Recognizing Human Activities from Partially Observed Videos

Yu Cao[1], Daniel Barrett[2], Andrei Barbu[2], Siddharth Narayanaswamy[2], Haonan Yu[2], Aaron Michaux[2]
Yuewei Lin[1], Sven Dickinson[3], Jeffrey Mark Siskind[2], Song Wang[1*]

[1] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA

[2] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

[3] Department of Computer Science, University of Toronto, Toronto ON, Canada

Abstract

Recognizing human activities in partially observed videos is a challenging problem and has many practical applications. When the unobserved subsequence is at the end of the video, the problem is reduced to activity prediction from unfinished activity streaming, which has been studied by many researchers. However, in the general case, an unobserved subsequence may occur at any time by yielding a temporal gap in the video. In this paper, we propose a new method that can recognize human activities from partially observed videos in the general case. Specifically, we formulate the problem into a probabilistic framework: 1) dividing each activity into multiple ordered temporal segments, 2) using spatiotemporal features of the training video samples in each segment as bases and applying sparse coding (SC) to derive the activity likelihood of the test video sample at each segment, and 3) finally combining the likelihood at each segment to achieve a global posterior for the activities. We further extend the proposed method to include more bases that correspond to a mixture of segments with different temporal lengths (MSSC), which can better represent the activities with large intra-class variations. We evaluate the proposed methods (SC and MSSC) on various real videos. We also evaluate the proposed methods on two special cases: 1) activity prediction where the unobserved subsequence is at the end of the video, and 2) human activity recognition on fully observed videos. Experimental results show that the proposed methods outperform existing state-of-the-art comparison methods.

1. Introduction

Human activity recognition aims at building robust and efficient computer vision algorithms and systems which can automatically recognize specific human activities from a sequence of video frames. Its applications include security, surveillance and human-computer interaction, etc. Early research on this problem [1, 5, 13, 11, 18] focused on a single person's simple actions, such as walking, running, and hopping. Recently, research on activity recognition has been extended to more complex activity scenarios which involve multiple persons interacting with each other or objects [15, 20, 24].

One widely used approach for human activity recognition is to train and classify the spatiotemporal features extracted from videos with different activities. Inspired by successful 2D scale-invariant image feature descriptors [12, 3], a variety of spatiotemporal feature detectors/descriptors have been developed [11, 5, 7, 10, 21, 22], and their robustness and effectiveness have been demonstrated in several successful human activity recognition methods [5, 13, 11, 18, 15, 14, 20, 9, 19]. In these methods, a sequence of 2D video frames are treated as a 3D XYT video volume in which interest points are located by finding local maxima in the responses of the feature detector, followed by calculating vectorized feature descriptors at each interest point. By using the bag-of-visual-words technique, spatiotemporal features within a video can be combined into a feature vector that describes the activity presented in the video.

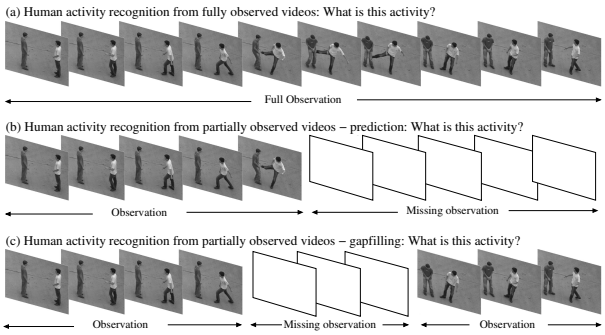


Figure 1. An illustration of the human activity recognition from fully and partially observed videos.

Previous research on human activity recognition usually focused on recognizing activities after fully observing the entire video, as illustrated in Fig. 1(a). However in prac-

*Corresponding author. Email: songwang@cec.sc.edu.

tice, partially observed videos may occur when video signals drop off, cameras or objects of interest are occluded, or videos are composited from multiple sources. The unobserved subsequence may occur any time with any duration, yielding a temporal gap as shown in Fig. 1(c). Recognizing activities in such temporal gaps is of particular importance in defense and security. For example, one of the four major themes in the DARPA Mind's Eye program[1] is to handle such gapped videos for activity recognition.

When the unobserved subsequence is at the end of the video, the problem is reduced to activity prediction from unfinished activity streaming, as illustrated in Fig. 1(b). Activity prediction has been studied by Ryoo [14]. Another example of related work on activity prediction is the max-margin early event detectors (MMED) [6], which try to detect the temporal location and duration of a certain activity from the video streaming. Recently, Kitani *et al*. studied a special activity prediction problem in [8], which tries to predict the walking path of a person in certain environments based on historical data. However, activity recognition from general gapped videos, as shown in Fig. 1(c), has not been well studied yet. Note that, in general, this is a different problem from activity prediction because the temporal gap may divide the video into two disjoint observed video subsequences and we need to combine them to achieve a reliable recognition.

In this paper, we propose a probabilistic formulation for human activity recognition from partially observed videos, where the posterior is maximized for the recognized activity class and the observed video frames. In our formulation, the key component in defining the posterior is the likelihood that the observed video frames describe a certain class of activity. In this paper, we take a set of training video samples (completely observed) of each activity class as the bases, and then use sparse coding (SC) to derive the likelihood that a certain type of activity is presented in a partially observed test video. Furthermore, we divide each activity into multiple temporal segments, apply sparse coding to derive the activity likelihood at each segment, and finally combine the likelihoods at each segment to achieve a global posterior for the activity. While video segments are constructed by uniformly dividing the video in SC, we also extend it to include more sparse coding bases constructed from a mixture of training video segments (MSSC) with different lengths and locations.

Using sparse coding with the constructed bases, the proposed methods can find closest video segments from different training videos when matching a new test video. Thus, the proposed methods don't require full temporal alignment between any pair of (training or test) videos, and they can handle the problems of 1) a limited number of training videos; 2) possible outliers in the training video data; and 3) large intra-class variations. We evaluate the proposed methods on several video datasets and compare their performance with several state-of-the-art methods. In the experiments, we not only evaluate the performance on general gapped videos, but also on fully observed videos without a gap and videos with a gap at the end (activity prediction).

The remainder of the paper is organized as follows. In Section 2, we present our probabilistic formulation of human activity recognition from partially observed videos. Section 3 introduces the likelihood component using a sparse coding (SC) technique followed by extending the SC to include more bases constructed from a mixture of segments (MSSC) with different temporal lengths. Experimental results and discussions are presented in Section 4, followed by conclusions in Section 5.

2. Problem Formulation

2.1. Human Activity Recognition from a Fully Observed Video

Given a fully observed video $\mathcal{O}[1 : T]$ of length T, where $\mathcal{O}[t]$ indicates the frame at time t, the goal is to classify the video $\mathcal{O}[1 : T]$ into one of P activity classes $\mathcal{A} = \{\mathcal{A}_p\}, p = 1, \ldots, \mathcal{P}$. A human activity is usually made up of a sequence of simpler actions, each of which may contain different spatiotemporal features. Therefore, we can divide the video $\mathcal{O}[1 : T]$ into a sequence of shorter video *segments* for spatiotemporal feature extraction. For simplicity, we uniformly divide the video $\mathcal{O}[1 : T]$ into M equal-length segments, where each segment $\mathcal{O}(t_{i-1} : t_i]$, with $t_i = \frac{iT}{M}$, corresponds to the i-th stage of the activity, with $i = 1, 2, \cdots, M$. For different videos, the length T might be different, and therefore the segments from different videos may have different lengths.

The posterior probability that an activity $\mathcal{A}_p$ is presented in the video $\mathcal{O}[1 : T]$ can be defined as $P(\mathcal{A}_p|\mathcal{O}[1 : T])$, which can be rewritten as:

$$\begin{aligned} P(\mathcal{A}_p|\mathcal{O}[1:T]) &\propto \sum_{i=1}^{M} P(\mathcal{A}_p, (t_{i-1}:t_i]|\mathcal{O}[1:T]) \\ &\propto \sum_{i=1}^{M} P(\mathcal{A}_p, (t_{i-1}:t_i]) P(\mathcal{O}[1:T]|\mathcal{A}_p, (t_{i-1}:t_i]). \end{aligned} \tag{1}$$

In this formulation, $P(\mathcal{A}_p, (t_{i-1} : t_i])$ is the prior of stage i of activity $\mathcal{A}_p$ and $P(\mathcal{O}[1 : T]|\mathcal{A}_p, (t_{i-1} : t_i])$ is the observation likelihood given activity class $\mathcal{A}_p$ in the i-th stage. Then the index of the recognized activity is

$$\begin{aligned} p^* = \arg\max_p \sum_{i=1}^{M} & P(\mathcal{A}_p, (t_{i-1}:t_i]) \cdot \\ & P(\mathcal{O}[1:T]|\mathcal{A}_p, (t_{i-1}:t_i]). \end{aligned} \tag{2}$$

[1] http://www.darpa.mil/Our_Work/I2O/Programs/Minds_Eye.aspx

2.2. Human Activity Recognition from a Partially Observed Video

A partially observed video can be represented by $\mathcal{O}[1:T_1] \cup [T_2:T]$, where frames $\mathcal{O}(T_1:T_2)$ are missing, as illustrated in Fig. 2. For simplicity, we assume that T_1 is always the last frame of a segment and T_2 is always the first frame of another segment. Otherwise, we can intentionally decrease T_1 to a nearest last frame of a segment and increase T_2 to a nearest first frame of a segment.

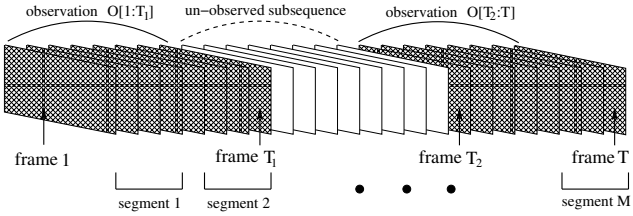


Figure 2. An illustration of a partially observed video (general case), where the unobserved subsequence is located in the middle of the video.

By following the formulation in Eqn. (1), the posterior probability that an activity $\mathcal{A}_p$ is presented in this partially observed video can be defined as:

$$\begin{aligned} P(\mathcal{A}_p|\mathcal{O}[1:T_1] \cup \mathcal{O}[T_2:T]) \propto \\ \omega_1 \sum_{i|t_i \le T_1} P(\mathcal{A}_p,(t_{i-1}:t_i]|\mathcal{O}[1:T_1]) \\ + \omega_2 \sum_{i|t_{i-1} \ge T_2} P(\mathcal{A}_p,(t_{i-1}:t_i]|\mathcal{O}[T_2:T]), \end{aligned} \tag{3}$$

where $\omega_1 = \frac{T_1}{T_1+T-T_2+1}$ and $\omega_2 = \frac{T-T_2+1}{T_1+T-T_2+1}$ reflect the proportionality between the length of $\mathcal{O}[1:T_1]$ and $\mathcal{O}[T_2:T]$. We can rewrite this as:

$$\begin{aligned} &P(\mathcal{A}_p|\mathcal{O}[1:T_1] \cup \mathcal{O}[T_2:T]) \propto \\ &\omega_1 \sum_{i|t_i \le T_1} P(\mathcal{A}_p,(t_{i-1}:t_i])P(\mathcal{O}[1:T_1]|\mathcal{A}_p,(t_{i-1}:t_i]) \\ &+ \omega_2 \sum_{i|t_{i-1} \ge T_2} P(\mathcal{A}_p,(t_{i-1}:t_i])P(\mathcal{O}[T_2:T]|\mathcal{A}_p,(t_{i-1}:t_i]). \end{aligned} \tag{4}$$

The index of the recognized activity is therefore:

$$p^* = \arg\max_p P(\mathcal{A}_p|\mathcal{O}[1:T_1] \cup \mathcal{O}[T_2:T]). \tag{5}$$

Notably, when $T_2 - 1 = T$ the problem is reduced to its special case – activity prediction. When $T_1 = T$, the problem is degenerated to the classic human activity recognition from fully observed videos. In practice, we can assume that the prior of $\mathcal{A}_p$ on each segment satisfies a uniform distribution, without favoring any special activity. We introduce the calculation of the likelihood component in the following section.

3. Likelihood

3.1. Likelihood calculation using sparse coding

Without loss of generality, in this section, we only consider the calculation of the likelihood $P(\mathcal{O}[1:T_1]|\mathcal{A}_p,(t_{i-1}:t_i])$, since $P(\mathcal{O}[T_2:T]|\mathcal{A}_p,(t_{i-1}:t_i])$ can be calculated in a similar way. The basic idea is to collect a set of training videos (completely observed) for activity class $\mathcal{A}_p$ and then define the likelihood $P(\mathcal{O}[1:T_1]|\mathcal{A}_p,(t_{i-1}:t_i])$ by comparing $\mathcal{O}[1:T_1]$ with the i-th segment of all the training videos. For each segment of a video, we use the bag-of-visual-words technique to organize its spatiotemporal features into a fixed-dimensional feature vector. For the i-th segment of the n-th training video, we denote its feature (row) vector, after applying the bag-of-visual-words technique, as $\mathbf{h}_i^n$. For the test video $\mathcal{O}[1:T_1]$, we also extract such a feature for stage i to be $\mathbf{h}_i^{\mathcal{O}}$.

One intuitive way to define the likelihood $P(\mathcal{O}[1:T_1]|\mathcal{A}_p,(t_{i-1}:t_i])$ is to first construct a mean feature vector, as used in [14], $\bar{\mathbf{h}}_i = \frac{1}{N}\sum_{n=1}^{N}\mathbf{h}_i^n$ for the i-th stage over all N training videos in class $\mathcal{A}_p$. Then, the likelihood $P(\mathcal{O}[1:T_1]|\mathcal{A}_p,(t_{i-1}:t_i])$ can be defined as:

$$P(\mathcal{O}[1:T_1]|\mathcal{A}_p,(t_{i-1}:t_i]) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-\|\mathbf{h}_i^{\mathcal{O}}-\bar{\mathbf{h}}_i\|^2}{2\sigma^2}}, \tag{6}$$

where $\|\mathbf{h}_i^{\mathcal{O}} - \bar{\mathbf{h}}_i\|$ represents the distance between the observed feature and the mean feature from the training video dataset in stage i.

However, simply using the mean feature vector as the unique activity model may suffer from two practical limitations. First, when the number of training videos is limited, the mean feature may not be a representative of the true 'center' of its activity class in feature space. Second, when outliers are accidentally included in the training dataset, e.g. the activity label for a training video is actually incorrect or one training video shows a large difference from the other training videos in the same activity class, the mean feature vector may not well represent the considered activity.

To alleviate these limitations, we propose to take feature vectors from training data as bases, with which we can use sparse coding to approximate features extracted from the testing (partially observed) video. The reconstruction error from sparse coding is used to replace the feature distance in Eqn. (6) for likelihood calculation.

Specifically, for segment i, we construct the bases matrix A_i using the segment-i feature vectors from N training videos:

$$A_i = \begin{pmatrix} \mathbf{h}_i^1 \\ \mathbf{h}_i^2 \\ \cdots \\ \mathbf{h}_i^N \end{pmatrix}. \tag{7}$$

Then the reconstructed sparse representation of $\mathbf{h}_i^{\mathcal{O}}$ can be written as $\tilde{\mathbf{h}}_i^{\mathcal{O}} = A_i\mathbf{x}^*$, where $\mathbf{x}^*$ is the linear combination coefficients of the sparse coding representation which can be derived by solving the following minimization problem:

$$\mathbf{x}^* = \min_{\mathbf{x}} \|\mathbf{h}_i^{\mathcal{O}} - \tilde{\mathbf{h}}_i^{\mathcal{O}}\|^2 + \lambda\|\mathbf{x}\|_0. \quad (8)$$

This minimization problem can be approximated by replacing the term $\|\mathbf{x}\|_0$ with $\|\mathbf{x}\|_1$ and then solved by L^1 minimization toolboxes such as [23, 2]. In this paper, we choose toolbox [23]. In particular, we use its Orthogonal Matching Pursuit (OMP) implementation. The original likelihood equation Eqn. (6) can then be rewritten as:

$$P(\mathcal{O}[1:T_1]|\mathcal{A}_p, (t_{i-1}:t_i]) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-\|\mathbf{h}_i^{\mathcal{O}} - A_i\mathbf{x}^*\|^2}{2\sigma^2}}. \quad (9)$$

By using the sparse coding as described above, the proposed SC method can automatically select a proper subset of bases for approximating the test video segments. This way, it can exclude outliers in the training set for likelihood calculation. In addition, for different video segments in the test video, the proposed SC method can identify segments from different training videos and use their linear combination for representing a test video segment. Compared to the mean activity model, the proposed method can substantially decrease the approximation error and to some extent, alleviate the problem of limited number of training videos. Note that, in the proposed method, different test videos and different segments from a test video will identify different bases and different coefficients for likelihood calculation. This is different from the support vector machine (SVM) classifiers where the support vectors (analogue to selected bases in SC) are fixed when the training is finished. In the experiments, we will show that the proposed SC method outperforms MMED [6], a structured SVM based method, on the activity prediction task.

3.2. Likelihood calculation using sparse coding on a mixture of segments

It is well known that, in practice, humans perform activities with different paces and overhead time. These phenomena introduce temporal intra-class variations. To handle such variations, we further extend SC to MSSC by including more bases that are constructed from a mixture of segments with different temporal lengths and temporal locations in the training videos.

More specifically, when calculating the likelihood of segment i of the test video, we not only take segment $(t_{i-1}, t_i]$ in the training video to construct a basis, but also take 8 segments in each training video to construct 8 more bases. As illustrated in Fig. 3, we take the j-th training video as an example. These 8 more segments are $(t_{i-2}, t_{i-1}]$, $(t_i, t_{i+1}]$, $(t_{i-2}, t_i]$, $(t_{i-1}, t_{i+1}]$, $(t_{i-2}, t_i]$, $(t_{i-1}, t_{i-1} + (t_i - t_{i-1})/2]$, $(t_{i-1} + (t_i - t_{i-1})/2, t_i]$, and $(t_{i-1} + (t_i - t_{i-1})/4, t_i - (t_i - t_{i-1})/4]$, which are all around segment $(t_{i-1}, t_i]$, but with varied segment lengths and small temporal location shifts. We expect that these additional bases can better handle the intra-class activity variation by approximating the test video segments more accurately.

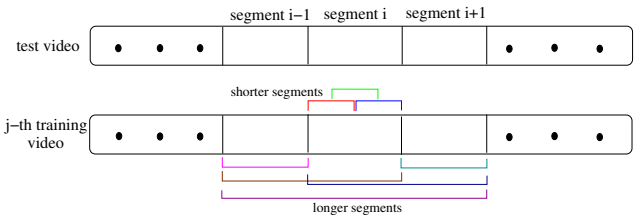


Figure 3. An illustration of a mixture of segments with different temporal lengths and shifted temporal locations.

4. Experiments

We test the proposed SC and MSSC methods on three human activity recognition tasks: 1) the special case – activity *prediction*, where the video gap is at the end of the video; 2) the general case – *gapfilling*, where the gap separates two observed subsequences; and 3) the degenerate case – full video *recognition*, where there is no gap along the video. Each task is evaluated on real datasets with different challenge levels. We implement the proposed methods in MATLAB, and use the Cuboids descriptors [5] as spatiotemporal features. We use the bag-of-visual-words technique to organize spatiotemporal features, in which a codebook with 800 words is generated by K-means clustering. In experiments, we consistently set $M = 20$, i.e., each activity (and each video) is uniformly divided into 20 temporal segments. We set $\sigma = 6$ (in Eqn. (9)) and $\lambda = 10^{-2}$ (in Eqn. (8)) for the proposed methods throughout the three recognition tasks.

We choose several state-of-the-art comparison methods, including Ryoo's human activity prediction methods (both non-dynamic and dynamic versions) [14], early event detector – MMED [6], C2 [7], and Action Bank [17]. Based on their applicability, we apply these methods (adapted if necessary) on all or a part of the three recognition tasks, which will be further explained in following sections. We also implement a baseline sparse coding (named after '*baseline*') method which concatenates features from different segments of a training video into a single feature vector as one row of the basis matrix and then directly applies sparse coding for recognition. More specifically, a (partially observed) test video is classified into an activity class that leads to the smallest reconstruction error. Furthermore, in order to clarify that the proposed SC and MSSC methods can perform better than voting methods which share a similar idea of using a subset of training samples for classification, we implement a KNN (K Nearest Neighbor) algorithm

as another comparison method. Specifically, against each training video, we apply Ryoo's methods (non-dynamic and dynamic versions) to calculate the posterior of the test (partially observed) video and then identify the K (K is the number of training videos in the same activity class) training videos against which the test video has largest posteriors. This way, we can apply a simple majority voting to the activity classes of K nearest training videos to classify the test video.

4.1. Evaluation on the special case: prediction

In the prediction task, we simulate incremental arrival of video frames (represented by observation ratio $[0.1, 0.2, \ldots, 1.0]$) as in [14], and evaluate the performance for each observation ratio. Ryoo's methods, the KNN methods, MMED and the baseline sparse coding method are selected for comparison since they can handle prediction.

For Ryoo's methods, since the original codes are not available publicly, we implement them by following [14]. And by tuning parameters in our implementation, we actually achieve comparable or even better performance than those reported in [14]. For MMED method, we use its published code and follow the settings in [6]. When recognizing a test video $\mathcal{O}$, it is concatenated with other test videos from other activity classes into a long video, with $\mathcal{O}$ at the end. MMED returns a subsequence in this long video. To adapt MMED from early event detection to human activity prediction, we set its minimum searching length to be the temporal length of $\mathcal{O}$, and the step length of the searching to be identical to the segment length in the proposed SC method. If the subsequence returned by MMED contains no less than 50% of the frames in the observed subsequence in $\mathcal{O}$, it is counted as a correct prediction.

We have three datasets for evaluating prediction: *UT-interaction #1*, *UT-interaction #2* [16] and *DARPA Y1*, a subset of videos from the Year-1 corpus of the DARPA Mind's Eye program [4]. In DARPA Y1, each video shows one of the 7 human activities: 'fall', 'haul', 'hit', 'jump', 'kick', 'push' and 'turn'. For each activity class, we collect 20 videos. DARPA Y1 is much more complex than the UT-interaction datasets in that 1) actor size in the same activity class varies significantly in different videos; 2) the overhead time for an activity varies from one video to another; 3) activities are recorded from different camera perspectives; 4) activity pace varies in different videos; and 5) backgrounds are more complex due to shadows and non-uniform illuminations.

As in [14], we use the leave-one-out cross validation for performance evaluation. There are 10 folds of cross validations on UT-interaction #1, #2; 20 folds of cross validations on DARPA Y1. For each test video, the result is a single human activity class out of all possible activities classes and we use the average accuracy over all cross validation tests and all activity classes as a quantitative metric of the performance.

Figure 4 shows the prediction results on these three test datasets. We can see that, the proposed SC and MSSC methods show comparable or better performance on these three datasets, especially on DARPA Y1 and when the observation ratio is not overly small. The baseline sparse coding method achieves good performance (close to SC and MSSC) on UT-interaction #1, #2 but not in DARPA Y1 because the activities in UT-interaction datasets show very small intra-class variations, while the proposed methods can better handle the large intra-class variations. MMED performs not as good as other methods because MMED is originally designed for early event detection, with a goal of localizing the starting and ending frames of an activity and this is different from the prediction task in this experiment, where our goal is to recognize the activity from a given observed subsequence.

4.2. Evaluation on the general case: gapfilling

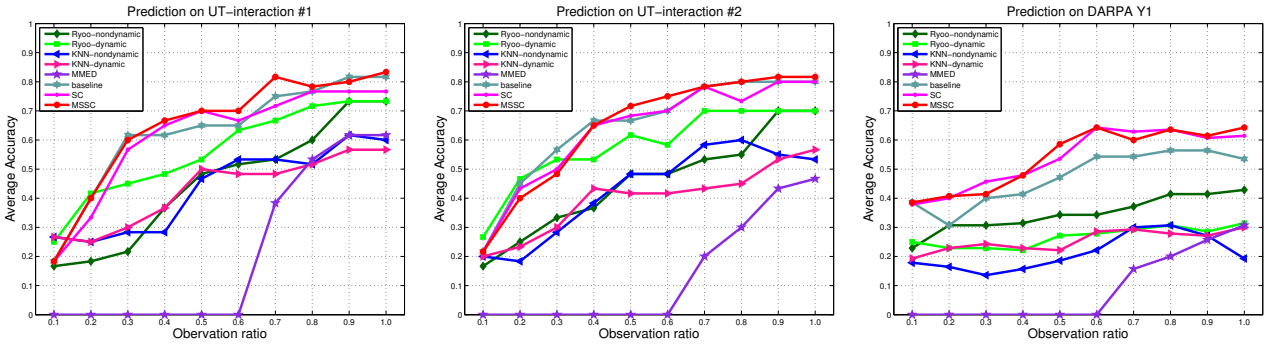
In the gapfilling task, we compare the proposed SC and MSSC methods with adapted Ryoo's methods (non-dynamic and dynamic versions), KNN (non-dynamic and dynamic versions) and the baseline sparse coding method. Specifically, we adapt Ryoo's methods to perform activity prediction on these two subsequences, and the gapfilling posterior score is the summation of prediction posterior scores on each observed subsequence. We use the same parameter setting for adapted Ryoo's methods as used in the prediction task.

We first perform gapfilling evaluation on UT-interaction #1,#2 and DARPA Y1 datasets. We intentionally replace a subsequence of frames from a test video by empty frames to create a partially observed video. To have a more reliable performance evaluation, we try different lengths and different temporal locations of empty subsequences for each test video. Specifically, for each test video $\mathcal{O}[1:T]$, we construct a non-observation interval $(\beta_1 T : \beta_2 T]$, where

$$(\beta_1, \beta_2) = \{[0.1, 0.2, \cdots, 0.9] \times [0.2, 0.3, \cdots, 0.9]\}, \tag{10}$$

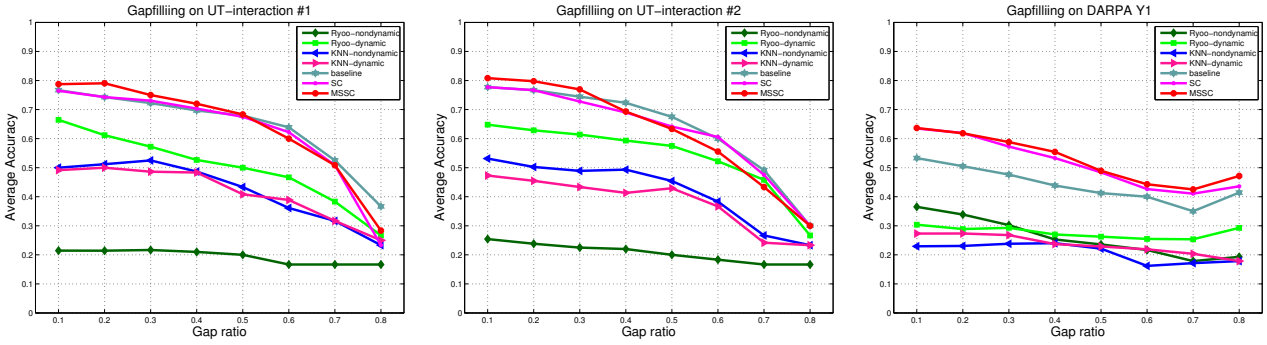
and $\beta_1 < \beta_2$. We further define non-observation ratio as $\hat{\beta} = \beta_2 - \beta_1$ which varies from 0.1 to 0.8 with the step of 0.1 according to Eqn. (10). We finally evaluate the accuracy rate in term of each possible non-observation ratio $\hat{\beta}$ by counting the percentage of the correctly recognized test videos with the non-observation ratio $\hat{\beta}$ over all folds (10 folds for UT-interaction #1, #2; 20 folds for DARPA Y1) of cross validations.

As shown in Fig. 5, we achieve a similar performance ranking as in the prediction evaluations. The proposed SC and MSSC methods achieve comparable or better performance when the gap ratio is not overly large. On the more



(a) (b) (c)

Figure 4. Prediction results on three datasets: (a) UT-interaction #1; (b) UT-interaction #2; and (c) DARPA Y1.



(a) (b) (c)

Figure 5. Gapfilling results on three datasets: (a) UT-Interaction #1; (b) UT-interaction #2; and (c) DARPA Y1.

complex DARPA Y1 dataset, the proposed methods clearly achieve better performance.

We further evaluate the proposed methods and comparison methods on more complex datasets: The DARPA Mind's Eye program provides a Year-2 evaluation corpus, which contains 4 sub-datasets of long test videos (each video has a length more than ten minutes) with multiple gaps. For brevity, we name this dataset as *DARPA Y2-Gapfilling* and the four sub-datasets are 'gapfilling short duration', 'gapfilling long duration','gapfilling natural description', and 'gapfilling natural recognition', respectively. This dataset is much more challenging than DARPA Y1 dataset due to: 1) the important action units are missing for the underlying activities in many cases; and 2) many activities are performed simultaneously by multiple actors. In DARPA Y2-Gapfilling, there are in total 267 gaps, with length from 122 to $2,239$ frames.

DARPA Mind's Eye program also provides three training sets (different from DARPA Y2-Gapfilling) to learn the model for each activity. These three training sets are 'C-D2b' (22 activity classes, totally $3,819$ training videos), 'C-D2c' (16 activity classes, totally $2,80$ training videos) and 'C-D2bc' (23 activity classes, totally $4,409$ training videos). We perform the proposed and comparison methods on all videos in DARPA Y2-Gapfilling with respect to these three training datasets, respectively. In our experiment, for each gap in DARPA Y2-Gapfilling, we construct test video clips by including a certain number of observed frames before and/or after this gap. For each gap, nine video clips are constructed with the gap appearing 'at the beginning', 'in the center' or 'at the end' of the video clip and counting for 20%, 40% or 60% of the clip length. This way, we construct a total of $2,403$ video clips with a gap and evaluate the recognition results against the human annotated ground-truth (may give multiple activities labels for a test gapped video clip). Precision-recall results (obtained by thresholding the posteriors) are shown in Fig. 6. We can see that the proposed SC and MSSC methods outperform the comparison methods in most of the test clips. However, the general performance is low, which indicates that the gapfilling on practical scenarios is far from a solved problem.

4.3. Evaluation on degenerate case: full-video recognition

For full-video recognition, we compare the proposed SC and MSSC methods with the baseline sparse coding, Ryoo's methods (non-dynamic and dynamic versions), C2 and Action Bank. Previous published recognition methods are mostly evaluated on short video clips where each of them contains a single activity, which cannot reflect real

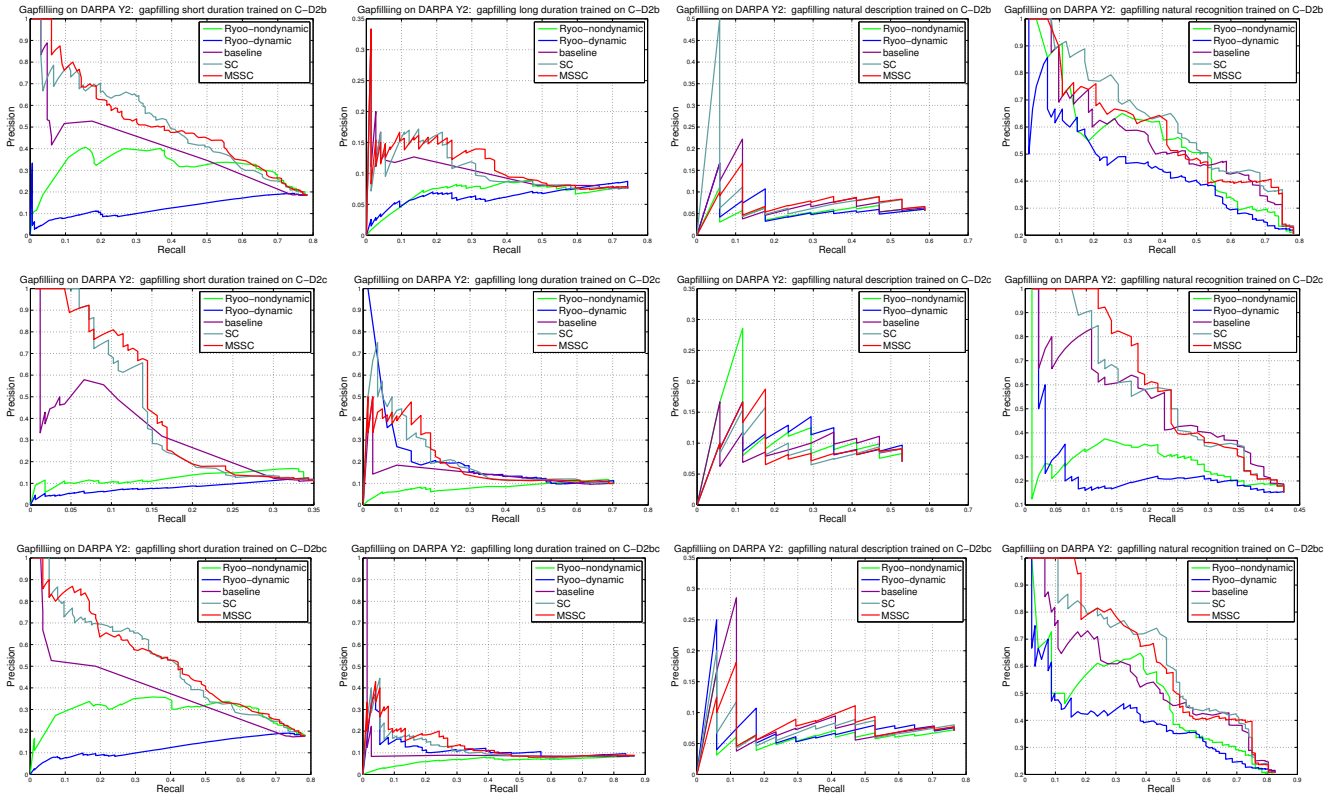


Figure 6. Gapfilling results on DARPA Y2-Gapfilling dataset. Row 1, 2 and 3 show the results by training on 'C-D2b', 'C-D2c' and 'C-D2bc', respectively. Column 1, 2, 3 and 4 show the results on 'gapfilling short duration', 'gapfilling long duration', 'gapfilling natural description', and 'gapfilling natural recognition', respectively.

performance in the practical scenarios. Thus, besides the full video recognition results on UT-interaction #1,#2 and DARPA Y1 provided in prediction task (see Fig. 4 at observation ratio 1.0), we further test these methods on *DARPA Year-2 Recognition*, a dataset provided by the DARPA Minds' Eye program for large-scale activity recognition evaluation. DARPA Year-2 Recognition contains 3 sub-datasets: 'description free form', 'description scripted', and 'recognition'. DARPA Y2-Recognition is very challenging because 1) the length of each video is around (mostly larger than) ten minutes (more than one and half hours in total); and 2) a large number of subsequences contain unrelated moving objects in the background.

We break the long video into partially overlapping short clips using sliding windows for activity recognition. For the proposed SC and MSSC methods, Ryoo's methods (non-dynamic and dynamic versions), and the baseline sparse coding method, we calculate the posteriors of each activity presented in each short clip. We normalize the posterior scores that an activity is present in each short clip and label the video clip with the activities that have posterior scores larger than a pre-set threshold τ. In the experiments, we choose $\tau = 0.05$ and C-D2b as the training set. For C2 and Action Bank methods, we use their default parameters to detect activities in each constructed video clip. We check the overlap between the sliding window with the recognized activities and the ground-truth labeling of the activities (starting and ending frames) using the intersection/union ratio. Given the identical activity label, we threshold this overlap ratio to get precision/recall values and then combine them into a F_1-measure, which is shown in Table 1. As shown in the table, the proposed SC, MSSC and baseline sparse coding methods achieve relatively better performance on two different ground-truth labelings. However, the general performance is very low and this indicates that there is still a long way to go to achieve good activity recognition in practical scenarios.

5. Conclusion

In this paper, we proposed novel methods for recognizing human activities from partially observed videos. We formulated the problem as a posterior-maximization problem whose likelihood is calculated on each activity temporal stage using a sparse coding (SC) technique. We further include more sparse coding bases for a mixture of varied-length and/or varied-location segments (MSSC) from the training videos. We evaluated the proposed SC and MSSC methods on three tasks: activity prediction, gapfilling and full-video recognition. The experimental results demonstrate that the proposed methods produce better perfor-

F_1-measures evaluated against ground-truth I												
Test datasets	'description free form'				'description scripted'				'recognition'			
Overlap thresholds	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7
Ryoo's dynamic	0.81%	0.68%	0.54%	0.33%	8.6%	8.6%	8.6%	8.6%	0.9%	0.67%	0.49%	0.28%
Ryoo's non-dynamic	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
baseline	**1.24%**	**1.11%**	**0.83%**	**0.53%**	15.45%	15%	14.09%	13.64%	**1.09%**	0.89%	0.72%	0.48%
MSSC	1.04%	0.92%	0.71%	0.46%	10.97%	10.97%	10.47%	9.98%	1%	0.84%	0.68%	0.45%
SC	1.23%	1.10%	0.82%	**0.53%**	**16.39%**	**15.85%**	**14.75%**	**14.21%**	**1.09%**	**0.90%**	**0.75%**	**0.5%**
C2	0.52%	0.26%	0.26%	0.26%	0%	0%	0%	0%	0%	0%	0%	0%
Action Bank	0.38%	0%	0%	0%	0.53%	0.18%	0.18%	0.18%	0.25%	0.25%	0.25%	0.25%
F_1-measures evaluated against ground-truth II												
Test datasets	'description free form'				'description scripted'				'recognition'			
Overlap thresholds	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7
Ryoo's dynamic	0.77%	0.66%	0.49%	0.26%	6.52%	4.35%	4.35%	4.35%	0.98%	0.74%	0.52%	0.29%
Ryoo's non-dynamic	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
baseline	0.93%	**0.8%**	**0.65%**	0.41%	6.83%	6.38%	5.92%	5.47%	**1.14%**	**0.99%**	**0.8%**	**0.51%**
MSSC	0.85%	0.72%	0.56%	0.37%	6%	5.5%	5.5%	5%	1.06%	0.93%	0.74%	0.46%
SC	0.91%	**0.8%**	**0.65%**	**0.42%**	**7.12%**	**6.58%**	**6.03%**	**5.48%**	1.13%	**0.99%**	**0.8%**	0.5%
C2	**0.97%**	0.32%	0%	0%	5.26%	5.26%	3.51%	1.75%	0%	0%	0%	0%
Action Bank	0.67%	0.67%	0.22%	0%	0.78%	0.59%	0.39%	0.2%	0.72%	0.48%	0.48%	0.24%

Table 1. Recognition results on DARPA Y2-Recognition dataset. The best F_1-measures on each test dataset at each overlap threshold are highlighted. The top and bottom tables show the results on two different sets of ground-truth labeling constructed manually.

mance than many state-of-the-art methods when the test datasets are complex. In contrast to many previous approaches, we conducted experiments on complex datasets that reflect practical scenarios. The results show that there is still a long way to go to achieve satisfactory recognition on such data.

Acknowledgments

This work was supported, in part, by AFOSR FA9550-11-1-0327, NSF IIS-1017199 and ARL under Cooperative Agreement Number W911NF-10-2-0060 (DARPA Mind's Eye).

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005. 1

[2] E. Candés and J. Romberg. L-1 magic package. http://users.ece.gatech.edu/∼justin/l1magic/downloads/l1magic-1.11.zip. 4

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 1

[4] DARPA. Video Dataset from DARPA Mind's Eye Program. http://www.visint.org, 2011. 5

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005. 1, 4

[6] M. Hoai and F. De la Torre. Max-margin early event detectors. In *CVPR*, pages 2863–2870, 2012. 2, 4, 5

[7] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, pages 1–8, 2007. 1, 4

[8] K. Kitani, B. D. Ziebart, J. A. D. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, pages 201–214, 2012. 2

[9] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, pages 99.1–99.10, 2008. 1

[10] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003. 1

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008. 1

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[13] J. Niebles, H. Wang, and F.-F. Li. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008. 1

[14] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, pages 1036–1043, 2011. 1, 2, 3, 4, 5

[15] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, pages 1593–1600, 2009. 1

[16] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010. 5

[17] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012. 4

[18] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, pages 32–36, 2004. 1

[19] P. Scovanner, S. Ali, and M. Shah. A 3-Dimensional SIFT descriptor and its application to action recognition. In *Proceedings of the 15th International Conference on Multimedia*, pages 357–360, 2007. 1

[20] D. Waltisberg, A. Yao, J. Gall, and L. V. Gool. Variations of a hough-voting action recognition system. In *ICPR*, 2010. 1

[21] G. Willems, T. Tuytelaars, and L. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, pages 650–663, 2008. 1

[22] S. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, pages 1–8, 2007. 1

[23] A. Y. Yang, A. Ganesh, Z. Zhou, A. Wagner, V. Shia, S. Sastry, and Y. Ma. L-1 Benchmark package. http://www.eecs.berkeley.edu/∼yang/software/l1benchmark/l1benchmark.zip. 4

[24] T. Yu, T. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forest. In *BMVC*, pages 52.1–52.12, 2010. 1