Hierarchical Graphical Models for Simultaneous Tracking and Recognition in Wide-Area Scenes

Nandita M. Nayak, Yingying Zhu, and Amit K. Roy-Chowdhury

***Abstract*—We present a unified framework to track multiple people, as well localize and label their activities, in complex long-duration video sequences. To do this, we focus on two aspects - the influence of tracks on the activities performed by the corresponding actors and the structural relationships across activities. We propose a two-level hierarchical graphical model which learns the relationship between tracks, relationship between tracks and their corresponding activity segments, as well as the spatiotemporal relationships across activity segments. Such contextual relationships between tracks and activity segments are exploited at both the levels in the hierarchy for increased robustness. An L1-regularized structure learning approach is proposed for this purpose. While it is well known that availability of the labels and locations of activities can help in determining tracks more accurately and vice-versa, most current approaches have dealt with these problems separately. Inspired by research in the area of biological vision, we propose a bi-directional approach that integrates both bottom-up and top-down processing, i.e., bottom-up recognition of activities using computed tracks and top-down computation of tracks using the obtained recognition. We demonstrate our results on the recent and publicly available UCLA and VIRAT datasets consisting of realistic indoor and outdoor surveillance sequences.**

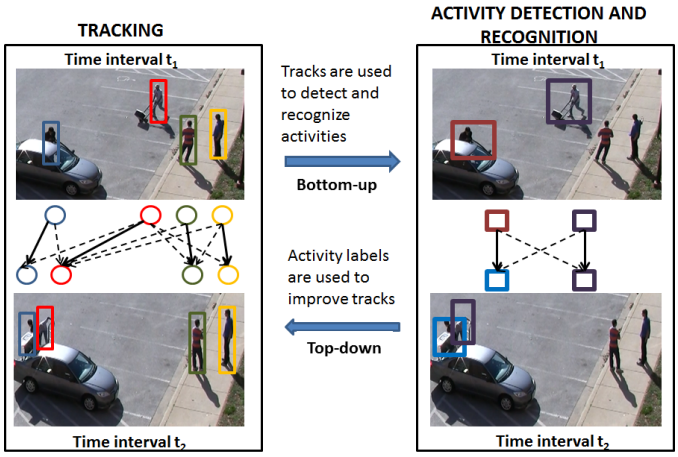

Fig. 1. Figure demonstrates the bi-directional processing of videos for integrated tracking and activity recognition. The bottom-up (or feedforward) processing involves detection and recognition using an initial set of tracks along with low level features and spatiotemporal context between activities. The top-down (or feedback) processing involves correcting the tracklet associations using the obtained labels.

I. Introduction

A continuous video consists of two inter-related components: 1) tracks of the persons in the video and 2) localization and labels of the activities of interest performed by these actors. Activity analysis of continuous videos involves solving both the tracking as well as recognition problems. In the past, most research on video analysis has treated these two problems separately. However, in the context of continuous videos, such as surveillance or sports videos, the solution to one problem can help in finding the solution to the other. Knowing the tracks can help in better detection and recognition of activities. Similarly, information about the location and labels of activities in a scene can help in determining the movement of people in the scene. Therefore, we propose a method which performs the two tasks in an integrated framework, modeling contextual relationships between tracks as well as activities using graphical models.

Research in the area of biological vision has shown that, the human visual system employs bi-directional (top-down as well as bottom-up) reasoning in analyzing and interpreting data of multiple resolutions [26]. This has been found to be particularly helpful in correcting errors due to false detections or noise. Applying these concepts to the analysis of continuous videos, we consider the task of obtaining recognition scores using tracks as a bottom-up (or feedforward) approach, while the task of correcting tracks using obtained recognition labels is treated as top-down (or feedback) processing. We alternate between both these steps resulting in a bi-directional algorithm that can help in increasing the accuracy of both these tasks.

The **main contribution** of our work is to propose a framework for simultaneous tracking, localization and labeling of activities in continuous videos, by integrating bottom-up and top-down processing along with automatic structure learning. Our approach can handle a varying number of actors and activities. In order to achieve this, we propose the following steps:

1) In the feedforward processing, the tracks are used to detect regions in the video where interesting activities are taking place. The activities in these detected regions are then recognized. The lower level nodes of the hierarchical graphical model captures relationships across tracklets, while the higher level nodes capture information across activities, also known as inter-activity context.
2) The detection and recognition of activities is carried out simultaneously using a 2-stage hierarchical Markov random field (HMRF) with L1-regularized structure

N. Nayak (email: nandita.nayak@email.ucr.edu) is with the Computer Science department at University of California, Riverside, CA, 92521 USA.

Y. Zhu (email: yzhu010@ucr.edu) and A. K. Roy-Chowdhury (email: amitrc@ee.ucr.edu) are with the Electrical Engineering department at University of California, Riverside, CA, 92521, USA.

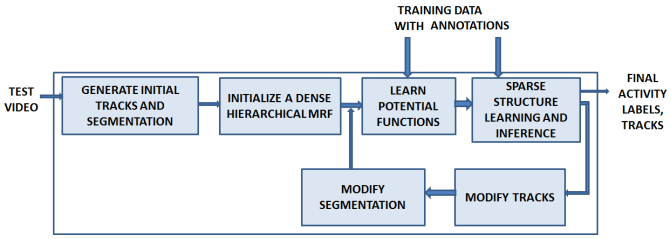

Fig. 2. Figure shows the illustration of our proposed method. Given a continuous video with computed tracklets, a set of tracks and activity segments are initialized. An HMRF model is built over the tracklets and segments. Edge potentials are learned on the annotated training data. Starting with a dense graph, L1-regularized structure learning gives a sparse set of edges. Inference on this graphical model provides a revised set of labels for the activities which can be fed back into the system to regenerate the tracks and rebuild the HMRF. The procedure is repeated until a stop criterion is reached. The tracks and labels of all segments are provided as output.

learning. We show that the structure learnt is sparse, and thus captures the most critical contextual information.

3) We use a bi-directional processing framework for learning the tracks and activities. The tracks computed in the previous step influence the structure of the hierarchical model and thereby the activities recognized. These activities are in-turn used to correct the tracks. We alternate between these two steps to arrive at the final solution for both these tasks. An illustration of the bi-directional computational framework in a continuous video is shown in Figure 1.

A. Overview

The illustration of our proposed method is shown in Figure 2. We have available a set of annotated training data with labeled tracks and activities, and a test video, for which the tracks as well as activities need to be discovered. We assume that we have with us a set of tracklets, which are short-term fragments of tracks with low probability of error. Tracklets have to be joined to form long-term tracks. In a multi-person scene, this involves tracklet association. Here, we use a basic particle filter for computing tracklets as mentioned in [28]. For the test video, it is assumed that each tracklet belongs to a single activity.

Pre-processing consists of computing tracklets and computing low level features such as space-time interest points in the region around these tracklets. Tracking involves association of one or more tracklets to tracks. Activity localization can now be defined as a grouping of tracklets into activity segments and recognition can be defined as the task of labeling these activity segments.

To begin with, we generate a set of match hypotheses for tracklet association and a likely set of tracks. An observation potential is computed for each tracklet using the features computed at the tracklet. Tracklets are grouped into activity segments using a standard baseline classifier such as multiclass SVM or motion segmentation.

Next, we construct a two-level Markov random field using the tracklets and activity segments. The first level nodes correspond to the tracklets and the second level nodes correspond to the activity segments. One or more tracklets can correspond to the same activity segment. Edges model relationships between nodes of the same level as well as nodes at different levels. This structure incorporates the context information between adjacent tracklets as well as across activity segments.

The dense HMRF has edges connecting each node to all other nodes within a certain spatiotemporal range. This gives us the initial graph on which we perform the learning and recognition.

The node features and edge features for the potential functions are computed from the training data. There are two tasks to be performed on the graph - choosing an appropriate structure and learning the parameters of the graph. Both these steps can be performed simultaneously by posing the parameter learning as an L1-regularized optimization [25]. The sparsity constraint on the HMRF ensures that the resulting parameters are sparse, thus capturing the most critical relationships between the objects. The parameters which are set to zero denote the edges which have been deleted from the graph. The non-zero parameters denote the parameters of edges retained after automatic structure discovery.

Inference on this graph provides the posterior probabilities for all nodes using information available at two resolutions. The activity labels are used in a top-down fashion to recompute the tracks. Activity segmentation on the recomputed tracks gives us a new set of nodes on which structure learning and inference is then repeated. Convergence is said to be achieved when the node labels and tracks do not change from one iteration to the next.

The output of the algorithm is a set of tracks, segments and the labels assigned to each segment.

II. Related Work

Reviews of related work in tracking and recognition can be found in [32][35]. We focus only on those that consider the problem of simultaneous localization and recognition. Simultaneous localization and classification of scenes in broadcast programs has been researched in the past in [29] but these scenes have distinctive breaks unlike continuous activity sequences. Localization and classification of single-person activities with distinctive breaks was performed in [8]. Activity localization and labeling of single person activities was also demonstrated in [4] but the system used concatenated short duration sequences which lack contextual information. We perform localization and classification on multi-person sequences in continuous videos and also explore the integration of tracking into the framework.

Simultaneous activity recognition and tracking has been studied in the context of interacting objects. The relations between interacting targets obtained from activity recognition is used in the tracking process using a relational dynamic Bayesian network in [27]. Simultaneous recognition of a collective activity and tracking of the multiple targets involved is performed in [5], [23]. However, these only deal with the motion relations between interacting persons and not across activities of the same person. They also do not look into the bi-directional processing in an EM framework.

Graphical models are commonly used to encode relationships in video analysis. Stochastic and context free grammars have been used to model complex activities in [21]. Variable length Hidden Markov models are used to identify activities with high amount of intra class variabilities in [31]. Co-occurring activities and their dependencies have been studied in [11]. Hierarchical MRF was used for image segmentation in [19]. In our work, we propose a hierarchical Markov random field framework which can handle varying number of actors and activities for the task of activity localization and recognition.

Spatio-temporal relationships have played an important role in the recognition of complex activities. Methods such as [22] and [37] explore spatio-temporal relationships at a feature level. Complex activities were represented as spatio-temporal graphs representing multi-scale video segments and their hierarchical relationships in [2]. Spatio-temporal context was represented using an MRF in [17], but activity locations were computed beforehand. The authors in [39] and [30] utilize context for recognition. These papers do not explore integration of higher and lower level representations. The method proposed in [34] utilizes a hierarchical model for context, however, it assumes that the activity locations are known and does not incorporate tracking. The authors in [16] and [12] explore relationships between simultaneous individual actions in a group activity but we consider the more general case where activities need not be directed towards having a collective objective. We also improve the tracks based on the recognition scores.

In applications such as activity recognition, the structure of the graph is difficult to determine. Prior approaches have either used fixed graphical models such as in [6], [16] or built graphs of known structures such as and-or graphs [20]. Recent approaches such as [39] have tackled this problem by using a greedy forward search to determine the best possible graph, thereby making the learning and inference very intensive. We learn an optimal set of structural relationships along with the parameters automatically in an L1-regularized learning framework as described in III-B. The L1-regularization ensures that a graph contains a sparse set of edges which represent the most critical contextual dependencies between nodes.

III. Hierarchical MRF (HMRF) Model

Consider a video to consist of a set of p tracklets resulting in tracks T. The tracklets can be grouped into a set of q activity segments along the tracks. We design a 2-level hierarchical MRF with nodes $X = X_t \bigcup X_a$, where the lower level nodes $X_t = \{x_{t_1}, x_{t_2} ... x_{t_p}\}$ correspond to tracklets and the higher level nodes $X_a = \{x_{a_1}, x_{a_2} ... x_{a_q}\}$ correspond to activity segments. The set of observed features obtained for a node x_i is denoted as y_i. There are three kinds of edges in the graph: edges connecting adjacent tracklets which belong to a valid track hypothesis, edges connecting tracklets to their corresponding activity segments, and edges connecting activity segments which are within a specified spatiotemporal distance of each other. A typical HMRF constructed over a continuous video is shown in Figure 3.

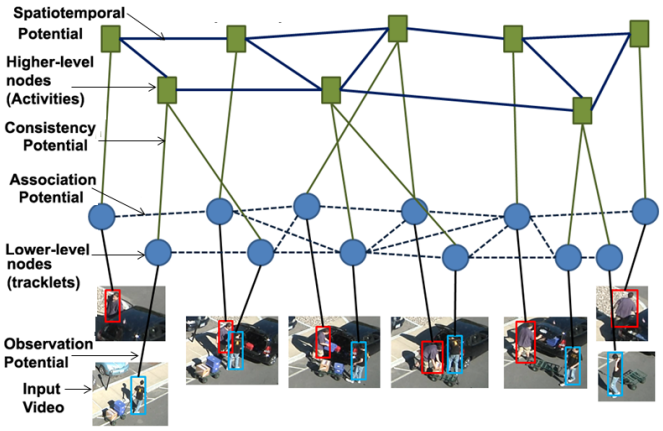

Fig. 3. Figure shows a typical HMRF over an activity sequence. Tracklets are extracted from a continuous video and form lower level nodes. Using an initial set of tracks, a segmentation of tracklets is performed to obtain activity segments. These form the higher level nodes. Edges model relationships between potentially associated tracklets, tracklets and their corresponding activity segments, and the spatiotemporal context information between activity segments. The node potentials and edge potentials are marked in the graph.

The overall energy function of the HMRF is given by

$$E(X_t, X_a, T) = \frac{1}{Z} \exp(-\Psi(X_a, X_t, T)), \tag{1}$$

$$\begin{aligned}\Psi(X_a, X_t, T) = &\sum_{x_{t_i}} \mathbf{w_o^{t_i}} \psi_o(x_{t_i}, y_{t_i}) + \sum_{x_{a_i}} \mathbf{w_o^{a_i}} \psi_o(x_{a_i}, y_{a_i}) \\ &+ \sum_{x_{t_i}} \sum_{x_{t_j} \in N(x_{t_i})} \mathbf{w_a^{t_i,t_j}} \psi_a(x_{t_i}, x_{t_j}) \\ &+ \sum_{x_{t_i}} \sum_{x_{a_j} \in N(x_{t_i})} \mathbf{w_c^{t_i,a_j}} \psi_c(x_{t_i}, x_{a_j}) \\ &+ \sum_{x_{a_i}} \sum_{x_{a_j} \in N(x_{a_i})} \mathbf{w_{st}^{a_i,a_j}} \psi_{st}(x_{a_i}, x_{a_j}),\end{aligned} \tag{2}$$

where $\psi_o(.)$ is the observation potential computed over both levels, $\psi_a(.)$ is the association potential, $\psi_c(.)$ is the consistency potential and $\psi_{st}(.)$ is the spatiotemporal context potential of the HMRF. Z is the normalization constant. Here, $\mathbf{w_o}$ is the model parameter for the observation potentials and $\mathbf{w_a}$, $\mathbf{w_c}$ and $\mathbf{w_{st}}$ are the corresponding model parameters for the edges of the graphical model, represented using similar superscripts. It is to be noted that for a multi-state model such as in this case, with the nodes taking n states, each edge parameter is a matrix of n^2 elements.

A. Computation of Potential Functions of HMRF

We will now describe the four kinds of potential functions mentioned above in detail.

1) Observation Potential: Each node of the graph (lower or higher level) is associated with an observation potential. At the lower level the observation potential is obtained from the image features associated with the tracklet corresponding to the node, while at the higher level, it is obtained from the image features of *all* the tracklets that link to the higher level

node. Here, we utilize space-time interest points [10] as well as object attributes [39] to learn a multi-class SVM classifier in a Bag-of-Words formulation. This is also referred to as the baseline classifier. The observation potential of a node x_i is therefore defined as

$$\psi_o^c(x_i, y_i) = -\log(P(x_i = c|y_i)), \tag{3}$$

where ψ_o^c is the observation potential for a node x_i (tracklet or activity segment) and y_i is its observed feature descriptor. It is to be noted that any other set of features or algorithms can also be used for the baseline classifiers.

2) Association Potential: The association potential is defined on the edges connecting tracklets which are hypothesized to be associated with each other. The association potential models the likelihood of association of two tracklets by measuring the compatibility of activities taking place in the two tracklets. The association potential for two tracklets belonging to activity class c_a and c_b is given by

$$\psi_a(x_{t_i}, x_{t_j}) = d_{ij}\mathbf{I}(x_{t_i}, x_{t_j}), \tag{4}$$

where $\mathbf{I}(a, b)$ is an indicator function which returns 1 if the features belonging to tracklet a and the features belonging to tracklet b map to the same activity label and 0 otherwise.

3) Consistency Potential: The consistency potential is defined on the edges connecting tracklets to their corresponding activity segments. This potential function models the compatibility in the hierarchy between the lower level nodes and the higher level nodes which contain the same spatio-temporal region. The consistency potential is given by

$$\psi_c(x_{t_i}, x_{a_j}) = \exp(-k_{ij})\mathbf{I}(x_{t_i}, x_{a_j}), \tag{5}$$

where k_{ij} is the difference in the observation potentials of x_{t_i} and x_{a_j}. $\mathbf{I}(.)$ is the indicator function which returns 1 if a tracklet belongs to the same activity class as the activity segment to which it corresponds and 0 otherwise.

4) Spatio-temporal Context Potential: The spatio-temporal context potential is defined on edges connecting the action segments in the graph. Actions which are within a spatio-temporal distance of each other are assumed to be related to each other. There are three components to this potential: the spatial component, the temporal component and the frequency component.

The spatial and temporal components are modeled as normal distributions whose parameters μ_s, σ_s, μ_t and σ_t are computed using the training data. The spatial and temporal centroid of x_{a_i} and x_{a_j} is given by (s_i, t_i) and (s_j, t_j). The spatial component models the probability of an activity belonging to a particular category given its spatial configuration with its neighbor. The spatial potential is defined as

$$\psi_s(x_{a_i}, x_{a_j}) = N_{sd}(\|s_i - s_j\|^2; \mu_s(c_i, c_j), \sigma_s(c_i, c_j)), \tag{6}$$

Similarly, the temporal component models the probability of an activity belonging to a particular category given its temporal distance with its neighbor. The temporal potential is defined as

$$\psi_t(x_{a_i}, x_{a_j}) = N_{td}(\|t_i - t_j\|^2; \mu_t(c_i, c_j), \sigma_t(c_i, c_j)). \tag{7}$$

where $\mu_s(c_i, c_j)$,$\sigma_s(c_i, c_j)$, $\mu_t(c_i, c_j)$ and $\sigma_t(c_i, c_j)$ are the parameters of the distribution of relative spatial and temporal positions of the activities, given their categories.

The frequency component is the probability of two activities being within a pre-defined spatio-temporal vicinity of each other. The association probability $F(a_i, a_j)$ is computed as a ratio of the number of times an activity category c_j has occurred in the vicinity of activity category c_i to the total number of times the category c_i has occurred. The value of F is varied between a minimum and maximum value, both of which is greater than 0 and less than or equal to 1. Therefore, the spatio-temporal potential is given by

$$\psi_{st}(x_{a_i}, x_{a_j}) = F(a_i, a_j)\psi_s(x_{a_i}, x_{a_j})\psi_t(x_{a_i}, x_{a_j}). \tag{8}$$

B. Structure Discovery using L1-regularized parameter learning

1) Motivation for automatic structure discovery: For the model described above, learning involves determining two factors - the structure of the model, i.e. learning the edges of the graph, and learning the parameters of the model corresponding to this structure.

The effectiveness of a graphical model depends on the structure as well as the parameters chosen for the model. In the case of unconstrained videos such as surveillance videos, the graph structure varies with the number of people and activities in the video. Prior approaches such as [16] have fixed the graph apriori or used dense graphs as an alternative. The disadvantage of a dense graph is that the number of parameters to be estimated in the model grows exponentially with the number of edges. This makes the computation of parameters statistically inefficient and the model inaccurate. In addition, it may not be practical to fix the graph in some applications.

Sparsity has widely been used in different applications where it is advantageous to have a small set of parameters that effectively model the data. In continuous videos with a variable number of activities and people, the total number of possible contextual relationships can be exponential in the number of activities. However, in reality, the number of activities which are actually related to each other tends to be a small subset of all possible relationships. For example, two people in the scene may be acting independently and may not influence actions performed by each other. Similarly, a preceding action may provide sufficient context to the next action, while the other relationships may not be as significant. Therefore, by learning a sparse set of parameters, and in turn a sparse graph, we can effectively retain those contextual relationships which tend to influence the recognition scores to a greater extent, while also reducing the computational complexity involved in solving a dense graph.

L1-regularized learning is a useful tool to select a sparse set of features which represent a particular data. Different methods of sparse dictionary learning such as deep Boltzmann machines [24], stacked auto-encoders and sparse coders [13] have been used to represent image data in the context of object recognition. These concepts have been extended to video data in approaches such as 3D convolutional neural networks [9] and independent subspace analysis [14]. Such approaches

have demonstrated competitive performances in classification. However, most computer vision approaches which have used L1-regularized optimization have only explored sparsity in feature representation and not in structure representation. In this section, we show how to extend the concept of sparse feature learning to estimating a sparse set of relationships between events in continuous videos. Here, we perform a simultaneous learning of the structure and parameters of the model using an L1-regularized learning.

2) Standard L1-regularization of parameters: : The graphical model consists of a set of potentials and a set of parameters. As described in Equation 2, there are three kinds of parameters described on the edges of the graph - $\mathbf{w_a}$, $\mathbf{w_c}$ and $\mathbf{w_{st}}$. The overall parameter vector of the model is therefore formed by concatenating the weight vectors of all the potential functions, given by $\mathbf{w} = [\mathbf{w}_a^T, \mathbf{w}_c^T, \mathbf{w}_{st}^T]^T$. We begin by computing the potential functions on a densely connected graphical model. While we could use a fully connected graphical model, here, we assume that nodes within a specified spatiotemporal distance can influence each other contextually. Therefore, we build a graph where every node is connected to all nodes within a specified spatiotemporal distance. The structure discovery using L1-regularization of parameters is carried out as given below.

We wish to learn the structure of a sparsely connected graph, which represents the contextual relationships in the data. We propose to do this by an setting a sparsity condition on the parameters. A sparse set of parameters also results in a sparse set of edges, since setting a parameter to zero sets the corresponding potentials of the energy function to 1. We also set a sparsity constraint on the node parameters for effective feature coding. The non-zero node parameters would specify the sparse node features which are chosen to model the activities. The non-zero edge parameters would specify the edges which encode important contextual information between activities. This is done by imposing a restriction on the L1-norm of the parameter vector. For a set of m training instances and n nodes in the graph, the L1-regularized learning problem can be given by

$$F = \min_{\mathbf{w}} -\sum_{k=1}^{m}[\sum_{i=1}^{n}[\mathbf{w}_o\psi_o(x_o, y_o) + \sum_{j\in N(i)} \mathbf{w}_{ij}\psi_e(x_i, x_j)]] + m\log Z(w) + \lambda|\mathbf{w}|_1 \tag{9}$$

Here, λ is the regularization parameter which decides the sparsity of the resultant solution. This poses the structure learning as an optimization problem. This is a useful formulation for learning the graphical model since it does not impose any constraint on the structure and is also much faster than the search based method of edge addition/deletion.

3) Group L1-regularization of parameters: : In the above formulation, each node can take as many states as the number of meaningful activities in the data. For multi-state nodes representing n activities, the potential function can take n^2 values for each edge. Each edge is therefore represented by an edge parameter $\mathbf{w}$ which is composed of a matrix of n^2 elements, given by w_{ij}, where $i, j \in \{1, 2, .., n\}$

We want to learn the edges of a graphical model, each edge parameter representing the joint distribution of a node given the neighbor. The edge is reduced to zero only if *all* elements of the edge are set to zero. This is achieved by the L2-regularization of the n^2 elements over each edge. However, each non-zero edge in this case tends to have all parameters set to non-zero elements. To introduce sparsity for the elements of the non-zero edges, we introduce the l1-regularization over this function. This leads us to the group l1-regularization, which is defined as the l1-regularization of l2-norm of $\mathbf{w}$. Since there are three kinds of edge potentials in the graph, we form three regularization factors for the three sets of edges with the flexibility to choose three different regularization parameters. The optimization function therefore reduces to

$$\begin{aligned} F = \min_{\mathbf{w}} &-\sum_{k=1}^{m}[\sum_{i=1}^{n}[\mathbf{w}_o\psi_o(x_o, y_o) + \sum_{j\in N(i)} \mathbf{w}_{ij}\psi_e(x_i, x_j)]] \\ &+ m\log Z(w) + \lambda_c \sum_{i\in E_c}\sum_{j\in N(i)} \|\mathbf{w_c}\|_2 + \lambda_a \sum_{i\in E_a}\sum_{j\in N(i)} \|\mathbf{w_a}\|_2 \\ &+ \lambda_{st} \sum_{i\in E_{st}}\sum_{j\in N(i)} \|\mathbf{w_{st}}\|_2 \end{aligned} \tag{10}$$

This function can be viewed as a sum of a differentiable convex function and a convex regularizer. We solve this using the Barizilai-Borwein spectral projection method [25]. This method views the equation as a constrained optimization problem, with a series of group constraints. In the group regularization, the constraint given in the form of $\sum_{\mathbf{g}} \lambda_g \mathbf{w}_g$, replaces the non-differentiable regularizer with a linear function. The function is solved using a variant of the projected-gradient method with a variable step size. Therefore, we now have a smooth optimization problem over a convex set.

The spectral projection method solves for the parameters in an iterative manner. In each iteration, the value of the parameters is changed in the direction of the projection of the current values on the function space, i.e.,

$$\mathbf{w_{k+1}} = \mathcal{P}_f(\mathbf{w_k} - \alpha\nabla F(\mathbf{w_k})), \tag{11}$$

where $\mathcal{P}_f$ represents a Euclidean projection and α is the step size. For details of the spectral projection method, please refer [25]. The final solution introduces sparsity for the edges of the graph using the L1 constraint on the groups, as well as within each group by minimizing the total number of parameters.

IV. Inference on the HMRF

Given an initial set of tracks, activity labels are obtained by inference on the HMRF using the learned parameters. Inference on a graphical model involves computing the marginal probabilities of the hidden or unknown variables given an evidence or an observed set of variables. There are two steps in our inference algorithm which are alternated in an EM framework to obtain the solution to the tracking and activity recognition problems. Using a set of pre-computed tracks, we obtain a set of activity labels in a bottom-up inference strategy. Next, using the obtained activities, tracks are re-computed in a top-down processing. These steps are explained in detail below.

A. Bottom-up Inference: From tracks to activities

Inference is the task of estimating labels of activities using the computed parameters. Due to the loopy nature of the graph, an exact solution is intractable. We consider an approximate objective to solve this optimization. A pseudo-likelihood function is computed by replacing the likelihood with univariate conditionals. A grouping of consecutive actions taking the same activity labels gives activity regions. Output of the algorithm is the labels of activities and the structure of the graphical model.

We choose the belief propagation method for inference on the graph. At each iteration, a node sends messages to its neighbor. All nodes are updated based on the messages from their neighbors. Consider a node $x_i \in V$ with a neighborhood $N(x_i)$. The message $m_{x_i,x_j}(x_j)$ sent by a node $x_i \in V$ to its neighbor $x_j \in V, (x_i, x_j) \in E$ can be given as

$$m_{x_i,x_j}(x_j) = \alpha \int_{x_i} \Psi(x_i, x_j)\Psi_o(x_i, y_i) \prod_{x_k \in N(x_i)} m_{x_k,x_i}(x_i)dx_i \tag{12}$$

Here $\Psi(x_i, x_j)$ is taken as the association, consistency or spatiotemporal potential depending on the level of the nodes which it connects. We solve the inference problem starting with the lower level nodes and propagate the message to the higher level nodes. The marginal distribution of each activity region is given by

$$p(x_i) = \alpha\psi_o(x_i, y_i) \prod_{x_j \in N(x_i)} m_{x_j,x_i}(x_i) \tag{13}$$

The spatio-temporal region is said to belong to that category which has the highest marginal probability.

We use the loopy belief propagation algorithm due to its proven excellent empirical performance [15]. However, other variational inference methods such as the mean-field approximation can also be used for inference.

B. Top-down Inference: From activities to tracks

Tracks are to be formed by associating non-overlapping tracklets. Knowledge about the activities a person conducts in a given time interval can help in estimating his position and thereby the tracklet association. Therefore, in addition to the cost due to feature similarities, the compatibility of two tracklets given the activities that are being performed by the actor in the spatiotemporal region represented by the tracklets, given by the association potential, is utilized in the tracklet association algorithm.

The tracklet association is posed as a min-cost network problem as given in [36]. For a set of tracklets $t_1, t_2...t_n$, a set of m tracks $T_1, T_2, .., T_m$ are to be identified, such that, each track contains one or more tracklets. This can be accomplished by finding a set of m possible paths between two tracklets t_i and t_j, given by $h^1_{ij}, h^2_{ij}, .., h^m_{ij}$ known as the match hypotheses. Each hypothesis is associated with a cost of matching, given by d^k_{ij}. The tracks are defined as a matching function $T(f)$ of a set of binary flow variables f, estimated as

$$\begin{aligned}\hat{f} =& \underset{f}{\text{argmin}} P(f, X_t) = \underset{f}{\text{argmin}} \sum_i d_{en} f_{en,i} + \sum_i d_{ex} f_{i,ex} \\ &+ \sum_{ij} d_{ij} f_{ij} + \sum_{ij} w_a^{c_i,c_j} \psi_a^{(c_i,c_j)}(x_{t_i}, x_{t_j}) f_{ij}\end{aligned} \tag{14}$$

Here, f represents the set of binary flow variables indicating whether the tracklet i is an entry point $f_{i,en}$ of a track, exit point $f_{i,ex}$ of a track or a transition f_{ij} to another tracklet. Therefore, $f_{en,i}, f_{ex,i}, f_{ij} \in \{0, 1\}$. Every node can either be an entry node, an exit node, or be associated with a neighboring tracklet j. Therefore, $f_{en,i} + \sum_j \sum f_{ji} = 1, f_{i,ex} + \sum_j \sum_j f_{ij} = 1$.

The first and second constraints are binary constraints that model the cost associated with the image or motion features for inflow and outflow, given by d_{en} and d_{ex} respectively, the third constraint d_{ij} models the cost of association of two tracklets based on image or motion similarities. This matching cost is given as a weighted combination of distance between the color histograms of the tracklets and the spatiotemporal distance between them. The fourth term models the association cost of two tracklets t_i and t_j performing actions c_i and c_j and models the compatibility between activities performed by the tracklets. This term integrates the information from the higher-level activity nodes to the inference of tracks.

The match hypotheses for a set of tracklets can be found using the K-shortest path algorithm [5]. An initial set of tracks are computed using just the binary constraints. Activity segmentation is conducted on these set of tracks by running a baseline classifier on the tracklets and grouping adjacent tracklets of a track belonging to the same activity into a single activity segment. The HMRF is constructed on these tracklets and activity segments. Using the obtained labels from recognition, the cost matrix is updated and the tracks are re-computed. The algorithm is repeated with the modified tracks.

C. Bi-directional processing for tracking and activity recognition

As explained in Section III, the activity labels of the HMRF can be obtained by maximizing the energy function $E(X_t, X_a, T)$ in Equation 2, or in other words, minimizing $\Psi(X_t, X_a, T)$, i.e.

$$\begin{aligned}\hat{X} &= \underset{X_t, X_a}{\text{argmax}} E(X_t, X_a, T) \\ &= \underset{X}{\text{argmin}} \Psi(X_t, X_a, T)\end{aligned} \tag{15}$$

This is dependent on knowing the tracks T which are used to compute the nodes and edges of the graph as seen from Equation 2. Alternately, the track association problem utilizes the association potential which requires the activity labels assigned to the tracklets as can be seen from Equation 14. We can see that both X and T are dependent on each other. We propose to solve the tracking and activity recognition problems simultaneously. Since both X and T are unknown, this can be solved as an expectation maximization problem by iterating between two steps.

Algorithm 1 Algorithm for integrated tracking, localization and labeling of activities in a test sequence using HMRF.

Input: $\mathcal{S}_{\mathcal{R}} = \{V_1 \dots V_{N_{\mathcal{R}}}\}$ Set of training videos containing activity annotations
A continuous test video containing one or more activities.
Output: Labels of activities $\{x_{a_1} \dots x_{a_q}\}$ and tracks $T_1 \dots T_k$
Initial tracking: Generate hypotheses on tracklets and get an initial estimates of tracks $f^{(1)}$.
Training: Train baseline classifiers $c_1 \dots c_N$ for N activities and model the association potential $\psi_a(x_{t_i}, x_{t_j})$, consistency potential $\psi_c(x_{t_i}, x_{a_j})$ and spatio-temporal potential $\psi_{st}(x_{a_i}, x_{a_j})$ between all pairs of activities using annotated training videos. Initialize hierarchical MRF G containing p tracklets and q activity segments. Perform L1-regularized structure learning to learn a set of sparse edges representing contextual information and model parameters w.

Testing:

1) Tracklets form the lower level node $\{x_{t_1}, x_{t_2} ... x_{t_p}\}$. Run baseline classifiers to compute labels l_{old} for all nodes and initial activity segmentation using current tracks $\{x_{a_1}, x_{a_2}, ... x_{a_q}\}$.
2) Compute observation potential $\psi_o(x_{t_i}, y_{t_i})$ for each tracklet and $\psi_o(x_{a_i}, y_{a_i})$ activity segment using the baseline classifiers.
3) **E-Step:** Run inference to generate posteriors and labels for all nodes l_{new}.
4) **M-Step:** Recompute association potential using current labels l_{new}. Solve Equation 14 using the revised potential and recompute tracks $f^{(new)}$.
5) Compute new localization using $f^{(new)}$. Rebuild graph.
6) Repeat the EM algorithm until $l_{old} = l_{new} \parallel n_{iter} = max_{iter}$
7) Output tracks $T(f)$ and current labels for $\{x_{a_1}, x_{a_2}, ... x_{a_q}\}$.

E-Step: The expectation step computes the conditional expectation of the node labels $X^{(p)}$ given the parameters of the HMRF and the current estimation of the tracks given by $f^{(p)}$. This can be shown to be obtained as the posterior probabilities of the graphical model given by $X^{(p)} = \underset{X}{\operatorname{argmin}} \Psi(X_t, X_a, T(f^{(p)}))$. This can be solved as described in Section IV-A.

Maximization Step: The maximization step revises the flow parameters given the current node labels. We recompute the spatiotemporal context potential between the tracklets for all hypotheses and recompute the flow variables as $f^{(p+1)} = \underset{f}{\operatorname{argmin}} P(f, X_t^{(p)})$. This can be solved as described in Section IV-B.

The overall algorithm of our proposed method is explained in Algorithm 1.

V. Experiments and Results

A. Dataset

The goal of our approach is to demonstrate bi-directional processing for tracking and activity recognition in continuous videos. Therefore, we perform experimentation on long duration realistic videos. We evaluate our system on two challenging, realistic datasets containing long duration activities: 1)The UCLA office dataset and 2)VIRAT ground dataset [18].

The UCLA office dataset [20] consists of indoor and outdoor videos of single and two-person activities. Here, we perform experiments on the lab scene containing close to 35 minutes of video captured with a single fixed camera in a room. We work on 10 single person activities:1 - enter room, 2 - exit room, 3 - sit down, 4 - stand up, 5 - work on laptop, 6 - work on paper, 7 - throw trash, 8 - pour drink, 9 - pick phone, 10 - place phone down. The first half of the data is used for training and the second half for testing. Each activity occurs 6 to 15 times in the dataset.

The VIRAT dataset is a state-of-the-art activity dataset with many challenging characteristics, such as wide variation in the activities and a high amount of clutter and occlusion. We work on the parking lot videos involving single vehicle activities, person and vehicle interactions, and people interactions. The length of the videos vary between $2 - 15$ minutes and containing up to 30 activities in a video. For every scene, the first half is used for training and the second half for testing.

We perform two sets of experiments on the VIRAT dataset, one on Release 1 and the other on Release 2 of the data. For Release 1, there are 6 activities which are annotated: 1 - loading, 2 - unloading, 3 - open trunk, 4 - close trunk, 5 - enter vehicle, 6 - exit vehicle. In release 2, additional 5 activities have been added: 7 - person carrying an object, 8 - person gesturing, 9 - person running, 10 - person entering facility and 11 - person exiting facility.

For both datasets, we perform two sets of experiments, one with the dense graphical model (without L1-regularized learning) and the other with the sparse graphical model. The dense model assumes that all nodes within a pre-defined distance of each other are connected by an edge.

B. Methodology

We evaluated Space-Time Interest Points (STIP) from [10] and dense trajectories from [33] as the features for our experiments. It was found that the STIP features perform better for the datasets which we have used. This could be due to the fact that we work on scenes captured at a distance. The dense trajectory approach was not suitable for distant scenes like the ones we are working with here. Since the persons involved in an activity occupy a small part of the frame, the dense trajectories would require a high spatial sampling to capture the activity successfully. The recognition accuracy using STIP features was found to be approximately 10% higher than that of dense trajectories. Therefore, we utilize STIP features in our experiments. Similarly, we choose the bag-of-words against other approaches such as String of feature graphs [7] for the baseline classifier because feature relationships are not as prominent in a distant scene and graph matching can be computationally expensive.

A radial basis function kernel has been used for the SVM. We have used libsvm in our experiments. The regularization values were chosen experimentally using cross validation. Half the training set was used to learn the model and we used a total of 10 iterations during cross validation. The object attributes used in this approach are the same as in [39]. The detection of activity segments is performed using a sliding window approach. We use windows of two sizes (60 and 90 frames) with 50% overlap on the pre-computed tracks. We have used STIP interest points with feature vector dimension of 300. A radial basis function kernel has been used for the SVM. We have used libsvm in our experiments. The regularization

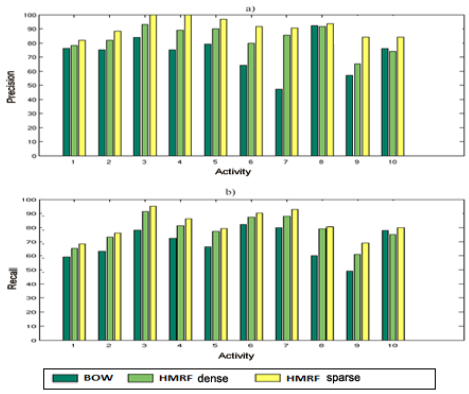

Fig. 4. The figure shows the precision and recall obtained on the UCLA office dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [10] as well as our context model with and without L1-regularization. The activities are mentioned in Section V-A.

values were chosen experimentally using cross validation. Half the training set was used to learn the model and we used a total of 10 iterations during cross validation. The activity classifier used is STIP along with multi-class SVM which has been used to compute the observation potential. The output of the activity classifiers run over the sliding windows provide recognition scores. These scores are then used to group regions into activity segments.

During training, we normalize all distances with respect to the scale of the video to make the approach invariant to scale. A threshold was set on the spatio-temporal distance between activities to initiate the dense graph. We used the distance threshold as a bounding box of 4 times the average dimensions of the person in the scene and a time threshold of 20 seconds. These values have been fixed experimentally. The graphical model is constructed on individual activity sequences. The regularization parameters experimentally determined where $\lambda_c = 3$, $\lambda_a = 3$, $\lambda_{st} = 4$.

To evaluate the accuracy of activity recognition, if there is more than a 40% overlap in the spatiotemporal region of a detected activity as compared to the ground truth and the labeling corresponds to the ground truth labeling, the recognition is assumed to be correct. Some examples of data which were correctly identified using our approach while incorrectly identified using a dense graphical model are shown in Figure 7.

C. Results on UCLA office dataset

For the UCLA dataset, we consider the single person activities. Comparison of the overall accuracy of our approach to [20] is shown in Table I. In [20], activity localization was assumed to be known. With simultaneous tracklet association and recognition - a significantly harder problem we get improved performance. The table shows the overall accuracy obtained with the dense graph (without L1-regularization) as well as with the sparse graph. It can be seen that the results have improved with the introduction of sparsity. This can be attributed to a better structure that captures the important contextual relationships. The details of events which have been

Method	**BOW**[10]	**Pei[20]**	**HMRF dense**	**HMRF sparse**
Accuracy	77.7	90.6	91.1	93.5

TABLE I
RECOGNITION ACCURACIES OF METHODS BOW [10], PEI [20], DENSE HMRF AND SPARSE HMRF FOR THE UCLA DATASET.

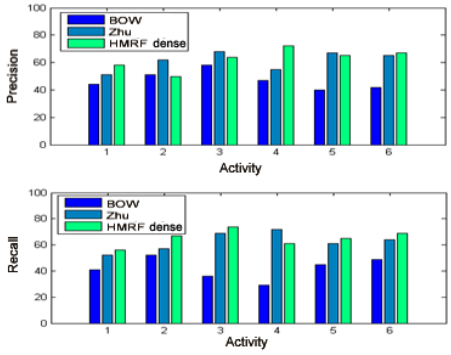

Fig. 5. The figure shows the precision and recall obtained on the VIRAT release 1 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [10] as well as Zhu et al [38]. The activities are listed in Section V-A.

classified in this data and the accuracy of recognition for each event have not been provided by the authors. Therefore, we provide per-event comparison to the baseline classifier, which is the Bag-of-Words. The values of precision and recall with and without L1-regularization are shown in Figure 4. It can be seen that the use of HMRF increases the recognition accuracy as compared to BOW in most cases.

D. Results on VIRAT dataset using dense graph

The classification results on VIRAT release 1 data using the dense graph is shown in Figure 5 and the results on VIRAT release 2 data is shown in Figure 6. Here, in addition to providing comparison with BOW, we also provide comparison against two recent approaches [1] and [38]. Authors in [38] utilize spatiotemporal context, while the authors in [1] utilize sum-product networks on low level features to localize foreground objects and label activities. However, both these approaches divide the video into shorter duration time clips for analysis. There is an improvement on using the HMRF as against the baseline classifiers (BOW). Our results are comparable to that in [1] and [38]. Although the overall accuracy is slightly lower with our approach, we consider the joint labeling of activities and tracking in our approach which is necessary for continuous videos, whereas the other methods deal with the labeling problem. Table II and III show the overall precision and recall values on VIRAT release 1 and reslease 2 data respectively using the dense graph. Figures 5 and 6 show comparison with [38] only since the recognition scores of individual activities are not given in [1].

E. Results on VIRAT dataset with sparse graph

The classification results on VIRAT release 1 data using L1-regularization is shown in Figure 8. The overall precision

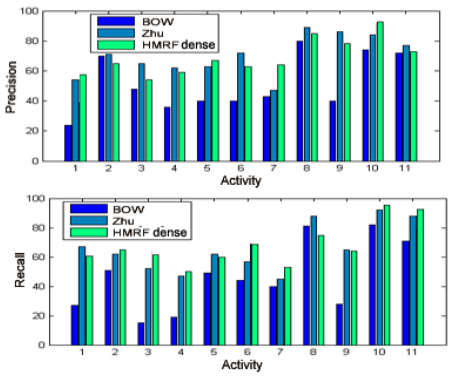

Fig. 6. The figure shows the precision and recall obtained on the VIRAT release 2 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [10] as well as Zhu et al [38]. The activities are listed in Section V-A.

Method	BOW[10]	Gaur[7]	Zhu[39]	dense HMRF	sparse HMRF
Precision	47.2	51.6	61.7	62.6	65.2
Recall	45.8	57.8	62.9	62.7	64.8

TABLE II
OVERALL PRECISION AND RECALL VALUES OF METHODS BOW [10], GAUR ET. AL[7], ZHU ET. AL [39] AND OUR APPROACH WITH THE DENSE AND SPARSE GRAPH FOR THE VIRAT RELEASE 1 DATASET.

and recall values with structure learning for VIRAT release 1 and comparison with recent approaches [7] and [39] is provided in Table II. It can be seen that the performance of our approach is better than the recent state-of-the-art methods for most activities. The overall performance is also better than that achieved with the dense graph. This improvement can be attributed to the improvement in structure, which captures the relationships across activities effectively.

Similarly, we compute the sparse graphical model and the activity recognition scores for VIRAT release 2 dataset consisting of 11 activities. The precision and recall values obtained are shown in Figure 9. It can be seen that our approach performed better than the current state-of-the-art methods. The overall accuracy of our method and other recent approaches is shown in Table III. Again, it can be seen that the use of sparse graph gives significant improvement in the overall accuracy as compared to the dense graph.

F. Structure Discovery

The sparse structure discovered by the L1-regularized learning for the parameters w_{ij} of an edge for VIRAT release 1 is shown in Figure 10 a). The structure represents the contextual relationships modeled in a parameter $\mathbf{w_k}$. It can be seen that 9 relationships out of 15 possible combinations of 6 activities were retained. In addition, it was also observed that the connections learnt could be intuitively justified as the contextual relationships between activities that are often observed in the training data. For example, loading and unloading are often related to opening and closing the trunk. These edges of the graph were retained, while some others, such as the edge connecting loading to unloading was deleted. Only about

Method	BOW[10]	Amer[1]	Zhu[39]	dense HMRF	sparse HMRF
Precision	50.3	72	71.5	67.4	74.9
Recall	52	70	73.1	69.5	76.7

TABLE III
PRECISION AND RECALL VALUES OF METHODS BOW [10], SPN [1] AND ZHU ET AL [38] AND OUR APPROACH USING THE DENSE AND SPARSE GRAPH FOR THE VIRAT RELEASE 2 DATASET.

Fig. 7. A few examples of activities which were incorrectly detected using a dense graphical model ($\lambda = 0$) and correctly discovered after the L1-regularized parameter learning. The advantage of learning a sparse graph is better representation of contextual information.

32.9% of the parameters were non-zero in the resulting model. The histogram of computed parameters is shown in Figure 10 b). We also demonstrate the sparse graph obtained by structure discovery in an activity sequence from VIRAT release 1 in Figure 11. A dense graphical model was constructed by adding an edge between every two nodes which had a spatio-temporal distance of less than half the maximum separation between activities in the sequence. After L1-regularization, those edges whose parameters have been set to zero were deleted resulting in the sparse graph.

For the 11 activities of VIRAT release 2, we demonstrate the contextual relationships captured in the parameter matrix $\mathbf{w}$ in Figure 12 a). Again, it was seen that our approach captured the contextual relationships which seemed most intuitive. For example, the activity running was mostly associated with people entering/exiting the facility or exiting a vehicle and opening a trunk. These relationships are seen in the resulting graph. The histogram of the computed parameters is also shown in Figure 12 b). From the histogram, it is evident that the parameters are very sparse, thereby eliminating edges of the graph. For one sequence of activities containing 7 meaningful activities, the resulting sparse graph after structure discovery is shown in Figure 13. About 31.3% of the parameters were retained after the L1-regularized learning.

G. Tracking results on VIRAT release 2

Two examples of tracking results are shown in Figure 14. In the first case, we have a sequence of activities performed by a single person in the presence of occlusion. While the absence of context terminates the track due to the presence of occlusion, the presence of feedback detects that a trunk has been opened and it is very likely that the same person would close the trunk. Therefore, track is not terminated. Similarly, the second example shows two persons loading a trunk. While there is an error in the tracks in the absence of feedback, it is seen that the addition of feedback takes into account the fact that the person getting out of the vehicle is very likely to enter the vehicle (as often witnessed in the training data) and corrects the tracks.

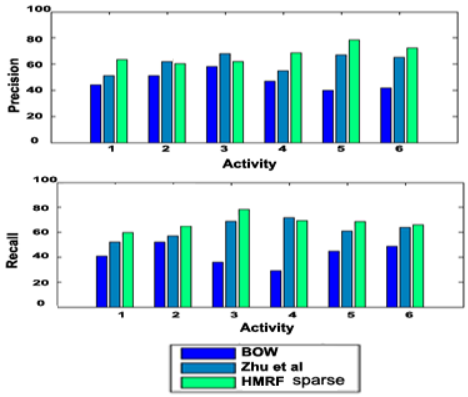

Fig. 8. The figure shows the precision and recall obtained on the VIRAT release 1 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [10] as well as Zhu et al [39]. The activities are listed in Section V-A.

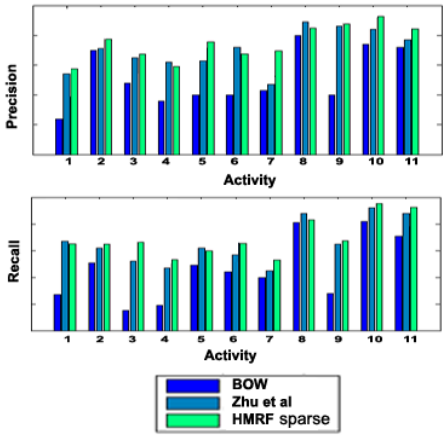

Fig. 9. The figure shows the precision and recall obtained on the VIRAT release 2 dataset with our approach. Comparison has been shown to the performance of baseline classifier BOW [10] as well as Zhu et al [39]. The activities are listed in Section V-A.

For a qualitative evaluation of tracking using our approach, there is no prior research which has provided results on tracking that we can compare with. Also, datasets that have been popular in the tracking community do not present activity recognition results. Therefore, we provide tracking results against the ground truth (GT) in Table IV. We compile the tracking results over 150 trajectories. The metrics used for

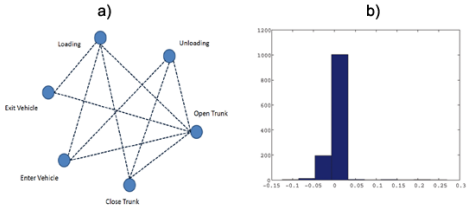

Fig. 10. The figure on the left shows the sparse contextual relationships discovered by L1-regularized learning on VIRAT 1 dataset. The edges corresponding to parameters which are set to zero have been deleted from the graph. The bar graph on the right shows the histogram of obtained sparse parameters.

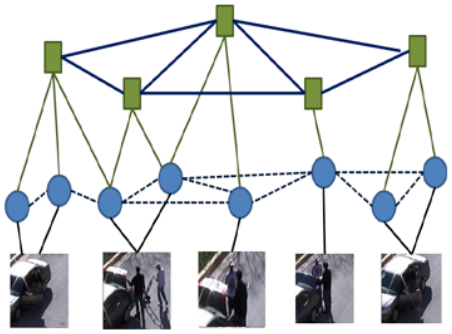

Fig. 11. For an activity sequence from VIRAT release 1 containing 5 activities, we show the sparse graphical model obtained after L1-regularized learning of parameters.

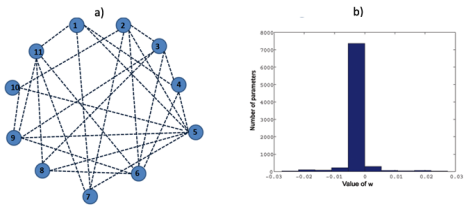

Fig. 12. Figure a) shows the sparse structure of the graph discovered by L1-regularized learning on 11 activities of VIRAT 2 dataset. The edges corresponding to parameters which are set to zero have been deleted from the graph. Figure b) shows the histogram of the learned parameters $\mathbf{w}$. From the histogram, it can be seen that $\mathbf{w}$ is sparse. The activity labels are the same as in Figure 9.

measuring the tracking accuracy are: Mostly tracked (MT): more than 80% of the track is correctly tracked; Mostly lost (ML) 20% or less tracked; Fragmented tracks (FT) Single track split into multiple IDS; ID switches (IDS) Switch between multiple tracks. It can be seen that there is a clear improvement in the tracking performance with the addition of bi-directional tracking.

H. Computational Time

It is well known that inference on a graphical model with loops is an NP-hard problem and can be tractable only with a bounded tree width [3]. While it can be solved in polynomial time in the size of the structure for select low-tree width graphs, in our case, we have an unbounded tree-width with multiple states that makes exact theoretical calculations on

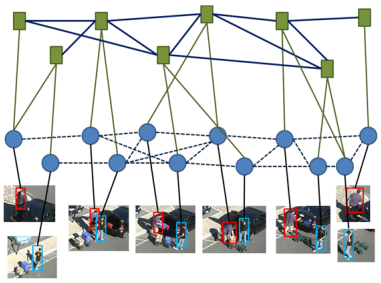

Fig. 13. For an activity sequence from VIRAT release 2 containing 7 activities, we show the sparse graphical model obtained after L1-regularized learning of parameters.

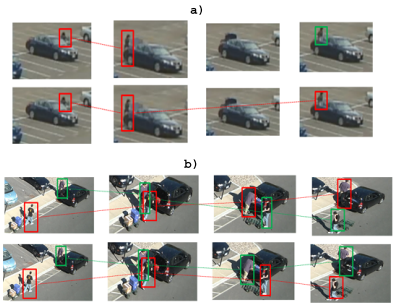

Fig. 14. The figure shows two examples where tracking is improved with the addition of context. The top row shows the tracking results without activity context while the bottom row shows the result with the addition of feedback. Red and green signify different tracks in each case. In the first case, it is seen that the track was wrongly terminated due to occlusion in the absence of context. In the second case, the tracklet association error was corrected with the addition of context.

Metric	One-Step Tracking	Bi-directional processing
GT	150	150
MT	105	121
ML	19	13
FT	36	14
IDS	35	22

TABLE IV
PRECISION AND RECALL VALUES OF METHODS BOW [10], AMER ET. AL[1] AND ZHU ET. AL [39] AND OUR APPROACH FOR THE VIRAT RELEASE 2 DATASET.

computational complexity very difficult. Also, the structure of the graph varies depending on the sequence. However, it can be said that, with the reduction in the number of edges and the introduction of sparsity, the tree-width as well as the number of loops in the graph is very likely to reduce, thereby achieving a speed-up in the performance. Experimentally, we run the approach on the dense graph (setting all values of λ to zero) and compare the taken for inference on the same set of activities using the sparse graph. Values were computed for 30 sequences containing 5 nodes on Matlab in Intel(R) Core(TM) i3 CPU @2.27GHZ . It was found that inference on the dense graph took 0.1248 seconds while the inference on the sparse graph with roughly 30% of the edges took only 0.0312 seconds. This clearly shows the improvement in speed due to sparsity. In summary, not only do we achieve higher accuracy in the graph discovery process, we do so with an order of magnitude less computational time.

VI. CONCLUSION

In this paper, we have presented a method which can perform tracking, localization and recognition of activities in continuous sequences in an integrated framework. The proposed framework uses an initial set of tracks for analysis of activities using a two-level hierarchical Markov random field. The lower nodes of the graph denote tracklets and the upper nodes denote activities. Spatio-temporal contextual relationships between activities and the influence of tracks on them has been modeled using the graph. The activity labels obtained in the bottom-up processing are in turn used to correct the errors in tracking in a top-down approach. We have demonstrated that the L1-regularized learning of parameters is a good substitute to alternate methods such as greedy forward search. The resulting graph was sparse and intuitively picked those edges which gave improved recognition scores as compared to the dense graph. The biologically inspired bi-directional processing is shown to be effective in improving the accuracy of tracking as well as activity recognition. This method can be extended to other applications which utilize graphical models for context representation

REFERENCES

[1] M. R. Amer and S. Todorovic. Sum-product networks for modeling activities with stochastic structure. In *Computer Vision and Pattern Recognition*, 2012.
[2] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *International Conference on Computer Vision*, 2011.
[3] V. Chandrasekaran, N. Srebro, and P. Harsha. Complexity of inference in graphical models. In *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence*, 2010.
[4] C.-Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *Computer Vision and Pattern Recognition*, 2012.
[5] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*, 2012.
[6] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Computer Vision and Pattern Recognition*, 2011.
[7] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. String of feature graphs analysis of complex activities. In *International Conference on Computer Vision*, 2011.
[8] M. Hoai, Z.-Z. Lan, and F. D. Torre. Joint segmentation and classification of human actions in video. In *Computer Vision and Pattern Recognition*, 2011.
[9] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
[10] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *ACM international conference on Image and video retrieval*, 2007.
[11] D. Kuettel, M. Breitenstein, L. V. Gool, and V. Ferrari. What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *Computer Vision and Pattern Recognition*, 2010.
[12] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *Computer Vision and Pattern Recognition*, 2012.
[13] Q. V. Le, M. A. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *International Conference on Machine Learning*, 2012.
[14] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition*, 2011.
[15] Y. Li and R. Nevatia. Key object driven multi-category object recognition, localization and tracking using spatio-temporal context. In *European Conference on Computer Vision*, 2008.
[16] V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *Computer Vision and Pattern Recognition*, 2011.
[17] N. Nayak, Y. Zhu, and A. Roy-Chowdhury. Exploiting spatio-temporal scene structure for wide-area activity analysis in unconstrained environments. *IEEE Transactions on Information Forensics and Security, Special Issue on Intelligent Video Surveillance*, 2013.
[18] S. Oh and et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Computer Vision and Pattern Recognition*, 2011.
[19] S. Park, S. Lee, I. Yun, and S. Lee. Hierarchical mrf of globally consistent localized classifiers for 3d medical image segmentation. *Pattern Recognition*, 2013.
[20] M. Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. In *International Conference on Computer Vision*, 2011.
[21] M. Ryoo and J. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Computer Vision and Pattern Recognition*, 2006.

[22] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *International Conference on Computer Vision*, 2009.

[23] L. S. D. S. Khamis, V. I. Morariu. A flow model for joint action recognition and identity maintenance. In *Computer Vision and Pattern Recognition*, 2012.

[24] R. Salakhutdinov and G. Hinton. A betterway to pretrain deep boltzmann machines. In *Neural Information Processing Systems*, 2012.

[25] M. Schmidt. *Graphical Model Structure Learningwith l1-Regularization*. PhD thesis, THE UNIVERSITY OF BRITISH COLUMBIA, Vancouver, 2010.

[26] M. Siegel. Integrating top-down and bottom-up sensory processing by somato-dendritic inteactions. *Journal of Computational Neuroscience*, 2000.

[27] V. K. Singh and R. Nevatia. Simultaneous tracking and action recognition for single actor human actions. *Visual Computer*, 2011.

[28] B. Song, T. Jeng, E. Staudt, and A. R. Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *European Conference on Computer Vision*, 2010.

[29] Y. Song. Mcmc-based scene segmentation method using structure of video. In *International Symposium on Communications and Information Technologies*, 2010.

[30] E. Swears, A. Hoogs, Q. Ji, and K. Boyer. Complex activity recognition using granger constrained dbn (gcdbn) in sports and surveillance video. In *Computer Vision and Pattern Recognition*, 2014.

[31] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition*, 2012.

[32] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008.

[33] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *Computer Vision and Pattern Recognition*, 2011.

[34] X. Wang and Q. Ji. A hierarchical context model for event recognition in surveillance video. In *Computer Vision and Pattern Recognition*, 2014.

[35] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74, 2011.

[36] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition*, 2008.

[37] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spation-temporal phrases for activity recognition. In *European Conference on Computer Vision*, 2012.

[38] Y. Zhu, N. Nayak, and A. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *IEEE Journal on Selected Topics in Signal Processing, Special Issue on Anomalous Pattern Discovery*, 2013.

[39] Y. Zhu, N. Nayak, and A. Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *Computer Vision and Pattern Recognition*, 2013.