2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Joint Detection and Recognition of Human Actions in Wireless Surveillance Camera Networks

Nikhil Naikal[1], Pedram Lajevardi[2] and Shankar. S. Sastry[1]

***Abstract*— Automatic recognition of human actions in video has been a highly addressed problem in robotics and computer vision. Majority of the recent work in literature has focused on classifying pre-segmented video clips, and some progress has also been made on joint detection and recognition of actions in complex video sequences. These methods, however, are not designed for wireless camera networks where the sensors have limited internal processing and communication capabilities.**

In this paper we present an efficient system for the joint detection and recognition of human actions using a network of wireless smart cameras. The foundation of our work is based on Deformable Part Models (DPMs) for detecting objects in static images. We have extended this framework to the single-view and multi-view video setting to jointly detect and recognize actions. We call this the Deformable Keyframe Model (DKM) and tightly integrate it within a centralized video analysis system. In our system, feature extraction is locally performed on-board wireless smart cameras, and the classification is performed at a base station with higher processing power. Our analysis demonstrates that this decoupling of the the recognition pipeline can significantly minimize the power and bandwidth consumed by the wireless cameras.

We experimentally validate our DKMs on two data sets. We first demonstrate the competitiveness of our algorithm by comparing its performance against other state-of-the-art methods, on a publicly available dataset. Then, we extensively validate our system on a novel dataset called the Bosch Multi-view Complex Action (BMCA) dataset. Our dataset consists of 11 actions continuously performed by 20 different subjects while being captured by cameras located at 4 different vantage points. In our experiments, we demonstrate that the presence of multiple-views improves the performance of action detection and recognition by about 15% over the single-view setting.

I. INTRODUCTION

Traditional Closed Circuit TV (CCTV) camera based surveillance systems typically consist of several wired cameras distributed within a building and the surrounding site, transmitting video streams to a control room as shown in Fig. 1. The security personnel employed are expected to monitor activity on all the video feeds, due to which several events can go unnoticed. Further, for applications such as indexing and retrieval of surveillance video, manual methods can be extremely time consuming, monotonous and stressful.

In the computer vision and robotics research communities, on the other hand, significant progress has been made in the areas of action recognition [1]-[8]. Most of this work has focussed on automatically recognizing human actions in publicly available datasets composed of pre-segmented video clips [1], [8]. The methods developed have primarily addressed variability in scale of the subjects, background clutter suppression and handling occlusions in the video clips [9]. These methods, however, are not directly applicable to surveillance systems as the temporal segmentation of continuous video is a challenging problem. Some recent works have focussed on partitioning the temporal segmentation and recognition process for long video sequences [10], [12]. However, these methods perform poorly, as low-level temporal cues are generally not discriminative enough for precisely partitioning the video. Some algorithms have also been developed for detecting and recognizing actions in generic video sequences [11], [23], [24]. These methods typically require a lot of processing at the image level, therefore making them hard to implement on a wireless smart camera.

Fig. 1. Typical CCTV control room with video feeds from several cameras

In this paper, we present a novel system for simultaneous detection and recognition of human actions in wireless smart camera networks. Our system is capable of handling video sequences captured by a single camera or multiple cameras with overlapping views. Our system is partitioned into distributed feature extraction (performed on the wireless smart cameras) and centralized spatiotemporal multi-view activity detection and recognition (performed at a base station). Each wireless camera in our system is capable of extracting, encoding and transmitting a descriptor vector corresponding to foreground objects of interest in every frame where motion is detected. At the base station, descriptor vectors from a single or multiple camera sources are fused within a graphical model framework for localizing and recognizing actions of interest. Our graphical model framework is based on the

[1]Nikhil Naikal and Shankar S. Sastry are with the Electrical Engineering and Computer Science Dept., University of California at Berkeley, Berkeley, CA - 94720, USA. {nnaikal,sastry}@eecs.berkeley.edu

[2]Pedram Lajevardi is with the Bosch Research and Technology Center, 4005, Miranda Ave., Palo Alto, CA - 94304, USA. Pedram.Lajevardi@us.bosch.com

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

famous Deformable Part Models (DPMs) for object detection in static images proposed by Felzenszwalb *et al.* [13]. We have extended the DPM framework to the spatiotemporal setting for both single and multiple view video streams. At its core, our algorithm replaces part appearance templates of the DPM by class-specific keyframes, and enforces spatiotemporal constraints between pairs of keyframes in the single-view setting. In the multiple-view setting, homography constraints [14] induced by the ground plane are used to enforce spatial connectivity between object regions in images from pairs of cameras.

The exposition of our paper is as follows. In section II we provide a brief literature review of activity recognition, while focussing on recent work that address similar problems as ours. We present our overall system pipeline in section III and discuss those basic elements of our pipeline that are drawn from previous work. The primary contribution of our paper is the centralized, multi-view spatiotemporal action detection and recognition algorithm and is presented in section IV. We validate the performance of our algorithm by performing experiments on standard and novel datasets, as presented in section V. Section VI provides a conclusion and an outline for future work.

II. LITERATURE REVIEW

Spatiotemporal bag-of-word representations are amongst the most popular approaches for action recognition because of their ease of use, and high discriminating capabilities [3], [8]. They have successfully been employed in both single view [10], [15], [16] and multi-view settings [6]. Although they work very well on temporally segmented video clips, they cannot be extended to action detection directly, as they ignore spatial and temporal relationships between discriminative templates. Further, detecting and describing spatiotemporal interest points would require significant processing, which can be a challenge for a low-power smart camera.

Other spatiotemporal template and filter based methods have also gained significant traction for action detection and recognition. Gorelick *et al.* [1] extract foreground silhouettes of moving people and use them to construct volumetric features for action recognition. Rodriguez *et al.* [4] use MACH filter responses to detect actions of interest. Ali & Shah [17] extract kinematic features from images to recognize actions. [9] provides an excellent survey of similar state-of-the-art methods. All these methods, however, require features extracted from every frame in a temporal volume. Thus, they would not work well within our framework, where the frequency of sampling images and transmitting extracted features needs to be variable in order to accommodate varying bandwidth constraints.

In the image based human pose and object detection literature, DPMs have gained a lot of popularity [13], [18]. Such human pose detectors have been fused with traditional image segmentation techniques to extract foreground pixels corresponding to people in static images [19]. Niebles *et al.* adapted the DPM framework to temporal action detection [11]. Tian *et al.* [23] and Lan *et al.* [24] have extended the framework to the spatiotemporal setting. While these methods are similar in spirit to our algorithm, their focus is on generic videos where no assumptions can be made regarding the background. Thus, these methods are very computationally intensive and cannot be easily adapted to wireless surveillance applications. Further, it becomes exponentially complex to extend their inference algorithms to multi-view scenarios, even after incorporating epipolar constraints.

Generative methods for activity recognition have been extensively addressed by the computer vision and control community. Sminchisescu *et al.* [25] have proposed conditional models for human action recognition. Fox *et al.* [26] and Tao *et al.* [27] have used Hidden Markov Models with Dirichlet and sparsity priors respectively for action and gesture recognition. Niebles *et al.* [15] have used Probabilistic Latent Semantic analysis for learning human actions in an unsupervised setting. Wang *et al.* [21] have used HMMs to recognize actions performed by gymnasts in multi-view settings. 3D exemplar based HMMs are used by Weinland *et al.* [22] to recognize actions in arbitrary views of camera networks. All these generative methods tend to perform poorly in the presence of actions that are not pre-defined during training. Further, in real surveillance settings, transition probabilities are very hard to estimate as different people being tracked might have different goals and destinations.

Some recent works on joint segmentation and recognition of human actions have addressed a problem related to ours. Shi *et al.* [28] introduce a Semi Markov model framework to capture the temporal structure of actions in video sequences. They present a structured learning framework to learn the parameters of their graphical model, and a Viterbi-style inference algorithm that works real-time on long video sequences. Hoai *et al.* [29] employ a similar approach with a multi-class SVM for learning model parameters and a slightly different cost function for inference. While these methods can be extended to our wireless surveillance camera setting, they still require transmission of every frame captured by the camera sensor, which can be challenging in resource constrained settings. Further, since their framework is purely temporal, multi-view information across pairs of cameras cannot be easily utilized.

III. SYSTEM PIPELINE

Our system consists of multiple smart cameras communicating wirelessly with a central processing station as shown in Fig. 2. In our current framework, we assume that all the cameras connected to the base station are viewing the same scene from different vantage points, and that images from all of them share some amount of overlap. We also assume that the cameras are time synchronized, and that minimal extrinsic calibration is available between pairs of cameras. The details of our calibration and the spatiotemporal multi-view recognition algorithm of the central processing station are presented in the section that follows.

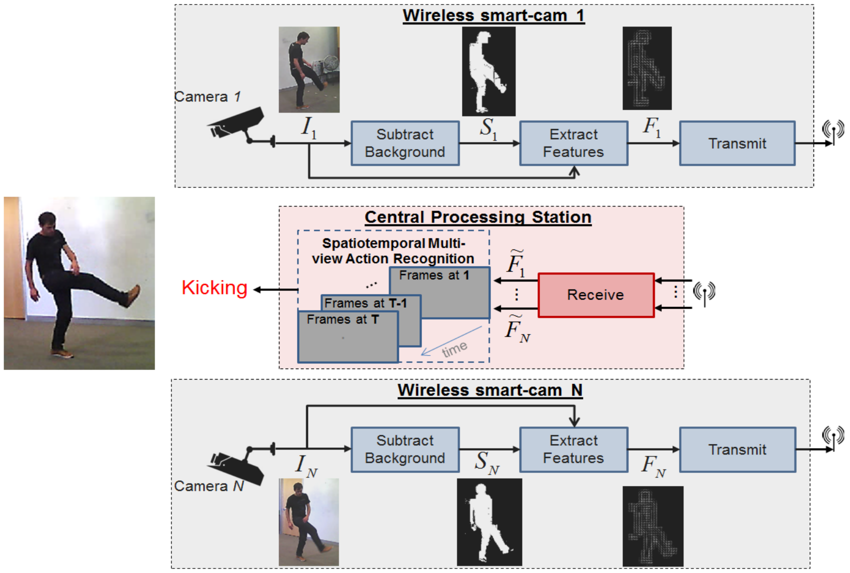


Fig. 2. System pipeline. See text for details.

Each wireless smart camera is capable of separating foreground objects, extracting gradient features for each object, and transmitting these features to the central processing station. We use an off-the-shelf background subtraction algorithm [30] to extract foreground object silhouettes in each camera. The size of the bounding box around the object can be used to determine the scale of the object. Nonetheless, it is impossible to uniquely disambiguate the size of the object and its distance from the camera as this information is lost during perspective projection. For our activity recognition application, however, we argue that the scale encodes sufficient information, as the distance covered by a smaller person translating closer to a camera can be comparable to that covered by a larger person further away.

Some activity recognition papers use features computed on silhouettes as inputs to their algorithms [1], [28], [29]. However, due to self occlusion, discriminative details within the object boundary can be lost when using silhouettes. For instance, this can be seen in the silhouette extracted by the first camera in Fig. 2, where the arm of the person is fully encapsulated by the boundary around his silhouette. In order to utilize maximum information available in each image, we extract HOG descriptors [31] within the bounding box around the foreground object. Specifically, we use the silhouette to extract the foreground pixels within the bounding box, and apply a grid to the foreground region. The number of rows and columns of the grid are kept constant for all foreground regions. In our experiments we have used 5 × 5 grids for each foreground object. HOG descriptors are extracted within each grid and vectorized to represent the appearance of the foreground object. These appearance descriptors along with the bounding box coordinates are subsequently transmitted wirelessly to the central processing station.

System Analysis: The processing performed on board each wireless camera is largely stabilized, making it amenable to deployment with minimal requirements for firmware updates. Even in situations where the number of action classes or the entire action recognition framework changes, the basic operations performed on the smart camera can remain unaltered. The primary purpose of feature extraction on-board the camera is to minimize the data transmitted. In the current framework, only 800 bytes (5 × 5 × 32) of data is transmitted for every object detected. Further, we can leverage the sparsity of the feature descriptors and utilize a compression scheme similar to that presented in [29]. In comparison, H.264 video compression provides an average bit rate of 64K bytes per image for 640 × 480 color images [32] (roughly 2 orders of magnitude higher) with more complex processing performed on the imaging platform. Although this analysis assumes that only one object is detected in any frame, this is still a conservative estimate of transmission savings, as there are going to be situations where no people are present or moving in front of the cameras. This leads us to believe that our system is an attractive wireless alternative for automated surveillance applications.

IV. SPATIOTEMPORAL MULTI-VIEW ACTION RECOGNITION

A. DEFORMABLE KEYFRAME MODEL (DKM)

Single-view Model: Our keyframe based action detection framework is closely related to the DPM model commonly used for object detection [13]. We represent a video sequence as D and any particular action as an N node directed graph, $G = (V, E)$. The nodes in the graph, V, correspond to keyframes. Any given node $i \in \{1...N\}$ has an anchor position $p_i = (x_i, y_i, t_i)$, where (x_i, y_i) represent the pixel location of the center of the bounding box around an object in the image, and t_i represents the frame number in the video

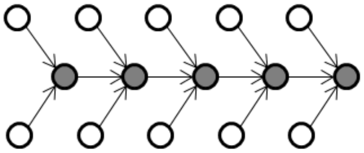


Fig. 3. Multi-view graphical model that represents any particular action. Filled nodes represent keyframes in reference camera, and empty nodes represent keyframes in other two cameras.

sequence. Edges in the graph, E, specify which pairs of keyframes are constrained to have relations. The framework is very general and edges in the graph need not be successive. For instance, jump edges can be used to connect nodes corresponding to repetitive keyframes in cyclical actions.

The score, $\boldsymbol{S}$, associated with a particular action model and keyframe-labeling can be written as [13]:

$$\boldsymbol{S}(p|D,\mathbf{w}) = \sum_{i\in V} \langle w_i, \phi^{app}(D,p_i)\rangle + \sum_{i,j\in E} \left\langle w_{ij}, \phi^{def}(p_i,p_j)\right\rangle \quad (1)$$

where, $\phi^{app}(D,p_i)$ is the HOG appearance descriptor of the object detected at frame t_i (see section III for details), and $\phi^{def}(p_i,p_j)$ models the deformation between pairs of frames. In the single-view setting, the deformation is given by $\phi^{def}(p_i,p_j) = [dx, dx^2, dy, dy^2, dt, dt^2]$, where $dx = x_i - x_j$, $dy = y_i - y_j$ and $dt = t_i - t_j$. For the right match, the keyframe appearance template, w_i, will have a maximum inner product response with the appearance descriptor at location p_i in the video D. The deformation weight w_{ij} models the Mahalanobis distance between the pairs of keyframes in the model, and its parameters need to be learned during training. We address the learning of appearance and deformation weights in section IV-C.

Multi-view Model: We extend our single-view keyframe model framework to incorporate multiple cameras capturing the same scene. In this case, we choose one reference camera, and all other cameras are connected to it, thereby yielding a directed graphical model as shown in Fig. 3. In this paper, we do not model the spatiotemporal relationship between nodes corresponding to each non-reference camera. This would, however, be a straightforward extension as the introduction of spatiotemporal edges in non-reference cameras would not introduce any cycles in our directed graph.

The score function for the multi-view setting remains the same as that in the single-view model in eqn. 1. The deformation function between frames captured at the same time instance from two views, however, needs to account for the epipolar constraints between the views. In most surveillance settings, it is common to have significant overlapping views of the ground plane on which people move about. We use the homography induced by this ground plane to enforce pairwise constraints between views. Specifically, we compute the ground plane homography, H_l^r, between any camera l in the network that shares scene overlap with the reference camera r. Since the homography is a linear transform that maps pixels in one view of a plane to another, it can be used to determine the distance between object detections across views. Further, since the people in the cameras' fields-of-view are in contact with the ground plane at most times, the centre of the line connecting the bottom corners of the bounding box detection around them can be used as a proxy for their 3D location in the scene. Although this assumption can be easily violated when people are closer to the camera, in surveillance applications, that is unlikely as cameras are intentionally positioned far from reach.

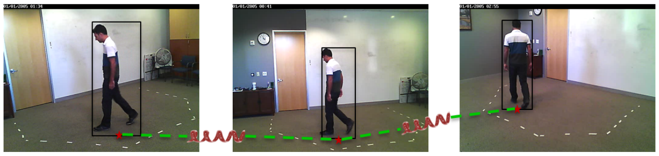


Fig. 4. Deformation constraints between reference view in the middle and two other cameras viewing the scene. Deformation cost modeled as spring connecting center of line between bottom corners of each bounding box, as they lie on the ground plane. All three images are captured at same time instant from three vantage points.

Given the homogeneous coordinates of a pixel, $f^l = (x^l, y^l, 1)^T$ on the ground plane in the l^{th} camera view, its position in the reference camera can be estimated as $\tilde{f^r} = H_l^r f^l$. The deformation function for the two views can then be given by $\phi^{def}(f_i^l, f_i^r) = [dx, dx^2, dy, dy^2]$, where, $[dx, dy] = (f^r - H_l^r f^l)^T$. Fig. 4 shows an example with deformation constraints between a reference camera and two other cameras on either side of it.

B. KEYFRAME SELECTION

Analogous to parts in DPMs for object detection, our deformable keyframe models use appearance templates corresponding to keyframes as node potentials. Thus, it is important for the same set of keyframes to be present in all samples of a given action, at least while learning the model parameters. We adopt the definition proposed by Bourdev & Malik [34] to define keyframes in our setting: Given a set of M training video samples $\{D_1, ..., D_M\}$ of any action, the goal of keyframe selection is to find a subset of N representative frames in each sample such that, similarly selected representative frames of actions are tightly clustered in 3D body configuration space. This process of supervised clustering of keyframes can easily be done using motion capture, where different subjects perform actions while simultaneously being recorded by a motion capture system to capture their 3D pose and a camera network to capture their appearance in multiple views. Using this method, we can automatically obtain ground-truth keyframe labelings $\{p_1, ..., p_M\}$, for all the video samples. In our experiments, however, we have manually annotated the keyframes as we were unable to find any publicly available complex action datasets captured using motion capture and traditional cameras.

C. LEARNING

We employ a structured learning [33] approach to train the parameters of our model for each action, $c \in \{1...C\}$, where

C is the total number of actions in our database. Given a set of M positive training examples $\{D_q\}$ $(q = 1, 2, ...M)$ for any action c, we are interested in learning the appearance (w_i^c's) and deformation parameters (w_{ij}^c's) given in eqn. 1 that would produce the correct labeling $\{p_q\}$. Since our scoring function (1) is linear in these parameters, it can be rewritten as

$$\boldsymbol{S}(p_q|D_q, \mathbf{w^c}) = \langle \mathbf{w^c}, \Phi(D_q, p_q) \rangle \,, \tag{2}$$

where, $\mathbf{w^c}$ is a vector that includes all the appearance and deformation parameters that need to be learned, and $\Phi(D_q, p_q)$ is the corresponding appearance and deformation energy due to a certain labeling p_q.

In our setting, we are also interested in discerning different actions from each other, so we need to learn models that can jointly detect and discriminate between different actions. We adopt a one-vs-all learning policy for each action, and learn the model parameters that can jointly detect and recognize any particular action given hard negative examples of other actions in the database.

We adopt the structural SVM framework of [33] and write our learning objective as,

$$\underset{\mathbf{w^c}, \{\xi_q\}, \{\eta_{q,q'}\} \geq 0}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w^c}\|^2 + \lambda_1 \sum_q \xi_q + \lambda_2 \sum_{q,q'} \eta_{q,q'} \tag{3}$$
$$\text{s.t.} \quad \forall q, \langle \mathbf{w^c}, \Phi(D_q, p_q) - \Phi(D_q, \tilde{p}) \rangle \geq \Delta(p_q, \tilde{p}) - \xi_q$$
$$\forall q, q', \langle \mathbf{w^c}, \Phi(D_q, p_q) - \Phi(D_{q'}, p_{q'}) \rangle \geq \Delta(p_q, p_{q'}) - \eta_{q,q'},$$

where, λ_1, λ_2 are user defined scaling parameters to minimize slack values in the optimization.

The first constraint in eqn. 3 implies that for the same class, any keyframe labeling $\tilde{p}$, other than the ground-truth labeling p_q, for the q^{th} data sample, needs to be penalized according to the loss function $\Delta(p_q, \tilde{p})$. The non-negative slack term ξ_q provides an extra level of robustness to account for some violation of the constraint. The second constraint implies that given any ground truth labeling p_q for the q^{th} sample of a particular action, any ground truth labeling $p_{q'}$ of the $\{q'\}^{th}$ sample of any other action sequence in the database will produce a lower score after filtering through another violation accommodating hinge-loss $\eta_{q,q'}$.

The objective of the loss function $\Delta(p_q, \tilde{p})$ is to reflect how well a particular labeling hypothesis $\tilde{p}$, matches the true labeling p_q. We have adopted a simple binary loss function with $\Delta(p_q, \tilde{p}) = 1$ if $\tilde{p} = p_q$, and $\Delta(p_q, \tilde{p}) = 0$ otherwise. We employ the cutting-plane algorithm described in [33] to solve our quadratic program (3).

Model bias: The learning procedure, however, does not produce weights of the same magnitude for each action class. Thus, the modeling score for each action class has an associated bias b^c, that needs to be estimated and subtracted from the final score during inference. In order to determine the bias for each action class, we apply the learned model for that action class to the training data samples and take the median of these scores as the bias, i.e.,

$$b^c = \text{median}\{\boldsymbol{S}(p_1|D_1, \mathbf{w^c}), ..., \boldsymbol{S}(p_M|D_M, \mathbf{w^c})\}. \tag{4}$$

D. INFERENCE

In our detection and recognition setting, given a query video sequence D, the inference problem is to find the best action c^*, and correspond labeling p^*, that maximizes the modeling score:

$$\{c^*, p^*\} = \underset{p, c \in \{1...C\}}{\operatorname{argmax}} \quad (\boldsymbol{S}(p|D, \mathbf{w^c}) - b^c). \tag{5}$$

Since our directed graph is a chain in the single-view and a tree in the multi-view scenarios, inference can efficiently be done via dynamic programming [13].

V. EXPERIMENTS

We evaluate our Deformable Keyframe Model (DKM) framework in three scenarios. In the first scenario, we test the discriminating capabilities of our model by performing whole-clip recognition. In the second scenario, we test the joint detection and recognition capabilities of our model in a controlled setting by synthesizing a complex action sequence by concatenating simple action video-clips. In the final scenario, we test our algorithm for joint detection and recognition of actions on a novel complex data set consisting of continuous actions performed by different subjects while being recorded by cameras placed at multiple vantage points.

A. Weizmann Simple Actions

The Weizmann dataset [1] is a popular dataset for validating action recognition algorithms, as it consists of short video clips captured under controlled conditions. It is composed of 10 action clips performed by 9 actors, all of whom remain un-occluded and at the same distance from the camera's focal plane. The background model of the scene is available, using which foreground silhouettes of the actors have been extracted for every frame of the video.

Keyframe selection: Automatic keyframe selection for the Weizmann dataset is challenging as there is no motion capture data available. Hence, we have manually selected keyframes for each action. A set of 5 keyframes have been manually selected for each action performed by every individual.

We follow the same testing procedure proposed by [1] for the dataset. As presented in section III, we extract HOG appearance descriptors for the foreground region in each frame, along with the coordinates of the bounding box. We use these features within our DKM framework and pick the action class that maximizes the modeling score (see eqn.5). Our DKM framework achieves **100%** recognition accuracy. This is comparable to the perfect recognition recognition reported by the authors of the dataset, and others who have also validated their methods after adopting the same testing procedure [1], [23], [35].

B. Weizmann Complex Actions

In order to validate the joint detection and recognition capability of our DKM, we synthesize complex actions by concatenating all the 10 actions performed by each of the subjects in the Weizmann dataset, thereby yielding 9 videos.

The order in which the actions are composed is chosen at random for each subject. The frame level features are still extracted using the method outlined in section III.

Our training methodology is similar to that employed by Hoai *et al.* [29]. We adopt a leave-one-out evaluation strategy: training on 8 sequences and testing on the left-out sequence. The models and associated bias for each action are learned using the procedure outlined in section IV-C.

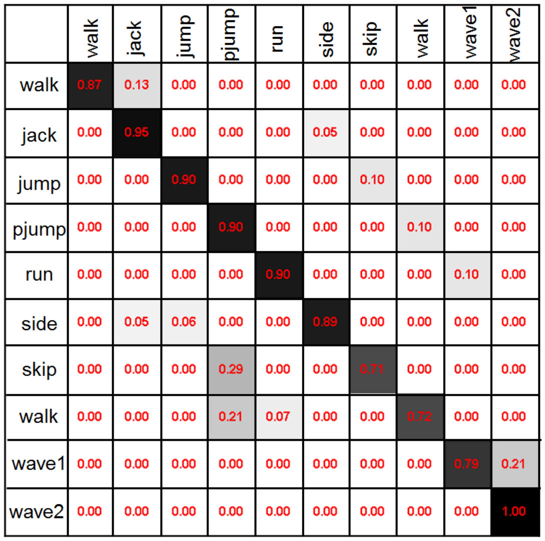
Our evaluation metric is also inspired by theirs. Specifically, we evaluate each of our models on a query synthesized video. Multiple detections are found by each action specific DKM, and all the overlapping detections with the highest score per class are retained. The temporal union of these detections provides a class specific segmentation of the query video sequence. At this point, the overall frame-level accuracy against the ground truth labels is calculated as the ratio of number of agreements over the total number of frames. It is important to note that this segmentation based metric is designed for joint segmentation and recognition algorithms such as [29] and it serves as a harder baseline evaluation metric for our detection and recognition algorithm.

Fig. 5 shows the confusion matrix for the joint segmentation and recognition of the 10 actions using the 9 complex video sequences. The average accuracy of our method is **86.28%**. Hoai *et al.* [29] report an average accuracy of 87.7%, which is just slightly higher than our accuracy. However, their focus is on joint segmentation and recognition, and their algorithm yields a label for every frame in the query video. In our detection based framework, there is no guarantee that all frames will be assigned a class label, as evidenced by the white regions in our qualitative segmentation results shown in Fig. 6. This leads us to believe that our method will perform well even in the presence of previously unseen action classes, but we have not yet tested this hypothesis.

C. Bosch Multi-view Complex Actions (BMCA) Dataset

In the literature, there exist several public datasets for activity recognition, but continuous action datasets for action detection are limited. Further, to the best of our knowledge, there are no publicly available multi-view action detection datasets with subjects performing several actions continuously. To aid in peer evaluation of distributed activity detection and recognition, we have constructed a multi-view video dataset called the BMCA dataset which will be available online.

The BMCA dataset consists of 11 actions performed back-to-back by 20 subjects. Each subject performs three to four trials of each action while facing different directions, and at different locations within the capture area. The subjects are continuously recorded using 4 time synchronized cameras arranged in the configuration shown in Fig. 7. The cameras capture color video at a frame rate of 10 Hz, thereby yielding 4 long video clips of roughly 12-15 minutes each. The location within the capture area and the direction to face while performing an action are decided by the subjects themselves. However, they are all instructed to maintain angular orientations of roughly $\{0^o, 90^o, 180^o$ and $270^o\}$ relative to the reference camera. In our setting, camera-2 is chosen as the reference view.



	walk	jack	jump	pjump	run	side	skip	walk	wave1	wave2
walk	0.87	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
jack	0.00	0.95	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00
jump	0.00	0.00	0.90	0.00	0.00	0.00	0.10	0.00	0.00	0.00
pjump	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.10	0.00	0.00
run	0.00	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.10	0.00
side	0.00	0.05	0.06	0.00	0.00	0.89	0.00	0.00	0.00	0.00
skip	0.00	0.00	0.00	0.29	0.00	0.00	0.71	0.00	0.00	0.00
walk	0.00	0.00	0.00	0.21	0.07	0.00	0.00	0.72	0.00	0.00
wave1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.79	0.21
wave2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Fig. 5. DKM performance on the Weizmann complex dataset. Confusion matrix shows joint segmentation and recognition accuracy of 10 actions at frame level. Off-diagonal numbers show frame misclassification rates. Average accuracy of 86.28% achieved on dataset.

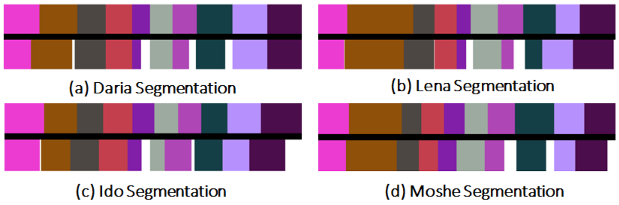


Fig. 6. Qualitative segmentation of four complex videos. For each segmentation, top row shows true class labels and bottom row shows estimated labels. Note the existence of white regions in the estimated labels at frames where no reliable detections were found. As expected, majority of the error occurs at segment boundaries. Image best viewed in color.

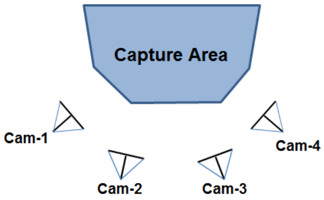
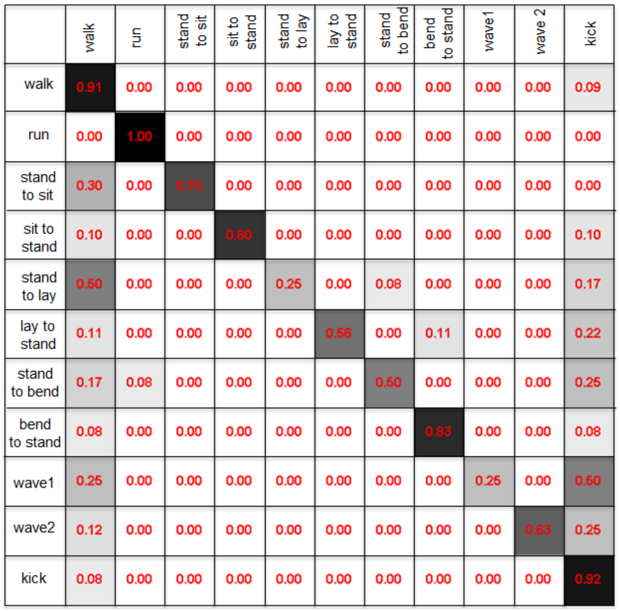


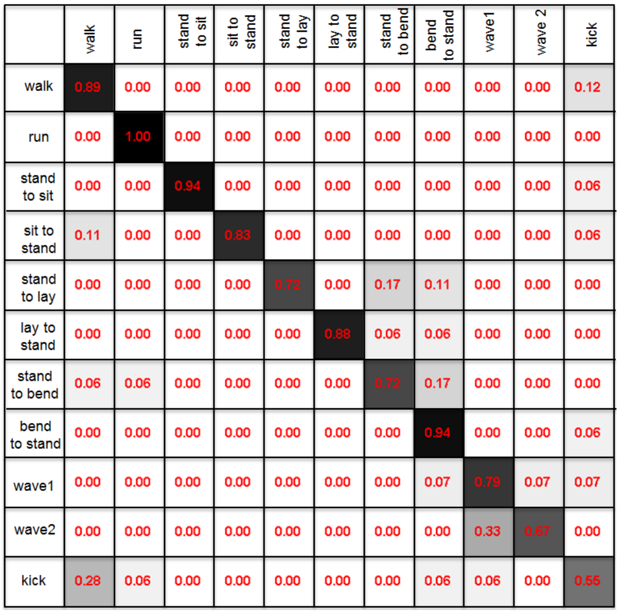
Fig. 7. Configuration of cameras used to create BMCA dataset. Cameras capture color video at 10HZ, and are time synchronized.

Keyframe selection: The background subtraction scheme presented in section III has been used to obtain bounding boxes around people in the dataset. We have manually annotated the dataset by providing the start and end times of each action and its associated action class labels. The keyframes for each action have also been manually selected. Fig. 10 shows the keyframe annotations of 3 subjects performing 3



	walk	run	stand to sit	sit to stand	stand to lay	lay to stand	stand to bend	bend to stand	wave1	wave 2	kick
walk	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09
run	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
stand to sit	0.30	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sit to stand	0.10	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.10
stand to lay	0.50	0.00	0.00	0.00	0.25	0.00	0.08	0.00	0.00	0.00	0.17
lay to stand	0.11	0.00	0.00	0.00	0.00	0.56	0.00	0.11	0.00	0.00	0.22
stand to bend	0.17	0.08	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.25
bend to stand	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.00	0.08
wave1	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.50
wave2	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.25
kick	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92

Fig. 8. Confusion matrix for single-view joint segmentation and recognition on BMCA dataset. Average accuracy is **66.74 %**.



	walk	run	stand to sit	sit to stand	stand to lay	lay to stand	stand to bend	bend to stand	wave1	wave 2	kick
walk	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12
run	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
stand to sit	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
sit to stand	0.11	0.00	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.06
stand to lay	0.00	0.00	0.00	0.00	0.72	0.00	0.17	0.11	0.00	0.00	0.00
lay to stand	0.00	0.00	0.00	0.00	0.00	0.88	0.06	0.06	0.00	0.00	0.00
stand to bend	0.06	0.06	0.00	0.00	0.00	0.00	0.72	0.17	0.00	0.00	0.00
bend to stand	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.00	0.00	0.06
wave1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.79	0.07	0.07
wave2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.67	0.00
kick	0.28	0.06	0.00	0.00	0.00	0.00	0.00	0.06	0.06	0.00	0.55

Fig. 9. Confusion matrix for multi-view joint segmentation and recognition on BMCA dataset. Average accuracy is **81.28 %**.

different actions.

Training: We have partitioned our dataset of 20 people into 5 training and 15 test sets. The 5 training sets include 11 actions performed at 4 angular orientations. Thus, we have learned 44 DKMs using the framework presented in section IV-C. We learn separate models for the single-view and multi-view experiments.

Testing: In order to validate our framework, we test our trained models on the 15 remaining test sets. As in the experiment for the Weizmann complex dataset, we employ the joint segmentation and recognition evaluation strategy. We only modify the segment labeling slightly so that all the detections corresponding to different orientations of the same action class are assigned the same label. We first evaluate the single-view DKM algorithm on the training videos captured by the reference camera. The results of our method is presented in the confusion matrix of Fig. 8. We obtain an average segmentation accuracy of **66.74%**. Although this accuracy is lower than that obtained on the Weizmann complex dataset, the BMCA dataset is a lot more challenging as it is longer and has more complex actions. In fact some of the actions are duals to others in the set; these include the ”stand to sit”, ”sit to stand”, ”stand to lay”, ”lay to stand”, ”stand to bend” and ”bend to stand” classes. Without the spatiotemporal constraints, it would be very hard to discriminate between these action duals. With the spatiotemporal constraints, however, there is no misclassification between action duals, as evidenced by the zero off-diagonal values in the confusion matrix.

Next, we evaluate the multi-view DKM algorithm on the same test sets by including the remaining camera views. The multi-view DKMs are evaluated on the test set using the same joint segmentation and recognition strategy used in the single-view case. The resulting confusion matrix is presented in Fig. 9. It is clear that the addition of multiple views significantly improves the action detection and recognition performance. Specifically, an average accuracy of **81.28 %** is achieved which represents a **14.54 %** increase in accuracy. We believe that incorporating more overlapping views around the capture volume can improve the accuracy even further, but have not yet tested this hypothesis.

VI. CONCLUSIONS

We have presented a framework for the joint detection and recognition of human actions on long, complex video sequences. Our method is well suited for situations where the camera sensors and the base station are connected only by a band-limited communication channel. We have made three primary contributions in this paper. The first includes a framework for feature extraction on a wireless smart camera that can minimize its power and bandwidth requirements. Our second contribution is the adaptation of the DPM object detection framework for single-view and multi-view action detection in continuous video, and our final contribution is a novel scheme to learn the bias and parameters of our deformable keyframe models. We have experimentally validated our algorithm on a publicly available dataset, and have demonstrated the competitiveness of our approach against state-of-the-art methods. Finally, we have introduced a novel multi-view continuous action data set called the Bosch Multiview Complex Action dataset and extensively validated the performance of our system using this dataset.

Our investigations have led us to several intriguing open problems for future investigation. First, our framework for evaluating multiple deformable keyframe models concurrently on the data by subtracting the model bias may deteriorate when more action classes are introduced. Perhaps

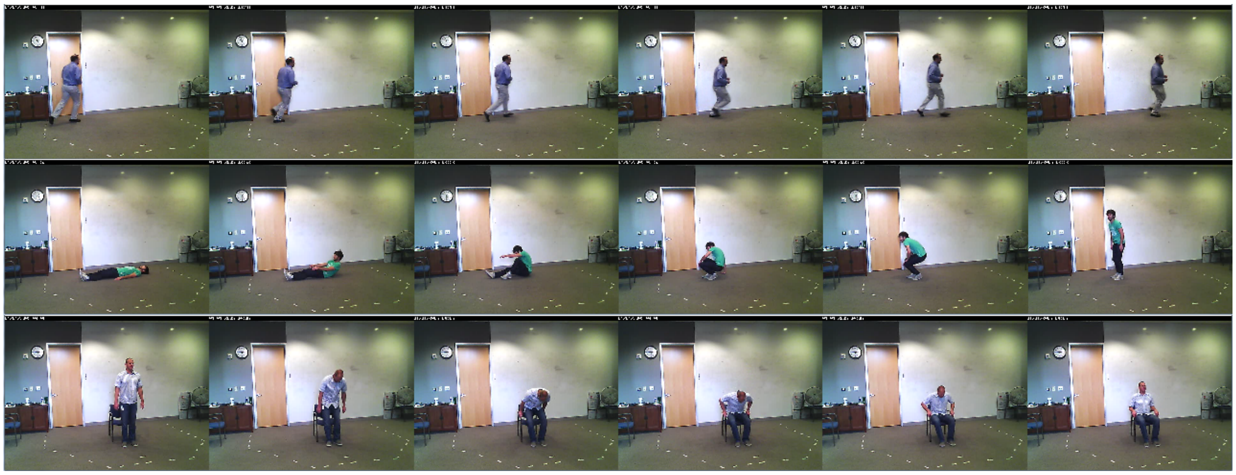


Fig. 10. Keyframes for a few actions in the BMCA dataset. The first row shows the 6 keyframes corresponding to the action ”run”. The second and third rows show the chosen keyframes for the actions ”lie-to-stand” and ”stand-to-sit” respectively.

a detection strategy similar to the generalized Hough transforms adopted by [34] could make the detectors more robust. Second, our best detection and recognition performance on our dataset is 82%. In order to successfully deploy such a system in real-world surveillance applications, the recognition rates have to be improved dramatically (e.g. $>$ 99%) with minimal false positives. Finally, robust techniques must be studied in order to deal with real world situations such as poor lighting, and occlusions in the scene. In such settings, a smart sensor selection scheme might have to be explored.

REFERENCES

[1] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, ”Actions as space-time shapes”, In PAMI, 2007.

[2] I. Laptev and P. Perez, ”Retrieving actions in movies”, In Proc. of ICCV, 2007.

[3] J. Liu, J. Luo and M. Shah, ”Recognizing realistic actions from videos ’in the wild’”, In Proc. of CVPR, 2009.

[4] M. Rodriguez, J. Ahmed and M. Shah, ”Action MACH: A spatiotemporal maximum average correlation height filter for action recognition”, In Proc. of ICPR, 2004.

[5] D. Weinland, R. Ronfard and E. Boyer, ”Free viewpoint action recognition using motion history volumes”, In Proc. of CVIU, 2006.

[6] C. Wu, A. Khalili and H. Aghajan, ”Multiview Activity Recognition in Smart Homes with Spatio-Temporal Features”, In Proc. of ICDSC, 2010.

[7] Y. Ke, R. Sukthankar and M. Hebert, ”Volumetric features for video event detection”, In IJCV, 2010.

[8] C. Schuldt, I. Laptev and B. Caputo, ”Recognizing human actions: A local SVM approach”, In Proc ICPR, 2004.

[9] P. Turaga, R. Chellappa and V. Subrahmanian, ”Machine recognition of human activities: a survey”, In Circuits and Systems for Video Technology, 2008.

[10] I. Laptev, M. Marsza, C. Schmidt and B. Rozenfeld, ”Learning realistic human actions from movies”, In Proc. CVPR 2008.

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

[12] S. Satkin and M. Hebert, ”Modeling the temporal extent of actions”, In Proc. of ECCV, 2010.

[13] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, ”Object detection with discriminatively trained part based models”, In PAMI, 2010.

[14] Y. Ma, S. Soatto, J. Kosecka and S. S. Sastry, ”An invitation to 3D vision: from images to geometric models”, Springer Verlag, 2003.

[15] J. C. Niebles, H. Wang and L. Fei-Fei, ”Unsupervised learning of human action categories using spatial-temporal words”, In IJCV, 2008.

[16] H. Wang, A. Klaser, C. Schmidt and L. Cheng-Lin, ”Action recognition by dense trajectories”, In Proc. of CVPR, 2011.

[17] S. Ali and M. Shah, ”Human action recognition in videos using kinematic features and multiple instance learning”, In PAMI, 2010.

[18] Y. Yang and D. Ramanan, ”Articulated pose estimation with flexible mixtures-of-parts”, In Proc. CVPR 2011.

[19] N. Naikal, D. Singaraju and S. S. Sastry, ”Using models of objects with deformable parts for joint segmentation and categorization of objects”, In Proc. of ACCV, 2012.

[20] N. Naikal, A. Yang and S. S. Sastry, ”Towards and efficient distributed object recognition system in wireless smart camera networks”, In Proc. of Fusion, 2010.

[21] Y. Wang, K. Huang, and T. Tan, ”Multi-view gymnast activity recognition with fused HMMs”, in Proc. of ACCV, 2007.

[22] D. Weinland, E. Boyer, and R. Ronfard, ”Action recognition from arbitrary views using 3D exemplars”, In Proc. of ICCV, 2007.

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

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

[25] C. Scminchisescu, A. Kanaujia, Z. Li and D. Metaxas, ”Conditional models for contextual human motion recognition”, In Proc. Of ICCV 2005.

[26] E. B. Fox, E. B. Sudderth, M. I. Jordan and A. S. Willsky, ”Nonparametric bayesian learning of switching linear dynamical systems”, In Proc. NIPS 2009.

[27] L. Tao, E. Elhamifar, S. Khudanpur, G. Hager and R. Vidal, ”Sparse Hidden Markov Models for Surgical Gesture Classification and Skill Evaluation”, In Proc of IPCAI, 2012.

[28] Q. Shi, L, Wang, L. Cheng and A. Smola, ”Discriminative human action segmentation and recognition using semi-markov model”, In Proc. of CVPR, 2008.

[29] M. Hoai, Z-Z. Lan, F.D.Torre, ”Joint segmentation and classification of human actions in video”, In Proc. of CVPR 2011.

[30] Z. Zivkovic, ”Improved adaptive Gaussian mixture model for background subtraction”, In Prof. of ICPR, 2004.

[31] N. Dalal and B. Triggs, ”Histograms of oriented gradients for human detection”, In Proc. of CVPR, 2005.

[32] T. Wiegrand, G. J. Sullivan, G. Bjontegaard and A. Luthra, ”Overview of the H.264/AVC video coding standard”, In CSVT, 2011.

[33] I. Tsochantaridis, T. Joachims, T. Hofmann and Y. Altun, ”Large margin methods for structured and interdependent output variables”, In JMLR, 2005.

[34] L. Bourdev and J. Malik, ”Poselets: Body part detectors trained using 3D human pose annotations”, In Proc. of ICCV, 2009.

[35] A. Fathi and G. Mori, ”Action recognition by learning mid-level motion features”, In Proc. of CVPR, 2008.