Online Motion Agreement Tracking

Zheng Wu
wuzheng@bu.edu
Jianming Zhang
jmzhang@bu.edu
Margrit Betke
betke@bu.edu

Computer Science Department
Boston University
Boston, USA

Abstract

This paper proposes a fast online multi-target tracking method, called motion agreement algorithm, which dynamically selects stable object regions to track. The appearance of each object, here pedestrians, is represented by multiple local patches. For each patch, the algorithm computes a *local* estimate of the direction of motion. By fusion of the agreements between a *global* estimate of the object motion and each *local* estimate, the algorithm identifies the object stable regions and enables robust tracking. The proposed patch-based appearance model was integrated into an efficient online tracking system that uses bipartite matching for data association. The experiments on recent pedestrian tracking benchmark sequences show that the proposed method achieves competitive results compared to state-of-the-art methods, including several offline tracking techniques.

1 Introduction

The performance of pedestrian tracking systems has steadily increased during the past few years. Two factors mainly contributed to the improvement: the advance of robust pedestrian detectors [10] and various extensions of the data association technique [3, 5, 7, 15, 16, 17, 19, 21]. In contrast to previous work, this paper focuses on the importance of the appearance model in an online setting, which is orthogonal to the approaches of previous studies on *multi-target* tracking, but is a key problem in *single* object tracking applications [1, 4, 13]. The proposed method uses an off-the-shelf pedestrian detector [9] and a standard Hungarian bipartite matching procedure for data association. We introduce a new patch-based representation of each target to be tracked along with a sequential update scheme, which we call "motion agreement tracking" (MAT). Our multi-target MAT algorithm is able to achieve competitive results on widely accepted benchmark sequences. It can be implemented easily and applied efficiently.

The development of algorithms for online tracking of pedestrians has experienced a slow progress beyond applications of the particle filter. The performance for this category of tracking algorithms is typically limited by the accuracy of the pedestrian detector used, which is often inadequate. Developing a high-performing pedestrian detector remains an unsolved problem in computer vision. To handle the often poor quality of detections returned by the detector, such as false alarms and missed detections, online tracking systems have been implemented with ad-hoc designs. The most recent works prefer to process the videos in a batch mode, where the data is first organized into sets of short track fragments [15, 19, 21] or represented by a graph structure [17, 20]. Based on these data structures, the algorithms typically compute a global objective function offline and search for the best set of tracks to minimize this function. Various constraints such as track continuity, mutual exclusion, and motion smoothness have been imposed in order to narrow the search space and correct the errors produced in the detection stage. Online tracking algorithms, however, are not necessarily inferior to their offline counterparts. They are typically more efficient and can easily encode and filter high-order states, such as object velocity and acceleration, or the joint state of all objects in the scene, etc. [14]. Offline tracking algorithms usually require an extensive tuning process for model selection, without which the numerically optimal solution to the designed objective function is not the desired solution.

The work described here was motivated by an evaluation of the pros and cons of online versus offline tracking algorithms. We wanted to investigate whether the performance results of a new, well-designed, online 2D tracker, like the MAT algorithm, can measure up to those of state-of-the-art offline algorithms. Our experiments show that our proposed online MAT algorithm indeed outperforms state-of-the-art offline algorithms for various benchmark videos. Given its efficiency and ease-of-use, our MAT algorithm is even valuable for tracking scenarios where its performance is expected to be inferior: The tracks it produces online may be used as valuable initializations for offline tracking algorithms. The main reason why our proposed online MAT algorithm performs well lies in its superior object appearance model. The proposed representation is robust to pose variations, which helps maintain the object identities and prevent track switch errors. Designing object appearance models for visual tracking has been extensively explored by the research community for *single-target tracking* applications. In addition to various online learning algorithms proposed to update the appearance model by re-training the underlying classification model [4, 13], patch-based appearance representation has also been shown to be more effective than the holistic model [8, 11, 18]. However, it is not straightforward to transfer these techniques to online *multi-target tracking* applications, given the high computational expense of maintaining a model for each target. Our method also adopts a patch-based representation by identifying patches whose local motion directions agree with the global motion of the object. It turns out that the appearance of such patches remain relatively stable with low variance throughout the tracking period. We designed the MAT algorithm so that the contributions of these stable patches lead to a collectively agreed motion estimate of the object, which can then be passed on to the data association step in the multi-target tracking framework.

In summary, our contribution to online multi-target visual tracking is to provide method, called MAT, for sequentially updating the appearance model of each target by indirectly evaluating the motion consistency among its local patches. We show that a distance measure based on appropriately re-weighted local patches will successfully reduce tracker errors especially that lead to track fragmentation and track switching. The design as an online tracker permits a real-time implementation. Its accuracy on widely-accepted benchmarks is highly competitive compared to the state-of-the-art techniques.

2 Method

2.1 Online Multi-target Tracking System Overview

Our Motion Agreement Tracking system is outlined below. A tracker can be in any of the four states: initialization, stable, lost and terminated. The management of the trackers is determined by optimal bipartite matching of object states and detections. The motion dynamics are modeled by a Kalman filter. Our new component is to identify and maintain a set of robust sub-regions (local patches) for the appearance of each object and adjust the distance measure accordingly, which will then be used by the data association step.

MOTION AGREEMENT TRACKING ALGORITHM

For each frame:

Given a new set of detected objects $\{X\}$, a list of current stable trackers $\{S\}$, previously lost trackers $\{L\}$ and trackers just initialized $\{U\}$. Each tracker i maintains the object's motion model by a Kalman filter, and an appearance model: a collection $\{A_i\}_k$ of local patches along with their weights $\{w_i\}_k$.

Step 1 **Cost computation:** Compute the weighted matching cost c between detections $\{X\}$ and trackers $\{S\}$, $\{L\}$, $\{U\}$ according to Eq. 3.

Step 2 **Optimal assignment:** Solve the bipartite matching problem with the Hungarian method and find the assignments between detections and trackers.

Step 3 **Tracker management:** Each unassigned tracker in S is declared to be lost and added to list L. Each re-assigned tracker in L is declared to be stable again and added to list S. A new tracker is initialized for each unassigned detection. If a new tracker in U has not become lost for the past τ_1 frames, it is added to the stable list S. If a tracker in L has been lost for τ_2 frames, it terminates itself.

Step 4 **Model update:** For each tracker i that has received a new detection, update its Kalman filter and its patch weights $\{w_i\}_k$ according to Eq. 2. Update the appearance model A_{ik} of patch k if it is not in an occlusion relationship. For every tracker, predict the position of the object in the next frame.

2.2 Appearance Model

We designed a person-specific appearance model A with a collection of local image patches by dividing the bounding box of a detected person into a grid representation, as shown in Fig. 1. Each local patch k is described by a 64-bin color histogram in HSV space. Different ways to generate these local rectangle-shaped patches are possible in our framework. Each patch is associated with a weight w_k, which is set to be uniform when the tracker is initialized. A high weight suggests that the local patch does not change significantly over time, and low weight that the patch belongs to the background or represents a fast changing part of the pedestrian.

2.3 Region Motion Agreement and Weights Update

When a detection is assigned to the tracker after solving the assignment problem (Step 2 in Tracking Algorithm), the updated Kalman filter returns a filtered estimate of the object's global motion vector v for the current frame. At the same time, each patch computes its own motion vector v_k. It is important to make this local motion estimation step independent of the global tracking procedure, as we prefer to update the appearance model independent of the global motion estimation. Here, for simplicity, we estimate the local displacement of each patch based on a similarity measure. We compare two popular measures in our system: the maximum normalized cross-correlation between the feature of the patch H_T and the feature

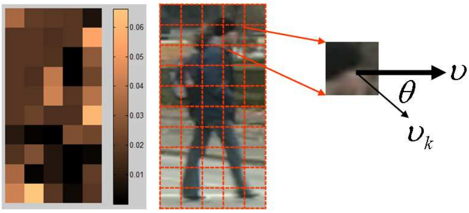

Figure 1: The model of a tracked pedestrian consists of a grid of local image patches (middle), its global motion vector v and a local motion vector v_k for each patch k (right). The angle θ_k between v and v_k is used to compute the influence, represented by weight w_k (left), of patch k in data association. A high weight is shown in light brown.

of the sub-image H_I; the minimum histogram intersection distance between H_T and H_I, since both of them are chosen to be histogram features.

Given the local motion estimates, our method evaluates each patch by checking the agreement between v_k and the global motion v. The intuition is that if v_k is similar to v, then this local patch moves along with the pedestrian, so it is more likely to be a stable region that does not undergo appearance change. Disagreement can be caused by local non-rigid deformation or the presence of background patches inside the bounding box. By focusing our effort on the most stable patches, we can construct a similarity measure that can distinguish between interacting objects. The level g of the agreement is computed by our implementation as follows:

$$\begin{aligned}\theta_k &= \cos^{-1}\left(\frac{v_k \cdot v}{||v_k||\,||v||}\right)\\ g_k &= \begin{cases} 2, & \text{if } \theta_k < \frac{\pi}{4} \\ 1, & \text{if patch is in an occlusion relationship (occluder or occluded)} \\ 0, & \text{otherwise} \end{cases}\end{aligned} \tag{1}$$

where θ_k is the angle between the two velocity vectors v and v_k as shown in Fig. 1. The motion score g is defined to be symmetric on both agreement and disagreement sides; the magnitude is not important here, as its contribution to the following weight update will be normalized. A score of $g_k = 1$ suggests a random guess. Using g_k, our algorithm updates the weight w_k associated with each patch k at time t as follows:

$$\begin{aligned}w_k(t) &= \frac{\alpha^{(t-1)} w_k^{(t-1)} + g_k,}{\alpha^{(t-1)} + \sum_k g_k}\\ \alpha^{(t)} &= \frac{\alpha^{(t-1)} + \sum_k g_k}{2},\end{aligned} \tag{2}$$

where α is a self-adaptive learning rate which controls how much the current estimates influence the update. A large value of α suggests a smooth update of the weights at the current frame, which happens when most local motion estimates agree with the global motion model. Both α and w are non-negative, and w is always normalized.

Finally, the distance function between an object i and a candidate detection j is given as:

$$c_{i,j} = \left(1 - \frac{|b_i \bigcap b_j|}{|b_i \bigcup b_j|}\right) + \lambda \sum_{k}^{\#\text{patches}} w_k \left(1 - \sum_{n}^{\#\text{bins}} \min(h_i^n, h_j^n)\right), \tag{3}$$

where first term evaluates the number of non-overlapping regions comprising i and j by computing one minus the ratio of overlap between their two bounding boxes (this number is relatively small, because the speed of pedestrian is typically slow in high-frame-rate video). The second term in Eq. 3 measures the difference of appearance of i and j by computing the intersection of the normalized HSV histogram bins, denoted as h_i^n, h_j^n, for each corresponding patch, and arriving at an agreement value by subtracting the individual contributions from one, weighing them by w_k, and summing them over all patches. Parameter λ balances the contributions of the motion and appearance terms and is set to be one empirically. This distance function is used in the data association step that determines the assignment of a detection to its corresponding tracker, as described in Sec. 2.1.

3 Experiments

We extensively tested our MAT system on recent popular datasets, which include 3 sequences from the PETS2009 dataset [1], 3 sequences from the TUD dataset [2], the Towncenter dataset [3], and 2 sequences from the ETH dataset [4]. For the first three datasets, we used a Matlab implementation of an off-the-shelf pedestrian detector [9] to provide the detection candidates, while for the ETH dataset we used publicly-available detection results to provide a fair comparison [5].

We used the standard CLEAR MOT metrics [6] to evaluate the 2D tracking performance. The Multiple Object Tracking Accuracy (MOTA) combines false positive rate, miss rate, and identity switch rate into a single number with ideal value 100%; Multiple Object Tracking Precision (MOTP) measures the average distance between the ground truth and the tracker output according to the region overlap criterion, where we chose 0.5 as the standard hit/miss threshold (the default aspect ratio of the bounding boxes given by the detector [9] is not perfectly aligned with the ground truth, and as a result, it may underestimate our precision on some of the sequences). To better assess the quality, we additionally report the numbers of Mostly Tracked (MT, ($\geq$ 80%) trajectories, Mostly Lost (ML, $\leq$ 20%) trajectories, track fragmentations (FM), and identity switches (IDS). In order to align with the results from the literature, the matches between the system-generated tracks and the ground truth are determined by a greedy search. Another common implementation with the bipartite matching method will generally give slightly higher scores.

We first tested on the 7 most popular sequences from the recent literature. Our quantitative results are shown in Table. 1. To analyze the effect of our appearance model, we developed a baseline method where all components are kept the same except that the model of object appearance does not use a grid of patches but is simplified to be a single HSV histogram of the entire bounding box. As a result, the motion agreement computation is not triggered. Throughout this experiment, the size of the grid is 6×3; α is initialized to be 0.1; and we choose histogram intersection as the similarity measure for local motion estimation. The online tracking algorithm by Zhang et al. [22] also has a rich representation that models the object holistically in multiple color spaces, which they call "template ensemble." The results were provided by the latest version of the tracker from the authors, given our

[1] http://www.cvg.rdg.ac.uk/PETS2009/
[2] http://www.d2.mpi-inf.mpg.de/datasets
[3] http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/
[4] http://www.vision.ee.ethz.ch/~aess/dataset/
[5] http://iris.usc.edu/people/yangbo/downloads.html

Data	Method	MOTA(%)	MOTP(%)	MT	ML	FM	IDS
S2L1	Baseline	90.8	74.3	23	0	11	10
(cropped)	Our MAT	92.8	74.3	23	0	11	8
	Zhang et al. [22]	91.0	66.1	22	0	16	10
	*Andriyenko et al. [2]	88.3	75.7	20	1	-	-
S2L2	Baseline	67.8	72.9	51	3	149	166
(cropped)	Our MAT	73.3	73.2	51	3	113	122
	Zhang et al. [22]	58.9	67.3	27	6	168	173
	*Andriyenko et al. [2]	60.2	60.5	25	8	-	-
S2L3	Baseline	55.6	69.9	18	8	48	58
(cropped)	Our MAT	58.3	69.7	21	8	39	41
	Zhang et al. [22]	42.2	64.9	10	14	36	34
	*Andriyenko et al. [2]	43.8	66.3	10	20	-	-
Stadtmitte	Baseline	75.1	70.0	9	0	2	3
	Our MAT	75.4	70.0	9	0	2	3
	Zhang et al. [22]	75.0	59.8	6	0	1	2
	*Andriyenko et al. [2]	68.6	64.0	5	0	-	-
Crossing	Baseline	90.2	76.8	11	0	6	10
	Our MAT	90.6	76.9	11	0	5	8
	Zhang et al. [22]	71.3	67.5	7	0	15	11
	*Breitenstein et al. [7]	84.3	71.0	-	-	-	2
Campus	Baseline	68.5	71.3	4	0	5	5
	Our MAT	68.5	71.3	4	0	5	5
	Zhang et al. [22]	74.7	68.0	6	0	4	3
	*Breitenstein et al. [7]	73.3	67.0	-	-	-	2
Towncenter	Baseline	69.4	68.7	139	18	462	222
	Our MAT	69.5	68.7	139	17	453	209
	Zhang et al. [22]	73.6	71.3	163	16	161	157
	*Benfold et al. [5]	64.8	80.4	-	-	-	259
	*Pellegrini et al. [16]	63.4	70.7	-	-	-	183

Table 1: Quantitative results on 7 publicly available sequences. Method indicated with "$*$" used its own pedestrian detector, and we list their results directly from published literature. Top score in each metric is highlighted in red. Note that we only track objects in a restricted area on PETS sequences, which is defined by Andriyenko et al. [2].

detection output. We also list several recent tracking techniques that have reported superior performance on the same sequence so that the readers can have a better view of the challenge of the data. These state-of-the-art techniques include batch energy minimization with an occlusion model by Andriyenko et al. [2], a tracker that encodes social behavior by Brendel et al., and two variants of a particle filter [5, 7]. As expected, our new appearance model mostly improves the tracking performance by reducing track fragmentation and ID switches, especially for a crowd with partial visibility (S2L2, S2L3). For "sparse situations," where pedestrians seldom interact, a strong appearance model cannot contribute much. The MAT algorithm also shows only marginal improvements if the pedestrian is always in severe, even complete occlusion (TUD dataset). Overall, our MAT algorithm achieves consistently good results across all sequences. Since it is conceptually simple and runs at 1–2 fp with a Matlab implementation, we plan to convert it to a real-time tracker and make it available, so it can serve as an efficient baseline algorithm for future studies.

To remove the effect of the pedestrian detector used, we conducted two additional experiments with the same detections as input. We also tested our system with different parameters. In particular, we chose two grid sizes, 6×3 and 10×5; three initial values of α, 0.1, 1 and 10; two similarity measures, the maximum normalized correlation and the minimum histogram intersection. In total, we analyzed the mean performance and its standard

Data	Method	MOTA(%)	MOTP(%)	MT	ML	FM	IDS
S2L1-full	Baseline	88.6	74.2	19	0	17	13
	Our MAT	90.1(±0.2)	74.3(±0.0)	19	0	17.5(±0.5)	10.5(±0.5)
	DP [17]	82.2	72.5	17	0	102	184
	DCT [3]	56.8	74.4	17	0	59	56
	DCT+DP [3]	77.0	74.5	16	0	63	58
S2L2-full	Baseline	69.5	72.5	31	0	188	200
	Our MAT	72.1(±0.6)	72.6(±0.0)	32(±0.0)	0.5(±0.5)	165.3(±4.9)	179.3(±4.4)
	DP [17]	54.9	73.1	8	2	394	501
	DCT [3]	35.7	69.4	4	0	492	525
	DCT+DP [3]	47.6	70.0	7	0	394	445
S2L3-full	Baseline	50.7	69.5	19	7	63	69
	Our MAT	52.5(±0.9)	69.5(±0.1)	16.4(±0.9)	7.5(±0.6)	60.8(±4.6)	62.9(±5.0)
	DP [17]	40.0	70.7	11	18	115	156
	DCT [3]	21.3	70.6	5	15	236	278
	DCT+DP [3]	32.4	70.7	7	15	97	103

Table 2: Quantitative results on PETS sequences. The MAT algorithm tracked all pedestrians in the videos. The performance of MAT with different system parameters is expressed in the form of mean(std). The top score in each metric is highlighted in red. Competing methods are evaluated by code from the authors' website with default parameter settings.

deviation from 12 system configurations. Given the same detection results, we compared our tracker with the batch energy minimization method (DCT) by Andriyenko et al. [3] and the batch network-flow method (DP) by Pirsiavash et al. [17] using their publicly available code. The DCT method requires a good initialization with tracklets. We used the tracks produced by the DP method as suggested in their paper. The results on the PETS dataset are shown in Table. 2. Again, we witnessed the consistent reduction of number of fragments and ID switches compared to our baseline tracker, and the performance is stable across different configurations. Our online MAT method performs surprisingly better than batch processing methods, which are more computationally expensive. In particular, the network-flow method has an inherent bias on the length of tracks it produces (its objective function tends to favor many small track fragments). It is also difficult to encode high-order state such as velocity or acceleration to the network which results in more ID switches than from Bayesian filter-based method. We also found the DCT energy minimization method has strong dependence on the initialization step in order to reduce the search space and avoid many local minima. Essentially, it is a trajectory-fitting method that focuses more on the smoothness of tracks. This limits its ability to handle irregular non-smooth motion patterns, which can be modeled more easily by the Bayesian filtering method. It also suffers from a model selection problem during its optimization procedure. Very often we saw that the solution that achieves higher tracking accuracy does not necessarily suggest a lower energy, which makes parameter tuning difficult.

Finally, we evaluated our trackers with detections provided by Yang et al. [21] on two sequences (Bahnhof and Sunny Day) from the ETH dataset. We chose 3 competing algorithms that report superior performance on these two sequences in literature: the batch network-flow method (DP) by Pirsiavash et al. [17] and two batch tracklet stitching methods (PIRMPT and CRF) [15, 21]. To make consistent comparisons with their reported results, we computed the metrics suggested by the authors [15, 21] using their software, which are slightly different from CLEAR MOT metrics. The results are shown in Table 3. We again found the inherent bias of the flow-based method to produce many track fragments. The two tracklet stitching methods learned a strong discriminate model with training examples extracted from a set of reliable tracklets. Our online tracker is not expected to achieve better performance than these complicated systems in terms of ID switches, but may provide good initialization to them.

Data	Method	Recall	Precision	MT(%)	ML(%)	FM	IDS
BAHNHOF	Our MAT	85.7	84.2	79.8	6.4	42	38
SUNNYDAY	Our MAT	78.9	75.8	83.3	6.7	3	7
All	Our MAT	84.5	82.6	80.6	6.5	45	45
	DP [17]	67.4	91.4	50.2	9.9	143	4
	PIRMPT [15]	76.8	86.6	58.4	8.0	23	11
	CRF [21]	79.0	90.4	68.0	7.2	19	11

Table 3: Quantitative results on the ETH dataset. The top score in each metric is highlighted in red. For details of the metrics, see [21].

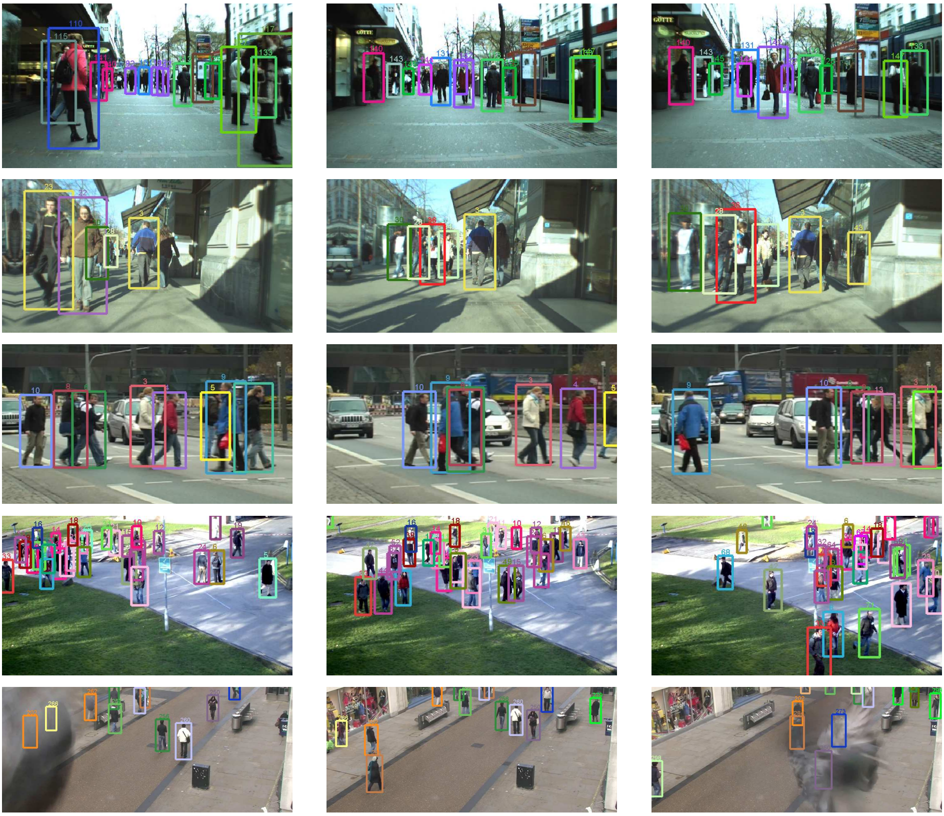

Figure 2: Sample images from our tracking results. Colors and numbers indicate tracks corresponding to different people.

4 Conclusion

We proposed an online multi-target tracking algorithm with a dynamic appearance model. The local regions that remain stable in time are discovered by a novel technique called “motion agreement tracking.” When a local motion estimate agrees with the global estimate, the algorithm considers such local patch to be stable and increases its weight to contribute a smaller value to the distance measure. We integrated our technique into an online tracking system and tested it extensively on popular tracking benchmarks. Our competitive results are particularly appealing since the technique is so efficient. They also suggest that the role of a proper appearance model may be more important than researchers used to think for the tracking application, where the majority of previous studies focuses on motion dynamics. The proposed motion agreement tracking algorithm can be utilized as a new baseline to help identify the challenges of future benchmarks and the limits of current tracking techniques.

Acknowledgements. This material is based in part upon work supported by the NSF (IIS-0910908, IIS-0855065 and IIS-1029430) and ONR (024-205-1927-5). We would like to thank Dr. Liliana Lo Presti and Prof. Stan Sclaroff for their assistance.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[2] A. Andriyenko, S. Roth, and K. Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *Proceeding of the 11th IEEE Workshop on Visual Surveillance*, pages 1839 – 1846, 2011.

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012.

[4] B. Babenko, Ming-Hsuan Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619 –1632, 2010.

[5] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, pages 3457–3464, 2011.

[6] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008.

[7] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI*, 33(9): 1820 –1833, 2011.

[8] L. Cehovin, M. Kristan, and A. Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *ICCV*, 2011.

[9] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

[10] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34(4):743–761, 2012.

[11] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011.

[12] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 34(7): 1409–1422, 2012.

[14] Z. Khan, T. Balch, and F. Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *PAMI*, 28(12):1960 – 1972, 2006.

[15] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking. In *CVPR*, 2011.

[16] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You will never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.

[17] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1–8, 2011.

[18] T. Vojir and J. Matas. Robustifying the flock of trackers. In *Computer Vision Winter Workshop*, 2011.

[19] Z. Wu, M. Betke, and T. H. Kunz. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *CVPR*, 2011. 8 pp.

[20] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR*, 2012.

[21] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012.

[22] J. Zhang, L. Lo Presti, and S. Sclaroff. Online multi-person tracking by tracker hierarchy. In *Proceeding of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2012.