Video-Based Action Detection Using Multiple Wearable Cameras

Kang Zheng(✉), Yuewei Lin, Youjie Zhou, Dhaval Salvi, Xiaochuan Fan, Dazhou Guo, Zibo Meng, and Song Wang

Department of Computer Science and Engineering, University of South Carolina, Room 3D19, 315 Main Street, Columbia, SC 29208, USA
{zheng37,lin59,zhou42,salvi,fan23,guo22,mengz,songwang}@email.sc.edu, songwang@cec.sc.edu

Abstract. This paper is focused on developing a new approach for video-based action detection where a set of temporally synchronized videos are taken by multiple wearable cameras from different and varying views and our goal is to accurately localize the starting and ending time of each instance of the actions of interest in such videos. Compared with traditional approaches based on fixed-camera videos, this new approach incorporates the visual attention of the camera wearers and allows for the action detection in a larger area, although it brings in new challenges such as unconstrained motion of cameras. In this approach, we leverage the multi-view information and the temporal synchronization of the input videos for more reliable action detection. Specifically, we detect and track the focal character in each video and conduct action recognition only for the focal character in each temporal sliding window. To more accurately localize the starting and ending time of actions, we develop a strategy that may merge temporally adjacent sliding windows when detecting durative actions, and non-maximally suppress temporally adjacent sliding windows when detecting momentary actions. Finally we propose a voting scheme to integrate the detection results from multiple videos for more accurate action detection. For the experiments, we collect a new dataset of multiple wearable-camera videos that reflect the complex scenarios in practice.

Keywords: Action detection · Multi-view videos · Focal character · Wearable cameras

1 Introduction

Video-based action detection, i.e., detecting the starting and ending time of the actions of interest, plays an important role in video surveillance, monitoring, anomaly detection, human computer interaction and many other computer-vision related applications. Traditionally, action detection in computer vision

K. Zheng and Y. Lin — Equal contribution.

L. Agapito et al. (Eds.): ECCV 2014 Workshops, Part I, LNCS 8925, pp. 727–741, 2015.
DOI: 10.1007/978-3-319-16178-5_51

is based on the videos collected from one or more fixed cameras, from which motion features are extracted and then fed to a trained classifier to determine the underlying action class [25,31,32,35]. However, using fixed-camera videos has two major limitations: 1) fixed cameras can only cover specific locations in a limited area, and 2) when multiple persons are present, it is difficult to decide the character of interest and his action from fixed-camera videos, especially with mutual occlusions in a crowded scene. In this paper, we consider a completely different approach where a set of temporally synchronized videos are collected from multiple wearable cameras and our main goal is to integrate the information from multiple wearable-camera videos for better action detection.

This new approach is applicable to many important scenarios. In a public crowded area, such as an airport, we can get all the security officers and other staff to wear a camera when they walk around for monitoring and detecting abnormal activities. In a prison, we can get each prisoner to wear a camera to collect videos, from which we may detect their individual activities, interactive activities, and group activities. Over a longer term, we may use these videos to infer the underlying social network among the prisoners to increase the security of the prison. In a kindergarten, we can get the teachers and the kids to wear a camera for recording what each of them sees daily, from which we can analyze kids' activities for finding kids with possible social difficulties, such as autism. We can see that, for some applications, camera wearers and action performers are different group of people, while for other applications, camera wearers and action performers can overlap. In our approach, we assume that the videos collected from multiple wearable cameras are temporally synchronized, which can be easily achieved by integrating a calibrated clock in each camera.

The proposed approach well addresses the limitation of the traditional approaches that use fixed cameras. 1) Camera wearers can move as he wants and therefore the videos can be collected in a much larger area; 2) Each collected video better reflects the attention of the wearer – the focal character is more likely to be located at the center of the view over a period of time and an abnormal activity may draw many camera-wearers' attention. However, this new approach also introduces new challenges compared to the approaches based on fixed cameras. For example, each camera is moving with the wearer and the view angle of the camera is totally unconstrained and time varying, while many available action recognition methods require the camera-view consistency between the training and testing data. In this paper, we leverage the multi-view information and the temporal synchronization of the input videos for more reliable action detection.

We adopt the temporal sliding-window technique to convert the action detection problem in long streaming videos to an action recognition problem over windowed short video clips. In each video clip, we first compensate the camera motions using the improved trajectories [32], followed by focal character detection by adapting the state-of-the-art detection and tracking algorithms [15,28]. After that, we extract the motion features around the focal character for action recognition. To more accurately localize the starting and ending time of an action, we develop a strategy that may merge temporally adjacent

sliding windows when detecting durative actions, and non-maximally suppress temporally adjacent sliding windows when detecting momentary actions. Finally, we develop a majority-voting technique to integrate the action detection results from multiple videos. To evaluate the performance of the proposed method, we conduct experiments on a newly collected dataset consisting of multiple wearable-camera videos with temporal synchronization. The main contributions in this paper are: 1) a new approach for action detection based on multiple wearable-camera videos. 2) a new dataset consisting of multiple wearable-camera videos for performance evaluation.

2 Related Work

Video-based action detection can usually be reduced to an action recognition problem, when the starting and ending frames of the action are specified – an action classifier is usually used to decide whether these frames describe the action. Three techniques have been used for this reduction: the sliding-window technique [11], which divides a long streaming video into a sequence of temporally overlapped short video clips, the tracking-based technique [18,38], which localizes human actions by person tracking, and the voting-based technique [2,38], which uses local spatiotemporal features to vote for the location parameters of an action. The sliding-window technique could be improved by using more efficient search strategy [39].

Most of the existing work on action recognition uses a single-view video taken by fixed cameras. Many motion-based feature descriptors have been proposed [1] for action recognition, such as space time interest points (STIPs) [20] and dense trajectories [31]. Extended from 2D features, 3D-SIFT [29] and HOG3D [19] have also been used for action recognition. Local spatiotemporal features [10] have been shown to be successful for action recognition and dense trajectories achieve best performance on a variety of datasets [31]. However, many of these features are sensitive to viewpoint changes – if the test videos are taken from the views that are different from the training videos, these features may lead to poor action recognition performance.

To address this problem, many view invariant methods have been developed for action recognition [17,27,41]. Motion history volumes (MHV) [34], histograms of 3D joint locations (HOJ3D) [36] and hankelets [22] are view invariant features. Temporal self-similarity descriptors show high stability under view changes [17]. Liu *et al.* [23] developed an approach to extract bag-of-bilingual-words (BoBW) to recognize human actions from different views. Recent studies show that pose estimation can benefit action recognition [37], e.g., key poses are very useful for recognizing actions from various views [6,24]. In [33], an exemplar-based Hidden Markov Model (HMM) is proposed for free view action recognition.

In multi-view action recognition, a set of videos are taken from different views by different cameras. There are basically two types of fusion scheme to combine the multi-view videos for action recognition: feature-level fusion and decision-level fusion. Feature-level fusion generally employs bag-of-words model to combine features from multiple views [40]. Decision-level fusion simply combines the

classification scores from all the views [26]. 3D action recognition approaches usually fuse the visual information by obtaining 3D body poses from 2D body poses in terms of binary silhouettes [5]. Most existing work on multi-view action recognition are based on videos taken by fixed cameras. As mentioned earlier, they suffer from the problems of limited spatial coverage and degraded performance in crowded scenes.

Also related to this paper is the egocentric video analysis and action recognition. For example, in [13,14] egocentric videos are used to recognize the daily actions and predict the gaze of the wearer. Similar to our work, they also take the videos from wearable cameras for action recognition. However, they are completely different from our work – in this paper, we recognize the actions of the performers present in the videos while the egocentric action recognition aims to recognize the actions of the camera wearers.

3 Proposed Method

3.1 Problem Description and Method Overview

We have a group of people, named *(camera) wearers*, each of whom wears a camera over head, such as Google Glasses or GoPro. Meanwhile, we have a group of people, named *performers*, each of whom performs actions over time. There may be overlap between wearers and performers, i.e., some wearers are also performers and vice versa. Over a period of time, each camera records a video that reflects what its wearer sees and the videos from all the cameras are temporally synchronized. We assume that at any time each wearer focuses his attention on at most one "focal character", who is one of the performers. The wearer may move as he wants during the video recording to target better to a performer or switch his attention to another performer. An example of such videos is shown in Fig. 1, where five videos from five wearers are shown in five rows respectively. For long streaming videos, the focal character in each video may change over time and the focal character may perform different actions at different time. Our goal of action detection is to accurately localize the starting and ending time of each instance of the actions of interest performed by a focal character by fusing the information from all the videos.

In this paper, we use the sliding-window technique to convert the action detection problem on a long streaming video into an action recognition problem on short video clips. Following sliding windows, a long-streaming video is temporally divided into a sequence of overlapped short video clips and the features from each clip are then fed into a trained classifier to determine whether a certain action occurs in the video clip. If yes, the action is detected with starting and ending frames aligned with the corresponding sliding window. However, in practice, instances of a same action or different actions may show substantially different duration time and it is impossible to exhaustively try different sliding-window lengths to match all possible action durations. In Section 3.4, we will introduce a new merging and suppression strategy to the temporally adjacent sliding windows to address this problem.

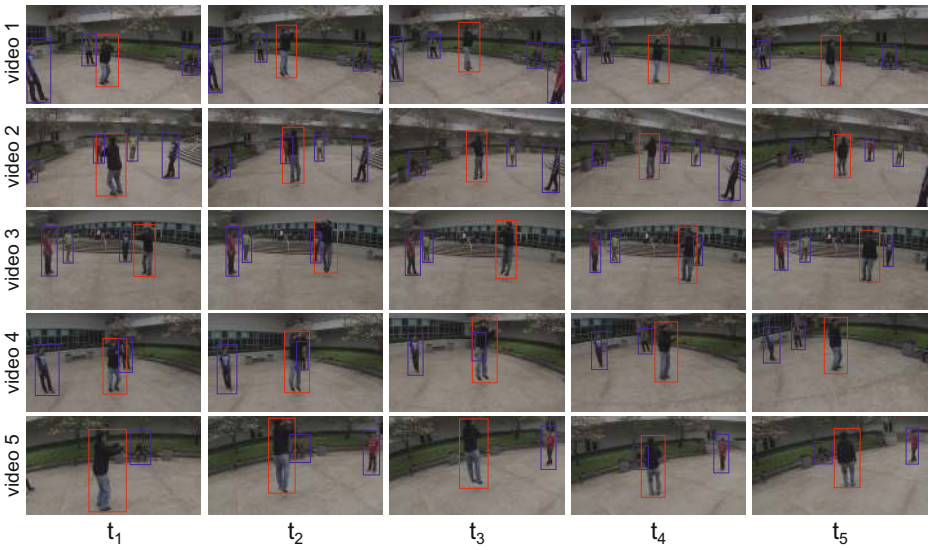

Fig. 1. An example of the videos taken by five wearable cameras from different views. Each row shows a sequence of frames from one video (i.e., from one wearer's camera) and each column shows the five frames with the same time stamp in the five videos respectively. In each frame, the focal character is highlighted in a red box and the blue boxes indicate the camera wearers, who wear a GoPro camera over the head to produce these five videos. Some wearers are out of the view, e.g., a wearer is not present in the video taken by his own camera. The same focal character is performing a *jump* action in these five videos.

When multiple temporally synchronized videos are taken for the same focal character, we can integrate the action detection results on all the videos for more reliable action detections. In this paper, we identify the focal character on each video, track its motion, extract its motion features, and feed the extracted features into trained classifiers for action detection on each video. In Section 3.5, we will introduce a voting scheme to integrate the action detection results from multiple synchronized videos.

Moving cameras pose new challenges in action recognition because the extracted features may mix the desired foreground (focal character) motion and undesired background (camera) motion. In this paper, we remove camera motions by following the idea in [32]. Specifically, we first extract the SURF features [3] and match them between neighboring frames using nearest neighbor search. Optical flow is also used to establish a dense correspondence between neighboring frames. Finally we estimate the homography between frames by RANSAC [16] and rectify each frame to remove camera motions.

After removing the camera motions, on each video clip we extract the dense trajectories and its corresponding descriptors using the algorithms introduced in [31,32]. Specifically, trajectories are built by tracking feature points detected

in a dense optical flow field [12] and then the local motion descriptors HOG [7], HOF [21] and MBH [8] are computed and concatenated as the input for the action classifier for both training and testing. We only consider trajectories with a length of no less than 15 frames. We use the standard bag-of-feature-words approach to encode the extracted features – for each feature descriptor, we use K-means to construct a codebook from $100,000$ randomly sampled trajectory features. The number of entries in each codebook is $4,000$. In the following, we discuss in detail the major steps in the proposed method, i.e., focal character detection, temporal merging and suppression for action detection and integrated action detection from multiple videos.

3.2 Focal Character Detection

By detecting the focal character, we can focus only on his motion features for more reliable action recognition. As discussed earlier, videos taken by wearable cameras facilitate the focal character detection since the wearers usually focus their attentions on their respective focal characters. In this paper we take the following three steps to detect the focal character in each video clip constructed by the sliding windows.

1. Detecting the persons in each video frame using the state-of-the-art human detectors [15], for which we use a publicly available software package[1].
2. Tracking the motion of the detected persons along the video clip using the multiple-object tracking algorithm [28] for which we also use a publicly available software package[2]. Given missing detections on some frames (e.g., red dashed box in Fig. 2), we need to link short human tracklets (e.g., solid curves in Fig. 2) into longer tracks (e.g., the long red track in Fig. 2).
3. Ranking human tracks in terms of a proposed attention score function and selecting the track with the highest score as the focal character, e.g., the long red track in Fig. 2.

In the following, we elaborate on the tracklet linking and the attention score function.

Tracklet Linking Let $\{T_1, \cdots, T_N\}$ be the N tracklets obtained by the human detection/tracking. Each tracklet is a continuous sequence of detected bounding boxes, i.e., $T_i = \left\{B_t^i\right\}_{t=t_1}^{t_2}$ where B_t^i represents 2D coordinates of the 4 corners of the bounding box in frame t, and t_1 and t_2 indicate the starting and the ending frames of this tracklet. The tracklet linking task can be formulated as a Generalized Linear Assignment (GLA) problem [9]:

[1] http://www.cs.berkeley.edu/~rbg/latent/index.html
[2] http://people.csail.mit.edu/hpirsiav/

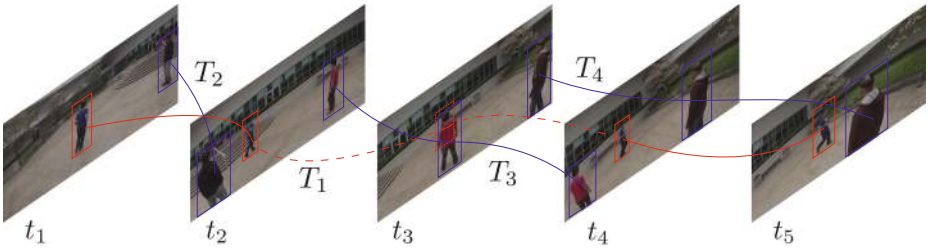

Fig. 2. An illustration of the focal character detection.

$$\begin{aligned} &\min_{X} \sum_{i=1}^{N} \sum_{j=1}^{N} D_{ij} X_{ij} \\ &s.t. \sum_{i=1}^{N} X_{ij} \leq 1; \sum_{j=1}^{N} X_{ij} \leq 1; X_{ij} \in \{0,1\} \end{aligned} \tag{1}$$

where $X_{ij} = 1$ indicates the linking of the last frame of T_i to the first frame of T_j and D_{ij} is a distance measure between two tracklets T_i and T_j when $X_{ij} = 1$.

Specifically, we define $D_{ij} = D_P(T_i, T_j) \times D_A(T_i, T_j)$ where D_P and D_A are the location and appearance distances between T_i and T_j respectively. The location distance D_P is defined by the Euclidean distance between the spatiotemporal centers of T_i and T_j in terms of their bounding boxes. The appearance distance D_A is defined by the sum of χ^2 distances between their intensity histograms, over all three color channels inside all their bounding boxes.

The GLA problem defined in Eq. (1) is an NP-Complete problem [9] and in this paper, we use a greedy algorithm to find a locally optimal solution [9]. By tracklet linking, we can interpolate the missing bounding boxes and achieve longer human tracks along the windowed video clip.

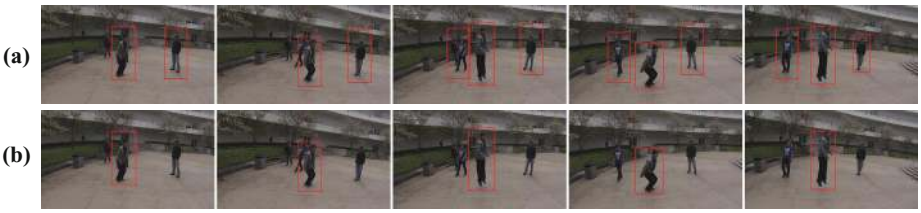

Fig. 3. An example of human detection and focal character detection. (a) Human detection results using Felzenszwalb detectors [15]. (b) Detected focal character.

Focal Character Detection. For each human track, we define an attention-score function to measure its likelihood of being the focal character for the wearer. Specifically, we quantify and integrate two attention principles here: 1)

the focal character is usually located inside the view of the camera wearer and the wearer usually moves his eyes (therefore his camera) to keep tracking the focal character. Mapped to the detected human tracks, the track of the focal character tends to be longer than the other tracks; 2) the focal character is usually located at a similar location in the view along a video clip. Based on these, we define the attention score $A(T)$ for a human track $T = \{B_t\}_{t=t_1}^{t_2}$ as

$$A(T) = \sum_{t=t_1}^{t_2} \exp\left\{-\left(\frac{(\bar{B}_{tx} - \mu_{Tx})^2}{\sigma_x^2} + \frac{(\bar{B}_{ty} - \mu_{Ty})^2}{\sigma_y^2}\right)\right\} \tag{2}$$

where (μ_{Tx}, μ_{Ty}) denotes the mean values of track T along the x and y axes, respectively, $(\bar{B}_{tx}, \bar{B}_{ty})$ denotes the center of the bounding box B_t in the track T at the frame t, and σ_x and σ_y control the level of the center bias, which we empirically set to $\frac{1}{12}$ and $\frac{1}{4}$ respectively in all our experiments. Given a set of human tracks in the video clip, we simply pick the one with the highest attention score as the track of the focal character, as shown in Fig. 3.

3.3 Action Recognition

In this section, we consider the action recognition on a short video clip generated by sliding windows. For both training and testing, we extract dense trajectory features only inside the bounding boxes of the focal character. This way, other irrelevant motion features in the background and associated to the non-focal characters will be excluded, with which we can achieve more accurate action recognition. Considering the large feature variation of a human action, we use a state-of-the-art sparse coding technique for action recognition [4]. In the training stage, we simply collect all the training instances of each action (actually their feature vectors) as the bases for the action class. In the testing stage, we extract the motion-feature vector of the focal character and sparsely reconstruct it using the bases of each action class. The smaller the reconstruction error, the higher the likelihood that this test video clip belongs to the action class. Specifically, let $\mathcal{T}$ be the feature vector extracted from a testing video clip. The likelihood of $\mathcal{T}$ belongs to action i is

$$L(i|\mathcal{T}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\|\mathcal{T} - \tilde{\mathcal{T}}_i\|^2}{2\sigma^2}\right). \tag{3}$$

where $\tilde{\mathcal{T}}_i = \mathcal{A}_i \mathbf{x}^*$ denotes the sparse coding reconstruction of feature vector $\mathcal{T}$ using the bases $\mathcal{A}_i$ in action class i and $\mathbf{x}^*$ is the linear combination coefficients of the sparse coding representation which can be derived by solving the following minimization problem:

$$\mathbf{x}^* = \min_{\mathbf{x}} \{\|\mathcal{T} - \mathcal{A}_i \mathbf{x}\|^2 + \alpha \|\mathbf{x}\|_0\}. \tag{4}$$

3.4 Action Detection

As mentioned before, the video clips used for action recognition are produced by sliding windows. In the simplest case, when an action is recognized in a video clip, we can take the corresponding sliding window (with the starting and ending frames) as the action detection result. However, in practice, different actions, or even the same action, may show different duration time. In particular, some actions, such as "handwave", "jump", "run" and "walk", are usually durative, while other actions, such as "sitdown", "standup", and "pickup", are usually momentary. Clearly, it is impossible to try all possible length sliding windows to detect actions with different durations. In this paper we propose a new strategy that conducts further temporal window merging or non-maximal suppression to detect actions with different durations.

We propose a three-step algorithm to temporally localize the starting and ending frames of each instance of the actions of interest. First, to accommodate the duration variation of each action, we try sliding windows with different lengths. Different from a momentary action that is usually completed in a short or limited time, a durative action may be continuously performed for an indefinite time. Thus, it is difficult to pick a small number of sliding-window lengths to well cover all possible durations of a durative action. Fortunately, durative actions are usually made up of repetitive action periods and the duration of each period is short and limited. For example, a durative "walk" action contains a sequence of repeated "footsteps". For a durative action, we select sliding-window lengths to cover the duration of the action period instead of the whole action.

Second, for each considered action class, we combine its action likelihood estimated on the video clips resulting from sliding windows with different lengths, e.g., l_1, l_2 and l_3 in Fig. 4, where the value of the curve labeled "window-length l_1" at time t is the action likelihood estimated on the video clip in the time window $[t - \frac{l_1}{2}, t + \frac{l_1}{2}]$ (centered at t with length l_1), using the approach we introduced above. To estimate a unified action likelihood at time t, a basic principle is that we pick the largest value at time t among all the curves, as shown by the point A in Fig. 4. In this example, l_1 is the most likely length of this action (or action period) at t. As a result, we can obtain the unified action likelihood curve $(U(t), S(t))$, where $U(t)$ is the maximum action likelihood over all tested different-length sliding windows centered at t and $S(t)$ is the corresponding window length that leads to $U(t)$.

Finally, based on the unified action likelihood (U, S), we perform a temporal merging/suppression strategy to better localize the starting and ending frames of the considered action. For a durative action, each sliding window may correspond to one of its action period. Our basic idea is to merge adjacent sliding windows with high action likelihood for durative action detection. Specifically, this merging operation is implemented by filtering out all the sliding windows with $U(t) < T_h$, where T_h is a preset threshold. This filtering actually leads to a set of temporally disjoint intervals in which all the t satisfy $U(t) \geq T_h$. For each of these intervals, say $[t_1, t_2]$, we take the temporal interval $[t_1 - \frac{S(t_1)}{2}, t_2 + \frac{S(t_2)}{2}]$ as a detection of the action. For a momentary action, we expect that it does not

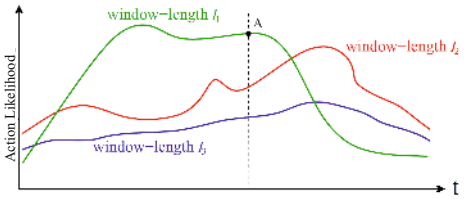

Fig. 4. An illustration of the estimated action likelihood using different-length sliding windows.

occur repetitively without any transition. We perform a temporal non-maximum suppression for detecting a momentary action – if $U(t) \geq T_h$ and $U(t)$ is a local maximum at t, we take the temporal interval $[t - \frac{S(t)}{2}, t + \frac{S(t)}{2}]$ as a detection of the action. This way, on each video, we detect actions of interest in the form of a set of temporal intervals, as illustrated in Fig. 5 where each detected action is labeled by red font.

3.5 Integrated Action Detection from Multiple Videos

In this section, we integrate the action detection results from multiple synchronized videos taken by different wearers to improve the accuracy of action detection. The basic idea is to use majority voting over all the videos to decide the underlying action class at each time. Note that here it is required that these videos are taken for a same focal character. As illustrated in Fig. 5, we take the following steps.

1. Temporally divide all the videos into uniform-length segments, e.g., segments (a) and (b) that are separated by the vertical dashed lines in Fig. 5. In this paper, we select the segment length to be 100 frames.
2. For each segment, e.g., segment (a) in Fig. 5, we examine its overlap with the temporal intervals of the detected actions in each video and label it with the corresponding action label or "no-action" when there is no overlap with any detected action intervals. For example, segment (a) is labeled "run" in Videos 1, 3, and 5, "walk" in Video 2, and "no-action" in Video 4.
3. For the considered segment, we perform a majority voting to update the action labels over all the videos. For example, on three out of five videos, segment (a) is labeled "run" in Fig. 5. We simply update the label of segment (a) to "run" on all the videos. When two or more actions are tied as majority, we pick the one with the maximum likelihood. For example, segment (c) is labeled "walk" on two videos and "run" on two other videos. We update the label of this segment to "run" on all the videos because "run" shows a higher likelihood.
4. After updating action labels for all the segments, update the action detection results by merging adjacent segments with the same labels, as shown in the last row of Fig. 5.

After these steps, the action detection results are the same for all the videos.

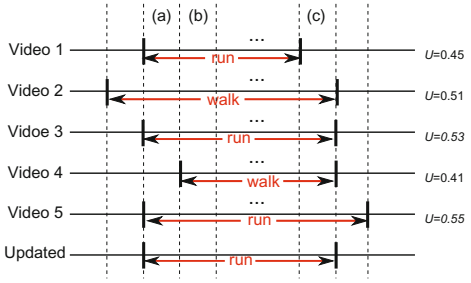

Fig. 5. An illustration of integrating action detection from multiple videos.

4 Experiments

We collect a new video dataset for evaluating the performance of the proposed method. In our experiment, we try two sliding-window lengths: 50 and 100 for computing the unified action likelihood. For comparison, we also try the traditional approach, where the motion features are extracted over the whole video without the focal character detection and the filtering of non-relevant features. Other than that, the comparison method is the same as the proposed method, including the type of the extracted features [31,32], camera motion compensation [32], and the application of sliding windows. For both the proposed method and the comparison method, we also examine the performance improvement by integrating the action detection results from multiple videos.

4.1 Data Collection

Popular video datasets that are currently used for evaluating action detection consist of either single-view videos with camera movement or multi-view videos taken by fixed cameras. In this work, we collect a new dataset that consists of temporally synchronized videos taken by multiple wearable cameras.

Specifically, we get 5 persons who are both performers and wearers and one more person who is only a performer. They perform 7 actions: *handwave*, *jump*, *pickup*, *run*, *sit-down*, *stand-up* and *walk* in an outdoor environment. Each of the 5 wearers mounts a GoPro camera over the head. We arrange the video recording in a way that the 6 performers alternately play as the focal character for the other people. As the focal character, each person plays the 7 actions once in the video recording. This way, we collected 5 temporally synchronized videos, each of which contains $5 \times 7 = 35$ instances of actions performed by 5 persons, excluding the wearer himself. The average duration of each action is about 18.4 seconds. We annotate these 5 videos for the focal characters and the starting and ending frames of each instance of the actions of interest, using the open video annotation tool of VATIC [30]. In our experiments, we use these 5 videos for testing the action detection method.

For training, we collect a set of video clips from two different views in a more controlled way. Each clip contains only one instance of the actions of interest.

Specifically, we get the same 6 persons to perform each of the 7 actions two or three times. In total we collected 204 video clips as the training data. The average length of the video clips in the training data is 11.5 seconds. The camera wearers are randomly selected from the five persons who are not the performer and the wearer may move his head to focus the attention on the performer in the recording. All the training videos (clips) are annotated with the focal characters for feature extraction and classifier training. Figure 6 shows sample frames of each action class from different cameras in our new dataset.

Fig. 6. Sample frames in the collected videos with annotated focal characters.

4.2 Independent Detection of Each Action

In this section, we conduct an experiment to detect each action independent of other actions. Specifically, we set the threshold T_h to 100 different values at the s-percentile of $U(t)$ over the entire video, i.e., $U(t) > T_h$ on $s\%$ of frames, where s continuously increases one by one from 1 to 100. Under each selected value of T_h, we perform temporal merging/suppression to detect each action independently. For each detected instance of an action (a temporal interval, e.g., D), if there exists an annotated ground-truth instance of the same action, e.g., G, with a temporal overlap $TO = \frac{|D \bigcap G|}{|D \bigcup G|}$, that is larger than $\frac{1}{8}$, we count this detection D to be a true positive. This way we can calculate the *precision*, *recall* and the *F-score*$= \frac{2 \times precision \times recall}{precision + recall}$. We pick the best *F-score* over all 100 selections of the threshold T_h for performance evaluation. From Table 1, we can see that the proposed method outperforms the comparison method on 6 out of 7 actions as well as the average performance. Note that in this experiment, the detected actions are allowed to temporally overlap with each other and therefore the integrated action detection technique proposed in Section 3.5 is not applicable. The performance reported in Table 1 is the average one over all the 5 videos.

4.3 Non-Overlap Action Detection

On the 5 collected long streaming videos, there is only one focal character at any time and the focal character can only perform one single action at any time. In this section, we enforce this constraint by keeping detecting at most one

Table 1. Performance (best *F-score*) of the proposed method and the comparison method when independently detecting each of the 7 actions on the collected multiple wearable-camera videos.

Methods	*handwave*	*jump*	*pickup*	*run*	*sitdown*	*standup*	*walk*	**Average**
Comparison	47.5%	48.0%	22.2%	48.4%	20.1%	13.1%	41.7%	35.5%
Proposed	55.0%	62.0%	19.6%	61.5%	22.2%	22.7%	60.7%	42.3%

action (the one with the highest action likelihood) at any time along the video. Specifically, we set the threshold T_h at the 99 percentile of $U(t)$. Then for each windowed short clip, we only consider it for the action with the highest action likelihood and the likelihood for the other actions is directly set to zero for this clip. After that, we follow the same temporal merging/suppression strategy to get the final action detection. Table 2 gives the *F-score* of the proposed method and the comparison method. It can be seen that the proposed method outperforms the comparison method in 3 out of 5 videos under two different definitions of the true positives – temporal overlap $TO > \frac{1}{4}$ and $TO > \frac{1}{8}$, respectively. We then further apply the technique developed in Section 3.5 to integrate the detection results from all 5 videos and the final detection performance is shown in the last column of Table 2. We can see that, by integrating detections from multiple videos, we achieve better action detection.

Table 2. Performance (*F-score*) of the proposed method and the comparison method when detecting all the 7 actions in a non-overlapping way on the collected multiple wearable-camera videos.

Methods		Video1	Video2	Video3	Video4	Video5	**Average**	**Integrated**
$TO > \frac{1}{4}$	Comparison	25.4%	14.3%	4.1%	22.2%	16.7%	16.5%	28.6%
	Proposed	22.6%	28.9%	13.2%	22.2%	20.8%	21.5%	35.4%
$TO > \frac{1}{8}$	Comparison	32.3%	23.8%	8.2%	27.2%	19.4%	22.2%	28.6%
	Proposed	26.4%	30.9%	20.8%	26.7%	29.2%	26.8%	38.0%

5 Conclusions

In this paper, we developed a new approach for action detection – input videos are taken by multiple wearable cameras with temporal synchronization. We developed algorithms to identify focal characters from each video and combined the multiple videos to more accurately detect the actions of the focal character. We developed a novel temporal merging/suppression algorithm to localize starting and ending time of both the durative and momentary actions. Image frames were rectified before feature extraction for removing the camera motion. A voting technique was developed to integrate the action detection from multiple videos. We also collected a video dataset that contains synchronized videos taken by multiple wearable cameras for performance evaluation. In the future, we plan

to enhance each of the steps of the proposed approach and develop algorithms to automatically identify subsets of videos with the same focal character.

Acknowledgement. This work was supported in part by AFOSR FA9550-11-1-0327 and NSF IIS-1017199.

References

1. Aggarwal, J., Ryoo, M.S.: Human activity analysis: A review. ACM Computing Surveys **43**(3), 16 (2011)
2. Bandla, S., Grauman, K.: Active learning of an action detector from untrimmed videos. In: ICCV (2013)
3. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
4. Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S., Siskind, J., Wang, S.: Recognize human activities from partially observed videos. In: CVPR (2013)
5. Chaaraoui, A.A., Climent-Pérez, P., Flórez-Revuelta, F.: An efficient approach for multi-view human action recognition based on bag-of-key-poses. In: Salah, A.A., Ruiz-del-Solar, J., Meriçli, Ç., Oudeyer, P.-Y. (eds.) HBU 2012. LNCS, vol. 7559, pp. 29–40. Springer, Heidelberg (2012)
6. Cheema, S., Eweiwi, A., Thurau, C., Bauckhage, C.: Action recognition by learning discriminative key poses. In: ICCV Workshops (2011)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
8. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
9. Dicle, C., Sznaier, M., Camps, O.: The way they move: tracking multiple targets with similar appearance. In: ICCV (2013)
10. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
11. Duchenne, O., Laptev, I., Sivic, J., Bach, F., Ponce, J.: Automatic annotation of human actions in video. In: CVPR (2009)
12. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003)
13. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: ICCV (2011)
14. Fathi, A., Li, Y., Rehg, J.M.: Learning to recognize daily actions using gaze. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 314–327. Springer, Heidelberg (2012)
15. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. TPAMI **32**, 1627–1645 (2010)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)

17. Junejo, I.N., Dexter, E., Laptev, I., Perez, P.: View-independent action recognition from temporal self-similarities. TPAMI **33**(1), 172–185 (2011)
18. Kläser, A., Marszałek, M., Schmid, C., Zisserman, A., et al.: Human focused action localization in video. In: SGA Workshop (2010)
19. Klaser, A., Marszałek, M., Schmid, C., et al.: A spatio-temporal descriptor based on 3D-gradients. In: BMVC (2008)
20. Laptev, I.: On space-time interest points. IJCV **64**(2–3), 107–123 (2005)
21. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
22. Li, B., Camps, O.I., Sznaier, M.: Cross-view activity recognition using hankelets. In: CVPR (2012)
23. Liu, J., Shah, M., Kuipers, B., Savarese, S.: Cross-view action recognition via view knowledge transfer. In: CVPR (2011)
24. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and Viterbi path searching. In: CVPR (2007)
25. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: ICCV Workshops (2009)
26. Naiel, M.A., Abdelwahab, M.M., El-Saban, M.: Multi-view human action recognition system employing 2DPCA. In: WACV (2011)
27. Parameswaran, V., Chellappa, R.: View invariance for human action recognition. IJCV **66**(1), 83–101 (2006)
28. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR (2011)
29. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional SIFT descriptor and its application to action recognition. In: ACM Multimedia (2007)
30. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. IJCV **101**(1), 184–204 (2013)
31. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
32. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
33. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3D exemplars. In: ICCV (2007)
34. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. CVIU **104**(2), 249–257 (2006)
35. Wu, S., Oreifej, O., Shah, M.: Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories. In: ICCV (2011)
36. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3D joints. In: CVPR Workshops (2012)
37. Yao, A., Gall, J., Fanelli, G., Van Gool, L.J.: Does human action recognition benefit from pose estimation?. In: BMVC (2011)
38. Yao, A., Gall, J., Van Gool, L.: A Hough transform-based voting framework for action recognition. In: CVPR (2010)
39. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
40. Zhang, T., Liu, S., Xu, C., Lu, H.: Human action recognition via multi-view learning. In: Proceedings of the Second International Conference on Internet Multimedia Computing and Service (2010)
41. Zheng, J., Jiang, Z.: Learning view-invariant sparse representations for cross-view action recognition. In: ICCV (2013)