2013 IEEE Conference on Computer Vision and Pattern Recognition

Context-Aware Modeling and Recognition of Activities in Video

Yingying Zhu   Nandita M. Nayak   Amit K. Roy-Chowdhury [*]
University of California, Riverside
yzhu010@ucr.edu   nandita.nayak.m@gmail.com   amitrc@ee.ucr.edu

Abstract

In this paper, rather than modeling activities in videos individually, we propose a hierarchical framework that jointly models and recognizes related activities using motion and various context features. This is motivated from the observations that the activities related in space and time rarely occur independently and can serve as the context for each other. Given a video, action segments are automatically detected using motion segmentation based on a nonlinear dynamical model. We aim to merge these segments into activities of interest and generate optimum labels for the activities. Towards this goal, we utilize a structural model in a max-margin framework that jointly models the underlying activities which are related in space and time. The model explicitly learns the duration, motion and context patterns for each activity class, as well as the spatio-temporal relationships for groups of them. The learned model is then used to optimally label the activities in the testing videos using a greedy search method. We show promising results on the VIRAT Ground Dataset demonstrating the benefit of joint modeling and recognizing activities in a wide-area scene.

1. Introduction

It has been demonstrated in [20] that context is significant in human visual systems. As there is no formal definition of context in computer vision, we consider all the detected objects and motion regions as providing contextual information about each other. Activities in natural scenes rarely happen independently. The spatial layout of activities and their sequential patterns provide useful cues for their understanding. Consider the activities that happen in the same spatio-temporal region in Fig. 1: the existence of the nearby car gives information about what the person (bounded by red circle) is doing, and the relative position of the person of interest and the car says that activities (a) and (c) are very different from activity (b). However, it is hard to tell what the person is doing in (a) and (c) - getting out of the vehicle or getting into the vehicle. If we knew that these activities occurred around the same vehicle along time, it would be immediately clear that in (a) the person is getting out of the vehicle and in (c) the person is getting into the vehicle. This example shows the importance of spatial and temporal relationships for activity recognition.

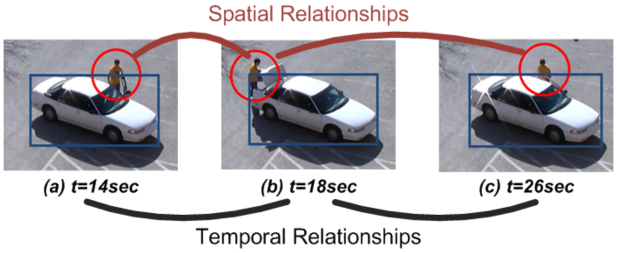


Figure 1: An example that demonstrates the importance of context in activity recognition. Motion region surrounding the person of interest is located by red circle and the vehicle with which he interacts is located by blue bounding box.

Many existing works on activity recognition assume that, the temporal locations of the activities are known [1, 19]. We focus on the problem of detecting activities of interest in *continuous* videos without prior information about the locations of activities. We provide an integrated framework that conducts multiple stages of video analysis, starting with motion localization. Then action segments, which are considered as the elements of activities, are detected using a motion segmentation algorithm based on the nonlinear dynamic model (NDM) in [5]. Finally, we learn a structural model that merges these segments into activities and generates the optimum activity labels for them.

Fig. 2 shows the framework of our approach. Given a video, we detect the motion regions using background subtraction. The segmentation algorithm aims to divide a continuous motion region into action segments, whose motion pattern is consistent and is different from its adjacent

[*] This work has been partially supported by NSF grant IIS-0712253 and a subcontract from Mayachitra Inc., through DARPA STTR award W31P4Q-11-C0042.

Y. Zhu and A. Roy-Chowdhury are with the Electrical Engineering Department and N. Nayak is in Computer Science at UCR.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.322

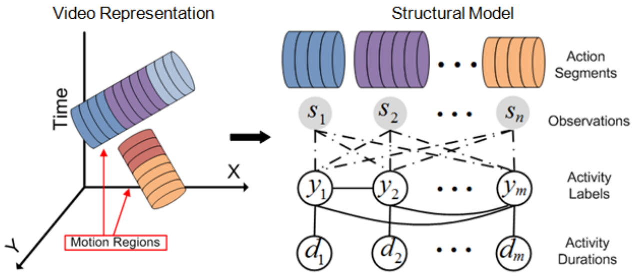


Figure 2: The left graph shows the video representation of an activity set with n motion segments and m activities (for testing, m needs to be determined by inference of the learned structural model). The right graph shows the graphical representation of our model. The gray nodes in the graph are the feature observations and the white nodes are the model variables. The dashed lines indicate that the connections between activity labels and the observations of action segments are not fixed, i.e., the structure of connections is different for different activity sets.

segments. The main challenge now is to develop a representation of the continuous video that respects the spatio-temporal relationships of the activities. To achieve this goal, we build upon existing well-known feature descriptors and spatio-temporal context representations that, when combined together, provide a powerful framework to model activities in continuous videos. Action segments that are related to each other in space and time are grouped together into activity sets. For each set, the underlying activities are jointly modeled and recognized by a structural model with the activity durations as the auxiliary variables. For the testing, the action segments, which are considered as the basic elements of activities, are merged together and assigned activity labels by inference on the structural model.

1.1. Main Contributions

The main contribution of this work is three-fold. (i) We combine low-level motion segmentation with high-level activity modeling under one framework. (ii) We jointly model and recognize the activities in video using a structural model, which integrates activity durations, motion features and various context features within and between activities into a unified model. (iii) We formulate the inference problem as a greedy strategy that iteratively searches for the optimum activity labels on the learned structural model. The greedy search decreases computational complexity of the inference process with negligible reduction to recognition accuracy.

2. Related Work

Many existing works exploring context focus on interactions among features, objects and actions [26, 23, 12, 2, 1], environmental conditions such as spatial locations of certain activities in the scene[16], and temporal relationships of activities [24, 17]. Space-time constraints across activities in a wide-area scene are rarely considered.

The work in [24] models a complex activity by a variable-duration hidden Markov model on equal-length temporal segments. It decomposes a complex activity into sequential actions, which are the context of each other. However, it considers only the temporal relationships, while ignoring the spatial relationships between actions. AND-OR graph [11, 21] is a powerful tool for activity representation. However, the learning and inference processes of AND-OR graphs become more complex as the graph grows large and more and more activities are learned. In [13, 14], a structural model is proposed to learn both feature level and action level interactions of group members. This method labels each image with an group activity label. How to smooth the labeling results along time is a problem and is not addressed in the paper. Also, these methods aim to recognize group activities and are not suitable in our scenario where activities cannot be considered as the parts of larger activities. In [27], a structural model is used to integrate motion features and context features in and between activities. However, there was no activity segmentation or modeling of the activity duration; only the regions with activity were detected. We propose an alternative method that explicitly models the durations, motion, intra-activity context and the spatio-temporal relationships between the activities and use them in the inference stage for recognition.

The inference method on a structural model proposed in [13, 14] searches through the graphical structure, in order to find the one that maximizes the potential function. Though this inference method is computationally less intensive than exhaustive search, it is still time consuming. As an alternative, greedy search has been used for inference in object recognition [7]. The novelty of this paper lies in developing a structural model for representing related activities in a video, and to demonstrate how to perform efficient inference on this model.

3. Model Formulation for Context-Aware Activity Representation

In this section, a structural activity model that integrates activity durations, motion features and various context features within and across activities is built upon automatically detected action segments to jointly model related activities in space and time.

3.1. Video Representation

Given a continuous video, background substraction [28] is used to locate the moving objects. Moving persons are identified by [9]. The bounding boxes of moving persons are used as the initialization of the tracking method developed in [22] to obtain local trajectories of the moving persons. Spatio-temporal interest points (STIP) features [15] are generated only for these motion regions. Thus, STIPs generated by noise, such as slight tree shaking, camera jitter and motion of shadows, are avoided. Each motion region is segmented into action segments using the motion segmentation based on the method in [5] with STIP histograms as the model observation. The detailed motion segmentation algorithm is described in Section 5.2.

3.2. Motion and Context Feature Descriptors

Assume there are $M+1$ classes of activities in the scene, including a background class with label 0 and M classes of interest with labels $1, ..., M$. We first define the concepts we use for the feature development. An activity is a 3D region consisting of one or multiple consecutive action segments. An agent is the underlying moving person along a trajectory. Motion region at frame n is a circular region surrounding the moving objects of interest in the n^{th} frame of the activity. Activity region is the smallest rectangular region that encapsulates the motion regions over all frames of the activity. Based on this, we can now encode motion and context information into feature descriptors.

Intra-activity motion feature descriptor Features of an activity that encode the motion information extracted from low-level motion features such as STIP features are defined as intra-activity motion features. We train a multi-SVM [4] classifier upon the detected action segments to generate the normalized confidence scores $s_{i,0}, ..., s_{i,M}$ of classifying the action segment i as activity classes $0, 1, ..., M$, such that $\sum_{j=0}^{M} s_{i,j} = 1$. We call the classifier as the baseline classifier. In general, any kind of classifier and low-level motion features can be used here. Given an activity, $\boldsymbol{x} = [\max_{i\in\aleph} s_{i,0}, ..., \max_{i\in\aleph} s_{i,M}]$ is developed as the intra-activity motion feature descriptor, where $\aleph$ is a list of action segments in the activity.

Intra-activity context feature descriptor Features that capture the relationships between the agents, as well as other interacting objects, are defined as intra-activity context features. Objects including vehicles, opening/closing entrance/exit doors of facilities, boxes and bags that overlap with the motion regions, are detected. Persons and vehicles are detected using the publicly available software [9]. Opening/closing entrance/exit doors of facilities, boxes and bags are detected using method in [6] with Histogram of Gradient as the low-level feature and binary linear-SVM as the classifier. These high-level image features will be used for the development of the context features within activities.

We define a set G of attributes related to the scene and the involved objects in activities of interest. G consists of N_G subsets of attributes that are exclusively related to certain image-level features. Since we work on the VIRAT dataset with individual person activities and person-object interactions, we use the following $(N_G = 6)$ subsets of attributes for the development of intra-activity context features in the experiments as shown in Fig. 3.

Attribute Subset	Associated Attributes
G_1	moving object is a person; moving object is a vehicle; moving object is of other kind.
G_2	the agent is at the body of the interacting vehicle; the agent is at the rear/head of the interacting vehicle; the agent is far away from the vehicle.
G_3	the agent disappears at the entrance of a facility; the agent appears at the exit of a facility; none of the two.
G_4	velocity of the agent (in pixel) is larger than a predefined threshold; velocity of object of interest is smaller than a predefined threshold.
G_5	the activity occurs in a parking area; the activity occurs in other areas.
G_6	a bag/box is detected on the agent; no bag/box is detected on the agent.

Figure 3: Example subsets of context attributes used for the development of intra-activity context features.

For a given activity, the above attributes are determined from image-level detection results. For frame n of an activity, we obtain $\boldsymbol{g}_i(n) = I(G_i(n))$, where $I(\cdot)$ is the indicator function. $\boldsymbol{g}_i(n)$ is then normalized so that its elements sum to 1. Fig. 4 shows an example of $\boldsymbol{g}_i(n)$.

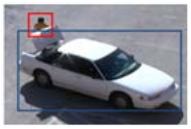


Figure 4: The image shows one frame of 'person unloading an object from a vehicle'. In the image, moving objects are the person and the vehicle, and the person is in the rear of the vehicle. So, for this frame, $\boldsymbol{g}_1(n) = [1 \quad 0 \quad 0]$ and $\boldsymbol{g}_2(n) = [0 \quad 1 \quad 0]$, where n is the frame number of this image in the activity.

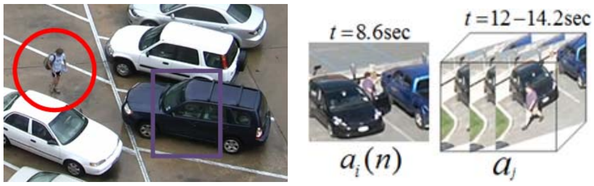
Let $\boldsymbol{g}_i = \frac{1}{N_f} \sum_{n=1}^{N_f} \boldsymbol{g}_i(n)$, where N_f is the total number of frames associated with the activity. The $\sum_{i=1}^{N_G} n_{G_i}$-bin histogram $\boldsymbol{g} = \frac{1}{N_G}[\boldsymbol{g}_1 \oplus \cdots \oplus \boldsymbol{g}_{N_G}]$ is the intra-activity context feature vector of the activity, where $\oplus$ denotes the vector concatenation operator.

Inter-activity context feature descriptor Features that capture the relative spatial and temporal relationships of activities are defined as inter-activity context feature. Define the scaled distance between activity a_i and a_j at the n^{th} frame of a_i as

$$r_s(a_i(n), a_j) = \frac{d(O_{a_i}(n), O_{a_j})}{R_{a_i}(n) + R_{a_j}}, \qquad (1)$$

where $O_{a_i}(n)$ and $R_{a_i}(n)$ denote the center and radius of the motion region of activity a_i at its n^{th} frame and O_{a_j} and R_{a_j} denote the center and radius of the activity region of activity a_j. $d(\cdot)$ denotes the Euclidean distance. Then, the spatial relationship of a_i and a_j at the n^{th} frame is modeled by $sc_{ij}(n) = bin(r_s(a_i(n), a_j))$ as in Fig. 5 (a). The normalized histogram $\boldsymbol{sc}_{a_i,a_j} = \frac{1}{N_f}\sum_{n=1}^{N_f} \boldsymbol{sc}_{ij}(n)$ is the inter-activity spatial feature of activity a_i and a_j.

Temporal context is defined by the following temporal relationships: n^{th} frame of a_i is before a_j, n^{th} frame of a_i is during a_j, and n^{th} frame of a_i is after a_j. $\boldsymbol{tc}_{ij}(n)$ is the temporal relationship of a_i and a_j at the n^{th} frame of a_i as shown in Fig. 5 (b). The normalized histogram $\boldsymbol{tc} = \frac{1}{N_f}\sum_{n=1}^{N_f} \boldsymbol{tc}_{ij}(n)$ is the inter-activity temporal context feature of activity a_i with respect to activity a_j.



(a) (b)

Figure 5: (a) The image shows an example of inter-activity spatial relationship. The red circle indicates the motion region of a_i at this frame while the purple rectangle indicates the activity region of a_j. Assume SC is defined by quantizing and grouping $r_s(n)$ into three bins: $r_s(n) \leq 0.5$ (a_i and a_j is at the same spatial position at the n^{th} frame of a_i), $0.5 < r_s(n) < 1.5$ (a_i is near a_j at the n^{th} frame of a_i), and $r_s(n) \geq 1.5$ (a_i is far away from a_j at the n^{th} frame of a_i). In the image, $r_s(n) > 1.5$, so, $\boldsymbol{sc}_{ij}(n) = [0 \quad 0 \quad 1]$. (b) The image shows one example of inter-activity temporal relationship. The n^{th} frame of a_i occurs before a_j. So, $\boldsymbol{tc}_{ij}(n) = [1 \quad 0 \quad 0]$.

3.3. Structural Activity Model

For an activity set $\boldsymbol{a}$ with n action segments, we assign an auxiliary duration vector $\boldsymbol{d} = [d_1, \cdots, d_m]$ ($\sum_{i=1}^{m} d_i = n$) and a label vector $\boldsymbol{y} = [y_1, \cdots, y_m]$. $y_i \in \{0, ..., M\}$ is the activity label of the i^{th} activity and d_i is its activity duration, for $i = 1, \cdots, m$. Thus, for $\boldsymbol{a} = [a_1, \cdots, a_m]$, a_i is the i^{th} activity in the set. Assume $\boldsymbol{x}_i \in R^{D_x}$ and $\boldsymbol{g}_i \in R^{D_g}$ to be the motion feature and intra-activity context feature of instance a_i, and D_x and D_g to be the dimension of $\boldsymbol{x}_i$ and $\boldsymbol{g}_i$ respectively. $\boldsymbol{\omega}_{d,y_i} \in R^{D_x}$, $\boldsymbol{\omega}_{x,y_i} \in R^{D_x}$ and $\boldsymbol{\omega}_{g,y_i} \in R^{D_g}$ are the weight vectors that capture the valid duration, motion and intra-activity context patterns of activity class y_i. $\boldsymbol{sc}_{ij} \in R^{D_{sc}}$ and $\boldsymbol{tc}_{ij} \in R^{D_{tc}}$ are the inter-activity context features associated with a_i and a_j. D_{sc} and D_{tc} are the dimensions of $\boldsymbol{sc}_{ij}$ and $\boldsymbol{tc}_{ij}$ respectively. $\boldsymbol{\omega}_{sc,y_i,y_j} \in R^{D_{sc}}$ and $\boldsymbol{\omega}_{tc,y_i,y_j} \in R^{D_{tc}}$ are the weight vectors that capture the valid spatial and temporal relationships of activity classes y_i and y_j. In general, dimensions of the same kind of feature can be different for each activity class/class pairs. Four potentials are developed to measure the compatibilities between the assigned variables $(\boldsymbol{y}, \boldsymbol{d})$ and the observed features of activity set $\boldsymbol{a}$.

Activity-duration potential measures the compatibility between the activity label y_i and its duration d_i for activity a_i. It is defined as

$$F_d\left(y_i, d_i\right) = d_i \boldsymbol{\omega}_{d,y_i}^T \boldsymbol{I}(d_i). \tag{2}$$

If d_{max} is the maximum duration of an activity, $\boldsymbol{I}(d_i)$ generates a $d_{max} \times 1$ vector with one for the $(d_i)^{th}$ element and zeros otherwise.

Intra-activity motion potential measures the compatibility between the activity label of a_i and the intra-activity motion feature x_i developed from the associated action segments as

$$F_x\left(y_i, d_i\right) = d_i \boldsymbol{\omega}_{x,y_i}^T \boldsymbol{x}_i. \tag{3}$$

Intra-activity context potential measures the compatibility between the activity label of a_i and its intra-activity context feature g_i as

$$F_g\left(y_i, d_i\right) = d_i \boldsymbol{\omega}_{g,y_i}^T \boldsymbol{g}_i. \tag{4}$$

Inter-activity context potential measures the compatibility between the activity labels of a_i and a_j and their spatial and temporal relationships sc_{ij} and tc_{ij} as

$$F_{sc,tc}\left(y_i, y_j, d_i, d_j\right) = d_i d_j (\boldsymbol{\omega}_{sc,y_i,y_j}^T \boldsymbol{sc}_{ij} + \boldsymbol{\omega}_{tc,y_i,y_j}^T \boldsymbol{tc}_{ij}). \tag{5}$$

Combined potential function $F(\boldsymbol{a}, \boldsymbol{y}, \boldsymbol{d})$ is defined to measure the compatibility between $(\boldsymbol{y}, \boldsymbol{d})$ of the activity set a and its features:

$$\begin{aligned} F\left(\boldsymbol{a}, \boldsymbol{y}, \boldsymbol{d}\right) = & \sum_{i=1}^{m} F_d\left(y_i, d_i\right) + \sum_{i=1}^{m} F_x\left(y_i, d_i\right) \\ & + \sum_{i=1}^{m} F_g\left(y_i, d_i\right) + \sum_{i,j=1}^{m} F_{sc,tc}\left(y_i, y_j, d_i, d_j\right). \end{aligned} \tag{6}$$

The optimum assignment of $(\boldsymbol{y}, \boldsymbol{d})$ for $\boldsymbol{a}$ maximizes the potential function $F(\boldsymbol{a}, \boldsymbol{y}, \boldsymbol{d})$.

4. Model Learning and Inference

4.1. Learning Model Parameters

We define the weight vector $\boldsymbol{\omega}$ as the concatenation of all the weight vectors defined above as

$$\boldsymbol{\omega} = [{\boldsymbol{\omega}_d}^T, {\boldsymbol{\omega}_x}^T, {\boldsymbol{\omega}_g}^T, {\boldsymbol{\omega}_{sc}}^T, {\boldsymbol{\omega}_{tc}}^T]^T, \tag{7}$$

where $\boldsymbol{\omega}_d$ is obtained by concatenating the w_{d,y_i} for all the $M+1$ activity classes. $\boldsymbol{\omega}_x$, $\boldsymbol{\omega}_g$, $\boldsymbol{\omega}_{sc}$ and $\boldsymbol{\omega}_{tc}$ are developed similarly. Thus, the potential function $F(a,y,d)$ can be converted into a linear function with a single parameter ω,

$$F(\boldsymbol{a},\boldsymbol{y},\boldsymbol{d}) = \boldsymbol{\omega}^T \boldsymbol{\Gamma}(\boldsymbol{a},\boldsymbol{y},\boldsymbol{d}), \tag{8}$$

where $\boldsymbol{\Gamma}(\boldsymbol{a},\boldsymbol{y},\boldsymbol{d})$, called the joint feature of activity set $\boldsymbol{a}$, can be easily obtained from (6).

Suppose we have P activity sets for training. Let the training set be $(\boldsymbol{A},\boldsymbol{Y},\boldsymbol{H}) = (\boldsymbol{a}^1,\boldsymbol{y}^1,\boldsymbol{d}^1),...,(\boldsymbol{a}^P,\boldsymbol{y}^P,\boldsymbol{d}^P)$, where $\boldsymbol{a}^i$ is the activity set, $\boldsymbol{y}^i$ is the label vector and $\boldsymbol{d}^i$ is the auxiliary vector. The loss function for assigning $\boldsymbol{a}^i$ with $(\widehat{\boldsymbol{y}}^i, \widehat{\boldsymbol{d}}^i)$, $\Delta(\boldsymbol{a}^i,\widehat{\boldsymbol{y}}^i,\widehat{\boldsymbol{d}}^i)$, equals the number of action segments that associate with incorrect activity labels (an action segment is mislabeled if over half of the segment is mislabeled). The learning problem can now be written as

$$\begin{aligned}\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \Bigg\{ & \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} - C\sum_{i=1}^{P} \boldsymbol{\omega}^T \boldsymbol{\Gamma}\left(\boldsymbol{a}^i,\boldsymbol{y}^i,\boldsymbol{d}^i\right) \\ & + C\sum_{i=1}^{P} \max_{(\widehat{\boldsymbol{y}}^i,\widehat{\boldsymbol{d}}^i)} \left[\boldsymbol{\omega}^T \boldsymbol{\Gamma}\left(\boldsymbol{a}^i,\widehat{\boldsymbol{y}}^i,\widehat{\boldsymbol{d}}^i\right) + \Delta(\boldsymbol{a}^i,\widehat{\boldsymbol{y}}^i,\widehat{\boldsymbol{d}}^i)\right] \Bigg\},\end{aligned} \tag{9}$$

where where C controls the tradeoff between the errors in the training model and margin maximization [3]. The problem in (9) can be converted to an unconstrained convex optimization problem [7] and solved by the modified bundle method in [25]. It iteratively searches for the increasingly tight quadratic upper and lower cutting planes of the objective function until the gap between the two bounds reaches a predefined threshold. The algorithm is effective because of its high convergence rate [25]. We set all weights related to background activities to be zeros.

4.2. Inference

With the learned model parameter vector $\boldsymbol{\omega}$, we now describe how to identify the optimum label vector $\boldsymbol{y}_{test}$ and duration vector $\boldsymbol{d}_{test}$ for an input instance $\boldsymbol{a}_{test}$. Suppose the testing instance has n action segments. Greedy forward search [7] is used to find the optimum labels and durations of the targeted activities. The potential function F is initialized as 0. We greedily instantiate $\boldsymbol{d}_i$ consecutive segments denoted as $\boldsymbol{a}_i$ that, when labeled as a specific activity class, can increase the weighted value of the compatibility function, F, by the largest amount. The algorithm stops when all the action segments are labeled. Algorithm 1 gives the overview of the inference process. The time complexity of the greedy search is $O(d_{max}Mn^2)$. While this greedy search algorithm cannot guarantee a globally optimum solution, in practice it works well to find good solutions for problems of our kinds [7].

Algorithm 1 Greedy Search Algorithm

```
Input:    Testing instance with $n$ action segments
Output:   Interested activities $A$, label vector $Y$ and the duration vector $D$

1. initialize $(A,Y,D) \leftarrow \{\emptyset,\emptyset,\emptyset\}$ and $F = 0$.
2. repeat
   $\Delta F(a_i,y_i,d_i)_{a_i \notin A} = \frac{F((A,Y,D)\cup(a_i,y_i,d_i)) - F(A,Y,D)}{d_i}$;
   $(a_i,y_i,d_i)^{opt} = \arg\max_{a_i \notin A} \Delta F(a_i,y_i,d_i)$;
   $(A,Y,D) \leftarrow (A,Y,D) \cup (a_i,y_i,d_i)^{opt}$;
3. end if $\Delta F(a_i,y_i,d_i)_{\forall a_i \notin A} < 0$ or $\sum_i {d_i}^{opt} = n$.
```

5. Experiment

To assess the effectiveness of our structural model in activity modeling and recognition, we perform experiments on the public VIRAT Ground Dataset [8]. We use the NDM method in [5] with the SVM classifier as the baseline (referred to as NDM + SVM) and integrate our context model with it. We compare our results with the popular activity recognition method, BOW+SVM [18], and recently developed methods - string of feature graphs (SFG) [10] and sum-product networks (SPN) [1].

5.1. Dataset

VIRAT Ground dataset is a state-of-the-art activity dataset with many challenging characteristics, such as wide variation in the activities and clutter in the scene. The dataset consists of surveillance videos of realistic scenes with different scales and resolution, each lasting 2 to 15 minutes and containing upto 30 events. The activities defined in Release 1 include 1 - person loading an object to a vehicle; 2 - person unloading an object from a vehicle; 3 - person opening a vehicle trunk; 4 - person closing a vehicle trunk; 5 - person getting into a vehicle; 6 - person getting out of a vehicle. We work on the all the scenes in Release 1 except scene 0002, and use half of the data for training and the rest for testing. Five more activities are defined in VIRAT Release 2 as: 7 - person gesturing; 8 - person carrying an object; 9 - person running; 10 - person entering a facility; 11 - person exiting a facility. We work on the all the scenes in Release 2 except scene 0002 and 0102, and use two-third of the data for training and the rest for testing.

5.2. Preprocessing and Feature Extraction

Motion regions involving only vehicles moving are excluded from the experiments since we are only interested in person activities and person-vehicle interactions. For the development of STIP histograms, $k = 500$ visual words and a 9-nearest neighbor soft-weighting scheme are used. For the SFG-based classifier, the size of each temporal bin used is 5 frames while other settings are the same as in [10].

A distance threshold of 2 times the height of the person and a time threshold of 15 seconds are used to group action segments into activity sets. We follow the description in Section 3.2 to develop the feature descriptors for each activity set. The first two sets of attributes in Fig. 3 are used for the experiments on Release 1, and all are used for the experiments on Release 2.

5.3. Motion Segmentation

We develop an automatic motion segmentation algorithm by detecting boundaries where the statistics of motion features change dramatically, and obtain the action segments. Let two NDMs be denoted as M_1 and M_2, and d_s be the dimension of the hidden states. The distance between the models can be measured by the normalized geodesic distance $dist(M_1, M_2) = \frac{4}{d_s \pi^2} \sum_{i=1}^{d} {\theta_i}^2$, where θ_i is the principal subspace angle (please refer to [5] for details on the distance computation).

A sliding window of size T_s, where T_s is the number of temporal bins in the window, is applied to each detected motion region along time. A NDM $M(t)$ is built for the time window centered at the t^{th} temporal bin. Since an action can be modeled as one dynamic model, the model distances between subsequences from the same action should be small, compared to those of subsequences from a different action. Suppose an activity starts from temporal bin k; the average model distance between temporal bin $j > k$ and k is defined as the weighted average distance between model j and neighboring models of k as

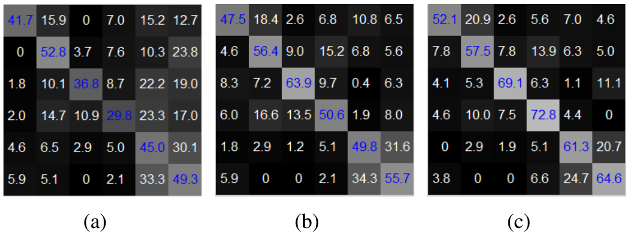
$$DE_k(j) = \sum_{i=0}^{T_d - 1} \gamma_i \cdot dist(M(k+i), M(j)), \qquad (10)$$

where T_d is the number of neighboring bins used, and γ_i is the smoothing weight for model $k+i$ that decreases along time. When the average model distance grows above a predefined threshold d_{th}, an action boundary is detected. Action segments along tracks are thus obtained. In order to demonstrate that the segmentation algorithm can automatically detect actions, we evaluate the performance on VIRAT Release 1. We synthesize continuous videos by concatenating video clips, each containing an activity.

Defining the segmentation accuracy as twice the absolute sum of deviations of estimated activity boundaries from the real ones normalized by the total number of total frames, the segmentation accuracy on VIRAT dataset Release 1 is $85.5 \pm 3.8\%$. We change the window size T_s from 50 to 70 with step size 5. The smaller the distance threshold d_{th} is the more number of action segments a complex activity may have.

5.4. Recognition Results on VIRAT Release 1

Fig. 6 shows the confusion matrix for the baseline classifier and our model with different kinds of features. As an example of the importance of context features, the baseline classifier often confuses “open a vehicle trunk” and “close a vehicle trunk” with each other. However, if the two activities happen closely in time in the same place, the first activity in time is probably “open a vehicle trunk”. This kind of contextual information within and across activity classes are captured by our model and used to improve the recognition performance.



41.7	15.9	0	7.0	15.2	12.7
0	52.8	3.7	7.6	10.3	23.8
1.8	10.1	36.8	8.7	22.2	19.0
2.0	14.7	10.9	29.8	23.3	17.0
4.6	6.5	2.9	5.0	45.0	30.1
5.9	5.1	0	2.1	33.3	49.3

(a)

47.5	18.4	2.6	6.8	10.8	6.5
4.6	56.4	9.0	15.2	6.8	5.6
8.3	7.2	63.9	9.7	0.4	6.3
6.0	16.6	13.5	50.6	1.9	8.0
1.8	2.9	1.2	5.1	49.8	31.6
5.9	0	0	2.1	34.3	55.7

(b)

52.1	20.9	2.6	5.6	7.0	4.6
7.8	57.5	7.8	13.9	6.3	5.0
4.1	5.3	69.1	6.3	1.1	11.1
4.6	10.0	7.5	72.8	4.4	0
0	2.9	1.9	5.1	61.3	20.7
3.8	0	0	6.6	24.7	64.6

(c)

Figure 6: Recognition Results for VIRAT Release 1. (a): Confusion matrix for the baseline classifier; (b): Confusion matrix for our approach using motion and intra-activity context features; (c): (b): Confusion matrix for our approach using motion and intra- and inter- activity context features.

We show the results on VIRAT Release 1 using precision and recall in Fig. 7. We have compared our results with the popular BOW+SVM approach, the more recently proposed String-of-Feature-Graphs approach [10] and the baseline classifier. Our approach outperforms the other methods. The results are expected since the intra-activity and inter-activity context give the model additional information about the activities beyond the motion information encoded in low-level features. SFG approach models the spatial and temporal relationships between the low-level features and thus takes into account the local structure of the scene. However, it does not consider the relationships between various activities and thus our method outperforms the SFGs. Fig. 8 shows examples that demonstrate the significance of context in activity recognition.

5.5. Recognition Results on VIRAT Release 2

We work on VIRAT Release 2 to further evaluate the effectiveness of the proposed approach. We follow the method defined above to get the recognition results on this dataset. Fig. 9 compares the recognition accuracy using precision and recall for different methods. We can see that the performance of our method is comparable to that in [1]. In [1], an SPN on BOW is learned to explore the context among motion features. However, [1] works on video clips, each containing an activity of interest with additional 10 seconds occurring randomly before or after the target activity instance, while we work on continuous video.

Activity Class	BOW+SVM [18]	SFG [10]	Baseline (NDM+SVM)	Our Method (1)	Our Method (2)
loading-object	44.2(42.8)	50.7(52.3)	43.6(41.7)	42.1(47.5)	**51.6**(**52.1**)
unloading-object	51.1(57.2)	57.1(55.4)	34.9(52.8)	61.3(56.4)	**62.7**(**57.5**)
opening-trunk	58.5(39.3)	38.4(50.3)	59.7(36.8)	64.2(63.9)	**68.5**(**69.1**)
closing-trunk	47.2(33.4)	60.0(61.2)	40.6(29.8)	44.4(50.6)	**55.2**(**72.8**)
getting-into-vehicle	40.4(48.2)	61.8(59.2)	32.7(45.0)	53.0(49.8)	**67.5**(**61.3**)
getting-out-of-vehicle	42.2(53.8)	41.6(68.0)	32.1(49.3)	49.6(55.7)	**65.2**(**64.6**)
Average	47.2(45.8)	51.6(57.8)	40.6(42.5)	52.4(53.8)	**61.7**(**62.9**)

Figure 7: Precision and recall (in parenthesis) for the six activities defined in VIRAT Release 1. Our method (1): the proposed structural model with motion feature and intra-activity context feature; our method (2): the proposed structural model with motion feature, intra-activity and inter-activity context features. Note that SVM+BOW works on video clips; while other methods work on continuous videos.

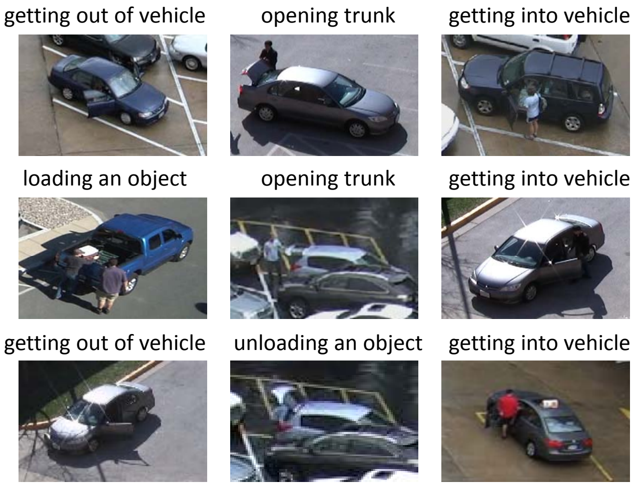
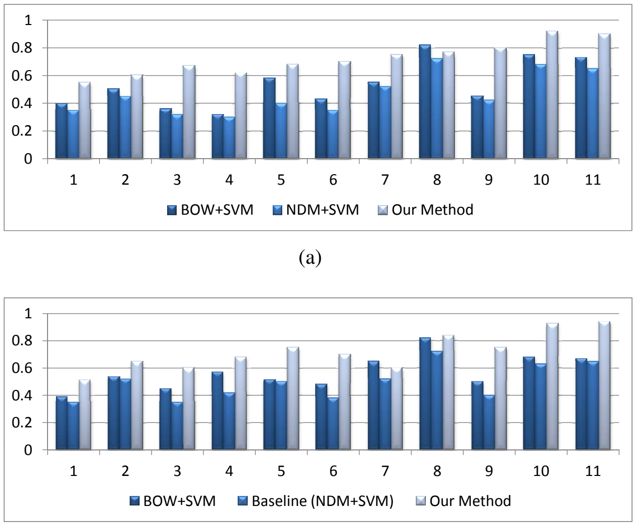


Figure 8: Example activities (from VIRAT Release 1) correctly recognized by baseline classifier (top), incorrectly by baseline classifier but corrected using intra-activity context (middle), and incorrectly recognized by baseline classifier and intra-activity context, but rectified using inter-activity context (bottom).

	BOW+SVM[18]	SPN[1]	Our Method
Precision	52.3	72	71.8
Recall	55.4	70	73.5

Figure 9: Precision and recall (in parenthesis) for different methods (averaged across activities).

Fig. 10 compares the precision and recall for the eleven activities defined in VIRAT Release 2 for BOW+SVM method, the baseline classifier, and our method. We see that by modeling the relationships between activities, those with strong context patterns, such as "person closing a vehicle trunk"(4) and "person running"(9), achieve larger performance gain compared to activities with weak context patterns such as "person gesturing"(7). Fig. 11 shows example results on activities in Release 2.



(a)

(b)

Figure 10: Precision (a) and recall (b) for the eleven activities defined in VIRAT Release 2.

6. Conclusion

In this paper, we present a novel approach to jointly model a variable number of activities in continuous videos. We have addressed the problem of automatic motion segmentation based on low-level motion features and the problem of high-level representations of activities in the scene. Upon the detected activity elements, we can build a high-level model that integrates various features within and between activities. It is expected that the proposed structural model can work with any other baseline classifiers. Our experiments demonstrate that joint modeling of activities, encapsulating object interactions and spatial and temporal relationships between activity classes, can significantly improve the recognition accuracy.

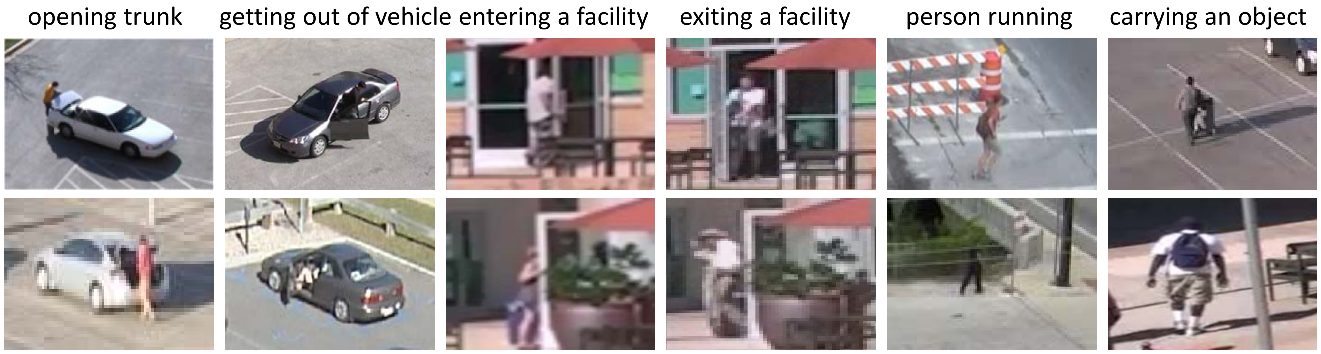


Figure 11: Examples (from VIRAT Release 2) in the bottom row show the effect of context features in correctly recognizing activities that were incorrectly recognized by the baseline classifier, while other examples of the same activities correctly recognized by the baseline classifier are shown in the top row.

References

[1] M. R. Amer and S. Todorovic. Sum-product networks for modeling activities with stochastic structure. In *CVPR*, 2012. 1, 2, 5, 6, 7

[2] Y. B. and F. L. Modeling mutual context of object and human pose in human object interaction activities. In *CVPR*, 2010. 2

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2nd edition, 2006. 5

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 2011. 3

[5] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, 2009. 1, 3, 5, 6

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3

[7] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. In *International Journal of Computer Vision*, 2011. 2, 5

[8] S. O. et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 5

[9] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, Release 4. http://people.cs.uchicago.edu/ pff/latent-release4/. 2, 3

[10] U. Guar, Y. Zhu, B. Song, and A. K. Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. In *ICCV*, 2011. 5, 6, 7

[11] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009. 2

[12] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009. 2

[13] T. Lan, Y. Wang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012. 2

[14] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010. 2

[15] I. Laptev. On space-time interest points. In *International Journal of Computer Vision*, 2005. 2

[16] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 2

[17] V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *CVPR*, 2011. 2

[18] Y.-G. J. G.-W. Ngo and J. Yang. Towards optimal bag of words for object categorization and semantic video retrieval. *ACM-CIVR*, 2007. 5, 7

[19] J. C. Niebles, H. Wang, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 1

[20] A. Oliva and A. Torralba. The role of context in object recognition. In *Trends in Cognitive Science*, 2007. 1

[21] Z. Si, M. Pei, B. Yao, and S. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *ICCV*, 2011. 2

[22] B. Song, T. Jeng, E. Staudt, and A. Roy-Chowdury. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, 2010. 2

[23] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009. 2

[24] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal stucture for complex event detection. In *CVPR*, 2012. 2

[25] C. H. Teo, Q. Le, A. Smola, and S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *SIGKDD*, pages 727–736, 2007. 5

[26] J. Wang, Z. Chen, and Y. Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, 2011. 2

[27] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. In *IEEE Journal of Selected Topics in Signal Processing*, pages 91–101, February 2013. 2

[28] Z.Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *ICPR*, 2004. 2