Study of Multi-Object Tracking

Course Final Report of Computer Vision II

Dapeng Sun
sundp@shanghaitech.edu.cn

Yisong Li
liys@shanghaitech.edu.cn

Kang Zhu
zhukang@shanghaitech.edu.cn

Abstract

Multi-Object tracking (MOT) is a challenging task of computer vision which has brand application. MOT has three components, which are object detection, object tracking and strategy of tracking by detection. Different with single object track, the critical task of MOT is data assocation. We study the work [20] that implement MOT with MDP, and we improve the work by changing the detector and the strategy of tracker. Finally ,we implement it on the MOT dataset and our video of real scene.

1. Introduction

Multi-object tracking **[MOT]** is a challenging, but practically important computer vision task which has broad application in many fields such as surveillance, robot navigation and autonomous driving. This task is to estimate the trajectories of multiple interacting object in a same scene(2D image plane or 3D object space) from a common video stream. What MOT doing is not all the same with single object tracking whose purpose is long-term tracking which automatically predict and determine the object's bounding box in every next frame of video stream, but focus on the data association that match detection responses across frames based on object detection with the existed and tracked objects.

The object of MOT is defined by its location and extent in a single frame. In every frame that follows, MOT task is to determine the all objects' corresponding location and extent informations or indicate not present. To acquire the goals, MOT has three critical components.

Object detection. The component is locate the objects in the frames using different approach. Object detection is the combinations of the region proposal and image classificaton. In a low-level, the task can be implement with the background modeling based on Gauss method to a scene where the position of camera is fixed. In a high-level, the task can be implement with the methods of learning that train a classifer to classify all patches of images to specific object or not. State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations, such as the work that SPPnet[10] and Fast R-CNN [5]. This component is the prerequisite for MOT. We use the work that Faster R-CNN[16] method achieved the better trade off between the performance and computional cost.

Object tracking. The component is estimate the motion of every single object in every frame. Trackers typically assume that the object is visible through-out the frame sequence. By means of mathematical statistics, using the representation of object in current frame, trackers predict the position of object in future frame by probability. This component is the prior consideration for MOT. To anticipate the long-term tracking of unknown objects in frames. We choosed Z.Kalal's work[13] that detection, learning and tracking framework (**TLD**) which exactly split and convert the task to tracking, learning, and detection for every single object in frames. In the work, using the methods of learning to correct the tracker and detector mutually with the appearances. The learning process is modeled as a discrete dynamical system and the conditions under which the learning guarantees improvement are found.

Strategy of tracking by detection. The component implement a series of strategies to link tracked targets to objects from a category detector. This task aims resolve ambiguities in associating object detections and overcome detection failures to acquire accurate and fixed trajectories of objects. This component is macro important indicator for MOT. In order to stably link noisy object detections on a frame with tracked objects. We choosed Xiang's method[20] for the online MOT problem as decision making in Markov Decision Processes (**MDP**) , where the lifetime of an object is modeled with a MDP. Learning a similarity function for data association is equivalent to learning a policy for the MDP, and the policy learning is approached in a reinforcement learning that integrate both advantages

of offline-learning and online-learning.

2. Related Work

Object detection. There are a large literature on object proposal methods, the most pravalent one is the Selective Search[19],which combines the strength of exaustive search and image segmentation. However most of them were adopted as external modules independently. Faster R-CNN introduce a novel method that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. [19] still use Selective Search as region proposal methods but trains CNNs end-to-end to classify the region proposals into object categaries or backgroud[6]. and [10] and [5] adjust the structure of the CNNs and proposed some novel training methods.However, CNNs mainly plays as a classifier, and it does not predict object bounds(except for refining by bounding box regression). Shared computation of convolutions has been attracting increasing attention for efficient, yet accurate, visual recognition. [16] introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals.

Object tracking. Recent a partion of study in object tracking has focused on describe the object with a bounding box and the motion is defined as a transformation that minimizes mismatch between the target template and the candidate patch. The template tracking can be either realized as static[4] or adaptive[15][17]. Adaptive discriminative trackers[3] [2] [7]can handle significant appearance changes, short-term occlusions, and cluttered background. The essential phase of adaptive discriminative trackers is the update: the close neighborhood of the current location is used to sample positive training examples, distant surrounding of the current location is used to sample negative examples, and these are used to update the classifier in every frame. However, these methods also suffer from drift and fail if the object disappear for a long time. To address these problems the update of the tracking classifier has been constrained by an auxiliary classifier trained in the first frame [9] or by training a pair of independent classifiers[18][22]. Our choosen work([13]) using a offine & online learning to acquire the stable result of tracking.

Strategy of tracking by detection. Recent study in MOT has focused on the tracking by detection strategy, which mainly aims to resolve the issue of data association in linking the objects. S. Avidan's work [1] and H. Grabner's work [8] make MOT task as a binary classification problem, where an ensemble of weak classifiers is trained online to distinguish between the object and the background and combined them into a strong classifier using Boosting. Khan's work [23] and Okuma's work [12] solve the data association problem as probabilistical problem using online methods. While the majority of methods ([14],[21],[11]) formulates MOT as a global optimization problem in a graph-based representation. The majority works has a critical component in each data association problem is a similarity function among objects. Our choosen work([20]) get this goal with a novel reinforcement learning algorithm online.

3. Methods

3.1. Object Detecion-Faster RCNN

Faster R-CNN, is composed of two modules. As **Figure** 1 shown, the first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector [5] that uses the proposed regions. The entire system is a single, unified network for object detection. Using the recently popular terminology of neural networks with attention mechanisms, the RPN module tells the Fast R-CNN module where to look.

A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score.We model this process with a fully convolutional network. Because our ultimate goal is to share computation with a Fast R-CNN object detection network [5], we assume that both nets share a common set of convolutional layers. In our experiments, we investigate the Simonyan and Zisserman model (VGG-16), which has 13 shareable convolutional layers.

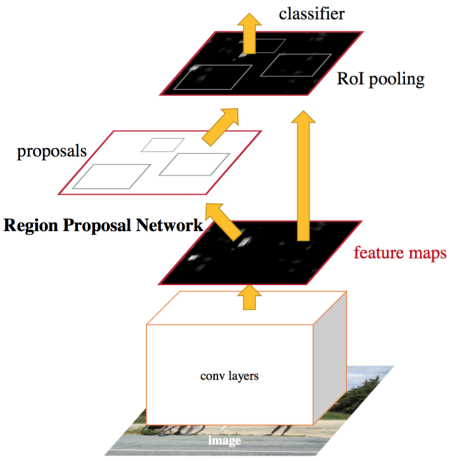

Figure 1. Faster R-CNN Architecture

To generate region proposals, we slide a small network over the convolutional feature map output by the last shared convolutional layer. This small network takes as input an $n \times n$ spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature ($256-d$ for ZF and $512-d$ for VGG, with ReLU fol-

lowing). This feature is fed into two sibling fully- connected layersa box-regression layer (reg) and a box-classification layer (cls). We use $n = 3$ in this paper, noting that the effective receptive field on the input image is large (171 and 228 pixels for ZF and VGG, respectively). This mini-network is illustrated at a single position in **Figure** 2 . Note that because the mini-network operates in a sliding-window fashion, the fully-connected layers are shared across all spatial locations. This architecture is naturally implemented with an $n \times n$ convolutional layer followed by two sibling 1×1 convolutional layers (for reg and cls, respectively).

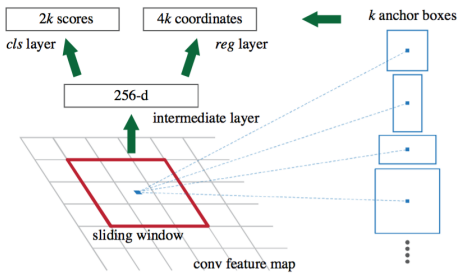

Figure 2. Region Proposal Network(PRN)

3.2. Single Object Tracking-TLD

TLD is a framework for the long-time single object on-line tracking in a video stream. From the name of the framework we can know that it contains three components:**Tracker**, assume that the frame-to-frame motion is limited and the object is always visible in all the video streaming, so the tracker can estimate the correct location of the object. But when the object disappears in the camera view the tracker will no longer work. **Detector** thinks all the frames in the video streaming are independent, so we can process them respectively. In each frame, we can detect some potential objects, but a detector may also make mistakes when it works. and **Learning** consider both the tracker and the detector synthetically, from the known result of the tracker and the detector to learn strategy in which tracker and detector can correct each other's mistakes. The correct procession is decided by the P-N experts, where the P analysis and correct the result of the detector while N expert analysis and correct the result of the result of the tracker. In **Sec 3.2.1**, we introduce the prerequisites of algorithm. In **Sec 3.2.2- Sec 3.2.4**, we formulate the critical components of TLD.

3.2.1 Prerequisites

For object tracking, the object only has two state: bounding box represent the location or a flag represent the object is not visible. The bounding box is described by its location and the size. A sequence of object states defines a trajectory of an object in the video streaming. Note that the trajectory is fragmented as the object may not be visible. Spatial similarity of two bounding box is measured using overlap, which is defined as a ratio between intersection and union. A single instance of the object's appearance is represented by an image patch p, similarity between two patches p_i,p_j is defined as

$$S(p_i, p_j) = 0.5(NCC(p_i, p_j) + 1),$$

where NCC is a Normalized Correlation Coefficient.

For both the tracker and the detector, we need a object model which is a data structure that represent the object and its surrounding observed so far. We can define the object model as a set of the positive and the negative patches,$M = \{p_1^+, p_2^+, \cdots, p_m^+, p_1^-, p_2^-, \cdots, p_n^-\}$where p^+ and p^- represent the object and the background patches respectively. Positive samples are added into the object model as the fixed order,$p_1^+$is the first positive patch added to the collection, while $p_m^+$is the last positive patch.

In order to measure the similarity, we define several similarity measures when given an arbitrary patch p and object model M:

- Similarity with the positive nearest neighbor, $S^+(p, M) = \max_{p_i^+ \in M} S(p, p_i^+)$.
- Similarity with the negative nearest neighbor, $S^-(p, M) = \max_{p_i^- \in M} S(p, p_i^-)$.
- Similarity with the positive nearest neighbor considering 50% earliest positive patches, $S^+_{50\%}(p, M) = \max_{p_i^+ \in M \wedge i < \frac{m}{2}} S(p, p_i^+)$.
- Relative similarity, $S^r = \frac{S^+}{S^+ + S^-}$. Relative similarity ranges from 0 to 1, high values mean more confident that the patch depicts the object.
- Conservative similarity, $S^c = \frac{S^+_{50\%}}{S^+_{50\%} + S^-}$. Conservative similarity ranges from 0to 1. High value of S^c mean more confidence that the patch resembles appearance observed in the first 50% of the positive patches.

3.2.2 Tracker

The tracker is based on the Median-Flow tracker, first of all, we need to represent the object with a bounding box, in order to track the object, we just need to estimate the location of the bounding box. W e can use a gride of 10×10 points in the bounding box, and compute the optical flow value of the 100 points, then estimate their reliability, and vote with 50% of the most reliable displacements for the motion of the bounding box. We have said that the tracker won't work

when the the object gets occluded or out of the camera view, so we need to identify these situations. Let d_i represent the displacement and d_m be the median displacement. A residual can be denote as $r = |d_i - d_m|$, when r is larger than 10, we can say the object gets occluded or out of the view. If the failure is detected, the tracker does not return any bounding box.

3.2.3 Detector

Offline detector. The critical offline detector has been introduced in **Sec 3.1**, which is used in our experiment.

Online detector-Nearest neighbor detector. The online detector scans the result of the off-line detector by a scanning-window and for each pitch decide whether it is an object. A straightforward approach is the Nearest Neighbor classifier as it involves evaluation of the relative similarity. A patch is classified as the object if $S^r(p, M) > \theta_{NN}$, where the θ is set empirically and its value is not critical. Otherwise, the patch can be seen as the background.

3.2.4 Learning

In TLD frame, the input is the first frame with the result of the detector and a video streaming, firstly, we can add the first object to the object model, and train a detector use the object model, then we can use the detector to classified the next frame of the video, now P-N experts will analysis and evaluate the detector, and correct the wrong result, and add the new data to the object model. Iterate above steps until convergence. The block diagram of the P-N learning is shown in **Figure** 3.

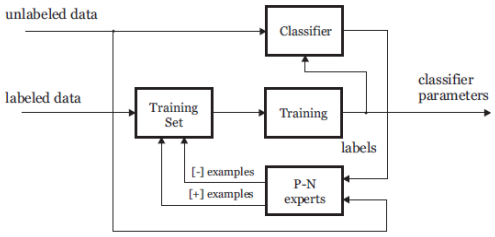

Figure 3. The block diagram of the P-N learning

Initialization. The learning component trains the initial detector using the examples generated as follows. The first frame statement the bounding box of the object, so we can select 1bounding boxes on the scanning grid that are closet to the initial bounding box, for each of the bounding box, we generate 20 warped versions by geometric transformations (shift,scale,rotation) and add them with Gaussian noise($\sigma = 5$) on pixels. So we can get 200 positive patches. Negative patches are selected from the surrounding of the initial bounding box.

P-expert. The goal of P-expert is to discover new appearances of the object and thus increase generalization of the object detector. P expert analyzes examples classified as negative, estimates false negative, that's mean P expert analyzes the patch which is the result of the tracker, if it is labeled positive, there is no problem, but if it is labeled negative by the detector, P expert corrects it as positive and adds it to the training data with positive label. In a word, P-expert is to identify reliable parts of the trajectory and use it to generate positive training examples. In every frame, the P-expert outputs a decision about the reliability of the current location (P-expert is an online process). If the current location is reliable, the P-expert generates a set of positive examples that update the object model and the online classifier.

N-expert. N-expert is to discover clutter in the background against which the detector should discriminate. N-expert analyzes examples classified as positive, estimates false positive. Because the TLD frame is to single object tracking, there should only one object each frame. Therefore, if the object location is known, all the other should be labeled negative. So for all the positive example, N-expert choose the most credible one, add it to the object model and label others as negative. The N-expert is applied at the same time as P-expert. The P-expert increases the classifiers generality. The N-expert increases the classifiers discriminability.

3.3. Multi-Object Tracking-MDP

In **Sec. 3.3.1**, we formulation the MDP modeling with MOT task and in **Sec 3.3.2**, we introduce the method using MDP strategy for online MOT task according Xiang's work [20].

3.3.1 Markov Decision Process (MDP) Modeling

In the framework, every object's lifecycle is defined as a MDP. As Figure 4 shown, MDP consist of three joint components which are State, Action & Transition Function, Reward Function.

Define the state $s \in S$ to represent the status of each target. As **Figure** 4 shown, we can find that $S = S_{active} \cup S_{Tracked} \cup S_{Lost} \cup S_{Inactive}$. Every subspace of State has infinity number of states. All state reached is encoded by the feature representation, such as the historgram of appearance, shape, location in frame, size and other priori information.

- "Active" state is the birth point in whole MOT process. the framework enters "Active" if an object detected by object detector or prepared of data. "Active" can transit to "Tracked" or "Inactive".
- "Inactive" state is the death point in whole MOT process. the framework enters "Inactive" if an object

can not be tracked unceasingly according some time threshold, such as lost for a long time. "Inactive" can be transited from "Active" or "Lost".

- "Tracked" state is the critical state that an object is tracked with feature representation by tracker. "Tracked" come from "Active", and go to itself in next frame or "Lost".
- "Lost" state is the ambigurous state that an object can not be tracked by tracker, since the object really be lost or the tracker failed currently. If the tracker restart in a definite time, "Lost" return to "Tracked", otherwise, it go to "Inactive".

Define the action $a \in A$ to represent the action of state transformation of each target in specific state. Define the state transition function $T : S\&A \longrightarrow S$ to measure the effect of specific action in specific state for every target. Given the current state and an action, we can get the specific new state.

Define the reward function function $R : S\&A \longleftrightarrow \mathcal{R}$ to calculate the reward value after transition. The reward function is used to evaluate the transition, which is unkonwn and need to be gotten by learning or other methods. In this method, we use a reinforcement learning to training.

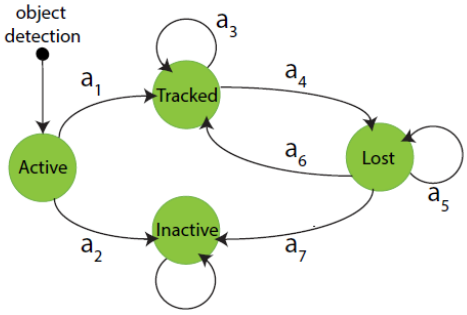

Figure 4. The MDP framework in MOT

.

In MDP, a policy $\pi : S \longrightarrow A$ is defined to make decising to determine which action to implement and which state to transit. The goal of policy learning to find the policy which has maximial value of reward function. Specifically, we need to design the policy in "Active", "Tracked" and "Lost". All the policy are classifers to acquire the action which can be learning with supervised learning.

In "Active", the policy is used to filter the noise objects by detector, and drift the noises to "Inactive" using support vector machine with the train data and feature representation. ***i.e.***, $R_{Active}(s, a) = y(a)(W^T\Phi(s) + b)$ where $y(a_1) = +1$, and$y(a_2) = -1$.

In "Tracked", the policy is designed to acquire a long-term tracking or turn into "Lost" if classifer label like this. The specifical learning method is similar to the TLD work [13] which is designed to the single object tracking in the **Sec 3.2** above.

In "Lost", the policy need to decide whether the state can return to "Tracked" or "Inactive" permanently. In this active, the process utmost reflect the importance of data assocication, which may make the "lost" object to be associate to some object in new frame, then the state come back to the "Tracked". The task is involve to reinforcement learning solved as soft-margin optimization problem to acquire a max-margin classifer for data association as following:

$$\begin{aligned} OBJ. \quad & \min_{W,b,\epsilon} \tfrac{1}{2}||W||^2 + C\sum_{k=1}^{M}\epsilon_k \\ S.T. \quad & y_k(W^T\Phi(t_k, d_k) + b) \geq 1 - \epsilon_k, \epsilon_k \geq 0, \forall k. \end{aligned}$$

where ϵ_k is the kth slack variable, C is a regularization parameter. t_k is the result of tracking, d_k is the result of detector. $\Phi(\bullet)$ with t_k and d_k is an vector represented with features. y_k is the label which is +1 if $a = a_6$, otherwise -1 if $a = a_5$.

3.3.2 MOT with MDPs

For MOT task, we can import all components of MDP (policy,reward function) of each object to generate MDPs network, In the actual scene of video stream, objects in "tracked" are processed to determine whether they should stay "tracked" or transit to "lost". Also we compute the similarity function between lost targets and object detections and use the reward function trained by reinforce learning with similarity score to acquire data assocation, According to the assignment, lost targets which are linked to some object detections are transferred to "tracked". Otherwise, they stay "lost". Overall, we initialize a MDP for each object detection which is not covered by any tracked target. **Algorithm** 1 describes our multi-object tracking algorithm using MDPs in detail according Xiang's work [20].

Algorithm 1 MOT with MDPs

input: A video sequence v and object detection $D = d_k$, where $k \in (1, N)$ and N is the number of patches of each frame for v,binary classifier (W, b) for data association.
output: Trajectories of objects $T = t_i$, where $i \in (1, M)$, and M is the number of frames.
Initialization:$T \longleftarrow \Phi$;
while Video frame is enable **do**
  for tracked object t_i in T **do**
    do MDP of t_i in "Tracked".
  end for
  for lost object t_i in T **do**
    Train the classification with reinforce learning jonit the data of t_i and d_k.
  end for
  Data association for all object in "Lost".
  for lost object t_i in T **do**
    do MDP of t_i in "Lost".
  end for
  Update the object with detector.
  for detected object d_k in D **do**
    do MDP of d_k in "Active".
  end for
end while

4. Experimental Results

As we improve the orignal MDP work with MOT [20], Changing the primitive objection detection methods to acquire the more precise result of detector in first step. Then, using TLD with the specific object we detecte to choose the principle objects to track. Assemble all tracking components with MDP to get the MOT. As **Figure** 5 shown that our results on the MOT dataset scene of AVG Town. Moreover, we capture some scene of real life on the campus of ShanghaiTech, as **Figure** 6 shown.

5. Conclusion

From the results, we found MOT with MDP,using detector of Faster RCNN and tracker of TLD has a stable effect in the scene camera moving or not. But under the condition of illumination change, the objects tracked maybe change their states to the "lost" ,then "inactive". Because the TLD using optical flow to predict the position of objects, it is sensitive to illumination conditions. In the future, we going to try using other feature representation or methods which is illumination invaranice to implement the MOT experiment and acquire more precise results.

Furthermore, we going to try the dataset and methods that can be used to do 3D tracking which aim to acquire more precise and differentiated representation of objects. Moreover, we can handle the occlusion better if we acquire the objects' depth. Many application can be implement because of that, such as the sports strategy mechine which can make policy rely on the status of two side by the tracking of athlete's moving and other conditions.

References

[1] S. Avidan. Ensemble tracking. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:494–501 vol. 2, 2005.

[2] S. Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007.

[3] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1631–1643, 2005.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

[5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260–267. IEEE, 2006.

[8] H. Grabner and H. Bischof. On-line boosting and vision. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:pages 260267, 2006.

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Computer Vision–ECCV 2008*, pages 234–247. Springer, 2008.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(9):1904–1916, 2015.

[11] B. H. J. C. Niebles and L. Fei-Fei. Efficient extraction of human motion volumes by tracking. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10)*, 2:pages 655662, 2009.

[12] N. D. F. J. J. L. K. Okuma, A. Taleghani and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. *2004 IEEE European Conference on Computer Vision (ECCV)*, page pages 2839, 2004.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[14] Y. L. L. Zhang and R. Nevatia. Global data association for multiobject tracking using network flows. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'08)*, 2:pages 18, 2008.

[15] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[16] 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.

[17] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[18] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[19] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[20] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. *2015 IEEE International Conference on Computer Vision (ICCV)*, 10.1109/ICCV.2015.534:4705–4713, 2015.

[21] C. H. Y. Li and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2:29532960, 2009.

[22] Q. Yu, T. B. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *Computer Vision–ECCV 2008*, pages 678–691. Springer, 2008.

[23] T. B. Z. Khan and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:pages 18051819, 2005.

Figure 5. AVG-TownCentre Data Set. MOT Result from frame 48 to 112 in 4 frames interval. The actual frame rate is 8 fps and the resolution is 1920×1080.

Figure 6. Real Scene On Campus of ShanghaiTech. MOT Result from frame 20 to 148 in 8 frames interval. The actual frame rate is 25 fps and the resolution is 1280×720.