Recognizing Human Actions with Outlier Frames by Observation Filtering and Completion

SHIH-YAO LIN, National Taiwan University
YEN-YU LIN and CHU-SONG CHEN, Academia Sinica
YI-PING HUNG, Tainan National University of the Arts

This article addresses the problem of recognizing *partially observed* human actions. Videos of actions acquired in the real world often contain corrupt frames caused by various factors. These frames may appear irregularly, and make the actions only partially observed. They change the appearance of actions and degrade the performance of pretrained recognition systems. In this article, we propose an approach to address the corrupt-frame problem without knowing their locations and durations in advance. The proposed approach includes two key components: *outlier filtering* and *observation completion*. The former identifies and filters out unobserved frames, and the latter fills up the filtered parts by retrieving coherent alternatives from training data. *Hidden Conditional Random Fields* (HCRFs) are then used to recognize the filtered and completed actions. Our approach has been evaluated on three datasets, which contain both fully observed actions and partially observed actions with either real or synthetic corrupt frames. The experimental results show that our approach performs favorably against the other state-of-the-art methods, especially when corrupt frames are present.

CCS Concepts: • **Computing methodologies → Activity recognition and understanding;**

Additional Key Words and Phrases: Human action recognition, outlier filtering, observation completion, early prediction, gap filling, and conditional random fields

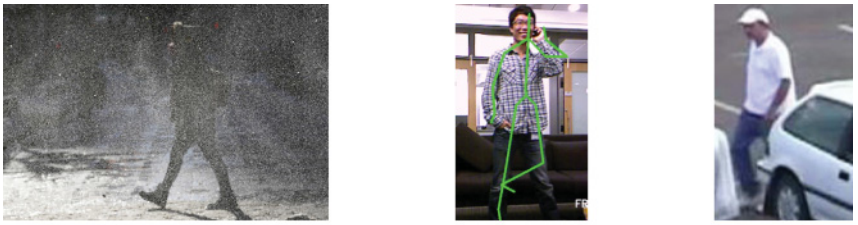
ACM Reference Format:
Shih-Yao Lin, Yen-Yu Lin, Chu-Song Chen, and Yi-Ping Hung. 2017. Recognizing human actions with outlier frames by observation filtering and completion. ACM Trans. Multimedia Comput. Commun. Appl. 13, 3, Article 28 (June 2017), 23 pages.
DOI: http://dx.doi.org/10.1145/3089250

1. INTRODUCTION

Human action recognition is essential to varied applications such as surveillance, health care, and human-computer interaction. Many existing efforts such as Andre [2013], Li et al. [2007], Lin et al. [2014, 2017], Tang et al. [2015], Vemulapalli et al. [2014], Zhang et al. [2011, 2015], and Zhao et al. [2014] focus on recognizing human actions in *fully observed* videos. Unfortunately, the assumption of full observation may

This work was supported in part by the Ministry of Science and Technology of the Republic of China under grants 104-2628-E-001-001-MY2, 105-2221-E-001-030-MY2, 105-2218-E-001-006, and 104-2221-E-002-050-MY3.
Authors' addresses: S.-Y. Lin is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan, 4005 Miranda Ave, Palo Alto, CA 94304, USA; email: shihyaolin@iis.sinica.edu.tw; Y.-Y. Lin is with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan, No 128, Section 2, Academia Road, Nangang, Taipei 11529, Taiwan; email: yylin@citi.sinica.edu.tw; C.-S. Chen is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, No 128, Section 2, Academia Road, Nangang, Taipei 11529, Taiwan; email: song@iis.sinica.edu.tw; Y.-P. Hung is with Tainan National University of the Arts, Tainan 720, Taiwan, 66, Daci Village, Guantian District, Tainan City, 72045 Taiwan; email: hung@csie.ntu.edu.tw.

DOI: http://dx.doi.org/10.1145/3089250



(a) (b) (c)

Fig. 1. Outlier frames caused by (a) noisy video signals, (b) skeleton inference errors, and (c) partial occlusions.

not be always held due to various factors including hardware limitations (e.g., signal loss or noise [Oshin et al. 2011]), software limitations (e.g., skeleton estimation errors [Chaaraoui et al. 2013]), and complex environments (e.g., partial occlusions [Ayvaci et al. 2012; Wang et al. 2009]). We consider video frames where the previously mentioned situations happen as *outliers*, which make the actions partially *unobserved*. Note that by unobserved frames, we do not mean that these frames are unobserved. Instead, we mean that the actions in these frames are (partially) unobserved. Figure 1 shows some examples of outlier frames. In this study, we present an approach for recognizing human actions with outlier frames.

Since outlier frames are inconsistent with the training data, they often cause significant performance degradation. Several studies [Shu et al. 2012; Wang et al. 2012; Shen et al. 2012; Weinland et al. 2010; Cao et al. 2013] have attempted to recognize actions that are not fully observed. However, they handle outlier frames by using extra domain knowledge or assume that the locations of outlier frames are known in advance. Thus, they may be less practical in real-world applications.

We refer to the task as *Partially Observed Action Recognition* (POAR). Some difficulties have arisen as follows. Firstly, we need to distinguish outlier frames in the video. Secondly, the observed parts may carry insufficient evidence and result in unreliable action predictions. To address these issues, we propose to divide the video of an action into observed and unobserved parts, and replace the unobserved part with a coherent alternative.

Specifically, our approach performs *outlier frame filtering* and *observation completion* to carry out POAR upon *Hidden-state Conditional Random Fields* (HCRFs) [Quattoni et al. 2007]. A video frame of a test action is considered an outlier if it is not similar to any frame in the training set. Although not all such frames are corrupt, they would cause performance degradation since they are not covered by the training data. The process of outlier frame filtering detects and removes these outliers.

After filtering out the outliers, the remaining frames of the action to be recognized are considered as the observed part of that action. This part may be composed of temporally separated segments and carries incomplete information for an accurate prediction. To handle the problems, extra knowledge is borrowed from training data. We treat the observed part as a query to the training data, retrieve training actions similar to the query, and replace the unobserved part with the alternative inferred from the retrieved actions. It follows that the *filtered and completed* actions can serve as inputs to HCRFs. In this way, our approach takes advantage of both information enrichment and temporal coherence regularization to facilitate POAR.

Our approach is evaluated on two benchmarks, the *UT-Interaction* dataset [Ryoo and Aggarwal 2010] and the *ArmGesture* dataset [Quattoni et al. 2007]. In view of the fact that most existing benchmarks are composed of fully observed actions without sufficient outliers, we collected a new dataset containing daily activities of 15 classes, where outlier frames are irregularly and naturally present. The experimental results

on the two benchmarks and the dataset we collected demonstrate that our approach can effectively detect outlier frames, seek the high-quality alternatives to the unobserved parts, and lead to a remarkable performance boost.

The main contribution of this work lies in that we develop a general approach to POAR. It infers the outlier frames and predicts the actions. It makes no assumption about the number, temporal locations, or durations of the outlier frames, and can work with various features such as those extracted from RGB images, depth maps, and skeleton structures. Two key components, outlier frame filtering and observation completion, are tightly coupled in our approach. The former identifies and removes the outlier frames. The latter borrows plausible alternatives to the filtered outliers. It turns out that our approach can robustly predict partially observed actions.

2. RELATED WORK

The literature on human action recognition is extensive. Our review focuses on those that recognize actions in videos because of their relevance to our approach. Owing to the advances in local descriptors, representing an action in a video as a collection of local patches or spatiotemporal cubes, for example [Laptev 2005; Maji et al. 2011], has become popular for its robustness to deformations and partial occlusions. However, the geometric and temporal layouts between local features are ignored.

To address this issue, several researches focus on modeling the spatial geometry and temporal coherence of the local features. Graph models such as *factorial Conditional Random Fields* (CRFs) [Wang and Suter 2007] and the *Hidden Markov Model* (HMM) [Chen and Aggarwal 2011] are used for their expressive power of relationship modeling.

Instead of handcrafted features, using features learned by *Convolutional Neural Networks* (CNNs) [Krizhevsky et al. 2012] has demonstrated its effectiveness in various applications such as object recognition [Li et al. 2017; Shih et al. 2017], human pose estimation [Cao et al. 2016; Chu et al. 2016], tracking [Carneiro and Nascimento 2013], and person reidentification [Xiao et al. 2016]. The success of CNNs also sheds light on video-based computer vision problems. Recent studies of action recognition, for example [Tran et al. 2015; Liu et al. 2016b; Donahue et al. 2015; Liu et al. 2016a; Feichtenhofer et al. 2016; Gan et al. 2015], focus on using deep learning frameworks for learning video representations. Gan et al. [2015] proposed a CNN-based method for high-level video event detection and key-evidence localization. Li et al. [2016] presented a deep architecture for human action recognition, which is capable of incorporating multigranularity information extracted from videos. Simonyan and Zisserman [2014] adopted a two-steam ConvNet framework that learns a spatial subnetwork and a temporal subnetwork jointly, and achieved very promising performance. However, most of these methods concentrate on recognizing fully observed actions. They are typically sensitive to outlier frames and suffer from performance degradation when outlier frames are present.

There have been research efforts on action recognition with *incomplete observation* caused by various issues such as partial occlusions [Shu et al. 2012; Wang et al. 2012], incorrect skeleton estimation [Shen et al. 2012], view changes [Weinland et al. 2010], and missing frames [Cao et al. 2013]. These works use issue-specific domain knowledge and/or assume that the locations of outlier frames are annotated. Thus, they would be impractical in complex environments or without manual annotation.

Early prediction, for example [Chen et al. 2011; Davis and Tyagi 2006; Hoai and De la Torre 2014; Jiang and Saxena 2014; Lan et al. 2014; Li and Fu 2014; Raptis and Sigal 2013; Ryoo 2011; Schindler and Van Gool 2008; Yu et al. 2012, 2015], aims to predict an *ongoing* action by inferring its beginning part. For example, Ryoo [2011] adopted both the *integral* and *dynamic* Bag-of-Words (BoW) to accomplish this task. Davis and Tyagi [2006] proposed an HMM-based probabilistic reliable-inference approach for

rapid human action detection. Chen et al. [2011] presented an approach to cluster human Motion Patterns (MPs) based on gait-trajectories, and predicted long-term future motion via matching current trajectories to classify MPs. Lan et al. [2014] introduced a new action representation, *hierarchical movemes*, to describe human movement at multiple levels of temporal intervals, and developed a maximum-margin learning method for predicting future actions. Lan et al. [2014] presented a max-margin early event detector that identifies the temporal location and duration of a certain action from the video stream. On the other hand, Cao et al. [2013] presented *gap-filling* to handle the unobserved subsequence occurring in an action. They firstly estimated the action likelihood at each observed subsequence and then inferred the global posterior of the whole activity. However, their approach assumes that the periods of unobserved subsequences have been given manually. This assumption makes the approach less practical for real-world applications.

HCRFs [Quattoni et al. 2007] are a powerful and discriminative model. HCRFs employ latent variables to model the hidden structures of observations and are effective for structured data prediction. HCRFs define the joint distribution over the class label and latent state labels conditionally on the observed data. Recent studies such as Zhang and Gong [2010], Song et al. [2012, 2013], and Wang et al. [2014] showed that HCRFs achieve more favorable performance than HMM and CRFs for action recognition.

However, HCRFs cannot work with incomplete observations or data labels. Some studies have attempted to tackle this limitation. Chang et al. [2009] presented an *incremental inference process* to infer HCRFs, and achieved facial expression prediction with incomplete observations. Chang et al. [2014] integrated multiple instance learning into HCRFs for addressing the uncertainty of data labels. Banerjee and Nevatia [2014] proposed a *Pose Filter based HCRFs* (PF-HCRFs) model, which combines a detection filter for finding the key poses in an action video, and a bag-of-words root filter for modeling the detected key poses. PF-HCRFs can infer the temporal locations of the key-poses even when the video frames are not fully observed. Both methods [Banerjee and Nevatia 2014; Chang et al. 2009] show the ability to work with incomplete observation. Instead, we show how to *complete the incomplete observation* and further improve the performance.

In this work, we address the problem of POAR. Regular (meaning fully observed here) action recognition can be considered a special case of POAR, if no unobserved part exists. POAR is reduced to early prediction when an unobserved subsequence occurs at the end of an action. Furthermore, our approach supports the detection of unobserved parts, and makes no assumptions about the temporal locations of outlier frames. Hence, it is general enough to carry out regular action recognition, early prediction, gap-filling, and even to recognize actions with arbitrary occurrence of outlier frames.

3. THE PROPOSED APPROACH

Our approach is described in this section. Consider a training set consisting of N fully observed actions, $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where each action instance is divided into T equal-length temporal segments or frames

$$\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,T}\}, \tag{1}$$

with $y_i \in \mathcal{Y}$ being its class label. $\mathcal{Y}$ is the set of action classes. In the following, a brief review of HCRFs is firstly provided. We then depict our two key components, outlier frame detection and observation completion, as well as their integration into HCRFs.

3.1. Action Recognition with HCRFs

Given an action $\mathbf{x}$, the CRFs [Sutton and McCallum 2007] model the conditional distribution of classes by $P(y|\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of model parameters. The posterior

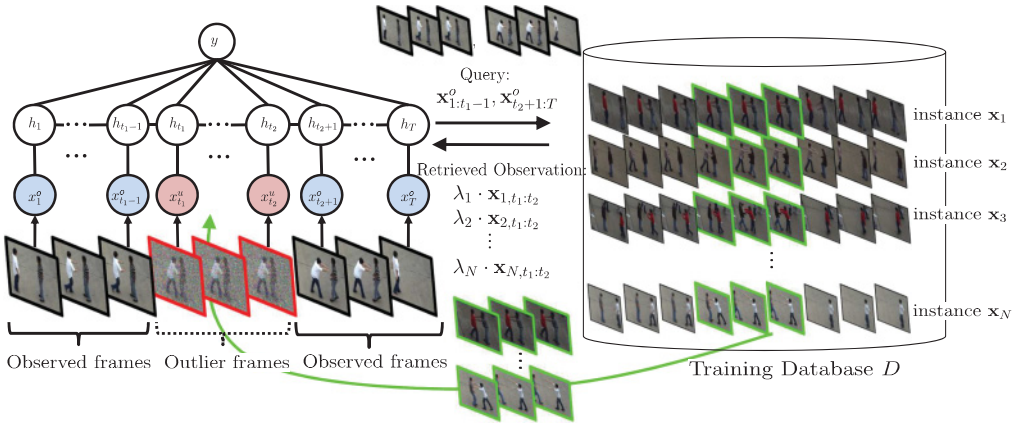


Fig. 2. Our approach for action recognition from partially observed videos. The left-hand side shows an HCRFs model trained by fully observed videos. Given a test video $\mathbf{x}$, our system employs the outlier-frame-filtering method (Section 3.2) to find the outlier frames (enclosed by red boundaries). The input video $\mathbf{x}$ is divided into an observed part $\mathbf{x^o}$ and an unobserved part $\mathbf{x^u}$. We then estimate $\mathbf{x^u}$ by employing $\mathbf{x^o}$ as queries to the training dataset. The estimated unobserved part $\bar{\mathbf{x}}^{\mathbf{u}}$ and the observed part $\mathbf{x^o}$ are then combined. The HCRFs are used to determine the action class from the combined sequence.

distribution $P(y|\mathbf{x}, \boldsymbol{\theta})$ of CRFs is a Gibbs distribution,

$$P(y|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\Psi(y, \mathbf{x}, \boldsymbol{\theta})\right), \tag{2}$$

where Ψ is the *potential function* that will be introduced later. $Z_{\mathbf{x}}$ is the *partition function* making $P(y|\mathbf{x}, \boldsymbol{\theta})$ a probability, that is,

$$Z_{\mathbf{x}} = \sum_{y' \in \mathcal{Y}} \exp\left(\Psi(y', \mathbf{x}, \boldsymbol{\theta})\right). \tag{3}$$

With the training set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the model parameter set $\boldsymbol{\theta}$ can be estimated by *maximizing the log-likelihood*, that is,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i, \boldsymbol{\theta}) - \frac{\|\boldsymbol{\theta}\|^2}{2\rho^2}, \tag{4}$$

where the first term is the log-likelihood of the training data, and the second term is added for regularization. The value of ρ is set to 0.5 in the experiments.

Instead of CRFs, we carry out partially observed recognition on HCRFs [Quattoni et al. 2007]. Specifically for an action instance $\mathbf{x}$, a set of hidden nodes, $\mathbf{h} = \{h_1, h_2, \ldots, h_T\}$, is created, one hidden node for each time stamp. We adopt a chain structure to model the dependence between these nodes as shown in the left-hand side of Figure 2. The conditional probability $P(y|\mathbf{x}, \boldsymbol{\theta})$ in HCRFs is given by

$$P(y|\mathbf{x}, \boldsymbol{\theta}) = \sum_{\mathbf{h}} P(y, \mathbf{h}|\mathbf{x}, \boldsymbol{\theta}) = \frac{\sum_{\mathbf{h}} \exp(\Psi(y, \mathbf{h}, \mathbf{x}, \boldsymbol{\theta}))}{\sum_{y', \mathbf{h}'} \exp(\Psi(y', \mathbf{h}', \mathbf{x}, \boldsymbol{\theta}))}. \tag{5}$$

We follow the original work of HCRFs [Quattoni et al. 2007] and define the potential function as

$$\Psi(y, \mathbf{h}, \mathbf{x}, \boldsymbol{\theta}) = \sum_{j=1}^{T} \phi(x_j) \cdot \theta_1(h_j) + \sum_{j=1}^{T} \theta_2(y, h_j) + \sum_{j=1}^{T-1} \theta_3(y, h_j, h_{j+1}), \tag{6}$$

where $\phi(x_j) \in \mathbb{R}^d$ is the feature representation of action $\mathbf{x}$ at the jth time stamp. $\theta_1(h_j) \in \mathbb{R}^d$ is the parameter vector of the jth hidden variable. The inner product of $\phi(x_j) \cdot \theta_1(h_j)$ represents the consensus between observation x_j and hidden state h_j. $\theta_2(y, h_j) \in \mathbb{R}$ and $\theta_3(y, h_j, h_{j+1}) \in \mathbb{R}$ measure the compatibility among the corresponding variables.

The hidden variables in Equation (6) are used to model the compatibility between the observations and the class labels. The semantic meaning of a hidden state can be considered a *key pose*. Thus, the first term in Equation (6), $\phi(x_j) \cdot \theta_1(h_j)$, can be interpreted as the consensus between frame x_j and pose h_j. The second term in Equation (6), $\theta_2(y, h_j)$, measures the compatibility between pose h_j and action class y. The third term in Equation (6), $\theta_3(y, h_j, h_{j+1})$, measures the compatibility between the successive poses and the action class. These hidden nodes are also used to enforce temporal smoothness.

With the training set D, the parameter set $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3\}$ can be learned by optimizing Equation (4). Efficient solvers, such as L-BFGS, can be applied to the optimization [Quattoni et al. 2007]. After learning, the HCRFs model $\boldsymbol{\theta}^*$ is obtained.

Feature extraction. In our work, $\phi(x_j) \in \mathbb{R}^d$ is the vector of the features extracted at time stamp j. The adopted features can vary from dataset to dataset. For example, they can be those extracted by applying the cuboid descriptor to RGB color images, or they can be the absolute 3D body joint locations extracted from a skeleton stream. The details of the features used in this work can be found in Section 4.2. Each x_j in this work is a set of successive frames within a small temporal window centered at time stamp j. Hence, $\phi(x_j)$ denote the features extracted from these frames. This implementation makes our approach tolerable to moderate misalignment or temporally nonuniform motions of actions. We still term x_j as a frame just for convenience in the following.

3.2. Outlier Frame Filtering

When an action $\mathbf{x}$ is given in the test phase, its label y is then predicted via HCRFs by

$$y = \arg\max_{y' \in \mathcal{Y}} \sum_{\mathbf{h}} P(y', \mathbf{h} | \mathbf{x}, \boldsymbol{\theta}^*). \tag{7}$$

However, as mentioned previously, outlier frames may occur in the test action $\mathbf{x}$. The extracted features $\{\phi(x_j)\}$ from outlier frames cannot be properly handled by model $\boldsymbol{\theta}^*$ in the potential function shown in Equation (6). This issue needs to be addressed because it typically causes performance degradation. Thus, our goal at this stage is to detect the outlier frames of action $\mathbf{x}$ by splitting it into two disjointed subsets $\mathbf{x^o}$ and $\mathbf{x^u}$, where $\mathbf{x^o} \cup \mathbf{x^u} = \mathbf{x}$ and $\mathbf{x^o} \cap \mathbf{x^u} = \emptyset$. The former subset comprises the observed part, and the latter consists of the unobserved part.

In this work, we consider the video frames of a test action as outliers if they are not similar to any frames in the training set. By this definition, false alarms may occur when the training set does not cover all the possible variations of actions. The learned HCRFs model does not work well on these unseen frames in the training phase. No matter whether they are true outlier frames or not, removing such frames before making the inference helps reduce the instability of the prelearned HCRFs model.

To this end, consider all the frames $S = \{x\}$ obtained in the training set D; with the adopted feature representation $\{\phi(x) \in \mathbb{R}^d\}$, the k-means clustering algorithm [MacQueen 1967] is applied to divide S into K disjointed groups, namely, $S = S_1 \cup S_2 \cup \cdots \cup S_K$. A multivariate Gaussian density is then used to model the frames in each group. That is, the density that frame x belongs to S_k is estimated by

$$p(x|S_k) = \mathcal{N}(\phi(x)|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{8}$$

where $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$ are the mean vector and covariance matrix of group S_k.

The cluster number K is set as the number of hidden states, which correspond to the key poses of actions. The number of hidden states is determined by using cross-validation in our experiments. For a test action $\mathbf{x} = \{x_t\}_{t=1}^{T}$, we framewise divide it into $\mathbf{x^o} \cup \mathbf{x^u}$ according to

$$x_t \in \begin{cases} \mathbf{x^u}, & \text{if } \max_k p(x_t|S_k) < \epsilon_k, \\ \mathbf{x^o}, & \text{otherwise,} \end{cases} \tag{9}$$

where ϵ_k is the threshold.

Determining the values of the thresholds $\{\epsilon_k\}_{k=1}^{K}$ in Equation (9) is nontrivial because they are usually data dependent, since outliers may be arbitrarily distributed. Outliers are those that are inconsistent with training data. Therefore, we set the thresholds by referencing training data so that at least 80% of the training frames are preserved. Test frames not covered by the training data are then treated as outliers and will be filtered out before prediction.

3.3. Observation Completion

Through outlier frame detection, action $\mathbf{x}$ is split into two parts, $\mathbf{x} = \mathbf{x^o} \cup \mathbf{x^u}$. To avoid the unfavorable influence caused by the outlier part $\mathbf{x^u}$, the objective for inference is changed from Equation (7) to

$$y = \arg\max_{y' \in \mathcal{Y}} \sum_{\mathbf{h}} P(y', \mathbf{h}|\mathbf{x^o}, \boldsymbol{\theta}^*), \tag{10}$$

where the outlier frames have been removed at this point. A practical way to predict $\mathbf{x}$ via Equation (10) is to accordingly remove the part where $\mathbf{x^u}$ involves, that is,

$$\Psi\left(y, \mathbf{h}, \mathbf{x^o}, \boldsymbol{\theta}\right) = \sum_{j \in \mathcal{O}} \phi\left(x_j\right) \cdot \theta_1\left(h_j\right) + \sum_{j \in \mathcal{O}} \theta_2\left(y, h_j\right) + \sum_{j \in \mathcal{O}, j+1 \in \mathcal{O}} \theta_3\left(y, h_j, h_{j+1}\right), \tag{11}$$

with the index set $\mathcal{O}$ defined as $\{t|x_t \in \mathbf{x^o}\}$.

Despite the feasibility, two problems arise when inferring from incomplete observation via Equation (10) with the modified potential function in Equation (11). First, the observed part $\mathbf{x^o}$ may carry insufficient evidence to make a reliable prediction. Second, the outlier frames may separate the observed part $\mathbf{x^o}$ into several isolated segments, making temporal consensus unavailable for the regularization of action inference.

To address the two problems, we reexploit the training data to compensate for the unobserved part. Specifically, the observed part $\mathbf{x^o}$ is considered the *query* to the training set $D = \{\mathbf{x}_i\}_{i=1}^{N}$ that serves as the *gallery*. We accordingly split the training actions $\{\mathbf{x}_i = (\mathbf{x}_i^{\mathbf{o}}, \mathbf{x}_i^{\mathbf{u}})\}_{i=1}^{N}$ based on $\mathcal{O}$, the index set of the observed frames in test action $\mathbf{x}$. Then, the estimated unobserved part $\tilde{\mathbf{x}}^{\mathbf{u}}$ of action $\mathbf{x}$ is determined by

$$\phi(\tilde{\mathbf{x}}^{\mathbf{u}}) \leftarrow \sum_{i=1}^{N} \lambda_i \phi\left(\mathbf{x}_i^{\mathbf{u}}\right), \tag{12}$$

where

$$\lambda_i = \frac{\exp(-d^2(\phi(\mathbf{x^o}), \phi(\mathbf{x}_i^{\mathbf{o}}))/\sigma^2)}{\sum_{j=1}^{N} \exp(-d^2(\phi(\mathbf{x^o}), \phi(\mathbf{x}_j^{\mathbf{o}}))/\sigma^2)}, \tag{13}$$

$d(\cdot, \cdot)$ is the used distance (or dissimilarity) function, and σ is a positive constant. The value of σ is empirically set to the average distance between training actions in the observed part. The rationale behind Equations (12) and (13) is that the similarity between two actions in one part implies their similarity in another part. As can be seen in Equation (13), if this test action is more similar to the ith training sample in

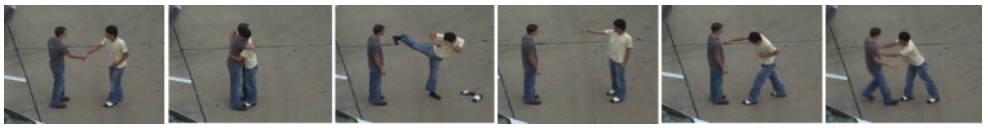


Fig. 3. The six categories of activities in the UT-Interaction dataset. From left to right, they are *hand-shaking, hugging, kicking, pointing, punching,* and *pushing*.

the observed part, the weight λ_i used in the composition is larger. The denominator of Equation (13) is used for weight normalization. The employed distance function $d(\cdot, \cdot)$ is dependent on the feature representations adopted. We use Kullback-Leibler (KL) divergence for histogram-based features and Euclidean distance for the rest. The details of the used distance functions are given in Section 4.2.

The label of test action $\mathbf{x}$ is then inferred by Equation (7) (with the potential function Equation (6)) by taking the *augmented action* $\tilde{\mathbf{x}} = \mathbf{x^o} \cup \tilde{\mathbf{x}}^{\mathbf{u}}$ as input. In this way, feature enrichment and temporal regularization are both attained, which is termed *observation completion* in our study. It makes the pretrained HCRFs model applicable to partially observed data and leads to remarkable performance improvement in POAR. The proposed observation completion and the improved performance distinguish our approach from previous ones for recognizing partially observed actions such as Cao et al. [2013], Davis and Tyagi [2006], Raptis and Sigal [2013], Hoai and De la Torre [2014], Lan et al. [2014], Ryoo [2011], Chang et al. [2009], and Banerjee and Nevatia [2014].

To conclude this section, we summarize how *outlier frame filtering* and *observation completion* work by showing the procedure of predicting a given testing action $\mathbf{x}$ step by step: (1) The action $\mathbf{x}$ is divided into the observed part $\mathbf{x^o}$ and the unobserved part $\mathbf{x^u}$ using outlier frame filtering via Equation (9). (2) The observed part $\mathbf{x^o}$ serves as the query to the training data, and we seek $\tilde{\mathbf{x}}^{\mathbf{u}}$, which is coherent to $\mathbf{x^o}$ and can act as an alternative to $\mathbf{x^u}$ via Equation (12). (3) Then, the *augmented action* $\tilde{\mathbf{x}} = \mathbf{x^o} \cup \tilde{\mathbf{x}}^{\mathbf{u}}$ is available. The prediction of action $\mathbf{x}$ is accomplished by taking the augmented action $\tilde{\mathbf{x}}$ as the input to the learned HCRFs in Equation (7).

4. EXPERIMENTAL SETUP

This section introduces our experimental settings. We first describe the datasets used for performance evaluation, including two benchmarks for action recognition and the dataset we collected, and afterward we discuss the adopted feature representation and the evaluation metric on each dataset.

4.1. Datasets for Performance Evaluation

The performance of our approach is evaluated on three datasets including the *UT-Interaction* [Ryoo and Aggarwal 2010] dataset, the *ArmGesture* [Quattoni et al. 2007] dataset, and our *CITI-DailyActivities3D* dataset.[1] These datasets comprise videos of different modalities such as RGB videos, depth maps, and 3D skeleton structures. They cover various activities ranging from single-person actions to multiperson interactions. Actions in the first two datasets are *clean*, whereas outlier frames are present in the CITI-DailyActivities3D dataset.

4.1.1. UT-Interaction Dataset. It is a collection of human interaction videos of six activity categories: *hand-shaking, hugging, kicking, pointing, punching,* and *pushing*. It is divided into two sets called UT-Interaction #1 and #2. Each has 60 videos with six types of human interactions and 10 videos per activity. Figure 3 shows some examples.

[1]CITI-DailyActivities3D dataset is available at citidatabase.shihyaolin.com.

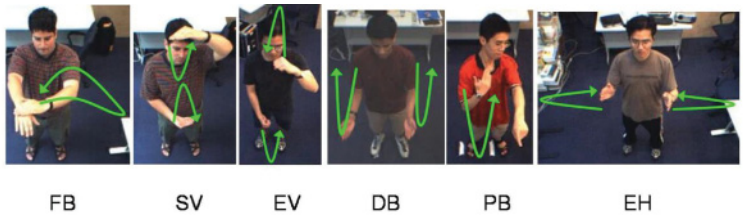


Fig. 4. The six categories of arm gestures in the ArmGesture dataset. From left to right, they are *Flip Back (FB)*, *Shrink Vertically (SV)*, *Expand Vertically (EV)*, *Double Back (DB)*, *Point and Back (PB)*, and *Expand Horizontally (EH)*.

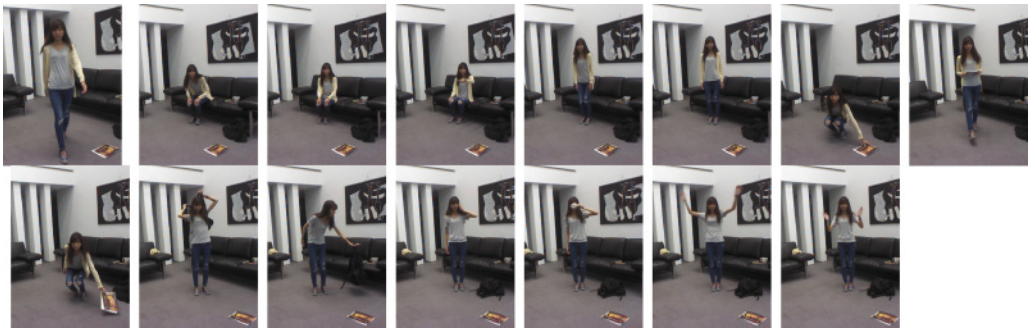


Fig. 5. The CITI-DailyActivities3D dataset. The figure shows one example from each of the 15 daily activities included in this dataset. The eight categories from left to right in the first row are *Walk, Sit down, Sit still, Use a TV remote, Stand up, Stand still, Pick up books,* and *Carry books*, respectively. The seven categories from left to right in the second row are *Put down books, Carry a backpack, Drop a backpack, Make a phone call, Drink water, Wave hand,* and *Cap*, respectively.

Both *segmented* and *unsegmented* versions of this dataset are available. Following the setting in Ryoo [2011], we use the segmented one for evaluation.

4.1.2. ArmGesture Dataset. It contains six types of arm-gesture sequences, including *Flip Back (FB)*, *Shrink Vertically (SV)*, *Expand Vertically (EV)*, *Double Back (DB)*, *Point and Back (PB)*, and *Expand Horizontally (EH)*, as shown in Figure 4. The video sequences were performed by 13 people with 120 sample sequences per class on average.

4.1.3. CITI-DailyActivities3D Dataset. Most existing benchmarks such as *MSR-ActionPair* [Oreifej and Liu 2013], *NATOPS* [Song et al. 2013], *UTKinect-Action* [Xia and Aggarwal 2013], *MSRC-12 Kinect gesture* [Fothergill et al. 2012], and *Cornell-Activity* [Sung et al. 2012] comprise videos with no or few corrupted frames. We collected this new dataset, where abundant outlier frames are irregularly and naturally present. It contains 15 daily activities: *walk, sit down, sit still, use a TV remote, stand up, stand still, pick up books, carry books, put down books, carry a backpack, drop a backpack, make a phone call, drink water, wave hand*, and *clap*, as shown in Figure 5.

The dataset has 482 sequences. Among them, 182 sequences contain outlier frames presenting in arbitrary locations and lasting for various durations. Ten actors, including eight males and two females, were recruited for building this dataset, and one of them is left-handed. Each activity is performed by each actor between two and five times. A Microsoft Kinect was used for the collection so that the RGB video, the depth maps, and the inferred skeletons of each activity sequence are all available. The skeleton structures in this work were extracted by using the `Kinect for Windows SDK`.

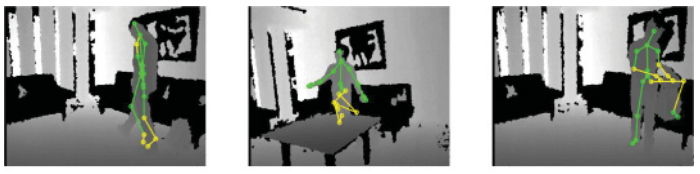


Fig. 6. Outlier frames in the skeleton streams caused by (a) self-occlusion, (b) object-occlusion, and (c) incorrect skeleton estimation.

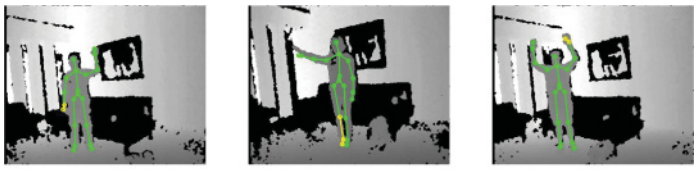


Fig. 7. An example of large intraclass variations. For the *wave hand* activity, the actors may wave their (a) left hands, (b) right hands, or (c) both hands.

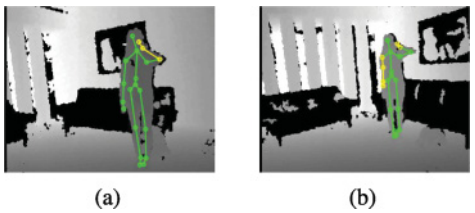


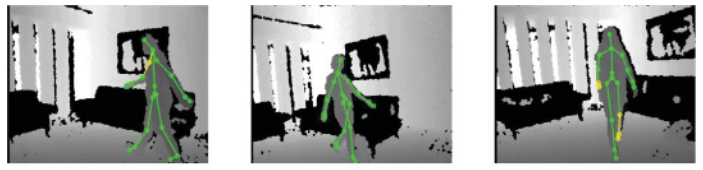
Fig. 8. An example of high interclass similarity. The skeleton structures are from (a) the *make a phone call* activity category and (b) the *drink water* activity category. These activities look very similar.

This dataset contains outlier frames, and several examples in the skeleton streams are shown in Figure 6, where the portions of the skeletons extracted with low confidence are drawn in yellow. It is seen that outlier frames could be caused by various reasons such as self-occlusion, object-occlusion, and incorrect skeleton estimation. This dataset also has large intraclass variations. Figure 7 gives an example of this case. For the *wave hand* activity, the actors may wave their left hands, right hands, or both. It contains high interclass similarity too. An example is displayed in Figure 8, where the skeletons of two activities *make a phone call* and *drink water* look very similar. In addition, unlike most daily activity datasets (e.g., Song et al. [2013], Sung et al. [2012], Oreifej and Liu [2013], and Fothergill et al. [2012]) where the actors are asked to face the camera, we did not include this requirement when constructing the CITI-DailyActivities3D dataset. The setting is thus more realistic and increases the difficulty owing to the perspective-projection variations. An example is given in Figure 9.

4.2. Feature Representation

These datasets were collected in diverse settings. The appropriate features to represent actions vary from dataset to dataset. The adopted feature representations are described next.

4.2.1. UT-Interaction Dataset. We follow the feature extraction setup in Ryoo [2011], where the *Spatial-Temporal Interest Points* (STIPs) are firstly detected for each action, and the *cuboid descriptor* [Dollár et al. 2005] is applied to describe each STIP. The



(a) (b) (c)

Fig. 9. An example of variations caused by perspective difference. (a)–(c) Three actions of the *walk* category are captured from different points of view.

Harris3D corner detector [Laptev 2005] is used for STIPs detection in this work. To produce a compact representation, actions are described by using the *BoW model* [Fei-Fei and Perona 2005], where the *codebook* is generated via k-means clustering with 800 codewords [Cao et al. 2013]. Each action is equally partitioned into $T = 20$ temporal segments. The KL divergence is used as the distance measure.

4.2.2. ArmGesture Dataset. The precompiled features are provided in this dataset, and are used in our experiments. The 2D tracked joint angles and 3D joint coordinates of shoulders and elbows are combined to form the feature representation. A 20-dimensional feature vector is used for characterizing a temporal segment. Euclidean distance is used.

4.2.3. CITI-DailyActivities3D Dataset. We use the absolute 3D body joint positions in the skeleton streams as the feature representation. $T = 30$ skeletons are uniformly sampled for each action in this dataset. To make the skeletons invariant to the absolute location, we perform the following preprocessing to normalize each skeleton. For location normalization, we transform the skeletal data from the world coordinate system to the person-centric coordinate system by setting the hip center as the origin. For scale normalization, we select a skeleton in this dataset as the reference, and normalize all the other skeletons so that their body part lengths are the same as that of the reference skeleton. For being robust to the viewpoint changes, we rotate each skeleton such that the ground plane projection of the vector from its left hip to its right hip is parallel to the global x-axis. Euclidean distance is used for dissimilarity estimation.

4.3. Evaluation Metric

For performance analysis and comparison, the evaluation metric for each dataset is described as follows.

4.3.1. UT-Interaction Dataset. We follow Cao et al. [2013], Ryoo [2011], and Banerjee and Nevatia [2014], and adopt *leave-one-sequence-out cross-validation* for performance evaluation.

4.3.2. ArmGesture Dataset. We follow Quattoni et al. [2007] and Song et al. [2012, 2013] where *fivefold cross-validation* is used as the performance measure.

4.3.3. CITIDailyActivities3D Dataset. We conduct three evaluation tasks on this dataset. The evaluation task #1 aims at evaluating the performance of approaches on fully observed videos. That is, neither the training nor the testing action videos contain outlier frames. The evaluation task #2 puts emphasis on the tolerance of approaches to outlier frames. Namely, the model of the approach for evaluation is learned on the clean training set, but tested on videos with outlier frames. Both the training and testing sets in the evaluation task #3 contain mixtures of clean and outlier frames. We will show that our approach can be further extended to address not only corrupt testing data but also corrupt training data. The *cross-subject test setting* is used in all

the tasks. We randomly split the 10 subjects into two equal-size groups. The actions of the subjects in one group serve as the training data, while the remaining actions act as the testing data. We then switch the two subject groups. The average performance is reported.

5. EXPERIMENTAL RESULTS

The proposed approach is evaluated in this section. Two sets of experiments are conducted. In the first set of experiments, we assume that the locations of outlier frames are *known in advance*. The outlier filtering mechanism of our approach is turned off for this set of experiments. We focus on evaluating the advantages of the proposed observation completion over existing approaches that work on actions with incomplete observation. Specifically, two settings, *early prediction* and *gap-filling*, are considered. The former involves recognizing actions with missing frames at the end of the sequences. The latter involves recognizing actions where the outlier (missing here) frames locate in the middle and thus, the observed frames are separated into two segments. The first set of experiments is conducted on the UT-Interaction and ArmGesture datasets. The CITI-DailyActivities3D dataset consists of actions with arbitrarily presenting outlier frames. It is inconsistent with early prediction and gap-filling.

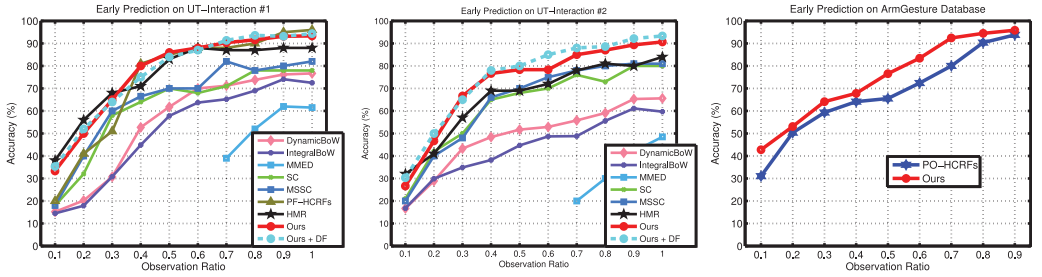
In the second set of experiments, the locations of outlier frames are *unknown*. This setting is more difficult and has not been considered in previous studies. The primary evaluation of interest is to check whether the proposed components, outlier filtering and observation completion, jointly work well. This set of experiments is conducted on two datasets. The outlier frames in the UT-Interaction dataset are manually synthetic, whereas those in the CITI-DailyActivities3D dataset are real. The ArmGesture dataset provides only the precompiled features instead of video frames. No outlier frames can be added. It is hence not adopted in this set of experiments.

5.1. Early Prediction

The positions of the unobserved frames in action videos are known before recognition in early prediction. Outlier frame filtering introduced in Section 3.2 is hence turned off. Only observation completion introduced in Section 3.3 is applied to retrieve a plausible alternative to the unobserved part. Next, we describe some of the representative approaches that can recognize actions with incomplete observation, and compare our approach with them.

5.1.1. Approaches for Comparison. For the UT-Interaction dataset, we chose the following state-of-the-art methods for comparison. These methods include *Integrate-BoWs* [Ryoo 2011], *DynamicBoW* [Ryoo 2011], *Max-Margin Early Event Detectors* (MMEDs) [Hoai and De la Torre 2014], *Sparse Coding* (SC) based method [Cao et al. 2013], *Mixture of Segments Sparse Coding* (MSSC) [Cao et al. 2013], *Pose Filter based Hidden Random Conditional Fields* (PF-HCRFs) [Banerjee and Nevatia 2014], and *Hierarchical Movemes Representation* (HMR) [Lan et al. 2014]. For the event detector MMED, we follow the setting in Cao et al. [2013] to evaluate its performance. All the previous methods have reported their results on both UT-datasets #1 and #2 except that PF-HCRFs [Banerjee and Nevatia 2014] provides the results on the UT-dataset #1 only. Motivated by the good performance of CNNs-based approaches to action recognition, the proposed approach is applied to the Deep learning-based Features (DF) extracted by using the two-stream architecture in Feichtenhofer et al. [2016]. The resultant approach is denoted by *Ours +DF*.

For the ArmGesture database, we compare our method with that in Chang et al. [2009], which infers *Partial Observation upon Hidden Conditional Random Fields* (PO-HCRFs). This scheme adopts Equation (11) to infer the action label when missing



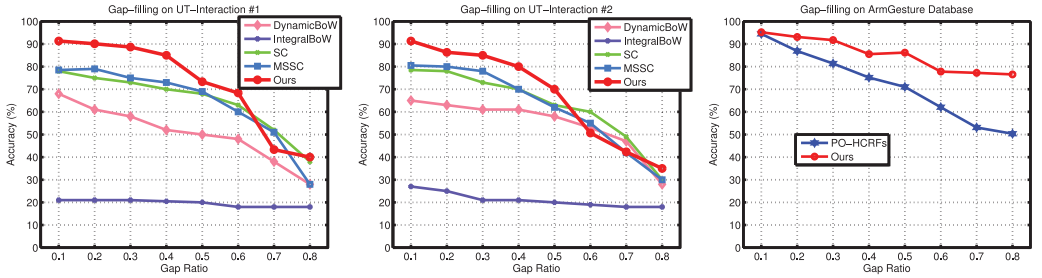
(a) (b) (c)

Fig. 10. Results of early prediction on (a) the UT-Interaction dataset #1, (b) the UT-Interaction dataset #2, and (c) the ArmGesture dataset.

frames occur at the end of the actions. The competing approach PO-HCRF is implemented by us, because its performance on the ArmGesture dataset has not been reported.

5.1.2. Results Analysis. Figures 10(a) and 10(b) summarize the performance of the competing approaches and our approach on the UT-Interaction datasets #1 and #2, respectively. The recognition rates of each approach with different fractions of the observed segments in videos, that is, *observation ratios*, are shown. Figure 10(a) shows that the HCRFs-based approaches such as our approach and PF-HCRFs [Banerjee and Nevatia 2014] perform better than the sparse coding based methods (e.g., SC and MSSC), and BoW approaches (e.g., DynamicBow and IntegralBow). This better performance occurs mainly because HCRFs employ hidden states to better enforce the implicit temporal coherence. Actions in the UT-Interaction dataset #2 are noisier than those in #1. In Figure 10(b), the sparse coding based methods, SC and MSSC, are robust to noises and achieve comparable performance to HMR [Lan et al. 2014] on the UT-Interaction dataset #2. However, since the likelihood at each action segment is estimated independently, SC or MSSC would neglect temporal coherence among the observed parts. Our approach employs temporal coherence information of the observed parts, and performs favorably against both SC and MSSC.

Our approach also outperforms the structural Support Vector Machine (SVM) [Finley and Joachims 2008] based approach MMED. The reason why MMED does not perform well here could be that it was designed to detect the starting and ending frames of the particular events. It is not fully consistent with our experimental setting where the starting and ending frames are known. Compared to PF-HCRFs and HMR, our approach achieves a higher performance in general (though is worse sometimes) as shown in Figures 10(a) and 10(b). This is because our approach recovers the unobserved part by referring to and borrowing information from the training data. The unobserved part completed in our approach carries richer and time-varying information. Then, by using both the observed part and the completed unobserved part, temporal regularization becomes attainable. Hence, our approach achieves favorable performance in comparison to other approaches. Note that when the observation ratio is low in Figure 10(a), the performance of our approach is lower than that of HMR because our approach uses the observed part to complete the unobserved part. If the observed duration is short, the quality of the completed unobserved part is degraded. Our method with deep learning-based features (DF) performs slightly better than with the ordinary cuboid descriptor-based BoW features. However, the performance gain is not significant. The reason may be that the two-stream network has the great power of fitting training data, and so it is more sensitive to the outlier frames that are present in testing.



(a) (b) (c)

Fig. 11. Results of gap-filling on (a) the UT-Interaction dataset #1, (b) the UT-Interaction dataset #2, and (c) the ArmGesture dataset.

Figure 10(c) shows that our method outperforms the PO-HCRFs in Chang et al. [2009] on the ArmGesture dataset. Partial observation in PO-HCRFs is used for the inference via Equation (11). Our approach instead uses not only the partially observed part but also the completed unobserved part for inference. Higher recognition rates are thus obtained.

5.2. Gap-Filling

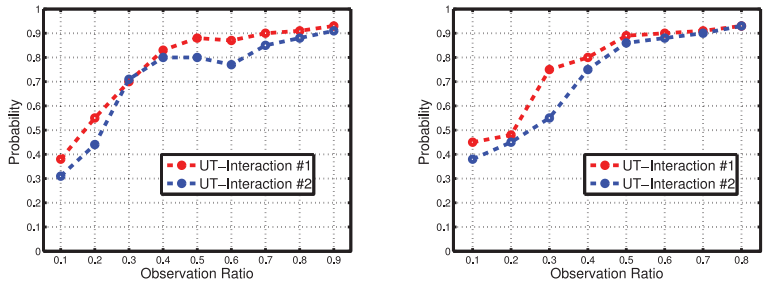
Unlike early-prediction, which has been studied extensively, there are fewer results on the gap-filling task, which was introduced by Cao et al. [2013] and is addressed under the assumption that the gap's location and duration are given.

5.2.1. Approaches for Comparison. Approaches including IntegralBow [Ryoo 2011], DynamicBow [Ryoo 2011], SC [Cao et al. 2013], and MSSC [Cao et al. 2013] have been evaluated on the UT-Interaction dataset and hence, are selected for comparison. Like early prediction, our approach is also compared to PO-HCRFs [Chang et al. 2009] on the ArmGesture dataset.

5.2.2. Results Analysis. Figures 11(a) and 11(b) report the performance of gap-filling by the competing approaches and our approach on the UT-Interaction datasets #1 and #2, respectively. In general, our approach works more favorably than those based on the sparse representation (SC and MSSC) and those based on the BoW representation (DynamicBow and IntegralBow). However, similar to the case of early prediction, when the gap ratio is higher, the performance of our approach drops owing to the same reason. Nevertheless, our method performs generally better than the other methods for both early-prediction and gap-filling.

The results in Figure 11(c) show that our method outperforms PO-HCRFs [Chang et al. 2009] remarkably on the ArmGesture dataset. Our approach performs better on the ArmGesture dataset than on the UT-Interaction dataset in the gap-filling scenario since the crucial gesture motions are typically present near the end of the actions. The results confirm that the proposed observation completion can effectively estimate the unobserved gap by using the earlier and end parts as the query, and lead to considerable performance gains on both datasets.

To gain insight into how observation completion improves the performance, we measure the *accuracy* of observation completion. Specifically, we compute the probability for the case that the retrieved alternative, that is, $\tilde{\mathbf{x}}^{\mathbf{u}}$ in Equation (12), is correct. We consider the alternative is correct if the training action with the largest weight in Equation (13) is of the same action category. Figures 12(a) and 12(b) show the probabilities of correct observation completion on the UT-Interaction dataset for early prediction and gap-filling, respectively. It can be observed that more than 70% of the unobserved



(a) (b)

Fig. 12. Probabilities of correct observation completion on the UT-Interaction dataset for (a) early prediction and (b) gap-filling.

parts are replaced by correct alternatives when the observation ratio is higher than 40% in both tasks. It is the main reason why our approach still works well even when abundant outliers are present.

5.3. POAR with Synthetic Outlier Frames

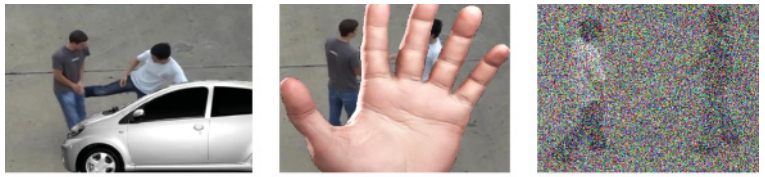
We evaluate our approach to POAR where the number, locations, and durations of outlier frame segments are unknown. This task has not been addressed by previous studies to the best of our knowledge. We jointly use the proposed algorithms for outlier frame filtering and observation completion to address this problem.

Specifically, we consider two scenarios. First, our approach is evaluated on actions with *synthetic* outlier frames. The outlier frames are artificially added to the testing actions in UT-Interaction dataset. Second, we test our approach on our CITI-DailyActivities3D dataset, where the outlier frames are *real* and occur naturally. The first scenario is explored in the following, while the second one is discussed in the next section.

5.3.1. Synthetic Outliers. For the first scenario, we randomly added outlier frames to action videos in the UT-Interaction datasets #1 and #2. The positions of the outlier frames are arbitrarily generated in the input videos. A wide range of *outlier ratio*, the proportion of outliers to all frames, ranging from 0.1 to 0.9 is considered. The types of the added outlier frames include frame signal noise, object occlusion, and camera occlusion. Figure 13 shows the examples of these outlier frames that we generated manually.

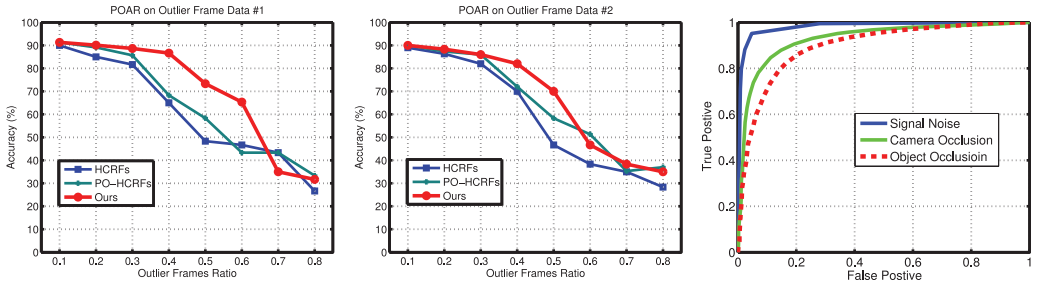
5.3.2. Approaches for Comparison. None of the previously adopted compared approaches is designed to work with unknown outlier frames. We compare our method with two baselines. The first baseline is HCRFs. Comparing our approach to this baseline measures the advantage of jointly using outlier frame filtering and observation completion for POAR. The other baseline is a variant of PO-HCRFs [Chang et al. 2009], in which the outlier frames are detected and removed by using our outlier-frame-filtering algorithm and then the action is inferred via Equation (11). We can examine whether observation completion helps improve the performance of POAR by comparing our approach with this baseline.

5.3.3. Results Analysis. Figures 14(a) and 14(b) show the performance of the two baselines and our approach on the UT-Interaction dataset. Compared to the two baselines, the performance of our method is more favorable in most cases. Our method and PO-HCRFs perform better than HCRFs in general, since they both try to detect and remove the outlier frames before predicting the actions. Compared to PO-HCRFs, our method



(a) (b) (c)

Fig. 13. Sample images of the outlier frames caused by (a) object occlusion, (b) camera occlusion, and (c) signal noise. The ratios of the noisy areas to the whole images are 34.5%, 53.7%, and 100% in the three cases, respectively.



(a) (b) (c)

Fig. 14. Results of POAR with unknown, synthetic outlier frames on (a) the UT-Interaction dataset #1 and (b) the UT-Interaction dataset #2. (c) ROC curves of outlier frame detection by using our outlier frame filtering algorithm.

achieves higher recognition rates in most cases. This result demonstrates the advantage of using observation completion. In the cases where the fractions of outlier frames are large, inference with the completed observation does not always lead to better results, because the inferred unobserved part may be incorrect. Our approach on average is remarkably superior to HCRFs and PO-HCRFs.

To individually evaluate the effectiveness of outlier filtering, Figure 14(c) demonstrates the Receiver Operating Characteristic (ROC) curves for the three types of outlier frames: object occlusion, camera occlusion, and signal noise. As shown in Figure 14(c), our outlier frame filtering algorithm achieves the true positive rates of 0.8 with false positive rates below 0.2 for all types of outlier frames.

5.4. POAR with Real Outlier Frames

Our approach here is evaluated on the CITI-DailyActivities3D dataset where the outlier frames are real and appear irregularly and naturally in the action sequences. Three tasks are conducted for evaluation on this dataset. In task #1, both training and testing actions are clean, that is, they have no outliers. Instead, task #2 and task #3 put emphasis on the tolerance of outliers. In task #2, the training actions are clean and the same as those in task #1, whereas testing actions are corrupted by outliers. In task #3, both the training and testing sets contain clean and corrupt actions. To deal with corrupted training data, we apply outlier filtering and observation completion to each training video in the same way of applying them to a testing video, except that this training video is temporarily removed from the training set when observation completion is performed. This avoids completing a training video by borrowing frames from itself.

In task #1, we check if our approach with extra components for outlier handling still performs well for clean actions. More importantly, we are interested in investigating

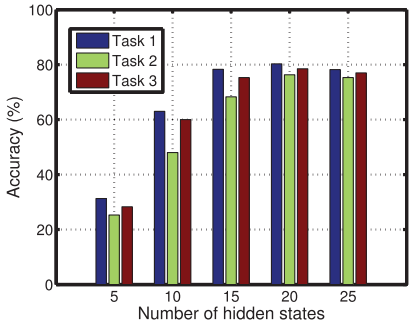


Fig. 15. The performance of our approach with different numbers of hidden states in HCRFs on all three tasks.

the performance differences between the first task and the other two tasks, which reveal the robustness of an approach against outliers.

Note that the value of K in k-means clustering in Equation (8) is set to the number of hidden states in HCRFs, because both of them correspond to the key poses of actions. We conduct an experiment to evaluate the sensitivity of our approach to the number of hidden states in HCRFs. Figure 15 shows the performance of our approach with different numbers of hidden states on all the three tasks. The results point out that a few hidden states suffice for getting stable performance.

5.4.1. Approaches for Comparison. We select 12 existing approaches for comparison, including the *Nearest Neighbor classifier* (NN), *k-Nearest Neighbor* (*k*NN) [Devanne et al. 2015], *Decision Tree* (DT), *Naïve Bayes Classifier* (NBC) [Chen et al. 2013], *Single-Hidden Layer Feedforward Neural Networks* (SLFNs) [Iosifidis et al. 2013], *Recurrent Neural Networks* (RNNs) [Martens and Sutskever 2011], HMM [Lv and Nevatia 2006; Piyathilaka and Kodagoda 2013], *Action Graph* (AG) [Li et al. 2008], HCRFs [Quattoni et al. 2007], *Hidden-state Conditional Neural Fields* (HCNFs) [Song et al. 2013], *Hierarchical Sequence Summarization model* (HSS) [Song et al. 2013], and the state-of-the-art approach by Gowayyed et al. [2013], where HCNFs are the Conditional Neural Fields (CNF) [Peng et al. 2009] with latent variables.

We are particularly interested in the comparison between our approach and the competing approach HCRFs [Quattoni et al. 2007]. Both approaches adopt hidden CRFs for classification. Their main difference is that HCRFs predict a given test action directly, whereas ours predicts that test action after it is processed by the proposed outlier frame filtering and observation completion. The performance gain of our approach over HCRFs [Quattoni et al. 2007] reveals its advantages.

Except for the approach in Gowayyed et al. [2013], all the approaches for comparison and ours adopt the 3D Joint Positions (JPs) of skeletal data as the feature representation. Note that the feature representation is not the one used in the original paper of each competing method. Nevertheless, using the same feature representation helps single out the effect of applying our approach to partially observed action recognition when comparing it with other methods. For graphical model-based classifiers, for example, HMM, HCRFs, and ours, the feature vector JP at each frame serves as the input to the corresponding observation node. For classifiers working on data in holistic representations, for example, NN, DT, and NBC, we concatenate the JP features extracted from all the frames. The approach by Gowayyed et al. [2013] adopts a scale- and speed-invariant body-joint-trajectory descriptor, namely, a *Histogram of Oriented Displacement* (HOD), for feature extraction. Furthermore, the feature representation

Table I. Accuracy Rates of Different Approaches on CITI-DailyActivities3D Task 1

Method	Accuracy (%)
NN Classifier	73.3
k-NN Classifier ($k = 5$) [Devanne et al. 2015]	69.6
Decision Tree	55.6
Naïve Bayes Classifier [Chen et al. 2013]	73.3
SLFNs (20 neurons) [Iosifidis et al. 2013]	74.3
Recurrent Neural Networks [Martens and Sutskever 2011]	77.3
Hidden Markov Model [Lv and Nevatia 2006; Piyathilaka and Kodagoda 2013]	73.3
Action Graph [Li et al. 2008]	71.5
Hidden-State CRFs [Quattoni et al. 2007]	80.3
Hidden-State Conditional Neural Fields [Song et al. 2013]	81.3
Hierarchical Sequence Summarization Model [Song et al. 2013]	82.3
Approach by Gowayyed et al. [2013] (16 Bins, 1 Level)	83.6
Approach by Gowayyed et al. [2013] (8 Bins, 3 Levels)	83.0
Ours	**80.3**

is compiled by using the *Fourier Temporal Pyramid* (FTP) to handle possible temporal misalignment and noisy observation.

5.4.2. Results on Task #1. Table I reports the accuracy rates of our approach and the 12 compared approaches on task #1, where all the training and testing videos are without corrupted frames. The accuracy of the NN classifier is 73.3%, while that of the kNN classifier is 69.6% with $k = 5$. The decision tree and Naïve Bayes classifier give recognition rates of 55.6% and 73.3%, respectively. When we set the number of neurons in the hidden layer as 20, methods SFLNs and RNNs achieve performances of 74.3% and 77.3%, respectively. The graphical model-based approaches including HMM, AG, HCRFs, HCNFs, and HSS give accuracies of 73.3%, 71.5%, 80.3%, 81.3%, and 82.3%, respectively. It can be observed that the accuracy of methods with hidden variables, such as RNNs, HMM, HCRFs, and HSS, achieve better performance. The reason may be that the hidden variables can better express the possible temporal variations. Thus, methods based on hidden variables are more robust to temporal misalignment. The state-of-the-art method [Gowayyed et al. 2013] achieves an accuracy of 83.6%.

Our approach achieves a recognition rate of 80.3%, which is superior or comparable to most competing approaches. It is worth noting that our approach and HCRFs give the same recognition rate on this task with the *clean* testing data. This result indicates that the additional steps, outlier frame filtering and observation completion, of our approach do not cause a drop in performance, even though they are designed to deal with outlier frames.

5.4.3. Results on Task #2. Table II reports the recognition results of our approach and the 12 approaches for comparison in the evaluation task #2. The difference between the two tasks is that the testing actions in task #2 contain outlier frames that occur arbitrarily with various temporal lengths. Note that the testing sets in the two tasks are different, and so the recognition rates in the two tasks cannot be compared. Nevertheless, the relative performance drops between the two tasks of different approaches can be compared. By comparing the results in Table I and Table II, all 12 competing approaches suffer from substantial performance drops ranging from 7.8% (=55.6%–47.8% in approach DT) to 21.7% (=73.3%–51.6% in approach HMM). We also observe that the problem of performance drop is even more dramatic in graphical model-based approaches such as HMM and HCRFs. These approaches are more expressive for temporal consistence modeling, and hence are more sensitive to noisy data accordingly. The Fourier temporal pyramid skeletal features used in Gowayyed et al. [2013] still show strong performance in this challenging task because the

Table II. Accuracy Rates of Different Approaches on CITI-DailyActivities3D Task 2

Method	Accuracy (%)
NN Classifier	57.6
k-NN Classifier ($k = 5$) [Devanne et al. 2015]	56.4
Decision Tree	47.8
Naïve Bayes Classifier [Chen et al. 2013]	64.8
SLFNs (20 neurons) [Iosifidis et al. 2013]	66.4
Recurrent Neural Networks [Martens and Sutskever 2011]	68.1
Hidden Markov Model [Lv and Nevatia 2006; Piyathilaka and Kodagoda 2013]	51.6
Action Graph [Li et al. 2008]	51.1
Hidden-State CRFs [Quattoni et al. 2007]	64.2
Hidden-State Conditional Neural Fields [Song et al. 2013]	62.6
Hierarchical Sequence Summarization Model [Song et al. 2013]	61.5
Approach by Gowayyed et al. [2013] (16 Bins, 1 Level)	66.1
Approach by Gowayyed et al. [2013] (8 Bins, 3 Levels)	69.9
Ours	**76.3**

Table III. Accuracy Rates of Different Approaches on CITI-DailyActivities3D Task 3

Method	Accuracy (%)
NN Classifier	63.0
k-NN Classifier ($k = 5$) [Devanne et al. 2015]	59.7
Decision Tree	45.1
Naïve Bayes Classifier [Chen et al. 2013]	62.8
SLFNs (20 neurons) [Iosifidis et al. 2013]	69.2
Recurrent Neural Networks [Martens and Sutskever 2011]	71.7
Hidden Markov Model [Lv and Nevatia 2006; Piyathilaka and Kodagoda 2013]	71.3
Action Graph [Li et al. 2008]	58.3
Hidden-State CRFs [Quattoni et al. 2007]	68.8
Hidden-State Conditional Neural Fields [Song et al. 2013]	66.3
Hierarchical Sequence Summarization Model [Song et al. 2013]	66.3
Approach by Gowayyed et al. [2013]	74.6
Ours	**78.5**

approach in Gowayyed et al. [2013] gives 69.9% recognition rate and outperforms the other competing approaches.

Our approach, with the developed outlier frame filtering and observation completion, can effectively address the recognition difficulties caused by outlier frames. It filters out the outlier frames, and retrieves alternative *inlier* frames from the training data to facilitate action prediction. It turns out that the performance drop is only 4.0% (=80.3%–76.3%), even if our approach is established upon graphical models. The achieved accuracy of 76.3% by our approach is significantly better than all 12 competing approaches.

5.4.4. Results on Task #3. Table III reports the recognition results of our approach and the competing approaches on the evaluation of task #3. The major difference between this task and the other two tasks is that some of training actions in this task are corrupted. The recognition rate of our approach on task #3 is higher than on task #2, because task #3 contains more training data including both clean and corrupt actions. The training and testing sets of this task are different from those of the other two tasks. Thus, the recognition rates of these approaches cannot be compared across tasks. Nevertheless, the performance rankings of all the approaches can be compared in the individual tasks. As shown in Table III, all 12 competing approaches suffer from performance degradation. The outlier frames in both training and testing data cause only a minor performance drop of our approach. Besides, the achieved accuracy of 78.5% by our approach remarkably outperforms all 12 competing approaches. It is

worth mentioning that some methods perform better on task #3 than task #2. The main reason is that similar outliers in the test actions have been included in the training actions.

6. SUMMARY AND CONCLUSION

This article presents an integrated action recognition approach that can both filter out outlier frames and infer the action label from a partially observed video. We first argue that the conventional HCRFs model has not provided a proper model to handle the case of corrupt frames, and it is difficult to reset the potential functions of the hidden and observation nodes without going back to the training data. We then introduce an outlier-frame-filtering approach to uncover the outlier frames, as well as the procedure of observation completion, which borrows plausible alternatives to the filtered outliers. Instead of retraining the HCRFs, we merely use the training data for feature-level temporal complementation. This is a simple but useful way to solve the problem. We also propose an inference algorithm to tackle the action recognition problem with incomplete observations based on the HCRFs model.

The partially observed problem considered in this article is general, which includes many problems as its special cases, for example, early prediction and gap-filling. Various practical issues could cause frames to be unreliable for inference (e.g., partial occlusion in a fraction of RGB videos and wrong axes extraction in depth videos), and this article presents a method handling the general problem setting of frame-level outliers. The proposed method is all-purpose because it does not try to identify the individual reasons but simply replaces the unreliable frames with those recalled from the training data. Our approach has been comprehensively evaluated on two benchmark datasets and one self-collected dataset where the outlier frames can be either synthetic or real, and the positions of these outliers can be either known or unknown. The promising experimental results show that our approach can achieve more favorable performance than existing approaches.

REFERENCES

Elisabeth Andre. 2013. Exploiting unconscious user signals in multimodal human-computer interaction. *ACM Trans. Multimedia Comput., Commun., Appl.* 9, 1s (2013), 48.

Alper Ayvaci, Michalis Raptis, and Stefano Soatto. 2012. Sparse occlusion detection with optical flow. *Int. J. Comput. Vis.* 97, 3 (2012), 322–338.

Prithviraj Banerjee and Ram Nevatia. 2014. Pose filter based hidden-CRF models for activity detection. In *Proc. Euro. Conf. Computer Vision*. 711–726.

Yu Cao, Daniel Barrett, Andrei Barbu, Swaminathan Narayanaswamy, Haonan Yu, Aaron Michaux, Yuewei Lin, Sven Dickinson, Jeffrey Mark Siskind, and Song Wang. 2013. Recognize human activities from partially observed videos. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2658–2665.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2016. Realtime multi-person 2D pose estimation using part affinity fields. *arXiv Preprint arXiv:1611.08050* (2016).

Gustavo Carneiro and Jacinto C. Nascimento. 2013. Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 11 (2013), 2592–2607.

Alexandros Andre Chaaraoui, José Ramón Padilla-López, and Francisco Flórez-Revuelta. 2013. Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices. In *Proc. Int' Conf. Computer Vision Workshops*. 91–97.

Feng-Ju Chang, Yen-Yu Lin, and Kuang-Jui Hsu. 2014. Multiple structured-instance learning for semantic segmentation with uncertain training data. In *Proc. Conf. Computer Vision and Pattern Recognition*.

Kai-Yueh Chang, Tyng-Luh Liu, and Shang-Hong Lai. 2009. Learning partially-observed hidden conditional random fields for facial expression recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*. 533–540.

Chia-Chih Chen and J. K. Aggarwal. 2011. Modeling human activities as speech. In *Proc. Conf. Computer Vision and Pattern Recognition*. 3425–3432.

Hongzhao Chen, Guijin Wang, and Li He. 2013. Accurate and real-time human action recognition based on 3D skeleton. In *Proc. Int'l. Conf. Optical Instruments and Technology*.

Zhuo Chen, Lu Wang, and Nelson H. C. Yung. 2011. Adaptive human motion analysis and prediction. *Pattern Recognition* 44, 12 (2011), 2902–2914.

Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Structured feature learning for pose estimation. In *Proc. Conf. Computer Vision and Pattern Recognition*.

James W. Davis and Ambrish Tyagi. 2006. Minimal-latency human action recognition using reliable-inference. *Image Vis. Comput.* 24, 5 (2006), 455–472.

Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Meroua Daoudi, and Alberto Del Bimbo. 2015. 3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold. *IEEE Trans. Cybernet.* 45, 7 (2015), 1340–1352.

Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. 2005. Behavior recognition via sparse spatio-temporal features. In *Proc. Int'l. Workshops on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. 65–72.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2625–2634.

Li Fei-Fei and Pietro Perona. 2005. A Bayesian hierarchical model for learning natural scene categories. In *Proc. Conf. Computer Vision and Pattern Recognition*, Vol. 2. 524–531.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional two-stream network fusion for video action recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*.

Thomas Finley and Thorsten Joachims. 2008. Training structural SVMs when exact inference is intractable. In *Proc. Int'l Conf. Machine Learning*.

Simon Fothergill, Helena M. Mentis, Pushmeet Kohli, and Sebastian Nowozin. 2012. Instructing people for training gestural interactive systems. In *Proc. Int'l. Conf. Human Factors in Computing Systems*. 1737–1746.

Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. 2015. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2568–2577.

Mohammad A. Gowayyed, Marwan Torki, Mohamed E. Hussein, and Motaz El-Saban. 2013. Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition. In *Proc. Int'l. Joint Conf. Artificial Intelligence*. 1351–1357.

Minh Hoai and Fernando De la Torre. 2014. Max-margin early event detectors. *Int. J. Comput. Vis.* 107, 2 (2014), 191–202.

Alexandros Iosifidis, Anastasios Tefas, and Ioannis Pitas. 2013. Dynamic action classification based on iterative data selection and feedforward neural networks. In *Proc. Euro. Conf. Signal Processing*. 1–5.

Yun Jiang and Ashutosh Saxena. 2014. Modeling high-dimensional humans for activity anticipation using Gaussian process latent CRFs. In *Robotics: Science and Systems*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems*. 1097–1105.

Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. 2014. A hierarchical representation for future action prediction. In *Proc. Euro. Conf. Computer Vision*. 689–704.

Ivan Laptev. 2005. On space-time interest points. *Int. J. Comput. Vis.* 64, 2–3 (2005), 107–123.

Chuanjun Li, S. Q. Zheng, and B. Prabhakaran. 2007. Segmentation and recognition of motion streams by similarity search. *ACM Trans. Multimedia Comput., Commun., Appl.* 3, 3 (2007), 16.

Kang Li and Yun Fu. 2014. Prediction of human activity by discovering temporal sequence patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 8 (2014), 1644–1657.

Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. 2016. Action recognition by learning deep multi-granular spatio-temporal video representation. In *Proc. ACM Conf. Multimedia Retrieval*. 159–166.

Wanqing Li, Zhengyou Zhang, and Zicheng Liu. 2008. Expandable data-driven graphical modeling of human actions based on salient postures. *IEEE Trans. Circuits Syst. Video Technol.* 18, 11 (2008), 1499–1510.

Xiao Li, Min Fang, Ju-Jie Zhang, and Jinqiao Wu. 2017. Learning coupled classifiers with RGB images for RGB-D object recognition. *Pattern Recognition* 61 (2017), 433–446.

Shih-Yao Lin, Yen-Yu Lin, Chu-Song Chen, and Yi-Ping Hung. 2017. Learning and inferring human actions with temporal pyramid features based on conditional random fields. In *Proc. Int'l. Conf. Acoustics, Speech, and Signal Processing*.

Yen-Yu Lin, Ju-Hsuan Hua, Nick C. Tang, Min-Hung Chen, and Hong-Yuan Mark Liao. 2014. Depth and skeleton associated action recognition without online accessible RGB-D cameras. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2617–2624.

Li Liu, Ling Shao, Xuelong Li, and Ke Lu. 2016a. Learning spatio-temporal representations for action recognition: A genetic programming approach. *IEEE Trans. Cybernetics* 46, 1 (2016), 158–170.

Li Liu, Yi Zhou, and Ling Shao. 2016b. DAP3D-Net: Where, what and how actions occur in videos? *arXiv Preprint arXiv:1602.03346* (2016).

Fengjun Lv and Ramakant Nevatia. 2006. Recognition and segmentation of 3-D human action using HMM and multi-class adaboost. In *Proc. Euro. Conf. Computer Vision*. 359–372.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. Berkeley Symp. Mathematical Statistics and Probability*, Vol. 1. 281–297.

Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. 2011. Action recognition from a distributed representation of pose and appearance. In *Proc. Conf. Computer Vision and Pattern Recognition*. 3177–3184.

James Martens and Ilya Sutskever. 2011. Learning recurrent neural networks with Hessian-free optimization. In *Proc. Int'l. Conf. Machine Learning*. 1033–1040.

Omar Oreifej and Zicheng Liu. 2013. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *Proc. Conf. Computer Vision and Pattern Recognition*. 716–723.

Olusegun Oshin, Andrew Gilbert, and Richard Bowden. 2011. Capturing the relative distribution of features for action recognition. In *Proc. Conf. Automatic Face and Gesture Recognition*. 111–116.

Jian Peng, Liefeng Bo, and Jinbo Xu. 2009. Conditional neural fields. In *Proc. Advances in Neural Information Processing Systems*. 1419–1427.

Lasitha Piyathilaka and Sarath Kodagoda. 2013. Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features. In *Proc. Int'l. Conf. Industrial Electronics and Applications*. 567–572.

Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. 2007. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.* (2007), 1848–1852.

Michalis Raptis and Leonid Sigal. 2013. Poselet key-framing: A model for human activity recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2650–2657.

M. S. Ryoo. 2011. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proc. Int'l. Conf. Computer Vision*. 1036–1043.

M. S. Ryoo and J. K. Aggarwal. 2010. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). (2010).

Konrad Schindler and Luc Van Gool. 2008. Action snippets: How many frames does human action recognition require?. In *Proc. Conf. Computer Vision and Pattern Recognition*. 1–8.

Wei Shen, Ke Deng, Xiang Bai, Tommer Leyvand, Baining Guo, and Zhuowen Tu. 2012. Exemplar-based human action pose correction and tagging. In *Proc. Conf. Computer Vision and Pattern Recognition*. 1784–1791.

Ya-Fang Shih, Yang-Ming Yeh, Yen-Yu Lin, Ming-Feng Weng, Yi-Chang Lu, and Yung-Yu Chuang. 2017. Deep co-occurrence feature learning for visual object recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*.

Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. 2012. Part-based multiple-person tracking with partial occlusion handling. In *Proc. Conf. Computer Vision and Pattern Recognition*. 1815–1821.

Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Proc. Advances in Neural Information Processing Systems*. 568–576.

Yale Song, Louis-Philippe Morency, and Randall Davis. 2012. Multi-view latent variable discriminative models for action recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2120–2127.

Yale Song, Louis-Philippe Morency, and Ronald W. Davis. 2013. Action recognition by hierarchical sequence summarization. In *Proc. Conf. Computer Vision and Pattern Recognition*. 3562–3569.

Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. 2012. Unstructured human activity detection from RGBD images. In *Proc. Int'l. Conf. Robotics and Automation*. 842–849.

C. Sutton and A. McCallum. 2007. *An Introduction to Conditional Random Fields for Relational Learning*. MIT Press.

Nick C. Tang, Yen-Yu Lin, Ju-Hsuan Hua, Shih-En Wei, Ming-Fang Weng, and Hong-Yuan Mark Liao. 2015. Robust action recognition via borrowing information across video modalities. *IEEE Trans. Image Process.* 24, 2 (2015), 709–723.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3D convolutional networks. In *Proc. Int'l. Conf. Computer Vision*. 4489–4497.

Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human action recognition by representing 3D skeletons as points in a lie group. In *Proc. Conf. Computer Vision and Pattern Recognition*. 588–595.

Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2012. Mining actionlet ensemble for action recognition with depth cameras. In *Proc. Conf. Computer Vision and Pattern Recognition*. 1290–1297.

Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2014. Learning actionlet ensemble for 3D human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 5 (2014), 914–927.

Liang Wang and David Suter. 2007. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *Proc. Conf. Computer Vision and Pattern Recognition*. 1–8.

Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. 2009. An HOG-LBP human detector with partial occlusion handling. In *Proc. Int'l. Conf. Computer Vision*. 32–39.

Daniel Weinland, Mustafa Özuysal, and Pascal Fua. 2010. Making action recognition robust to occlusions and viewpoint changes. In *Proc. Euro. Conf. Computer Vision*. 635–648.

Lu Xia and J. K. Aggarwal. 2013. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proc. Conf. Computer Vision and Pattern Recognition*. 2834–2841.

Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. 2016. Learning deep feature representations with domain guided dropout for person re-identification. *arXiv Preprint arXiv:1604.07528* (2016).

Gang Yu, Junsong Yuan, and Zicheng Liu. 2012. Predicting human activities using spatio-temporal structure of interest points. In *Proc. ACM Conf. Multimedia*. 1049–1052.

Gang Yu, Junsong Yuan, and Zicheng Liu. 2015. Propagative Hough voting for human activity detection and recognition. *IEEE Trans. Circ. Syst. Video Technol.* 25, 1 (2015), 87–98.

Bo Zhang, Nicola Conci, and Francesco G. B. De Natale. 2015. Segmentation of discriminative patches in human activity video. *ACM Trans. Multimedia Comput., Commun., Appl.* 12, 1 (2015), 4.

Jianguo Zhang and Shaogang Gong. 2010. Action categorization with modified hidden conditional random field. *Pattern Recognit.* 43, 1 (2010), 197–203.

Lei Zhang, Zhi Zeng, and Qiang Ji. 2011. Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation. *IEEE Trans. Image Process.* 20, 9 (2011), 2401–2413.

Xin Zhao, Xue Li, Chaoyi Pang, Quan Z. Sheng, Sen Wang, and Mao Ye. 2014. Structured streaming skeleton—A new feature for online human gesture recognition. *ACM Trans. Multimedia Comput., Commun., Appl.* 11, 1s (2014), 22.

Received September 2016; revised April 2017; accepted April 2017