The motion histograms are first ℓ_1-normalized and then compared with histogram intersection, $s = \delta_1(h_{\mathrm{M}i}, h_{\mathrm{M}j})$. The histograms are efficiently propagated through the hierarchy of super-voxels. Denoting with $r_t = r_i \cup r_j$ the super-voxel obtained by merging the super-voxels r_i and r_j, we have:

$$h_{\mathrm{M}k} = \frac{\Gamma(r_i) \times h_{\mathrm{M}i} + \Gamma(r_j) \times h_{\mathrm{M}j}}{\Gamma(r_i) + \Gamma(r_j)} \tag{3}$$

where $\Gamma(r)$ denotes the number of pixels in super-voxel r. The size of the new super-voxel r_t is $\Gamma(r_k) = \Gamma(r_i) + \Gamma(r_j)$.

Similarity by Color (s_{C}) *and texture* (s_{T}) In addition to motion, we also consider similarity based on color and texture. Both h_{C} and h_{T} are identical to the histograms considered for selective search in images (Uijlings et al. 2013), be it that we compute them on super-voxels rather than super-pixels. The histograms are computed from color and intensity gradient for each given super-voxel:

- The color histogram h_{C} captures the HSV components of the pixels included in a super-voxel;
- h_{T} encodes the texture or gradient information of a given super-voxel.

The method of similarity computation and the process of merging for color and texture is the same as for motion: describe each super-voxel with a histogram and compare the two by histogram intersection.

Similarity by Size (s_Γ) *and Fill* (s_{F}) The similarity $s_\Gamma(r_i, r_j)$ aims at merging smaller super-voxels first:

$$s_\Gamma(r_i, r_j) = 1 - \frac{\Gamma(r_i) + \Gamma(r_j)}{\Gamma(\text{video})} \tag{4}$$

where $\Gamma(\text{video})$ is the size of the video (in pixels). This tends to produce super-voxels, and therefore Tubelets, of varying sizes in all parts of the video (recall that we only merge contiguous super-voxels).

The last similarity measure s_{F} measures how well super-voxels r_i and r_j fit into each other. We define $B_{i,j}$ to be the tight bounding cuboid enveloping r_i and r_j. The similarity is given by:

$$s_{\mathrm{F}}(r_i, r_j) = \frac{\Gamma(r_i) + \Gamma(r_j)}{\Gamma(B_{i,j})}. \tag{5}$$

After each merge, we compute the new similarities between the resulting super-voxel and its neighbors. As illustrated in the following two figures. Figure 4 illustrates the method on a sample video. Each color represents a super-voxel and after every iteration a new super-voxel is added and two are removed. After 1000 iterations, observe that two Tubelets (blue and dark green) emerge around the action of interest in the beginning and the end of the video, respectively. At iteration 1720, the two corresponding super-voxels are merged. The novel Tubelet (dark green) resembles the yellow ground-truth sequence of bounding-boxes. This exhibits the ability of our method to group super-voxels both spatially *and* temporally. Also importantly, it shows the capability to sample an action proposal with boxes having very different aspect ratios. This is unlikely to be coped by sliding-subvolumes or even approaches based on efficient sub-window search. Figure 5 depicts another example, with

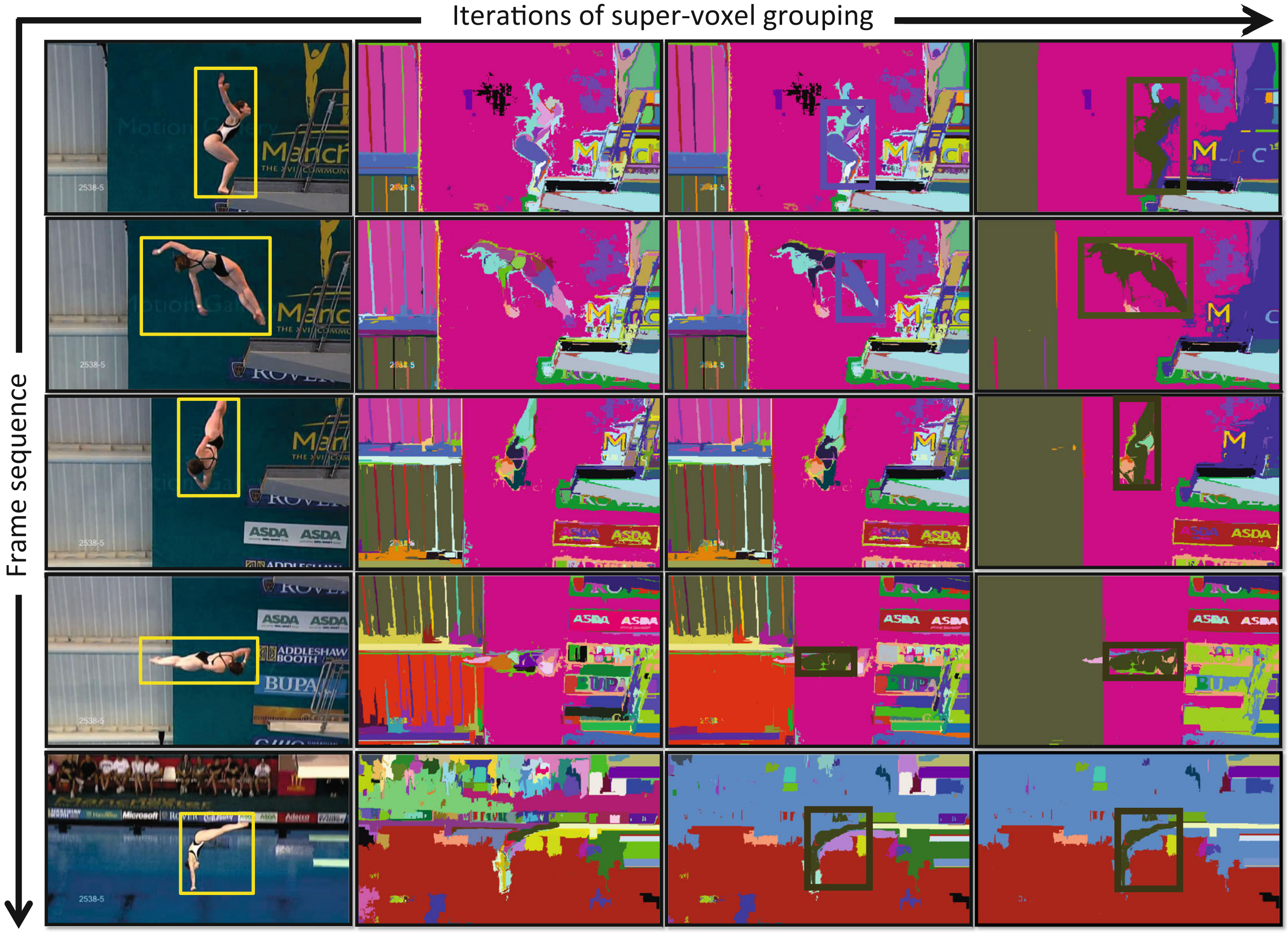


Fig. 4 Illustration of hierarchical grouping of super-voxels into Tubelets. *Left column* a sampled sequence of frames (1st, 15th, 25th, 35th, 50th) associated with the action 'Diving'. The *yellow bounding boxes* represent the ground-truth sequence. *Column 2*: the initial video segmentation used as input to our method. The *last two columns* show the two junctures of the iterative grouping algorithm. A Tubelet close to the action is also represented by *bounding boxes* in these two columns. Observe how close it is to the ground-truth in the last column despite the varying aspect ratios in different frames (Color figure online)

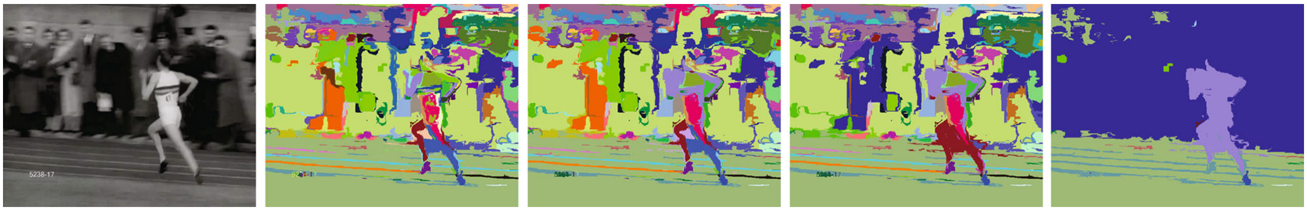


Fig. 5 Example for the action 'Running': the *first two images* depict a video frame and the initial super-voxel segmentation used as input of our approach. The *next three images* represent the segmentation after a varying number of merge operations

a single frame considered at different stages of the algorithm. Here the initial super-voxels (second image in first row) are spatially more decomposed because the background is cluttered both in appearance and in motion (spectators cheering). Even in such a challenging case our method is able to group the super-voxels related to the action of interest.

3.4 Pruning and Spatiotemporal Refinement of Tubelets

Pruning Proposals We apply two types of pruning to reduce the number of proposals leading to a more compact set of Tubelet action proposals with minimal impact on the recall.

Motion Pruning The first type of pruning is based on the amount of motion. Long videos that have much background clutter due to unrelated actors/objects, usually result in many irrelevant Tubelet proposals. We filter them based on their motion content, which we quantify by the number of motion trajectories (Wang and Schmid 2013). For each video, we rank the Tubelet proposals based on the number of trajectories, keep the top P proposals and the top ten percent of the rest. This is to ensure that at least a minimal number of proposals are retained from each video.

Overlap Pruning The second type of pruning is based on mutual overlaps of the action proposals. Many proposals have very high alignment or overlaps between them, all practically representing the same part of the video. To eliminate such redundant proposals we keep only one in a set of many highly overlapping ones. We do not select this proposal and simply pick the first one from the set. It is particularly useful when there is a large number of action proposals per video.

Spatiotemporal Refinement A super-voxel and therefore a Tubelet capturing an actor/object can continue to extend further even after the action is completed as shown in the top row of Fig. 6a. Tubelets are generated from super-voxels that generally follow an object or an actor and hence can be irregular in shape spatially, sometimes leading to sudden changes in the size of consecutive bounding boxes. We propose to handle the above two problems of weak temporal localization and non-smooth spatial localization by temporal and spatial refinement.

Temporal Refinement In order to deal with the overly long Tubelets we propose to temporally sample or segment them. For this we devise a method that can segment each proposal into smaller sub-sequences with tighter temporal boundaries, without increasing the total number of proposals too much. This temporal refinement is applied to one proposal at a time. Consider an action proposal of B boxes (i.e., extending over B frames) and ith box has $nrTraj(i)$ trajectories passing through it (where $i = 1 \dots B$). Now, we represent each box by two values, (a) relative location $= \frac{i}{B}$ and (b) relative motion content $= \frac{nrTraj(i)}{nrTraj_{max}}$. Here, $nrTraj_{max}$ is the maximum number of trajectories passing through any of the B boxes. The boxes that are temporally close to each other (i.e. similar relative location) and also have similar relative

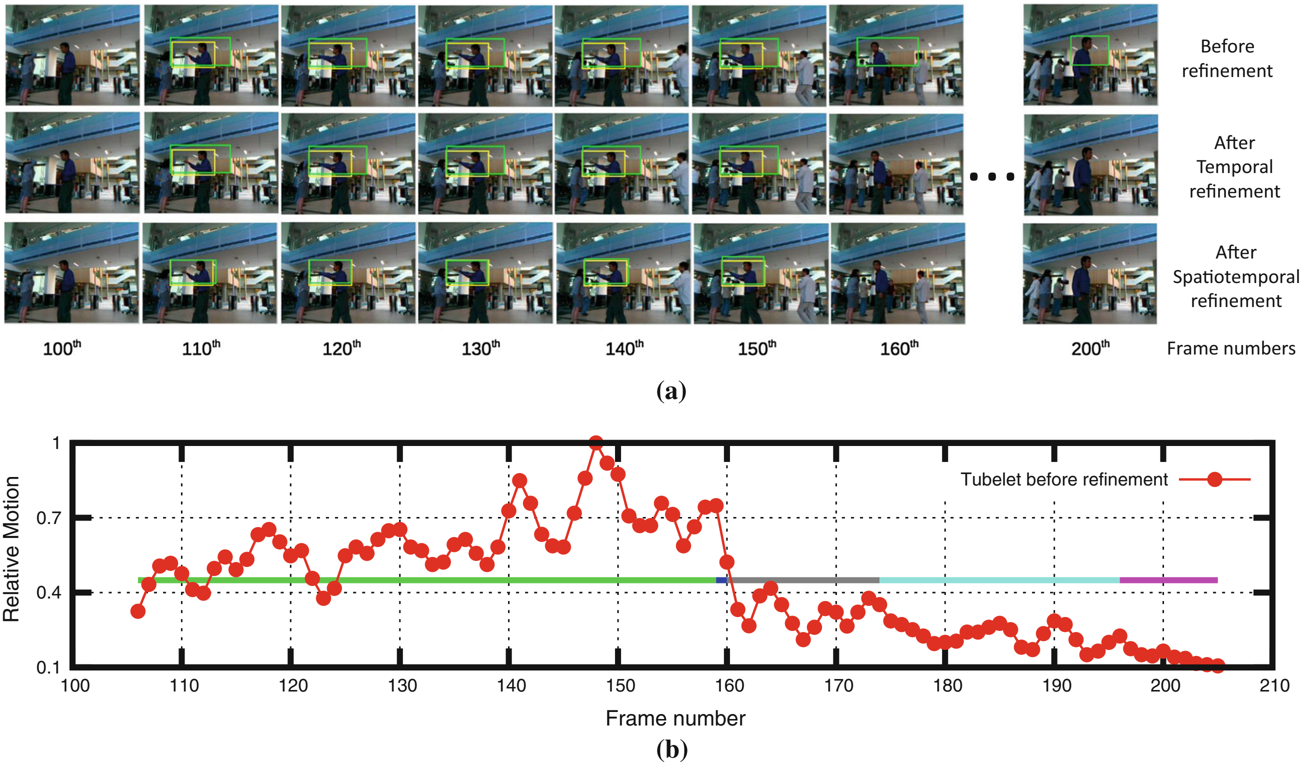


Fig. 6 **a** Impact of spatiotemporal-refinement of Tubelets: the *first row* shows an untrimmed video of about 900 frames. The ground-truth action is an instance of 'Boxing' from frame 108 to frame 156, as *bounded by the yellow boxes*. The *green boxes in the top row* show one of the best Tubelet action proposals obtained for this video. While it aligns well with the ground-truth spatially, it fails temporally as it continues beyond 200 frames. With temporal refinement in the second row, we are able to sample a sub-sequence that localizes the action temporally well also. *Third row* shows further improvement by spatial refinement. **b** Relative motion for Temporal refinement: relative motion is plotted versus time for the above shown example of untrimmed video and Tubelet proposal before refinement. The patterns changes before, during and after action. This is captured by *k-means* clustering ($k = 5$) leading to five segments (shown in *five colors*) or *six cuts* in the long proposal. One of resulting segments shown in *green* aligns well temporally to the action and corresponds to the Tubelet shown as *green boxes* in the *second row* of sub-figure **a** (Color figure online)

motion content are expected to belong to the same action instance. These are grouped together by *k-means* clustering ($k = 5$), leading to five clusters or segments (sometimes non-continuous) of bounding boxes. Each segment has an initial box and a terminal box. The initial box with the smallest frame number forms the first cut and the five terminal boxes make the other five cuts. The cuts are illustrated in Fig. 6b. All combinations of these six cuts are used to segment the initial proposal into fifteen sub-sequences. Then, very short proposals with temporal length less than thirty are filtered out. In practice, this increases the number of proposals by a factor ten. Therefore, we precede and follow temporal sampling by Overlap pruning, to restrict the total number of proposals. The impact of temporal refinement is shown in the second row of Fig. 6a.

Spatial Refinement We apply spatial refinement of proposals, to steer the super-voxels closer to the shape of the action rather than the objects/actor and also to avoid sudden changes in sizes of bounding boxes and thus have smoother sequence of boxes. First, to align the boxes closer to action we modify them such that they are not void of motion trajectories at the boundaries. In each box, the minimum and maximum of x and y coordinates of intersecting trajectories are computed and the box is restricted to $[x_{min} - N, y_{min} - N, x_{max} + N, y_{max} + N]$. The margin N is set equal to 5% of the frame width. Second, we apply weighted linear regression on width, height, x and y coordinates of the top left corner of the boxes. A LOWESS (locally weighted scatterplot smoothing)(Cleveland 1979) is used to estimate smoothed values of the four quantities. This is done over a local span of a few frames, typically a fifth of the proposal length. The impact of spatial refinement after temporal refinement is shown in the last row of Fig. 6a.

4 Datasets and Evaluation Criteria

4.1 Datasets

UCF Sports This dataset consists of 150 videos of actions extracted from sports broadcasts with realistic actions captured in dynamic and cluttered environments (Rodriguez et al. 2008). This dataset is challenging due to many actions with large displacement and intra-class variability. Ten action categories are represented, for instance 'diving', 'swinging bench', 'horse riding', etc. We use the disjoint train-test split of videos (103 for training and 47 for testing) suggested by Lan et al. (2011). The ground truth is provided as sequences of bounding boxes enclosing the actors. The area under the ROC curve (AUC) is the standard evaluation measure used, and we follow this convention.

MSR-II and KTH MSR-II dataset consists of 54 videos recorded in a crowded environment with many people moving in the background. Each video contains multiple actions of three types: 'boxing', 'hand clapping' and 'hand waving'. An actor appears, performs one of these actions, and walks away. A single video has multiple actions (5–10) of different types, making the temporal localization challenging. Bounding subvolumes or cuboids are provided as the ground truth. Since the actors do not change their location, it is equivalent to a sequence of bounding boxes. The localization criterion is subvolume-based, so we follow (Cao et al. 2010) and use the tight subvolume or cuboid enveloping Tubelet. Precision-recall curves and average precision (AP) are used for evaluation (Cao et al. 2010). As standard practice, this dataset is used for cross-dataset experiments with KTH (Schüldt et al. 2004) as training set.

UCF101 The UCF101 dataset by Soomro et al. (2012) is a large action recognition dataset containing 101 action categories of which 24 are provided with localization annotations, corresponding to 3204 videos. Each video contains one or more instances of same action class. It has large variations (camera motion, appearance, scale, etc.) and exhibits much diversity in terms of actions. Three train/test splits are provided with the dataset, we perform all evaluations on the first split with 2290 videos for training and 914 videos for testing. Mean average precision is used for evaluation.

Example frames of some of the action classes are shown in Fig. 7 for each dataset.

4.2 Evaluation Criteria for Action Proposals

To evaluate the quality of action proposals, we compute the upper bound on the localization accuracy, as previously done to evaluate the quality of object proposals (Uijlings et al. 2013), by the Mean Average Best Overlap (MABO) and maximum possible recall. In this subsection, we extend these measures from objects in images to actions in videos. This requires measuring the overlap between two sequences of boxes instead of two boxes.

Overlap or Localization Score In a given video V of F frames comprising m instances of different actions, the i^{th} ground truth sequence of bounding boxes is given by $gt^i = (B_1^i, B_2^i, \dots B_F^i)$. If there is no action of i^{th} instance in frame f, then $B_f^i = \emptyset$. From the action proposals, the j^{th} proposal formed by a sequence of bounding boxes is denoted as, $dt^j = (D_1^j, D_2^j, \dots D_F^j)$. Let $OV_{i,j}(f)$ be the overlap between the two bounding boxes in frame, f, which is computed as intersection-over-union. The localization score between ground truth Tubelet gt^i and a Tubelet dt^j is given by:

$$S(gt^i, dt^j) = \frac{1}{|\Gamma|} \sum_{f \in \Gamma} OV_{i,j}(f), \tag{6}$$

Fig. 7 Example video frames showing action classes from the UCF Sports, MSR-II and UCF101 datasets

where Γ is the set of frames where at least one of B^i_f, D^j_f is not empty. This criterion generalizes the one proposed by (Lan et al. 2011) by taking into account the temporal axis.

Mean Average Best Overlap (MABO) The Average Best Overlap (ABO) for a given class c is obtained by computing for each ground-truth annotation $gt^i \in G^c$, the best localization from the set of action proposals $T = \{dt^j | j = 1 \ldots m\}$:

$$\text{ABO} = \frac{1}{|G^c|} \sum_{gt^i \in G^c} \max_{dt^j \in T} S(gt^i, dt^j). \tag{7}$$

The mean ABO (MABO) summarizes the performance over all the classes.

Maximum Possible Recall (*Recall*) Another measure for quality of proposals is maximum possible recall. It is computed as the fraction of ground-truth actions with best overlap of greater than the overlap threshold (σ) averaged over action classes. We compute it with a very stringent localization threshold $\sigma = 0.5$.

Note that adding more proposals can only increase the MABO and Recall (scores are maintained if added proposals are not better). So, both MABO and Recall must be considered jointly with the number of proposals.

Action Localization An instance of action, gt^i, is considered to be correctly localized by an action proposal, dt^j, if the action is correctly predicted by the classifier and also the overlap/localization score is greater than the overlap threshold, i.e., $S(gt^i, dt^j) > \sigma$.

5 Experiments: Quality of Tubelets

In this section, we first analyze and evaluate the three stages of Tubelet extraction on the training set of the UCF Sports dataset. The initial step, super-voxel segmentation, is discussed in Sect. 5.1. Then, we evaluate different grouping functions over the initial set of super-voxels in Sect. 5.2 and also show that segmenting iMotion maps is complementary to segmenting input video frames. In Sect. 5.3, we evaluate the impact of spatiotemporal refinement and pruning on all three datasets. In all our evaluations, we do not use any additional constraint to keep proposals that last for the entire length for videos that are trimmed for actions. Finally, in Sect. 5.4 we compare Tubelets with the state-of-the-art. We evaluate Tubelets with modern representations for action localization in Sect. 6.

5.1 Super-Voxel Segmentation

Here, we evaluate the graph-based segmentation of video and the graph-based segmentation of $iMotion$ maps. Note that the objective of this experiment is not to compare, but to show that graph-based segmentation by Xu and Corso (2012), either on video or $iMotion$ maps, makes sense as initial super-voxels for Tubelets. We set parameters as follows: $\sigma = 0.5$, merging threshold of two nodes, $\theta = 200$, minimum segment size $smin = 500$, bigger c and $smin$ would mean larger (and hence fewer) segments. In Table 2, we present MABO,

Table 2 Quality of initial super-voxels by applying the graph-based segmentation by Xu and Corso (2012) on RGB video frames and on a sequence of *iMotion* maps for the UCF Sports train set

Segmenting	MABO	Recall	# Super-voxels	Time (s)
Video	36.2	17.3	862	379
iMotion maps	48.6	53.2	118	69

We report MABO, Recall (at $\sigma = 0.5$), number of initial super-voxels, and average execution time per video. Note the competitive performance of super-voxel segmentation on *iMotion* maps

Table 3 Evaluation of super-voxel groupings with *video segmentation* on the training set of UCF Sports

Super-voxel grouping	MABO	Recall	#Proposals
Single grouping function			
Motion	56.2	64.3	**299**
Color	47.3	42.0	483
Texture	44.6	36.2	381
Size	47.8	45.8	918
Fill	50.9	50.4	908
Motion + *Size* + *Fill*	57.2	65.5	719
Texture + *Size* + *Fill*	52.6	57.5	770
All-but-motion	53.4	53.6	672
All	58.1	66.7	656
Multiple grouping functions			
Union set, Φ	**62.0**	**74.7**	3254

Among the similarity measures, the ones based on *iMotion*: *Motion*, *Motion* + *Size* + *Fill* and *All* perform the best while generating a reasonable number of proposals. The union of the five selected grouping functions, Φ, further increases the MABO and Recall

Bold value indicates highest MABO/Recall and lowest #Proposals

Recall, number of super-voxels and computation time. The relatively efficient graph-based segmentation limits the number of super-voxels, while achieving a reasonable MABO. Segmentation of iMotion maps leads to higher MABO and Recall, fewer initial super-voxels and lower computation time. However super-voxels from video segmentation are also critical and complementary as we show in the next experiments.

5.2 Super-Voxel Grouping

We evaluate super-voxel groupings in Tables 3 and 4 for video and *iMotion* segmentations respectively. Nine grouping functions are considered that use one or more of the five similarity measures defined in Sect. 3.3: *Motion*, *Color*, *Texture*, *Size* and *Fill*. Five of these use only one similarity measure, while the other four use multiple similarities. Here, *All-but-motion* is Color + Texture + Size + Fill and *All* is Motion + Color + Texture + Size + Fill, the rest are self-explanatory. We first evaluate these 9 grouping functions in both the tables. In Table 3, the best performing groupings are the ones that involve the *iMotion* similarity measure: *Motion*, *Motion* + *Size* + *Fill* and *All*. Note that although the same set of n (=862) initial super-voxels are given as input to each grouping function, they lead to different number of new proposals, ($< n - 1$). This is because the proposals that are too small or have zero-motion are discarded during iterative grouping as explained in Sect. 3.3. For instance, *Motion* needs only 299 proposals per video to achieve a MABO of 56.2% and Recall of 64.3%. This is because *iMotion* brings most of the motion content in fewer super-voxels and the majority of the resulting super-voxels are too small or have zero-motion, and hence are discarded.

Table 4 Evaluation of super-voxel groupings with *segmentation of iMotion maps* on the training set of UCF Sports

Super-voxel grouping	MABO	Recall	#Proposals
Single grouping function			
Motion	52.9	66.9	90
Color	51.1	60.5	93
Texture	51.2	62.5	**81**
Size	52.2	63.5	158
Fill	52.7	61.9	155
Motion + *Size* + *Fill*	54.2	70.8	129
Texture + *Size* + *Fill*	53.9	67.8	145
All-but-motion	54.5	71.3	127
All	55.1	74.5	123
Multiple grouping functions			
Union set, Φ	**56.8**	**77.0**	624

The grouping functions containing the *iMotion* similarity measure again prove to be the most successful, though not as much as in Table 3. The union set, Φ, achieves a high MABO and Recall with only 624 proposals per video

Bold value indicates highest MABO/Recall and lowest #Proposals

Multiple Grouping Functions After trying several combinations on the training set of UCF Sports, we select 5 best grouping functions: *Motion*, *Fill*, *Motion* + *Size* + *Fill*, *All-but-motion* and *All*. We collect the proposals from these five selected grouping functions into a Union set Φ. Collecting proposals from multiple grouping functions significantly increases the MABO and Recall to 62.0 and 74.7% respectively. Considering that a common localization score threshold (σ) used in the literature is 0.2 (Lan et al. 2011; Tian et al. 2013), these MABO values and Recall at $\sigma = 0.5$ are very promising. Thus obtained set of Tubelets with input video segmentation and *Union set*, Φ, is from now on referred to as T_{vid}.

Super-voxel groupings with *segmentation of iMotion maps* are evaluated in Table 4. Here, the grouping functions containing the *iMotion* similarity measure again prove to be the most successful, though not as much as in the case of

Table 5 Combining of Tubelets from video segmentation and $iMotion$ segmentation, $\mathrm{T}_{vid} \cup \mathrm{Ti}_{Motion}$

Super-voxel grouping	MABO	Recall	#Proposals
Motion	63.9	80.9	**390**
Fill	62.2	77.5	1062
Motion + Size + Fill	65.1	86.4	848
All-but-motion	65.0	86.0	799
All	66.6	91.3	779
Union set, Φ	**69.5**	**93.6**	3878

Numbers are reported for the five selected grouping functions as well as their union set, Φ. The combination leads to significant improvement of MABO and Recall, showing the two sets of Tubelets from two video segmentations complement each other

Bold value indicates highest MABO/Recall and lowest #Proposals

video segmentation. It is because by segmenting $iMotion$ maps motion information is already utilized to some extent. *Fill* also leads to good MABO and Recall with just 155 proposals. The union set, Φ, achieves a good MABO of 56.8% and Recall of 77.0%, which even outperforms the Recall obtained with video segmentation by 2.3%. Although the best MABO with segmentation of $iMotion$ maps is lower than that for video segmentation, the number of proposals required is only 624 on average, which is lower than the 3254 proposals from video segmentation. This is a considerable reduction, which is in particular useful for long videos where the number of proposals can be high. Moreover, segmenting $iMotion$ maps is faster, which is again of interest when operating on longer videos. This set of Tubelets obtained by segmenting iMotion maps and *Union set*, Φ, is from here on referred to as Ti_{Motion}.

After analyzing segmentations from input video and $iMotion$ maps separately, we now combine the Tubelets from both, resulting proposal set denoted by $\mathrm{Ti}_{Motion} \cup \mathrm{T}_{vid}$. As reported in Table 5, the MABO increases up to 69.5% and Recall reaches 93.6%. This is an improvement of ∼7% in MABO and ∼16% in Recall over the individual best of video and $iMotion$ segmentations. The experiments till this point are conducted on the training set of UCF Sports. This validates the set of grouping functions, Φ, and that the two Tubelet sets Ti_{Motion} and T_{vid} complement each other for localizing actions. We fix this setting for the experiments to follow.

5.3 Pruning and Spatiotemporal Refinement

In this section, we evaluate the impact of pruning and spatiotemporal refinement on the quality of action proposals of UCF Sports, MSR-II and UCF101. The validation for grouping functions and segmentation is already done on the training set of UCF Sports. Now, we report results when considering *all* the videos of these three datasets, to be comparable with the numbers reported by other methods. Before moving to results, we provide the implementation details of pruning and spatiotemporal refinement.

Table 6 Impact of pruning and spatial refinement of Tubelets on UCF sports: even after motion pruning the MABO and Recall are maintained with only ∼26% of proposals

	MABO	Recall	#Proposals
$\mathrm{T}_{vid} \cup \mathrm{Ti}_{Motion}$	69.3	93.5	3432
+Motion pruning	69.3	93.5	884
+Overlap pruning	67.5	90.5	289
+Spatial refinement	67.5	91.9	289

With overlap pruning the number of proposals goes down further to ∼8% of the original number, with a small loss in MABO and Recall scores. The loss is compensated by spatial refinement of Tubelets

Implementation Details For motion pruning we set $P = 50$, so that at least fifty proposals are retained from each video. Also, motion pruning is only applied to T_{vid}, since proposals from Ti_{Motion} are expected to have enough motion content. Overlap pruning is similar to non-maximum suppression, but applied without classification scores and therefore can affect the recall. To minimize its impact on Recall, we set a high overlap threshold of 0.8 for overlap based pruning. For spatial refinement, we set N equal to 5% of the frame width.

UCF Sports In Table 6, we evaluate the impact of pruning and spatial refinement on MABO, Recall and the average number of proposals per video for UCF Sports dataset. The results for $\mathrm{T}_{vid} \cup \mathrm{Ti}_{Motion}$ for all 150 videos of UCF Sports is similar to that on its train set. Now, with motion pruning there is no loss of MABO and Recall while only ∼26% of original proposals are used. Further, with overlap pruning number of proposals further goes down to ∼8% of original number with a small loss in MABO and Recall. Finally, with spatial refinement of Tubelets there is small improvement of Recall. Altogether, with pruning and spatial refinement we are able to decrease the number action proposals by a factor 12 with only a modest loss in MABO and Recall.

MSR-II The MSR-II dataset has untrimmed videos with multiple instances of different types of actions in the same video. This poses additional challenges for temporal localization, which is experimentally illustrated in Table 7. The table reports MABO and Recall for Tubelet set T_{vid} after motion pruning for spatiotemporal localization and also spatial-only localization. Overlap score for spatiotemporal case is computed according to Eq. 6 as done in all other results. For spatial localization, we compute only for the frames where ground-truth proposal is present, i.e., we do not penalize overlap score for temporal misalignment. MABO doubles and the Recall shoots from 2.2 to 81.3% for spatial-only localization, which means that our Tubelets very well locate the actions

Table 7 Spatial localization versus spatiotemporal localization on untrimmed videos of MSR-II: spatial only localization leads to much better Recall, which indicates that the low Recall is due to weak temporal localization

Localization	MABO	Recall	#Proposals
Spatiotemporal	28.2	2.2	2342
Spatial only	60.9	81.3	2342

This calls for temporal refinement of Tubelets

Table 8 Impact of pruning and spatial refinement of Tubelets on MSR-II: pruning by motion maintains the MABO and Recall while reducing the proposals to only a quarter of the initial set

	MABO	Recall	#Proposals
$\mathrm{T}_{vid} \cup \mathrm{Ti}_{Motion}$	36.9	5.1	25,962
+Motion pruning	36.7	5.1	6560
+Temporal refinement	46.0	35.2	7287
+Spatial refinement	48.9	47.4	7287

Temporal refinement has a positive impact on proposal quality with Recall increased by 30%. Finally, with spatial refinement another improvement of ∼12% is achieved. Spatiotemporal refinement is important for this dataset

spatially but extends to the frames where there is no action of interest. This is due the tendency of super-voxels to continue to cover the actor even when the action is completed. We overcome this limitation by temporal refinement.

In Table 8, in addition to pruning and spatial refinement, we also report for temporal refinement to improve temporal localization. First, motion pruning maintains the MABO and Recall while reducing the number of proposals to only a quarter of initial number. This pruning needs to precede temporal refinement to limit the number of proposals. Second, temporal refinement leads to a massive improvement of 30.1% in Recall and 9.3% in MABO. Note that temporal refinement also includes overlap pruning to filter-out newly added very similar proposals. Also, to limit the number of proposals temporal refinement is exclusively applied to 'T_{vid} + Motion pruning', which means only overlap pruning is applied to 'Ti_{Motion} + Motion pruning'. Finally, with spatial refinement another huge improvement of ∼12% is achieved in Recall along with ∼3% improvement in MABO.

Overall, we achieve an improvement of 12% of MABO and 42.3% of Recall while decreasing the number of proposals by about 72% compared to the initial set, $\mathrm{T}_{vid} \cup \mathrm{Ti}_{Motion}$. The gain due to temporal refinement is easy to understand for this dataset of untrimmed videos. However, we also get impressive boost by spatial refinement that is much more than we get for the other two datasets. We attribute this to the exploitation of information from motion trajectories, which is paramount for MSR-II as noted before in van Gemert et al. (2015); Chen and Corso (2015). Localizing actions is more challenging when multiple untrimmed actions happen simultaneously in the same frames. We analyze Tubelets for such cases in Fig. 8. Temporally, Tubelets sometimes miss action for a few frames or continue for a few extra frames, but it does find multiple actions in the same frame consistently. In general, temporal localization pose a bigger challenge than localizing multiple actions in the same frame. Overall, Tubelet does well to handle these challenging cases.

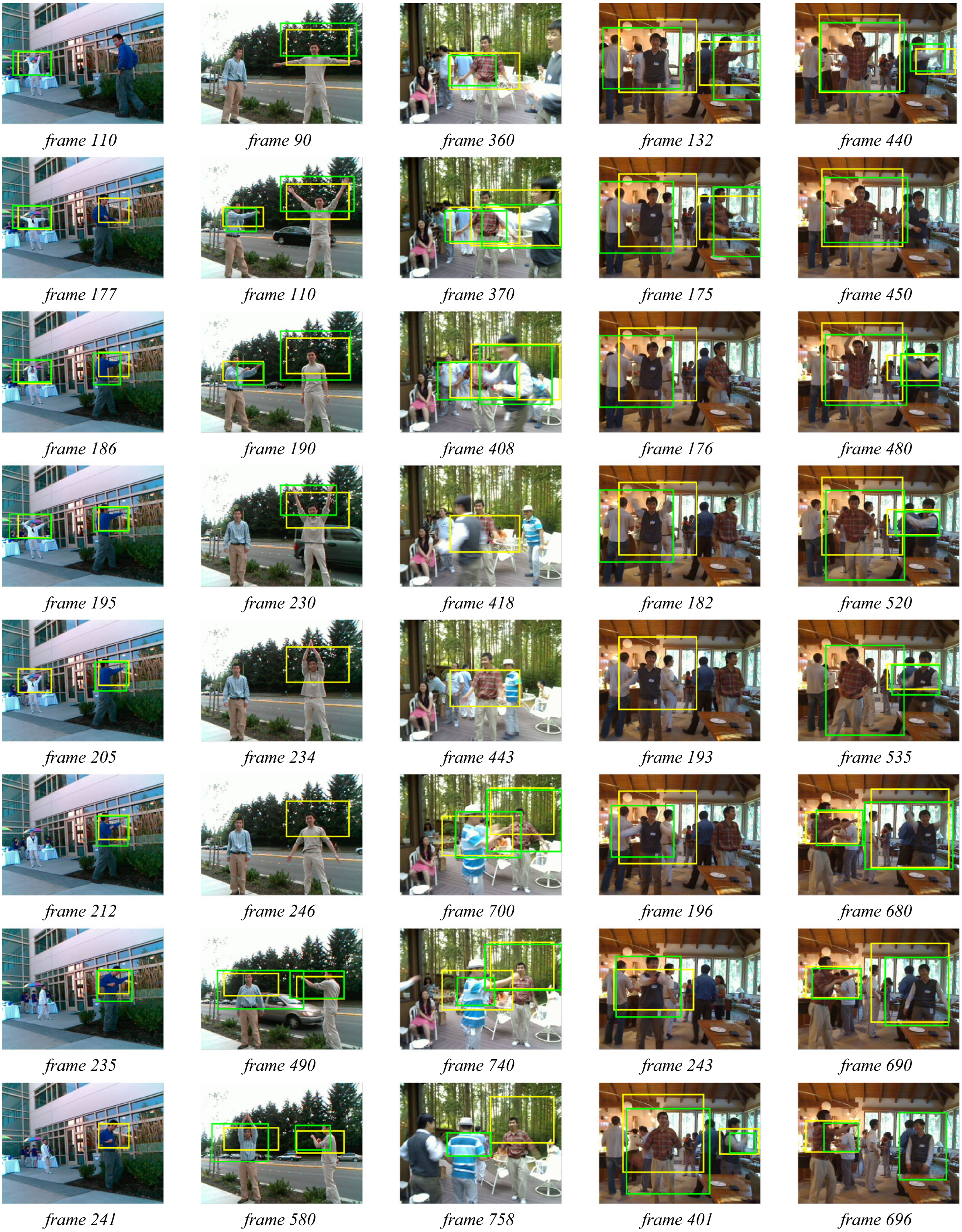
UCF101 In Table 9, we report the impact of pruning and spatial refinement on MABO, Recall and the average number of proposals per video for UCF101 dataset. Motion pruning also works well on the 3204 videos of UCF101, compressing the number of proposals by a factor of four, while maintaining MABO and Recall. Further, with overlap pruning number of proposals goes down to ∼ 9% of original number with a small loss in MABO and Recall. With favourable spatial refinement, eventually, final set of Tubelets achieve same performance as by $\mathrm{T}_{vid} \cup \mathrm{Ti}_{Motion}$, but with about 10 times fewer proposals.

Timings In Table 10, we report execution times per video for all stages of Tubelet generation. We focus on MSR-II as it is the only dataset containing all proposed stages (including temporal refinement). On average there are 766.9 frames per video. The experiments were performed on Intel(R) Xeon(R) CPU, 2.90GHz.

Conclusions In Tables 6, 8 and 9, we show many proposals are filtered out by motion and overlap pruning; and the boost provided by spatial/temporal refinement. For all the three datasets motion pruning filters out a large fraction of proposals, leading to a fourfold decrease. Temporal refinement, only applicable to MSR-II, boosts MABO (+9.3%) and Recall (+30.1%), while keeping the number of proposals limited because of the overlap pruning that is part of temporal refinement. Overlap pruning also leads to a threefold decrease in the number of proposals for UCF sports and UCF101, while losing 2% to 3% in Recall and less than 2% in MABO. Spatial refinement pushes Recall up by about 2% for UCF sports and UCF101. Its contribution to MSR-II is even more serious, leading to a 12% gain in Recall.

5.4 Comparison with State-of-the-Art Methods

In Table 11, we compare our Tubelets with alternative unsupervised action proposals from the literature. We also include average recall suggested for object detection by Hosang et al. (2015). While this metric is not common yet for action proposals evaluation, we anticipate it will be important for future reference. With a relatively small set of 289 proposals we outperform all the other approaches on UCF Sports. On MSR-II, we outperform the previous best approach of van Gemert et al. (2015). It is interesting to note the improvement in



frame 110 *frame 90* *frame 360* *frame 132* *frame 440*

frame 177 *frame 110* *frame 370* *frame 175* *frame 450*

frame 186 *frame 190* *frame 408* *frame 176* *frame 480*

frame 195 *frame 230* *frame 418* *frame 182* *frame 520*

frame 205 *frame 234* *frame 443* *frame 193* *frame 535*

frame 212 *frame 246* *frame 700* *frame 196* *frame 680*

frame 235 *frame 490* *frame 740* *frame 243* *frame 690*

frame 241 *frame 580* *frame 758* *frame 401* *frame 696*

Fig. 8 Tubelets on multiple instances of untrimmed actions of MSR-II: *first* and *second columns* show three co-occurring instances of 'boxing' and 'waving' actions. In the *third column*, there are two co-occurrences of 'boxing' and 'clapping'. The last video sequence is shown in *two columns* with eight action instances. In the initial part, co-occurring instances of 'waving' and boxing' are shown, followed by 'clapping' and then multiple simultaneous instances of 'boxing' and 'waving'. Overall, Tubelet does well, occasionally missing actions temporally but robust in capturing simultaneous actions

Table 9 Impact of pruning and spatial refinement of Tubelets on UCF101: motion pruning leads to ~1% loss in MABO and Recall while filtering out 75% of the proposals

	MABO	Recall	#Proposals
$T_{vid} \cup T_{iMotion}$	42.6	33.4	5410
+Motion pruning	41.7	32.5	1298
+Overlap pruning	40.9	30.6	472
+Spatial refinement	42.3	32.8	472

With overlap pruning the number of proposals goes down further to ~9% of the original number with a small loss in MABO and Recall. This loss is compensated by spatial refinement leading to the same performance with ten times fewer proposals

MABO and Recall over the initial version of our approach (Jain et al. 2014), indicating the value of spatiotemporal refinement and pruning. On UCF101, we achieve MABO and Recall comparable to the method of van Gemert et al. (2015), be it that we need five times less proposals. Overall, Tubelets provides state-of-the-art quality while balancing the number of proposals. Next, we evaluate the action localization abilities of Tubelets when combined with modern representations.

6 Experiments: Action Localization

In this section we evaluate our approach for action localization on UCF Sports, MSR-II and UCF101. For positive training examples, we use the ground-truth and our Tubelets that have localization score or overlap greater than 0.7 with the ground-truth. Negative samples are randomly selected by considering Tubelets whose overlap with ground-truth is less than 0.15. This scheme is followed for UCF Sports and UCF101. In case of MSR-II cross-dataset evaluation is employed, the training samples consist of the clips from KTH dataset while testing is performed on the Tubelets from the videos of MSR-II. We apply power normalization followed by ℓ_2 normalization before training with a linear SVM. One round of retraining on "hard-negatives" was enough as additional rounds did not improve performance further. Again, there is no retraining in case of MSR-II, only initial classifier trained on videos from KTH dataset are used.

We first give details of the representations used to encode each Tubelet and show their impact on the UCF Sports dataset. Then, we compare our action localization results with the state-of-the-art methods on each of the three datasets.

6.1 Tubelet Representations

We capture motion information by the four local descriptors computed along the improved trajectories (Wang and Schmid 2013). To represent the local descriptors, we use bag-of-words or Fisher vectors. A Tubelet is assigned the trajectories that have more than half of their points inside the Tubelet. For the third representation, we use features from a Convolutional Neural Network layer and average pool them over the frames. Below we explain these three representations.

Bag of Words (BoW) The local descriptors are vector quantized and pooled into a bag-of-words histogram. We set the vocabulary size to $K = 500$. This is the least expensive (and expressive) of the three representations.

Table 10 Average execution times per video for all stages of Tubelet generation on MSR-II

	Time (s)	Implementation
Initial segmentation		
Video	1264.1	By Xu and Corso (2012)
iMotion	236.1	By Xu and Corso (2012)
Grouping		
Video	7652.4	C with Matlab
iMotion	314.2	C with Matlab
Pruning proposals		
Motion pruning	105.8	C with Matlab
Overlap pruning (Video)	226.8	Matlab
Overlap pruning (*iMotion*)	134.3	Matlab
Spatiotemporal refinement		
Temporal refinement	19.3	C with Matlab
Spatial refinement	2164.2	Matlab
Motion trajectories	203.8	By Wang and Schmid (2013)
Total time	12,321	

On average there are 766.9 frames per video

Table 11 Comparing quality of action proposals against state-of-the-art

	MABO	Recall	AvgRecall	#Proposals
UCF sports				
Jain et al. (2014)	62.7	78.7	–	1642
Oneata et al. (2014a)	55.6	68.1	–	3000
van Gemert et al. (2015)	64.2	89.4	–	1449
Puscas et al. (2015)	62.2	–	–	340
Tubelets	**67.5**	**91.9**	37.5	**289**
MSR-II				
Jain et al. (2014)	34.8	3.0	–	4218
van Gemert et al. (2015)	47.9	44.3	–	**6706**
Tubelets	**48.9**	**47.4**	10.4	7287
UCF 101				
van Gemert et al. (2015)	40.0	**35.5**	–	2299
Tubelets	**42.3**	32.8	9.2	**472**

Our Tubelets outperform all other approaches on these three datasets with a modest number of proposals. Our Recall on UCF101 is slightly behind the approach of van Gemert et al. (2015), be it they use five times more proposals

Bold value indicates highest MABO/Recall and lowest #Proposals for a each dataset

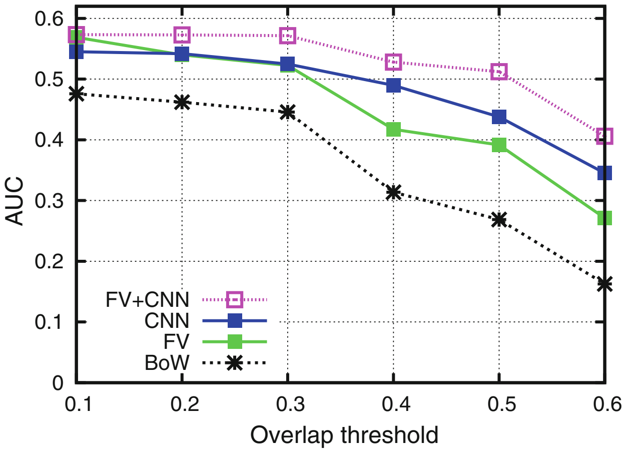


Fig. 9 Comparing representations: bag-of-words, Fisher vector and CNN features on UCF Sports, performance is measured by AUC for σ from 0.1 to 0.6, following (Tian et al. 2013). The best AUC is obtained when both Fisher vector and CNN features are combined for the Tubelet representation

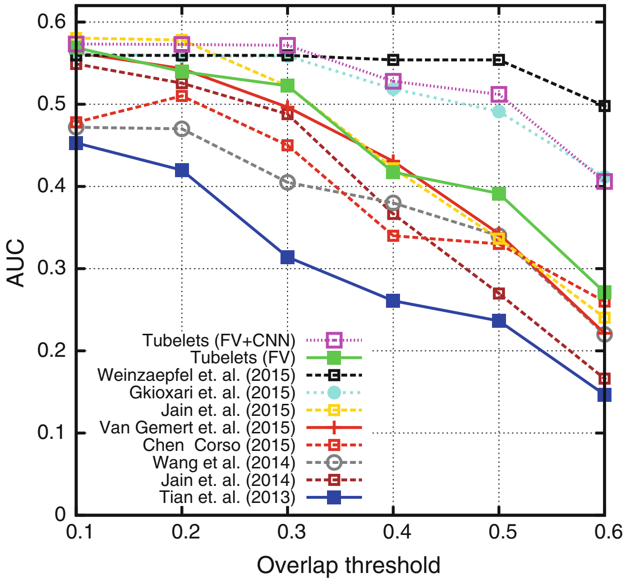


Fig. 10 Comparison with state-of-the-art methods on UCF Sports, performance is measured by AUC for σ from 0.1 to 0.6

Fisher Vectors (FV) We first apply PCA on the local descriptors and reduce the dimensionality by a factor of two. Then 256,000 descriptors are selected at random from a training set to estimate a Gaussian Mixture Model with K (=128) Gaussians. Each video is then represented by $2DK$ dimensional Fisher vector, where D is the dimension of the descriptor after PCA. Finally, we apply power and ℓ_2 normalization to the Fisher vector as suggested in (Perronnin et al. 2010). The feature computation is reasonably efficient but the memory requirement would be a bottleneck if the number of proposals is high (e.g.>5000). Fisher vectors have been used for temporal action localization by Oneata et al. (2014b) and for spatiotemporal action localization by van Gemert et al. (2015).

Convolutional Neural Network (CNN) We use an in-house implementation of GoogLeNet (Szegedy et al. 2015), trained on ImageNet over 15k object categories (Jain et al. 2015b) without fine-tuning. The features are extracted from the fully-connected layer (before softmax2) of the network, which is a 1024 dimensional vector to represent a bounding box in a frame. Since a Tubelet is a sequence of bounding boxes, the

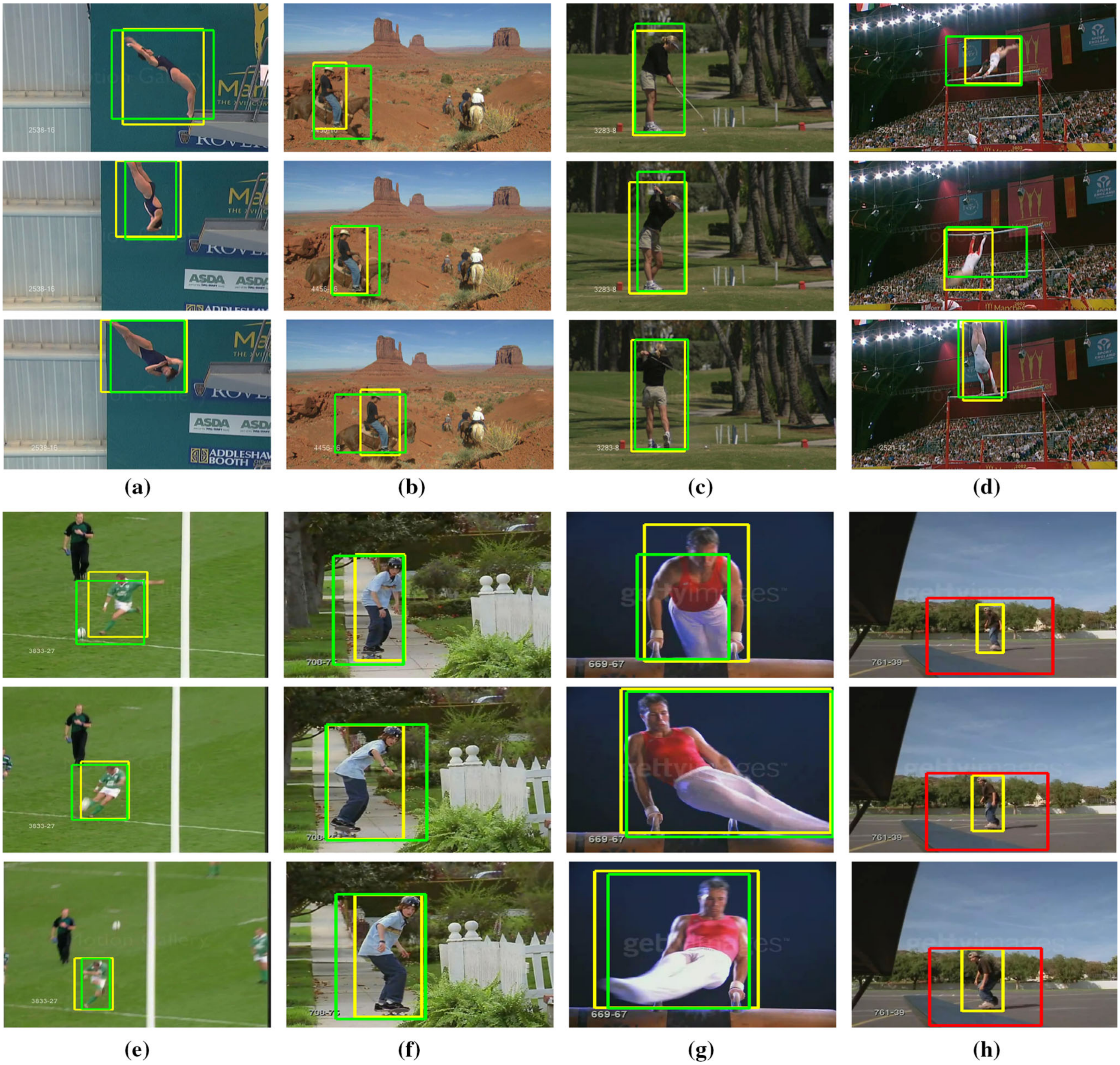


Fig. 11 Localization results shown as a sequence of bounding boxes (UCF-Sports): ground-truth is shown in *yellow*, correctly localized detections in *green* and poorly localized ones in *red*. Caption below each sequence reports the class detected. **a** Diving. **b** Riding-horse. **c** Golf. **d** Swinging-bar. **e** Kicking. **f** Skating. **g** Swinging-bench. **h** Skating (Color figure online)

final representation for it is obtained by averaging the feature vectors for the sampled frames (2 frames per second). Here, the memory requirement is limited, and feature computation is the costly operation, motivating the need for a compact set of action proposals.

Comparing Representations We now analyze the impact of the above three Tubelet representations on the UCF Sports dataset, following the process described in Sect. 4.2. Following popular practice, we use area under ROC curve (AUC) as the evaluation measure, as common for this dataset. Figure 9 compares the performance of the various Tubelet representations for a varying overlap threshold. We observe a clear improvement when moving from BoW to FV, to CNN and eventually the combination of FV and CNN, especially for higher thresholds ($\sigma \geq 0.4$).

6.2 Comparison with State-of-the-Art Methods

We now compare our approach with state-of-the-art methods on the three datasets.

Table 12 Comparison with state-of-the-art methods on MSR-II: average precision (AP) and mean AP are reported

Method	Boxing	Clapping	Waving	mAP
Cao et al. (2010)	17.5	13.2	26.7	19.1
Tian et al. (2013)	38.9	23.9	44.7	35.8
Jain et al. (2014)	46.0	31.4	85.8	54.4
Yuan et al. (2011)	64.9	43.1	64.9	55.3
Wang et al. (2014)	41.7	50.2	80.9	57.6
Yu and Yuan (2015)	67.4	46.6	69.9	61.3
Mosabbeb et al. (2014)	72.4	56.9	81.1	70.1
van Gemert et al. (2015)	67.0	78.4	74.1	73.2
Chen and Corso (2015)	**94.4**	73.0	**87.7**	**85.0**
Tubelets	72.4	**79.9**	84.4	78.9

Highest AP for a class or mean AP are shown in bold

UCF Sports In Fig. 10, we compare the performance of our method with the best reported results in the literature. In (Jain et al. 2015b), the previous version of Tubelets were represented with FV and CNN features, hence for comparison we use Tubelets represented with FV + CNN (combined with late fusion). The boost over Jain et al. (2015b), relying on segmentation of video frames only, shows the importance of segmenting $iMotion$ maps as well. Tubelets represented with FV + CNN is competitive to the methods of Gkioxari and Malik (2015) and Weinzaepfel et al. (2015) and outperforms all other approaches. In terms of mean average precision at overlap threshold of 0.5 as reported by Gkioxari and Malik (2015) (75.8%) and Weinzaepfel et al. (2015) (90.5%), we score lower with 68.5%. Since van Gemert et al. (2015) uses only the FV representation, for fair comparison we also

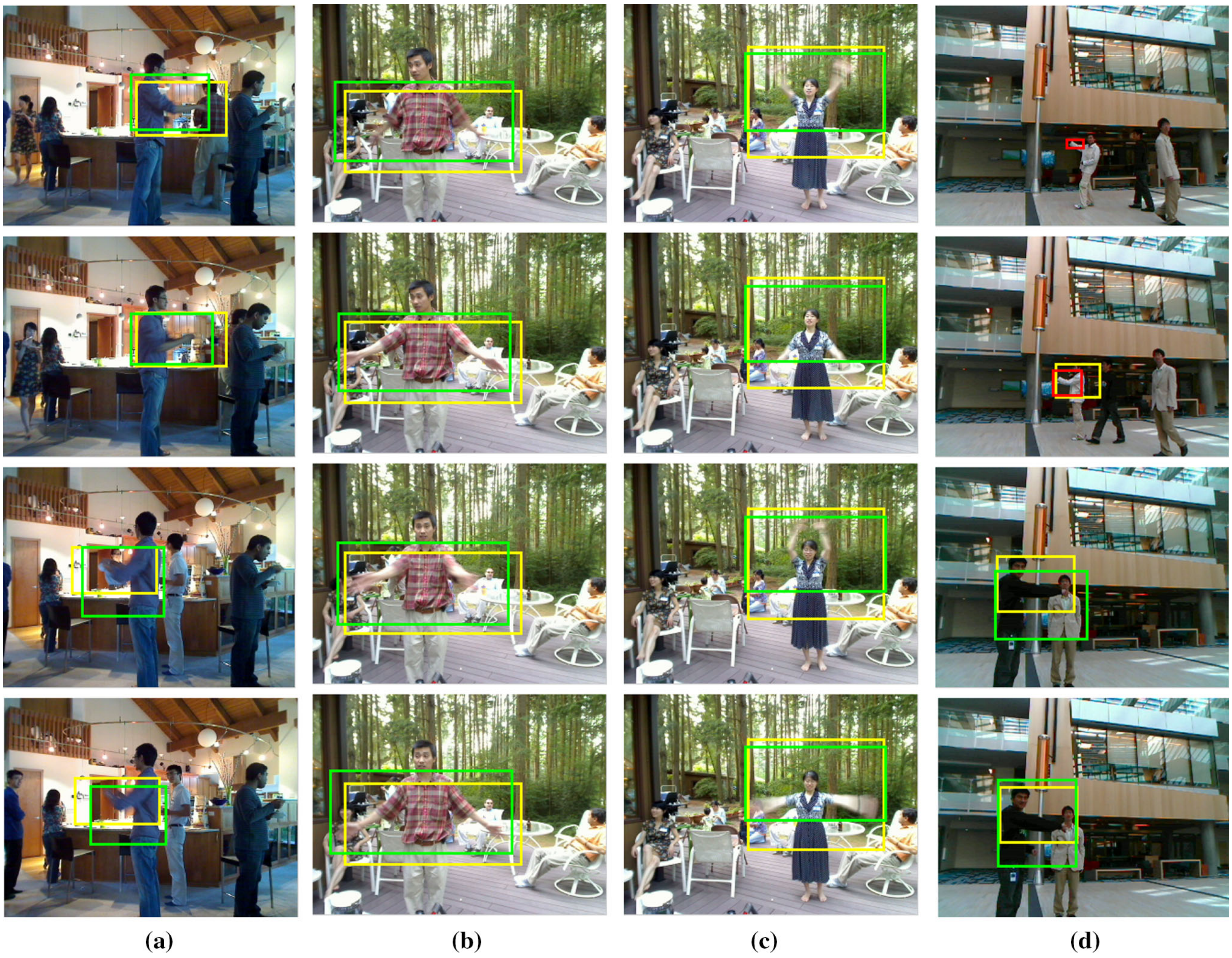


Fig. 12 Localization results shown as a sequence of bounding boxes (MSR-II): ground-truth is shown in *yellow*, correctly localized detections in *green* and poorly localized ones in *red*. Two instances of 'Boxing' being correctly localized are shown in the *first column*. The *middle two columns* show successful results for 'Clapping' and 'Waving' actions. *Last column* shows a failure case of poor localization of an instance of 'Boxing', while the second instance in the video is localized well. **a** Boxing. **b** HandClapping. **c** HandWaving. **d** Boxing (Color figure online)

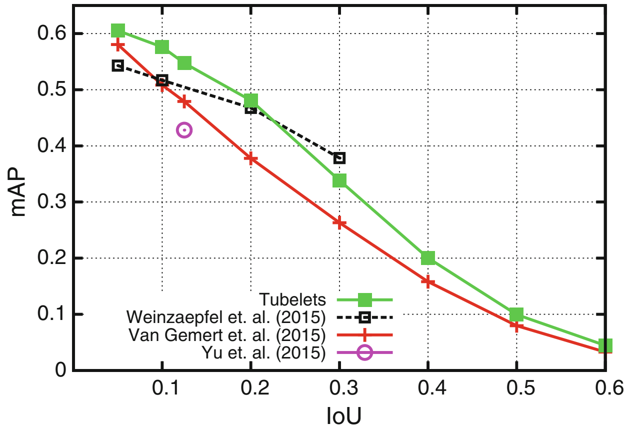


Fig. 13 Comparison with state-of-the-art methods on UCF101: Tubelets are obtained using the selected five grouping functions and represented with FV. Performance is measured by mAP for σ from 0.1 to 0.6

include Tubelets with a FV representation, which does better for most of the thresholds. Figure 11 shows some examples of action localizations from UCF Sports.

MSR-II This dataset is designed for cross-dataset evaluation. Following standard practice, we train on KTH dataset and test on MSR-II. While training for one class, the videos from other classes are used as the negative set. We use the FV representation to be more comparable with the competitive work of (van Gemert et al. 2015), which also generates action proposals in an unsupervised manner like Tubelets. In Table 12, we compare with several state-of-the-art methods; mean average precision (mAP) along with the APs for the three classes are reported. Following the usual practice on this dataset we report results for an overlap threshold of 0.125. Apart from Chen and Corso (2015), our approach outperforms all other methods by 5% of mAP or more. Chen and Corso (2015) very well utilizes information from motion trajectories and samples action proposals by clustering over a space-time trajectory graph. Motion trajectory based approaches are particularly well-suited for MSR-II dataset, as observed with our spatiotemporal refinement of Tubelets and also in (van Gemert et al. 2015). Similarly, the approach of Chen and Corso (2015) that is mainly focused on trajectories lead to excellent performance on MSR-II but its performance on UCF Sports is modest (Fig. 10). Finally, compared to the Tubelets in Jain et al. (2014), we improve mAP by 24.5%. Again, we claim the importance of using both input video frames and $iMotion$ maps for segmentation and spatiotemporal refinement of Tubelets. Figure 12 shows some examples of localizations for MSR-II.

UCF101 UCF101 is much larger than the other two datasets, with 24 action classes, and is currently the most challenging dataset for classification of proposals. Again, we represent Tubelets with FV following (van Gemert et al. 2015). In Fig. 13, we report mAPs for different overlap thresholds and compare Tubelets with three other approaches that report results on UCF101 dataset. Despite the use of human detection, the approach by Yu and Yuan (2015) is about 10% behind our method for an overlap threshold of 0.125. Weinzaepfel et al. (2015) uses bounding-box level action class supervision while generating proposals. Despite their additional supervision and use of two-stream CNN features, we achieve better mAP for 3 out of 4 overlap thresholds. Using box-level annotations their approach performs better at higher overlap thresholds. For limited number of classes they have an edge, whereas our approach would be more useful for generating proposals when the number of classes increases. The only other approach that uses proposals generated in an unsupervised manner, as we do, is APT by van Gemert et al. (2015). Tubelets outperform their approach while requiring only about a fifth of proposals (see Table 11).

Figure 14 displays some examples of action localizations from UCF101. With 24 classes this dataset offers larger variety in types of actions. Poor localization (shown in red) mainly happens in case of multiple actors, when during the action one of the actors gets occluded (see 'Salsa Spin'). Typically, in that case, Tubelets often encapsulates both actors together. However, the varying aspect ratios, diverse locations in the video frames, speed of action and multiple actors are well captured by our action proposal method.

7 Conclusions

We presented an unsupervised approach to generate proposals from super-voxels for action localization in videos. This is done by iterative grouping of super-voxels driven by both static features and motion features, motion being the key ingredient. We introduced independent motion evidence to characterize how the action related motion deviates from the background. The generated $iMotion$ maps provide a more efficient alternative for segmentation. Moreover, $iMotion$-based features allow for effective and efficient grouping of super-voxels. Our action proposals, Tubelets, are action class independent and implicitly cover variable aspect ratios and temporal lengths. We showed, for the first time, the effectiveness of Tubelets for action localization in Jain et al. (2014). In this paper, $iMotion$ maps are presented with further insights and the segmenting $iMotion$ maps is shown complementary to segmenting input video frames. Additionally, we introduced spatiotemporal refinement and pruning of Tubelets. Spatiotemporal refinement overcomes the tendency of super-voxels to sometimes follow the actor even after the action is completed. This led to improved MABO and Recall scores, especially on the untrimmed videos of MSR-II (Table 8), while pruning kept the number of Tubelets limited. The impact of these and the other components of

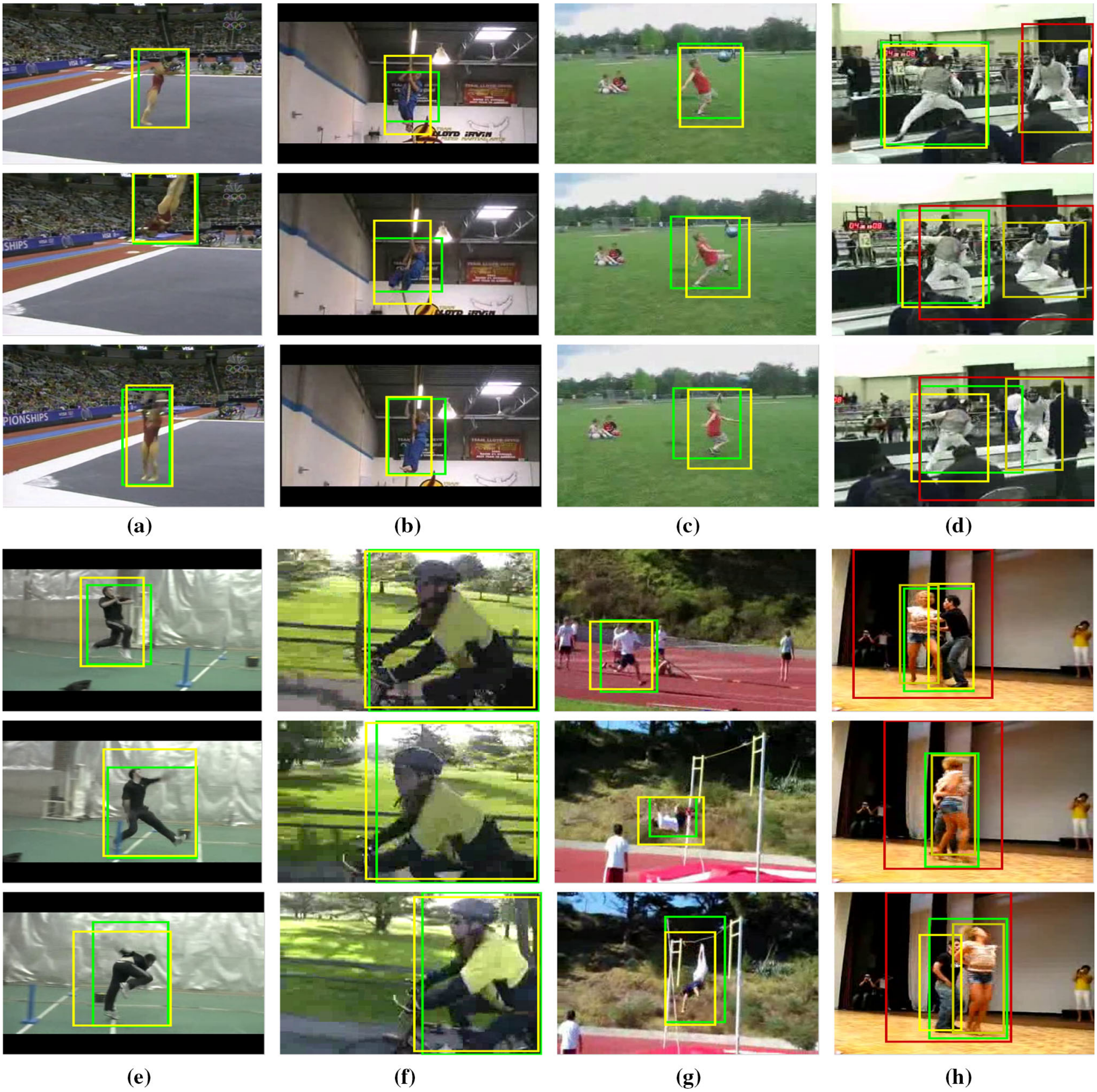


Fig. 14 Localization results shown as a sequence of bounding boxes (UCF101): ground-truth is shown in *yellow*, correctly localized detections in *green* and poorly localized ones in *red*. Caption below each sequence reports the class detected. In case of multiple actors ground-truth boxes are shown *darker* for the second actor. Poor localization mainly happens in such cases when during the action one of the actors gets occluded (see Salsa spin) and typically Tubelet often encapsulates both actors together. With 24 classes UCF101 offers larger variety in types of action, which is well captured by our action proposal method. **a** Floor gymnastics. **b** Rope climbing. **c** Soccer juggling. **d** Fencing. **e** Cricket bowling. **f** Biking. **g** Pole vault. **h** Salsa spin (Color figure online)

Tubelet generation are extensively evaluated in our experiments.

We evaluate our method for both action proposal quality and action localization. For action proposal quality, Tubelets beat all other existing approaches on the three datasets with much fewer number of proposals (Table 11). For action localization, our method leads to the best performance on UCF101 and second best on UCF Sports and MSR-II. The method of Chen and Corso (2015) gets best mAP for MSR-II but its performance on UCF Sports is rather modest. Similarly, Weinzaepfel et al. (2015) does well on UCF Sports and UCF101 but being supervised in generating proposals is not easy to apply on MSR-II. Ours is the only method that delivers excellent performance on both the trimmed videos of UCF Sports and UCF101 as well as the untrimmed videos of MSR-II.

References

Brox, T., & Malik, J. (September 2010). Object segmentation by long term analysis of point trajectories. In *Proceedings of the European conference on computer vision*.

Cao, L., Liu, Z., & Huang, T. S. (June 2010). Cross-dataset action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Chen, W., & Corso, J. J. (2015). Action detection by implicit intentional motion clustering. In *Proceedings of the IEEE international conference on computer vision*.

Chen, W., Xiong, C., Xu, R., & Corso, J. (2014). Actionness ranking with lattice conditional ordinal random fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 748–755).

Cheron, G., Laptev, I., & Schmid, C. (December 2015). P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, *74*(368), 829–836. doi:10.1080/01621459.1979.10481038.

Dalal, N., & Triggs, B. (June 2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Delaitre, V., Laptev, I., Sivic, J. (2010). Recognizing human actions in still images: A study of bag-of-features and part-based representations. In *BMVC*.

Derpanis, K. G., Sizintsev, M., Cannons, K. J., & Wildes, R. P. (2013). Action spotting and recognition based on a spatiotemporal orientation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(3), 527–540.

Dollar, P., Rabaud, V., Cottrell, G., & Belongie, S. (October 2005). Behavior recognition via sparse spatio-temporal features. In *Visual surveillance and performance evaluation of tracking and surveillance*.

Everts, I., van Gemert, J. C., & Gevers, T. (2014). Evaluation of color spatio-temporal interest points for human action recognition. *IEEE Transactions on Image Processing*, *23*(4), 1569–1580.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, *59*(2), 167–181.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Girshick, R. B., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(1), 142–158.

Gkioxari, G., & Malik, J. (2015). Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2015). What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(4), 814–830.

Huber, P. J. (1981). *Robust statistics*. New York: Wiley.

Jain, M., Jégou, H., & Bouthemy, P. (June 2013). Better exploiting motion for better action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jain, M., van Gemert, J. C., Jégou, H., Bouthemy, P., & Snoek, C. G. M. (June 2014). Action localization by tubelets from motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jain, M., van Gemert, J. C., Mensink, T., & Snoek, C. G. M. (2015a). Objects2action: Classifying and localizing actions without any video example. In *Proceedings of the IEEE international conference on computer vision* (pp. 4588–4596).

Jain, M., van Gemert, J. C., & Snoek, C. G. M. (June 2015b). What do 15,000 object categories tell us about classifying and localizing actions? In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jain, M., Jégou, H., & Bouthemy, P. (2016). Improved motion description for action classification. *Frontiers in ICT*, *2*, 28.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., & Schmid, C. (2012). Aggregating local descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(9), 1704–1716.

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., & Black, M. J. (December 2013). Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*.

Kang, K., Ouyang, W., Li, H., & Wang, X. (June 2016). Object detection from video tubelets with convolutional neural networks. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Ke, Y., Sukthankar, R., & Hebert, M. (October 2005). Efficient visual event detection using volumetric features. In *Proceedings of the IEEE international conference on computer vision*.

Kim, M., & Pavlovic, V. (2010). Structured output ordinal regression for dynamic facial emotion intensity prediction. In *European conference on computer vision* (pp. 649–662). Springer.

Kläser, A., Marszalek, M., & Schmid, C. (September 2008). A spatio-temporal descriptor based on 3d-gradients. In *Proceedings of the British machine vision conference*.

Kläser, A., Marszałek, M., Schmid, C., & Zisserman, A. (2012). Human focused action localization in video. In *Trends and topics in computer vision* (pp. 219–233).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

Kwak, S., Cho, M., Laptev, I., Ponce, J., & Schmid, C. (2015). Unsupervised object discovery and tracking in video collections. In *ICCV*.

Lampert, C. H., Blaschko, M. B., & Hofmann, T. (June 2008). Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Lan, T., Wang, Y., & Mori, G. (November 2011). Discriminative figure-centric models for joint action localization and recognition. In *Proceedings of the IEEE international conference on computer vision*.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2), 107–123.

Ma, S., Zhang, J., Ikizler-Cinbis, N., & Sclaroff, S. (2013). Action recognition and localization by hierarchical space-time segments. In *Proceedings of the IEEE international conference on computer vision* (pp. 2744–2751).

Maji, S., Bourdev, L., & Malik, J. (2011). Action recognition from a distributed representation of pose and appearance. In *Proceedings* *of the IEEE conference on computer vision and pattern recognition* (pp. 3177–3184).

Manen, S., Guillaumin, M., & Van Gool L. (2013). Prime object proposals with randomized Prim's algorithm. In *Proceedings of the IEEE international conference on computer vision*.

Mosabbeb, E. A., Cabral, R., De la Torre, F., & Fathy, M. (2014). Multilabel discriminative weakly-supervised human activity recognition and localization. In *ACCV*.

Ng, J. Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4694–4702).

Odobez, J.-M., & Bouthemy, P. (1995). Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, *6*(4), 348–365.

Oneata, D., Verbeek, J., & Schmid, C. (December 2013). Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the IEEE international conference on computer vision*.

Oneata, D., Revaud, J., Verbeek, J., & Schmid, C. (2014a). Spatio-temporal object detection proposals. In *Proceedings of the European conference on computer vision*.

Oneata, D., Verbeek, J., & Schmid, C. (2014b). Efficient action localization with approximately normalized fisher vectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Perronnin, F., & Dance, C. R. (2007). Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Perronnin, F., Sánchez, J., & Mensink, T. (September 2010). Improving the fisher kernel for large-scale image classification. In *Proceedings of the European conference on computer vision*.

Piriou, G., Bouthemy, P., & Yao, J.-F. (2006). Recognition of dynamic video contents with global probabilistic models of visual motion. *IEEE Transactions on Image Processing*, *15*(11), 3417–3430.

Prest, A., Leistner, C., Civera, J., Schmid, C., & Ferrari, V. (2012). Learning object class detectors from weakly annotated video. In *CVPR*.

Puscas, M., Sangineto, E., Culibrk, D., & Sebe, N. (2015). Unsupervised tube extraction using transductive learning and dense trajectories. In *Proceedings of the IEEE international conference on computer vision*.

Raptis, M., Kokkinos, I., & Soatto, S. (June 2012). Discovering discriminative action parts from mid-level video representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Rodriguez, M. D., Ahmed, J., & Shah, M. (June 2008). Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, *105*(3), 222–245.

Schüldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local svm approach. In *Proceedings of international conference of pattern recognition*.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *NIPS*.

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, (2012). URL arxiv.org/abs/1212.0402.

Soomro, K., Idrees, H., & Shah, M. (2015). Action localization in videos through context walk. In *Proceedings of the IEEE international conference on computer vision*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tian, Y., Sukthankar, R., & Shah, M. (June 2013). Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tran, D., & Yuan, J. (June 2011). Optimal spatio-temporal path discovery for video event detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tran, D., & Yuan, J. (December 2012). Max-margin structured output regression for spatio-temporal action localization. In *Advances in neural information processing systems*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4489–4497).

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., & Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, *104*(2), 154–171.

van de Sande, K. E. A., Snoek, C. G. M., & Smeulders, A. W. M. (2014). Fisher and vlad with flair. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

van Gemert, J. C., Jain, M., Gati, E., & Snoek, C. G. M. (2015). APT: Action localization proposals from dense trajectories. In *Proceedings of the British machine vision conference*.

Viola, P. A., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137–154.

Wang, H., & Schmid, C. (December 2013). Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*.

Wang, H., Kläser, A., Schmid, C., & Liu, C.-L. (June 2011). Action recognition by dense trajectories. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, H., Kläser, A., Schmid, C., & Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60–79.

Wang, H., Oneata, D., Verbeek, J., & Schmid, C. (2015a). A robust and efficient video representation for action recognition. *International Journal of Computer Vision, 119*(3), 1–20.

Wang, L., Qiao, Y., & Tang, X. (2014). Video action detection with relational dynamic-poselets. In *Proceedings of the European conference on computer vision*.

Wang, L., Qiao, Y., & Tang, X. (2015b). Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4305–4314).

Wang, Y., & Mori, G. (2011). Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(7), 1310–1323.

Weinzaepfel, P., Harchaoui, Z., & Schmid, C. (2015). Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*.

Xu, C., & Corso, J. J. (2012). Evaluation of super-voxel methods for early video processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yu, G., & Yuan, J. (2015). Fast action proposals for human action detection and search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yuan, J., Liu, Zicheng, & Wu, Y. (June 2009). Discriminative subvolume search for efficient action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yuan, J., Liu, Z., & Ying, W. (2011). Discriminative video pattern search for efficient action detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(9), 1728–1743.