A Directed Sparse Graphical Model for Multi-Target Tracking

Mohib Ullah and Faouzi Alaya Cheikh
Department of Computer Science (IDI)
Norwegian Univeristy of Science and Technology, Norway
{mohib.ullah,faouzi.cheikh}@ntnu.no

Abstract

We propose a Directed Sparse Graphical Model (DSGM) for multi-target tracking. In the category of global optimization for multi-target tracking, traditional approaches have two main drawbacks. First, a cost function is defined in terms of the linear combination of the spatial and appearance constraints of the targets which results a highly non-convex function. And second, a very dense graph is constructed to capture the global attribute of the targets. In such a graph, It is impossible to find reliable tracks in polynomial time unless some relaxation and heuristics are used. To address these limitations, we proposed DSGM which finds a set of reliable tracks for the targets without any heuristics or relaxation and keeps the computational complexity very low through the design of the graph. Irrespective of traditional approaches where spatial and appearance constraints are added up linearly with a given weight factor, we incorporated these constraints in a cascaded fashion. First, we exploited a Hidden Markov Model (HMM) for the spatial constraints of the target and obtain most probable locations of the targets in a segment of video. Afterwards, a deep feature based appearance model is used to generate the sparse graph. The track for each target is found through dynamic programming. Experiments are performed on 3 challenging sports datasets (football, basketball and sprint) and promising results are achieved.

Keywords: Sparse graph, Hidden markov model, Deep feature, Dynamic programming.

1. Introduction

In computer vision, multi-target tracking is an active field of research with application including but not limited to video surveillance [5], crowd analysis [30], robotics navigation, human behavior analysis [9], Computer Generated Images (CGI), to name a few. In the context of Tracking-by-Detection, target tracking can be divided into two discrete steps i.e. target localization and target association. Usually, target localization is achieved through a discriminative [22, 23] or generative [15] classifier. While target association is a highly complex problem and different techniques have been proposed in the literature. In a nutshell, target association can be classified into recursive and non-recursive techniques. In the class of recursive approaches, In the early days of computer vision, Joint Probabilistic Data Association Filtering (JPDAF) [8] is used for association. More recently, Markov Chain Monte Carlo (MCMC) based sampling techniques [31, 18] are exploited to solve association problem. MCMC based approaches help to represent the non-linear multi-modal distribution of the target state but as the number of targets increases in the scene, a higher number of samples are needed to represent the posterior distribution appropriately. Therefore, they are not feasible for real world scenarios. Different than sampling based approach, A Kalman filter [4] is used in [29] where HoG features model the appearance of targets. Moreover, the association between the targets are established through Hungarian algorithm. Yu et al. [33] come up with an Exchange Object Context (EOC) model where the contextual information of the targets are exploited for tracking. They define a cost function in terms of target appearance, shape, the movement and the background and used Hungarian algorithm for the optimization. These approaches are considered local because only two frames are used to associate the targets using bipartite graph matching [12].

Such approaches are computationally efficient but they are not robust for long-term tracking. Compared to local approaches, global approaches use a number of frames for association. They work best for long term tracking but at the cost of high computational complexity. In a nutshell, global approaches have two drawbacks. First, they add spatial and appearance constraints linearly with a given weight factor and second, they use a fully connected dense graph for finding reliable tracks. Due to the dense nature of the graph, finding a reliable track in the graph is NP-hard and usually some relaxation or heuristics are used in the literature to solve the optimization problem in deterministic time. Berclaz et al. [3] proposed a 3D tracking technique where the ground floor is discretized into disjoint grid and

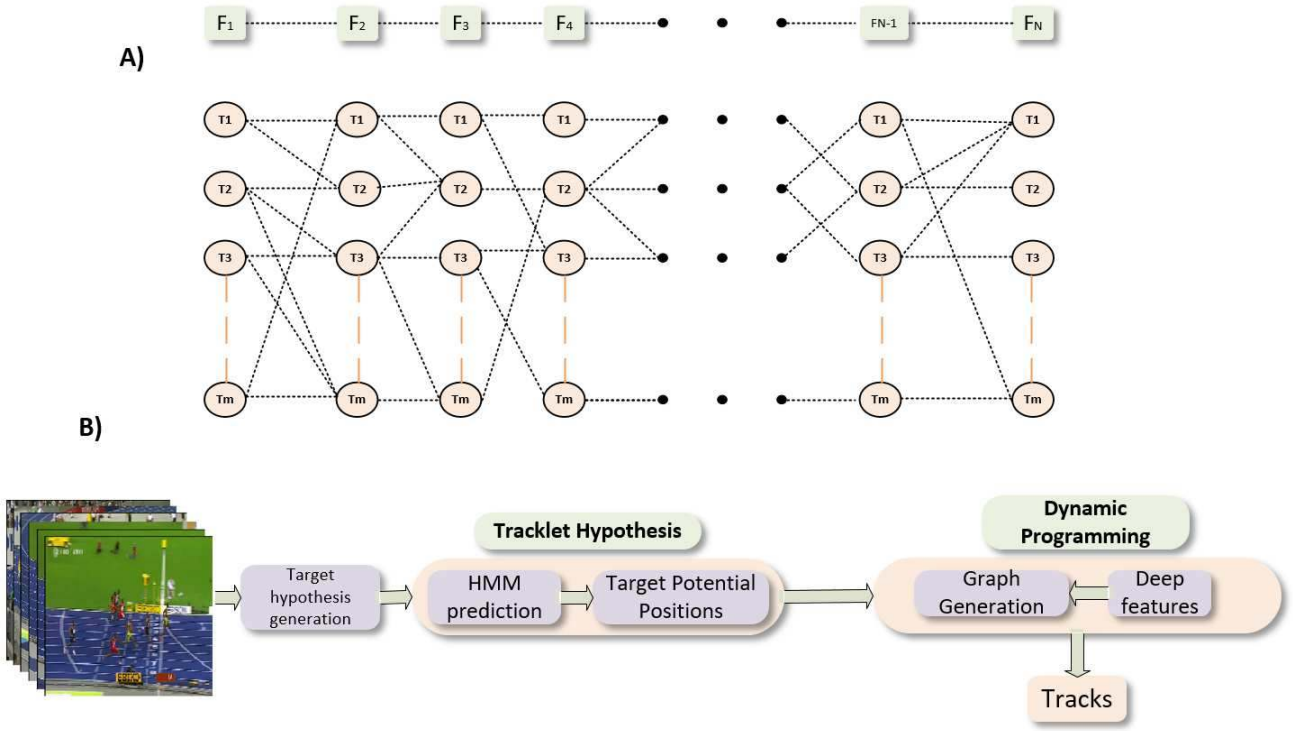

Figure 1: (A) HMM model is used to predict the possible locations of target in each frame. F_1 to F_n corresponds to the frame number. Where T_1 and T_m is the first and m^{th} target location, respectively. (B) HMM model helps to find the possible location of targets. Deep feature based appearance model is exploited to define the edge cost of the directed graph. Consequently, dynamic programming is used to get a reliable track for each target.

K-shortest path algorithm is used to find reliable tracks for the targets. Russell et al. [25] introduced a directed acyclic hyper-graph which can capture the long range interaction among the targets in the scene. The trajectories of the targets are optimized through global MAP criteria and complexity of the algorithm is linear w.r.t the number to targets, and the number of frame considered for the optimization. However, the algorithm requires an upper bound on the number of targets to be tracked in the scene. Roshan et al. [24] formulate tracking as a graph clique problem and proposed a two stage tracking framework. Initially, video is divided into clusters and local association is established within clusters through Tabu-search which gave the tracklets. In the 2nd stage, association is established among the clusters to obtain the final smooth trajectories of the target. Afshin et al. [6] used the same approach but a generalized optimization scheme was adopted and all the trajectories in the temporal window are obtained at once. This approach works well if the number of targets are small. If the number of target increases, the number of edges in the fully connected graph increases exponentially and optimization becomes very difficult as shown in Fig 2. Similarly, Zhu et al. [36] proposed a multi-stage tracking framework for multi-target tracking. Initially, Ada-boost is used to get the initial movement of the targets and get the trajectory fragments. Then Hungarian algorithm is used to optimized these fragments. And finally, energy minimization based extrapolation algorithm is exploited to achieve final trajectories. Likewise, Anton et al. [2] introduced a complicated cost function which models all the attributes of the targets and used conjugate gradient decent to solve the non-convex optimization problem. They introduce transdimentional jump moves to avoid the local minima. Similarly, Zhang et al. [35] formulated multi-target tracking as network flow problem and used conditional random field to find the reliable tracks.

The backbone of these approaches is the generation of dense graph. In the dense graph, a target could be associated to a completely irrelevant counterpart just because of the structure of the graph. Although the cost function penalizes the targets which are far away but irrelevant association are considered in the optimization scheme which makes it intractable. Motivated by this fact, we adopted a different approach to construct the graph. Spatial and appearance constraints are salient features for target association but rather than combining them in a single cost func-

tion, we use these features in a cascaded fashion. The key contributions of the paper are two-folds:

- We define a better graphical model for multi-target tracking. In the context of spatial constraints, HMM is exploited to reduce the search space for target association.
- The edge cost of the graph is defined through deep features and dynamic programming is used to find the optimal set of trajectories for the targets.

The rest of the paper is organized in the following way: In section 2, a brief overview of the proposed approach is given. In section 3, HMM is explained that helps in finding most probable locations of the targets. Section 4 explain a similarity metric that is used to define the similarity between the targets. Section 5 elaborate deep feature based appearance model that define the edge cost of the graph. In section 6, the graphic model formulation and dynamic programming based optimization strategy is explained. Quantitative results are given in section 7 and section 8 summarizes the paper with concluding remarks.

2. Proposed Method

The block digram is given in Fig 1. The input to the algorithm is the target hypothesis in each frame. Target hypothesis can be generated with a detector or manual annotation. For the spatial constraints, we adopted a Hidden Markov Model (HMM). The nodes in the graph are assumed to have linear Gaussian distribution. A constant velocity model is adopted as the dynamic model of the HMM. Each target is associated with an individual HMM. Hence, highly probable positions are predicted for each target. This scheme helps to reduce the solution space and while constructing the graph, rather than connecting nodes densely, only a subset of nodes are connected which are predicted by HMM. In Fig. 2, a graphical depiction of different graphical structure are given. While constructing the graph, targets are modeled through deep features and the similarity score of target across the frames is used as the edge cost. The similarity between two targets are found through mutual information. We process the video sequence as a whole. Once the graph is constructed with proper edge cost, dynamic programming formulation proposed in [20] is used to find reliable track for each target. In the following sections, each step of the proposed DSGM is explained.

3. Hidden Markov Model

A Hidden Markov Model (HMM) is a probabilistic sequence model which maps a set of observations to corresponding labels based on the likelihood probability. The sequence of observations could be words, sentences or pedestrian in the context of tracking. The model calculates the probability over possible labels and choose the one which maximizes the probability. For tracking, the set of observations is the targets location in a frame and the labels are the unique IDs that are assigned to it. For generating a sparse graph, HMM suits the best because most probable location of a target is obtained by combining the prior knowledge of the state and the current observations. State of an observation corresponds to a spatial location of target in the 2D space. Mathematically,

$$x_t = f_t(x_{t-1}, v_{t-1}) \tag{1}$$

x_k corresponds to a spatial location of target in the 2D space, v_{k-1} is the process noise and f_k is a function which transforms the previous state of a target to its current state. In a video sequence, the observation of a target is obtained sequentially, therefore, the goal of HMM is to estimate the optimal target state x from the available observation i.e.

$$z_t = h_t(x_t, n_t) \tag{2}$$

n_t is the measurement noise and h_t is a function which relates the observation z_t to the target state. Hence, HMM recursively approximate target state x_t at time t, considering the observation $z_{1:k}$. In probabilistic terms, the goal of state estimation can be translated to estimating the following posterior distribution:

$$p(x_t|z_{1:t}) \tag{3}$$

It is assumed that the initial pdf $p(x_o|z_o)$ is known. For target tracking, it essentially shows the initial location of a target. With the given information, the posteriors pdf $p(x_t|z_{1:t})$ is recursively obtained in two steps: prediction and update.

By generalizing the prior assumption, the pdf $p(x_{t-1}|z_{1:t-1})$ is assumed to be known. The prediction step uses the equation 1 to obtain the prior pdf at time t. In the Bayesian sense, equation 1 can be approximated through Chapman-Kolmogorov equation as follow;

$$p(x_t|z_{1:t-1}) = \sum_{x_{t-1}} p(x_t|x_{t-1}, z_{1:t-1}) \tag{4}$$

By first order Markovian assumption,

$$p(x_t|z_{1:t-1}) = \sum_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|z_{1:t-1}) \tag{5}$$

In the update step, the observation z_t is used to find the posterior pdf at time t as ;

$$p(x_t|z_{1:t}) = p(x_t|z_{1:t-1}, z_t) \tag{6}$$

Using Bayes' rule

$$p(x_t|z_{1:t}) = \frac{p(x_t|z_{1:t-1})p(z_t|x_t)}{p(z_t|z_{1:t-1})} \quad (7)$$

$$p(x_t|z_{1:t}) \approx \sum_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})p(z_t|x_t) \quad (8)$$

The denominator $p(z_k|z_{1:k-1})$ is a constant w.r.t to the state vector x and can be ignored. $p(x_t|x_{1:t-1})$ correspond to the previous known state of target and treated as the prior. For motion of the targets, a constant velocity model is assumed which approximates $p(x_{t-1}|z_{1:t-1})$. Similarly, the observation model which is the targets location in each frame is modeled by $p(z_t|x_t)$. Equations 5 and 8 form the theoretical basis of HMM. Further details are beyond the scope of this paper, interested readers my refer to [21, 32]. We instantiate a HMM model to each target. In section 4, an approach based on mutual information is explained that is used to establish congruity among the observations with corresponding model.

4. Mutual Information

Let's assume two targets $t_{1,k}$ and $t_{2,(k+1)}$ at time instant k and $(k+1)$ in a video sequence. Mutual information (MI) is a similarity metric that measure the mutual dependence between $t_{1,k}$ and $t_{2,(k+1)}$. Intuitively, it measures the information two targets share [19]. If $t_{1,k}$ and $t_{2,(k+1)}$ are completely different from each other, then knowing one can't give any information about other and hence mutual information would be zero. In the other case, when $t_{1,k}$ and $t_{2,(k+1)}$ are very similar, one can be used to represent the other. In this case, mutual information would give the entropy of $t_{2,(k+1)}$ (or $t_{1,k}$). Among different possible representation of MI, representation in terms of Kullback-Leibler divergence [14] is suitable for our tracking application. For the sake of convince, we can drop the subscript k and $(k+1)$ in the mathematical formulation and can write as,

$$M(t_1, t_2) = D_{KL}(p(t_1, t_2)||p(t_1) \times p(t_2)) \quad (9)$$

Where D_{KL} is the Kullback-Leibler divergence. $p(t_1, t_2)$ is the joint distribution of the targets and modeled by concatenating the feature vectors of the two targets. Similarly, $p(t_1)$ and $p(t_2)$ is the marginal distribution and corresponds to the feature vector of the targets. Equation. 9 can be simplified through marginalization and conditioning as;

$$M(t_1, t_2) = \sum_{t_2} p(t_2) \sum_{t_1} p(t_1|t_2) \log \frac{p(t_1|t_2)}{p(t_1)} \quad (10)$$

$$= \sum_{t_2} p(t_2) D_{KL}(p(t_1|t_2)||p(t_1))$$

$$M(t_1, t_2) = \mathbb{E}_{t_2}\{D_{KL}(p(t_1|t_2)||p(t_1))\}$$

Thus, MI is the expectation of the Kullback-Leibler divergence of the univariate distribution $p(t_1)$ of t_1 from the conditional distribution $p(t_1|t_2)$ of t_1 given t_2. Hence, MI helps to establish the correspondence between the targets location and the corresponding HMM model. In section 5, deep feature is explained that is exploited to define the $p(t_1)$, $p(t_2)$ and $p(t_1, t_2)$.

5. Deep Features

Over the past few years, techniques based on deep neural networks (DNNs) has become the method of choice for traditional computer vision problems such as image classification, object detection, speech recognition, to name a few. For vision tasks, the superior performance of such network comes from the fact that they learn high level features from the visual data through statistical learning over a large amount of labeled data set. Essentially, such DNN learns an optimal representation of the input space. The higher is the quality and quantity of labeled data, the better representation it learns. Different architectures of DNN are proposed in recent years and they have been the winners [13, 34, 28, 11] of Image-net challenge since 2012. DNNs are essentially hierarchical models and based on the depth and organization of different layers, various architectures have been proposed [13, 34, 28, 11, 27]. A common trend in these architectures are, the deeper the network, the better the performance. Depth of the network means introducing more layers in the network, which introduces more non-linearity in the network and as a result helps in mitigating over-fitting problem. Moreover, by increasing the depth of the network, features become more salient and gives better target representation [10]. For example, ResNet [11], the 2015 winner of Imagenet challenge is 20 times deeper than AlexNet [13] and 8 times deeper than VGGNet [27]. However, by increasing depth of the network i.e. introducing more layers in the network, the complexity of network increases. An optimal solution is to keep a balance between the depth of the network against the performance of the network.

A typical DNN is trained end to end but its architecture can be divided into two functional blocks i.e. feature extraction and classification. Feature extraction is done through learned filters while fully connected layer is responsible for the classification. In [26], the authors has shown that even if a Convolutional Neural Network (CNN) is trained on a generic data-set and used as a generic feature extractor, it still gives better performance than hand-crafted features. Inspired from this concept, we fine-tune a CNN model on our dataset and use it to model the appearance of the targets. Next section 5.1 briefly explain the concept of fine tuning.

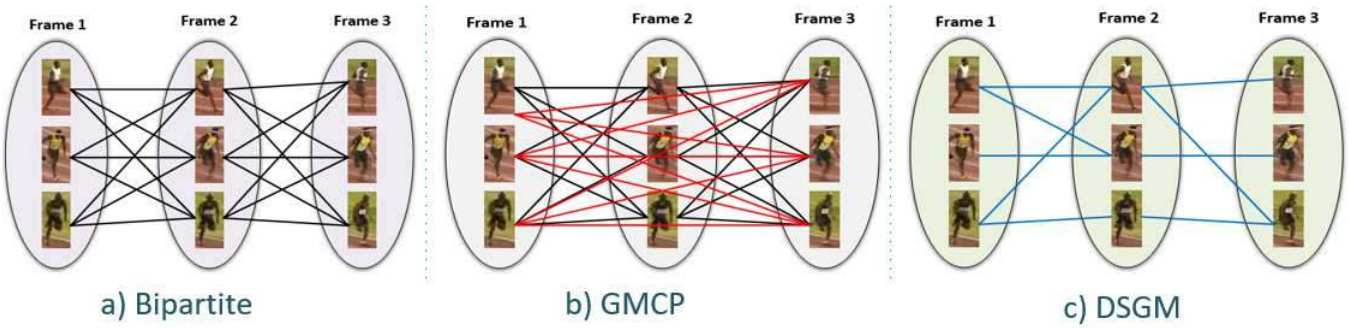

Figure 2: (a) Target association based on bipartite graph matching [29]. Each target in the current frame t is connected to all the other targets in the next frame. (b) Target association based on GMCP [24]. In this setting, a set of frames are considered and all the target in each frame are connected to all the other targets in the set of frames. (c) Target association based on proposed DSGM. It is clear from the graphical depiction that due to the spatial constraints, the number of edges have been reduced compared to bipartite and GMCP.

5.1. Transfer Learning

Any machine learning algorithm works under the assumption that the training and test data are drawn from same feature space and has the same distribution. However, in most real world applications, this is not true. If the distribution of the data changes, the brute force approach is to retrain the whole model on the new data. This approach is feasible for simple models. However, when it comes to deep architecture, millions of training samples are needed to retrain the network. This is clearly not feasible and in most cases it's not even possible to collect such large amount of data. For such problems, transfer learning is the optimal tool. In our case, we truncated the full network into its functional blocks and removed the classification part of the CNN. We keep the lower layers of the network intact because lower layers in a CNN are responsible for extracted low layer features which are common in every object. However, we re-train the higher layer with our comparatively smaller dataset so that the feature extraction of CNN is fine-tuned to our specific tracking problem. We use the fined tuned CNN to model the appearance of the targets and to calculate the probabilities $p(t_1)$, $p(t_2)$ and $p(t_1, t_2)$ as explained in 4.

6. Global Optimization

The HMM model approximates the most probable locations for the targets. It helps to avoid the redundant and irrelevant connection in the directed acyclic graph that is generated from the whole video sequence to approximate smooth trajectories for the targets. In a nutshell, HMM provides a ground for introducing sparsity in the graph. Once the graph is generated, we apply dynamic programming formulation proposed in [20] to get the trajectories for the targets. In Fig. 2, a visual comparison is given which shows the graphical structure of bipartite graph matching, the GMCP [24] and our proposed DSGM. It is obvious from the graph that both bipartite and GMCP have many redundant connections. However, due to HMM, DSGM has very few connections. In the next section 6, the formulation of DSGM is explained.

6.1. Graphical Model

Once the potential locations of the targets are known in all the frames, the tracking problem is formulated as min-cost flow [35]. It is an optimization problem [1] and the idea is to figure out the most reasonable way of delivering a certain amount of flow through a network of nodes and edges. Let $G = (V, E)$ be a directed acyclic graph with a set of V nodes and E edges. Each edge $e_{i,j} \in E$ has a cost $p_{i,j}$ which is the price of a flow moving from node $n_i \in V$ to $n_j \in V$. Each edge $e_{i,j} \in E$ may have a capacity $c_{i,j}$ which shows the amount of flow that can pass through the edge. The optimization variable in the min-cost flow problem is the edge flow. The flow on an edge $e_{i,j} \in E$ can be expressed as $f_{i,j}$. Then the optimization model can be written as:

$$Minimize \sum_{(i,j)\in E} p_{i,j} f_{i,j} \tag{11}$$

while in vector form:

$$Minimize \sum \mathbf{p} \times \mathbf{f} \tag{12}$$

where $\mathbf{p}$ is the vector of all edge cost and $\mathbf{f}$ the vector of all edge flows. Usually, the optimization of eq. 12 is subjected to

$$\sum_{i:(i,j)\in E} f_{i,j} - \sum_{j:(j,i)\in E} f_{j,i} = n(i) \tag{13}$$

for all $i \in E$. The constraint of eq. 13 is known as mass balance constraint and $n(i)$ indicates the net flow of a

node. At a time instant t, the capacity of an edge is set to 1 which enforces the unique label constraint for each target. The nodes in the graph represent the targets in the scene and modeled through deep features 5 and the edge cost $p_{i,j}$ is calculated through a mutual information that is explained in section 4.

6.2. Bayesian Inference

For the inference and optimization, we followed similar formulation like [20]. The vector valued state variable s indicates the spatial position ρ of a target at time instant τ.

$$s = (\rho, \tau) \qquad s \in V \tag{14}$$

where V is the set of all nodes in the graph. The track of a single target can be written as a set of state vector $L = \{s_1, s_2, ..., s_n\}$. The tracks of all the targets can be represented by a super set $Z = \{L_1, L_2, ..., L_M\}$ where M is the total number of targets in a video sequence. It is assumed that the tracks of the targets are independent from each other. Due to spatio-temporal dependence, each track can be seen as a variable length Markov chain. In the context of Bayesian inference, the posterior distribution of the set of trajectories can be written as:

$$P(Z|O) = P(Z)P(O|Z) \tag{15}$$

$P(O|Z)$ is the observation model that shows the probability of observing M targets at O locations given the set Z. O is the set of locations of all the targets.

6.2.1 Prior

Given track independence assumptions which states that the tracks of targets are independent from one another, each track can be seen as a variable length Markov chain and the distribution can be written as:

$$P(Z) = P(L_1, L_2, ..., L_M) \tag{16}$$

In compact form, it can be written as,

$$P(Z) = \prod_{i=1}^{M} P(L_i) \tag{17}$$

where

$$P(L) = P_s(s_1)P_e(s_n)\left\{\prod_{n=1}^{N-1} P(s_{n+1}|s_n)\right\} \tag{18}$$

$P_s(s_1)$ is the probability of trajectory L starting at s_1 and $P_e(s_n)$ corresponds to the probability of a track ending at s_n. Similarly, $P(s_{n+1}|s_n)$ is the node transition probability which is obtained through deep features. The edge cost $p_{i,j}$ that is computed in 4 corresponds to this probability and a transition is only executed if it is higher than certain threshold ϵ.

Datasets	Methods	MOTA	MOTP
AFL	Anton et al. [16]	32.0%	64.1%
	Milan et al. [17]	29.7%	63.3%
	Dicle et al. [7]	16.7%	60.8%
	Proposed DSGM	28.4%	62.8%
Sprint	Proposed DSGM	35.9%	52.0%
Basketball	Proposed DSGM	25.8%	43.7%

Table 1: Quantitative results of our DSGM. The results show that our method perform better than [7] and gives comparable performance to [16, 17] both on MOTA and MOTP.

6.2.2 Observation Model

Let O is the set of all the spatial locations of targets in a video segment. Our aim is to find $P(O|Z)$. Due to the nature of our tracking problem, the following two conditions are imposed:

- A target hypothesis can only be associated to a single track L.
- At a given time instant t, a spatial location can only be occupied by a single target i.e. $L_i \cap L_j = \emptyset$ where $i \neq j$.

Both conditions ensure a unique label for each target. Mathematically,

$$P(O|Z) = \prod_{L \in Z} \prod_{s \in L} P_t(o_s) \prod_{i \in V \setminus L} P_{nt}(o_i) \tag{19}$$

Where $P_t(o_s)$ is the probability of a spatial location belonging to a target and $P_{nt}(o_i)$ is the probability of a location being background. Our aim is to find the optimal set of trajectories with the minimum cost. In other words, the aim is to find the maximum a posteriori (MAP) of the tracks given the target observation O in all the frames.

$$Z^* = \arg\max_{Z} P(Z)P(O|Z) \tag{20}$$

Eq. 20 can be seen as the dual of eq. 12. For the optimization of the trajectories, we follow the algorithm of [20].

7. Experiment

The approach is tested on 3 sport games (sprint, football and baseball). Few frames are shown in Fig 3. There is large

variation in the appearance of the targets and due to severe articulations, the target representation changes substantially from frame to frame. Basketball dataset consists of 725 frames with a frame rate of 20. Similarly, bolt sequence consists of 350 with same frame rate. The football dataset AFL has comparatively low resolution and low frame rate with a total of 299 frames. The target hypothesis in each frame are generated with manual annotation with a bounding box enclosing the target. For the deep features, Alexnet [13] is fine-tuned on our own dataset with Matconvnet toolbox. The processing is performed on Intel core i7 with 16 GB RAM. The threshold ϵ is set to 0.6. Quantitative results are shown on MOTA/MOTP metrics. Tracking results are presented in Table 1 along with comparative results from the literature. The results of [7, 16, 17] are taken from the corresponding papers.

8. Conclusion

We propose a Directed Sparse Graphical Model (DSGM) for multi-target tracking which finds a set of tracks for the targets without assuming any heuristics or relaxation. Due to fewer connections, the computational complexity is very low for estimating the trajectories in the graph. Different from traditional approaches, we incorporated spatial and appearance constraints in a cascaded fashion. The spatial constraints are imposed through a HMM which finds the most probable locations of the targets in a segment of video. While the appearance constraints helps to find the edge cost of the directed acyclic graph. The appearance of targets are modeled through deep features. The track for each target is found through dynamic programming. Experiments are performed on 3 challenging sports dataset (football, basketball and sprint) and promising results are achieved.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows*. Elsevier, 2014. 5

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *IEEE conference on computer vision and pattern recognition*, pages 1926–1933, 2012. 2

[3] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011. 1

[4] G. Bishop, G. Welch, et al. An introduction to the kalman filter. *ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 8(27599-23175):41, 2001. 1

[5] S. Coşar, G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, and F. Brémond. Toward abnormal trajectory and event detection in video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):683–695, 2017. 1

[6] A. Dehghan, S. Modiri Assari, and M. Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *IEEE conference on computer vision and pattern recognition*, pages 4091–4099, 2015. 2

[7] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *IEEE International Conference on Computer Vision*, pages 2304–2311, 2013. 6, 7

[8] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 807–812, 1980. 1

[9] D. Gowsikhaa, S. Abirami, and R. Baskaran. Automated human behavior analysis from surveillance videos: a survey. *Artificial Intelligence Review*, 42(4):747–765, 2014. 1

[10] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*, 2015. 4

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4

[12] M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey. *50 years of integer programming 1958-2008: From the early years to the state-of-the-art*. Springer Science & Business Media, 2009. 1

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4, 7

[14] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

[15] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–36, 2006. 1

[16] A. Milan, R. Gade, A. Dick, T. B. Moeslund, and I. Reid. Improving global multi-target tracking with local updates. In *Springer European Conference on Computer Vision*, pages 174–190, 2014. 6, 7

[17] A. Milan, K. Schindler, and S. Roth. Detection-and trajectory-level exclusion in multiple object tracking. In *IEEE conference on computer vision and pattern recognition*, pages 3682–3689, 2013. 6, 7

[18] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, 2009. 1

[19] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005. 4

[20] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of

(a) Frame 4 (b) Frame 87 (c) Frame 155 (d) Frame 234 (e) Frame 296

(f) Frame 4 (g) Frame 57 (h) Frame 115 (i) Frame 284 (j) Frame 316

(k) Frame 28 (l) Frame 60 (m) Frame 139 (n) Frame 207 (o) Frame 283

Figure 3: Tracking results of DSGM. Top to bottom: datasets basketball (a-e), sprint (f-j), and AFL (k-o). Targets are given a unique integer ID.

objects. In *IEEE conference on computer vision and pattern recognition*, pages 1201–1208, 2011. 3, 5, 6

[21] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP magazine*, 3(1):4–16, 1986. 4

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[24] A. Roshan Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. *Springer European Conference on Computer Vision*, pages 343–356, 2012. 2, 5

[25] C. Russell, L. Agapito, and F. Setti. Efficient second order multi-target tracking with exclusion constraints. In *BMVC*, pages 1–11. Citeseer, 2011. 2

[26] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 4

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 4

[29] M. Ullah, F. A. Cheikh, and A. S. Imran. Hog based real-time multi-target tracking in bayesian framework. In *IEEE international conference on advanced video and signal based surveillance*, pages 416–422, 2016. 1, 5

[30] M. Ullah, H. Ullah, N. Conci, and F. G. De Natale. Crowd behavior identification. In *IEEE international conference on image processing*, pages 1195–1199, 2016. 1

[31] F. Wang, P. Li, X. Li, and M. Lu. Ordered over-relaxation based langevin monte carlo sampling for visual tracking. *Neurocomputing*, 220:111–120, 2017. 1

[32] G. Welch and G. Bishop. An introduction to the kalman filter. 1995. 4

[33] H. Yu, L. Qin, Q. Huang, and H. Yao. Online multiple object tracking via exchanging object context. *Neurocomputing*, 292:28–37, 2018. 1

[34] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Springer European conference on computer vision*, pages 818–833, 2014. 4

[35] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *IEEE conference on computer vision and pattern recognition*, pages 1–8, 2008. 2, 5

[36] S. Zhu, C. Sun, and Z. Shi. Multi-target tracking via hierarchical association learning. *Neurocomputing*, 208:365–372, 2016. 2