End-to-end learning of motion, appearance and interaction cues for multi-target tracking

Amir Sadeghian
Stanford University
amirabs@stanford.edu

Khashayar Khosravi
Stanford University
khosravi@stanford.edu

Alexandre Robicquet
Stanford University
arobicqu@stanford.edu

Abstract

The task of Multi-Object Tracking (MOT) largely consists of locating multiple objects at each time frame, and matching their identities in different frames yielding to a set of object trajectories in a video frame. There are several cues used for representing the individuals in a crowded scene. We demonstrate that in general, fusing appearance, motion, and interaction cues together can enhance the level of performance on the MOT task. In this paper we combined appearance, motion and interaction cues in one deep unified framework. An important contribution of this work is a generic scalable MOT method that can fuse rich features from different dynamic or static models.

1. Introduction

Multiple Object Tracking (MOT), or Multiple Target Tracking (MTT), plays an important role in computer vision and is a crucial problem in scene understanding. The objective of MOT is to produce trajectories of objects as they move around the image plane. MOT covers a wide range of application such as pedestrians on the street [26, 19] sports analysis (e.g. sport players in the court [14, 17, 24], bio tracking (birds [15], ants [9], fishes [20, 21, 5], cells [16, 12], and etc), robot navigation, or autonomous driving.). In crowded environments occlusions, noisy detections (false alarms, missing detections, non-accurate bounding), and appearance variability (Same person, different appearance or different people, same appearance) are very common. As a result, multi object tracking has become challenging task in computer vision.

Recent works have proven that tracking objects jointly and taking into consideration their interaction in addition to their appearance can give much better results in crowded scenes. The focus of this paper is to marry the concepts of appearance model, object motion, and object interactions to obtain a robust and scalable tracker than works in crowded scenes. We propose an online unified deep neural network tracker that jointly learn to reason on a strong appearance model, strong individual motion model, and object interactions (dynamic scene knowledge). In this project we will not study the interaction model and only focus on appearance and motion model.

Our strong appearance model is a Siamese convolutional neural network (CNN) that is able to find occlusions and similarity of objects in different time frames in addition to object bounding box prediction in next time frame. We also use two seperate Long Short-Term Memory (LSTM) model for our motion prior and interactions model that tracks the motion and trajectory of objects for longer forecasting period (suited in presence of long-term occlusion). These models extract appearance cues, motion priors, and interactive forces which are critical parts of the MOT problem. We then integrate these parts into a coherent system using a high-level LSTM that is responsible to reason jointly on different extracted cues. We show our model is able to fuse and use different data modalities and get a better performance. The magic is this scalability, one can add another cue component (e.g. exclusion model) to the model and finetune the model to reason jointly on the new cue and previous cues.

2. Related Work

In recent years tracking has been successfully extended to scenarios with multiple objects [18, 11, 8, 23]. Different from single object tracking approaches which have been constructing a sophisticated appearance model to track single object in different frames, multiple object tracking does not mainly focus on appearance model. Although appearance is an important cue but in crowded scenes relying only on appearance can lead to a less accurate MOT system. To this end, different works have been improving only the appearance model [6, 3], some works have been combining the dynamics and interaction between targets with the tar-

get appearance.

2.1. Appearance model

Technically, appearance model is closely related to visual representation features of objects. Depending on how precise and rich the visual features are, they are grouped into three sets of single cue, multiple cues, and deep cue. Because of efficiency and simplicity single cue appearance model is widely used in MOTs. Many of single cue models are based on raw pixel template representation for simplicity [25, 2, 22, 19], while color histogram is the most popular representation for appearance modeling in MOT approaches [4, 11, 28]. Other single cue approaches are using covariance matrix representation, Pixel comparison representation, or SIFT like features. The multi cues approaches combines different kinds of cues to make a more rebust appearance model. The final appearance cue used in tracking is the deep visual reperesentation of objects. These high-level features are extracted by deep neural networks mostly convolutional neural networks trained for a specific task [7]. Our model shares some characteristics with [7], but differs in two crucial ways: first, we are learning to handle occlusion and solve the re-identification task in addition to David's work that is bounding box regression only. We output the similarity score (same object or not) and bounding box. Second, there are differences in the overall architecture, e.g. the number of fully connected layers on top of two networks for fusing, loss function, inputs and outputs and hence the training and testing procedure is different since we want to address re-identification as well as bounding box to help tracking.

2.2. Motion model

Object motion model describes how an object moves. Motion cue is very important for multiple object tracking since knowing the potential position of objects in the future frames will reducing search space and help the appearance model on better detectation of similar objects. Popular motion models used in multiple object tracking are divided into linear motion models and Non-linear motion models. As the name "linear motion" indicates objects following the linear motion model move with constant velocity. This simple motion model is the is the most popular model in MOT [3]. There are many cases that linear motion models can not deal with, in this cases non-linear motion models are proposed to produce a more accurate motion model for objects [27]. We present a new Long Short-Term Memory (LSTM) model which jointly reasons based on the past movements of an object and predicts the future trajectorys of that object [1].

3. Multi Object Tracking Framework

As shown in Figure 1, MOT involves three primary components. Our model includes modeling of appearance, motion, and interaction. These components will be described in more details.

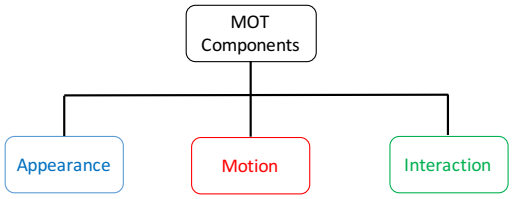

Figure 1. MOT components

3.1. Appearance

In this section, we now describe the appearance model that we integrate into our framework for multi-object tracking. As we recall, our problem is fundamentally based on addressing the challenge of data association: that is, given a set of targets T_t at time step t, and a set of candidate detections D_{t+1} at timestep $t+1$, we would like to compute all of the valid pairings that exist between members of T_t and D_{t+1}.

The idea underlying our appearance model is that we can compute the similarity score between a target and candidate detection based on purely visual cues. More specifically, we can treat this problem as a specific instance of *re-identification*, where the goal is to take pairs of bounding boxes and determine if their content corresponds to the same person. We thus desire our appearance model to recognize the subtle similarities between input pairs, as well as be robust to occlusions and other visual disturbances.

To approach this problem, we construct a Siamese Convolutional Neural Network (CNN), whose structure is depicted in Figure 2. Let BB_i and BB_j represent the two bounding boxes we wish to compare – in our case, BB_i might be a target bounding box at frame t, and BB_j would be a candidate detection at frame $t+1$. We first crop the images containing BB_i and BB_j to contain only the bounding boxes themselves, while also ensuring that we include some amount of the surrounding image context. The network then accepts the raw content within each bounding box and passes it through its layers until it finally produces a 500-dimensional feature vector for each of the two inputs.

Let ϕ_i and ϕ_j thus be the final hidden activations extracted by our network for bounding boxes BB_i and BB_j. In order to compute the similarity, we then simply concatenate the two vectors to get a 1000-dimensional vector $\phi = \phi_i || \phi_j$, and pass this as input to a final fully-connected layer. We lastly apply a Softmax classifier, which outputs the probabilities for the positive and negative classes, where

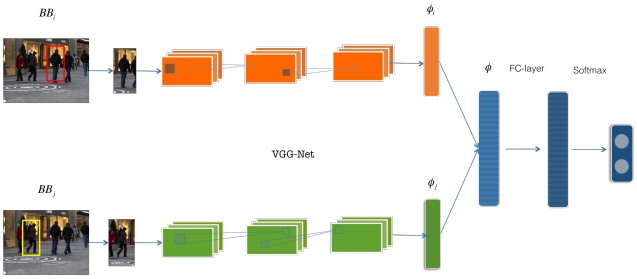

Figure 2. Our appearance model

positive indicates that the inputs match, and negative indicates otherwise.

The actual network structure we use for this challenge consists of the 16-layer VGG net, which won the ImageNet 2014 localization challenge. In our case, we begin with the pre-trained weights of this network, but remove the last fully-connected layer so that the network now outputs a 500-dimensional vector.

We then fine-tune this network by training the overall network on positive and negative samples extracted from our training sequences. For positive pairs, we use instances of the same target that occur in different frames. For negative examples, we use pairs of different targets that may span across all frames.

We trained this model on MOT3D dataset which contains 2 scenes with more than 950 frames that contain more than 5500 objects. We extracted more than 100k of positive and negative samples. We did the training on one scene and validated on the other scene. The result was 84 percent accuracy on the binary classification problem of positive/negative pairs. We used CUHK03 dataset [13] as the sanity check for our prediction. This dataset contains 13164 images of 1360 pedestrians and contains 150k pairs. FPNN method which got rank 1 of identification MAP rate were able to achieve 19.89 percent accuracy. Our method achieves 18.61 percent of accuracy and outperforms several other methods such as LDM, KISSME, SDALF.

3.2. Motion

The second component of our overall framework is the inclusion of an independent motion prior for each target. The intuition is that the previous movements for a particular target can strongly influence what position a target is likely to be at during a future time frame.

Additionally, a nuanced motion prior can help our model when tracking objects that are occluded or lost, since it provides a heuristic as to where these objects might generally be located. Thus, formulating a sophisticated model for the motion prior of a target will be valuable in achieving robust performance during tracking.

We can therefore use this information to aid us in the task of data association, in which we can match members of T_t and D_{t+1} based on which detections are closest to the motion prior's next predicted location for each target.

To thus incorporate this information, we construct a Long Short-Term Memory (LSTM) network over the 3D *velocities* of each target. More concretely, let $(x_0^i, y_0^i, z_0^i), (x_1^i, y_1^i, z_1^i), \ldots (x_t^i, y_t^i, z_t^i)$ represent the 3d trajectory of the i-th target from the timestep 0 through timestep t. Assuming a point $(x_{t+1}^i, y_{t+1}^i, z_{t+1}^i)$, we want to see whether this point belongs to the trajectory of i-th target. Let use define the velocity of target i at the j-th timestep i to be $\mathbf{v_j^i} = (vx_j^i, vy_j^i, vz_j^i) = (x_j^i - x_{j-1}^i, y_j^i - y_{j-1}^i, z_j^i - z_{j-1}^i)$. This can be done by assigning an score to this point and seeing whether it is large enough or not. For this purpose, we train our LSTM to accept as inputs the velocities of a single target for timesteps $1, \ldots, t$ and produces H-dimensional outputs. We also pass the $t+1$ velocity vector (which we wish to determine whether it corresponds to a true trajectory or not) from a fully-connector layer that brings it to H-dimensional vector space. The last LSTM output is then concatenated with this vector and the result is passed to another fully connector layer which brings the $2H$ dimensional vector to the space of k features. Finally, another fully connector layer, reduces the dimension to 2 which will be used as the $0/1$ classification problem during the training.

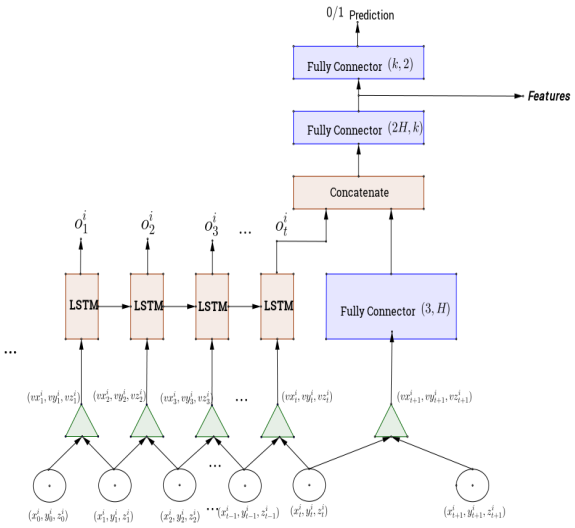

Figure 3. Our 3D motion prior model

Note that training occurs from scratch, and weights are shared across all targets. Once we train the network, then given a query target i at timestep t', the LSTM will output a predicted velocity $v_{t'+1}^i$. We can then simply add the velocity to the query target's position at t' in order to compute the motion prior's predicted position for frame $t'+1$. That

is,

$$(x^i_{t'+1}, y^i_{t'+1}, z^i_{t'+1}) = (x^i_{t'} + v x^i_{t'+1}, y^i_{t'} + v y^i_{t'+1}, z^i_{t'} + z^i_{t'+1})$$

We therefore obtain the predicted position from the motion prior, and can use this to filter out candidate detections that are not sufficiently close to the prior.

For training this model, we used MOT3D dataset, which only consists of true trajectories. We considered trajectories of length $t + 1 = 7$ and we assumed $H = 128$. For each true trajectory, we changed the last element of it by a randomly chosen object among all other objects that exist at the same frame. By doing this we were able to reach to the same number of invalid trajectories as the valid trajectories (it is not good to have unbalanced distributions for training). After training this model, we were able to achieve the accuracy of 95 percent for the $0/1$ classification problem.

3.3. Integration

Given these three components of our framework for Multi-object Tracking, we now describe the method by which we integrate these parts into a coherent system. To recall, we have identified appearance cues, motion priors, and interactive forces as critical parts of the MOT problem. We believe a sophisticated framework should merge these pieces together in an elegant way. You can find the graphical model of our approach in figure 4. Each human has an appearance edge and motion edge, and between every pairs of humans there is an interaction edge.

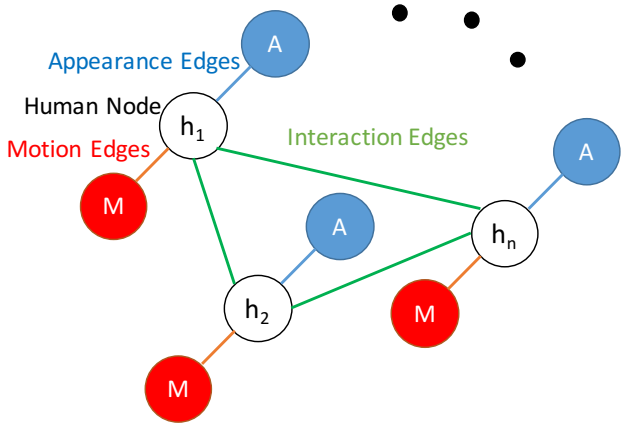

Figure 4. The graphical model of our approach

Our overarching model is a Long Short-Term Memory network which we construct over the already pre-trained appearance, motion, and interaction modules. This LSTM is trained to perform the task of data association: once again, suppose we are at timestep t and wish to determine whether target i is matched to a detection d found in timestep $t + 1$. We then train the LSTM to output the probabilities of whether the t and d correspond to the same object.

The inputs to the LSTM are feature vectors that we extract from our individual models. Let ϕ_A represent the hidden activations extracted from our appearance model before the final fully connected layer of the network, where we input the bounding boxes surrounding target i and detection d. Let ϕ_{Mj} be the hidden state of the Motion Prior LSTM extracted at timestep j, and likewise let ϕ_{Ij} be the hidden feature vector of the Interaction model extracted at timestep j. Then, the input to our integrator is given by

$$\phi_j = \phi_A || \phi_{Mj} || \phi_{Ij}$$

where we thus concatenate the individual feature vectors output by the modules. Therefore, when we set up the model we use these features as inputs to the LSTM and train it to output either a positive or negative label for each timestep (indicating whether there is a valid match) using a standard Softmax classifier and cross-entropy loss.

An important point to note is that we train this LSTM without fine-tuning the weights of the individual components of the framework, which are each in fact trained separately. The overall model, composed by the previous components is illustrated by figure 5 and the output of the model is a similarity score which is used as a weight for the edges of matching graph for matching the detections between time frames.

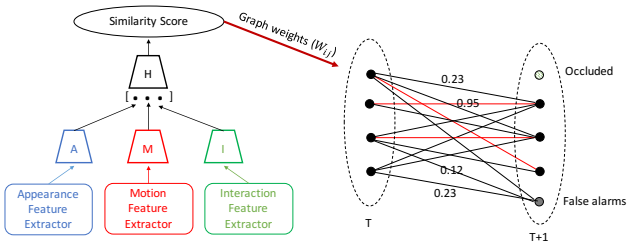

Figure 5. Our overall model

For training this model, we used the pretrained components described in previous sections and fine-tune the whole model end to end using MOT3D dataset.

4. Experiments

In this section, we now describe our various experiments and results, and then later peform a qualitative analysis on model's performance.

4.1. Baselines

We first discuss the various baselines that we use to establish a standard for comparison against our more nuanced model.

- **Markov Decision Process Tracker**

  In [23], authors demonstrated success on 2D Multi-object tracking by formulating the tracking problem as a Markov Decision Process (MDP). They represented every target as being in either an active, tracked, lost, or inactive states, and learned the appropriate transition probabilities and rewards based on extracted features. In order to evaluate this method on the 3D challenge, we project the bottom-midpoint of the predicted 2D bounding boxes to the ground plane (using the provided calibration parameters given in the data sequences).

- **MDP Tracker with Linear Motion Prior**

  Though the MDP described above can obtain reasonable results on the problem of multi-object tracking, we additionally incorporate a simple linear 3D motion prior into the feature vectors associated with each state in the MDP. More specifically, we use the normalized distance between a candidate detection and the motion prior prediction as a feature in that state.

- **MDP Tracker with LSTM Motion Prior**

  As a final baseline experiment, we realize that incorporating a simple linear motion prior may be too simplistic of an approach to accurately model the movements of a target. A more reasonable method is to use an LSTM similar to our own motion model to output the predicted 3D coordinates for every target, and then use these values in the feature vectors as described above.

We report the results using the proposed method on the 3D MOT 2015 Benchmark which includes the PETS09-S2L21 and the AVG-TownCentre2 sequences. The sensitivity of the method to the omission of single variables is evaluated on the PETS09-S2L1 dataset (available for training in the 3D MOT 2015 Benchmark). The corresponding results of an evaluation in 3D image space (correct detection requires at least 50% intersection-over-union score with the reference) and in 3D world coordinates (correct detection requires at most 1m offset in position) are reported in following section, respectively.

4.2. Datasets

We test our tracking framework on the Multiple Object Tracking Benchmark [10] for people tracking. The MOT Benchmark collects widely used video sequences in the MOT community and some new challenging sequences. We evaluate the proposed algorithm on MOT3D challenge which provides the 3D coordinate of position of the feet of people into the 3D world. It consists of two publicly available datasets: a crowded town center, and the well known PETS2009 dataset.

4.3. Results

The accuracy and results of each component of our system is described at each of the experimental sections. Here we see the final results of the tracker in table 6 for results of our tracker compared to other baselines on MOT3D challenge. The last 3 rows are our cross validation on MOT challenge training set.

Tracker	MOTA (H)	MOTP (H)	MT (H)	ML (L)
DBN (State of art) - 1st	**51.1**	**61.0**	**28.7%**	17.9%
KalmanSFM (Baseline) - 5th	25.0	53.6	6.7%	14.6%
Yu's 3D	45.5	**61.0**	6%	**3%**
Appearance only **(Cross validation)**	38.1	54.1	15%	20%
LSTM only **(Cross validation)**	28.9	48.3	9%	28%
Appearance and LSTM (Cross validation)	40.3	57.1	16%	19%
Appearance and LSTM (MOT Challenge)	28.3	51.7	29.1%	21.8%

Figure 6. Primary Results on MOT3D challenge

5. Conclusion

This paper proposes a deep neural network designed for multi object tracking. Quantitative results show that the tracking performance is superior to the baseline tracking methods. Our tracker can also be fine-tuned for various applications by providing more training videos of certain types of objects. Overall, our real-time neural network tracker opens up many possibilities for different applications and extensions, allowing us to learn from several cues used for representing the individuals in a crowded scene. We demonstrate that in general, fusing appearance and motion cues together can enhance the level of performance on the MOT task. We show our model is able to fuse and use different data modalities and get a better performance. One of the main advantage of our tracker to others is the scalability, one can add another cue component (e.g. exclusion model) to the model and finetune the model to reason jointly on the new cue and previous cues.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces.

[2] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Computer Vision–ECCV 2008*, pages 1–14. Springer, 2008.

[3] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009.

[4] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *Computer Vision–ECCV 2010*, pages 553–567. Springer, 2010.

[5] E. Fontaine, A. H. Barr, and J. W. Burdick. Model-based tracking of multiple worms and fish. In *ICCV Workshop on Dynamical Vision*. Citeseer, 2007.

[6] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, volume 1, page 6, 2006.

[7] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 FPS with deep regression networks. *CoRR*, abs/1604.01802, 2016.

[8] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV 2008*, pages 788–801. Springer, 2008.

[9] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *Computer Vision-ECCV 2004*, pages 279–290. Springer, 2004.

[10] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. arXiv: 1504.01942.

[11] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1683–1698, 2008.

[12] K. Li, E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, and P. G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical image analysis*, 12(5):546–566, 2008.

[13] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.

[14] C.-W. Lu, C.-Y. Lin, C.-Y. Hsu, M.-F. Weng, L.-W. Kang, and H.-Y. M. Liao. Identification and tracking of players in sport videos. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, pages 113–116. ACM, 2013.

[15] W. Luo, T.-K. Kim, B. Stenger, X. Zhao, and R. Cipolla. Bi-label propagation for generic multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2014.

[16] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen. Tracking in cell and developmental biology. In *Seminars in cell & developmental biology*, volume 20, pages 894–902. Elsevier, 2009.

[17] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking-linking identities using bayesian network inference. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2187–2194. IEEE, 2006.

[18] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[19] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009.

[20] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. *VISAPP (2)*, 2008:514–519, 2008.

[21] C. Spampinato, S. Palazzo, D. Giordano, I. Kavasidis, F.-P. Lin, and Y.-T. Lin. Covariance based fish tracking in real-life underwater environment. In *VISAPP (2)*, pages 409–414, 2012.

[22] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1948–1955. IEEE, 2012.

[23] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *International Conference on Computer Vision (ICCV)*, pages 4705–4713, 2015.

[24] J. Xing, H. Ai, L. Liu, and S. Lao. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *Image Processing, IEEE Transactions on*, 20(6):1652–1667, 2011.

[25] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE, 2011.

[26] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1233–1240. IEEE, 2011.

[27] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1918–1925. IEEE, 2012.

[28] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer, 2012.