Adaptive Multiple Appearances Model Framework for Long-Term Robust Tracking

Shuo Tang, Longfei Zhang(✉), Jiapeng Chi, Zhufan Wang, and Gangyi Ding

School of Software, Beijing Institute of Technology, Beijing, China
{tang_shuo,longfeizhang,dgy}@bit.edu.cn

Abstract. Tracking an object in long term is still a great challenge in computer vision. Appearance modeling is one of keys to build a good tracker. Much research attention focuses on building an appearance model by employing special features and learning method, especially online learning. However, one model is not enough to describe all historical appearances of the tracking target during a long term tracking task because of view port exchanging, illuminance varying, camera switching, etc. We propose the Adaptive Multiple Appearance Model (AMAM) framework to maintain not one model but appearance model set to solve this problem. Different appearance representations of the tracking target could be employed and grouped unsupervised and modeled by Dirichlet Process Mixture Model (DPMM) automatically. And tracking result can be selected from candidate targets predicted by trackers based on those appearance models by voting and confidence map. Experimental results on multiple public datasets demonstrate the better performance compared with state-of-the-art methods.

Keywords: Dirichlet process mixture model · Appearance model · Object tracking

1 Introduction

Robust object tracking in a long term is a challenging task in computer vision. There have been many trackers proposed by different researchers [1–3] that employ different types of visual information and learned features to build the appearance models as the base of the tracking, e.g. color histogram in Meanshift, multiple features in particle filter [4], Haar-like in MIL [5], etc.

However, it is still not enough to represent the tracking target with one appearance model, even online updating model or patch dictionary model and so on, while the target in internal and external variations. Internal variation includes pose changing, motion, shape deformation, illumination variation, etc. And External variation includes background changing, covered by foreground objects. Tackle this problem, an appearance model set is needed for describing the historical appearances of tracking target.

Therefore, we propose a novel nonparametric statistical method to model the appearance of the target as combination of multiple appearance models. Each

Y.-S. Ho et al. (Eds.): PCM 2015, Part I, LNCS 9314, pp. 160–170, 2015.
DOI: 10.1007/978-3-319-24075-6_16

model describes an typical appearance character under specific situation, and clustered by Dirichlet Process Mixture Model (DPMM) [7] framework dynamically unsupervised.

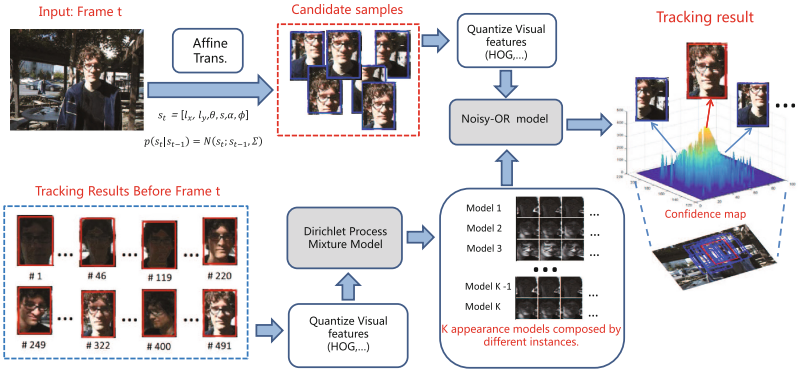


Fig. 1. The framework of the adaptive multiple appearance model tracking.

The Framework of our system is shown in Fig. 1. Experimental results on several public datasets show that AMAM tracking system is applicable to multiple camera system and indoor and outdoor climates tracking system, and outperform several state-of- the-art trackers.

The rest of the paper is organized as follows. Section 2 overviews some of the related works. Section 3 describes the proposed AMAM algorithm. We present experimental validation on several public datasets in Sect. 4 and conclude the paper in Sect. 5.

2 Related Works

In long term tracking task, the biggest challenge is drifting problem. Tackle this problem, appearance models tolerance range need to be enhanced. Ensemble tracking [9] and the Multiple Instance Learning boosting method (MIL) [5] using positive sample and negative samples of tracking targets to train classifiers. Semi-online boosting [10] using both unlabeled and labeled tracking candidate target to train classifiers online. Fragment-based tracking [16] coupled with a voting map can accurately track the partially occluded target. However, historical information is ignored when updating classifiers or models. Dictionary learning [15] was employed to using the linear combination to represent the dynamic appearance of the target and handles the occlusion as a sparse noise component. However, spatial and temporal information are lost when algorithm performing. Appearance representation learned by In our model, we build appearance model set to keep the spatial information of tracking target and tracking system could keep the temporal information as well. at the same time, all efficient

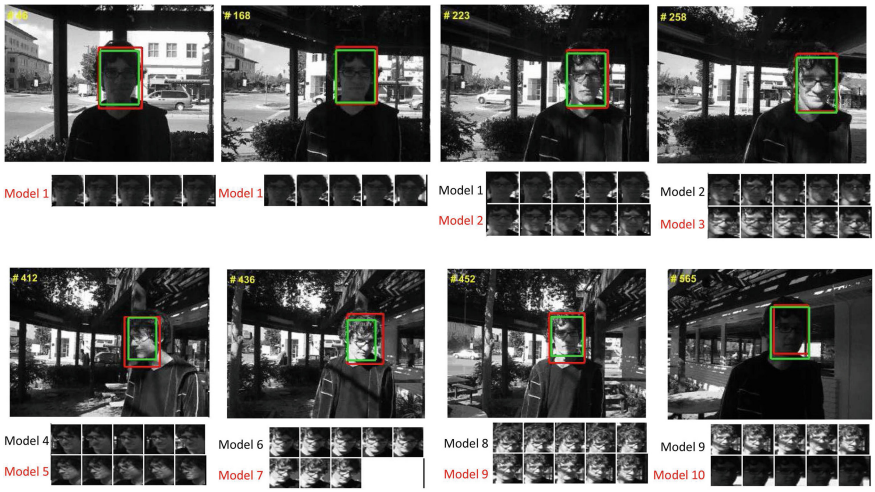


Fig. 2. The procedure of AMAM framework working.

appearance model can be employed in this framework including sparse coding, dictionary learning and learned target descriptions by deep learning or other machine learning methods.

3 The Framework of Adaptive Multiple Appearances Model Tracking

In this section, we describe the common framework of adaptive multiple appearance model. In first, we present the Dirichlet Process Mixture model (DPMM), which are employed to organize the adaptive appearance set. After that, we describe the tracking system based on AMAM framework.

3.1 Dirichlet Process Mixture Model

The Dirichlet process (DP) is parameterized by a base distribution H which has corresponding density $h(\theta)$, and a positive scaling parameter $\alpha > 0$. We denote a DP and suppose we draw a random measure G from a DP, and independently draw N random variables θ_n from G, this can be described as follows:

$$G|\{\alpha, H\} \sim DP(\alpha, H) \tag{1}$$

$$\theta_n \sim G,\ n \in \{1, \ldots, N\}$$

As shown by [8], given N independent observations $\theta_i \sim G$, the posterior distribution also follows a DP:

$$p(G \,|\, \theta_1, \ldots, \theta_N, \alpha, H) \sim Dir(\alpha H(A_1) + n_1, \ldots, \alpha H(A_r) + n_r)$$

$$\sim DP\left(\alpha+N,\frac{1}{\alpha+N}\left(\alpha H+\sum\nolimits_{i=1}^{N}\delta_{\theta_i}\right)\right) \tag{2}$$

where $n_1 n_2, ..., n_r$ represent the number of observations falling in each of the partitions $A_1 A_2, ... A_r$ respectively,N is the total number of observations, and δ_{θ_i} represents the delta function at the sample point θ_i.

3.2 Model Inference

Given N observations $X = \{X_i\}_{i=1}^{N} (X_i \in N^d)$, each $X_i = \{x_i\}_{i=1}^{d}$ represents a quantized d-dim HOG feature, and x_i is the histogram quantized bin counts, which is a quantized integer. Let z_i indicate the cluster or appearance model, associated with the i^{th} observation which is represented by quantized HOG feature. As shown in Fig. 1, we would like to infer the number of latent clusters or different appearances underlying those observations, and their parameters θ_k. Since the exact computation of the posterior is infeasible especially when data size is large, we resort to a variant of MCMC algorithms, namely, the collapsed Gibbs sampler [7] for faster approximate inference.

We choose multinomial distribution $F(\theta)$ to describe HOG features of observations, and the cluster prior $H(\lambda)$ is a Dirichlet distribution which is conjugate to $F(\theta)$. Given fixed cluster assignments z_{-i} for other observations, the posterior distribution of z_i factors as follows:

$$p\left(z_i \mid z_{-i}, X, \alpha, \lambda\right) \propto p\left(z_i \mid z_{-i}, \alpha\right) p\left(X_i \mid X_{-i}, z, \lambda\right) \tag{3}$$

The prior $p\left(z_i \mid z_{-i}, \alpha\right)$ is given by the Chinese restaurant process (CRP).

$$p\left(z_i \mid z_{-i}, \alpha\right) \sim \frac{1}{\alpha+N-1}\left(\sum_{k=1}^{K} N_k^{-i}\delta\left(z_i, k\right)+\ \alpha\delta\left(z_i, \bar{k}\right)\right) \tag{4}$$

The $\bar{k}$ denotes one of the infinitely many unoccupied clusters or new appearances. N_k^{-i} is the total number of observations in cluster k except observation i.

For the K clusters to which z_{-i} assigns observations, the likelihood of Eq. (3) is shown as follows:

$$p\left(X_i \mid z_i = k, z_{-i}, X_{-i}, \lambda\right) = p\left(X_i \mid \left\{X_j \mid z_j = k, j \neq i\right\}, \lambda\right) \tag{5}$$

Because dirichlet distribution $H(\lambda)$ is conjugate to multinomial distribution $F(\theta)$, $\theta = (p_1, p_2, ..., p_d)$ and $\{X_i\}_{i=1}^{N} \sim Mult\left(p1, p2, ..., pd\right)$, we can get a closed-form of predictive likelihood expression for each cluster or appearance k as follows:

$$p\left(X_i \mid z_i = k, z_{-i}, X_{-i}, \lambda\right) = \frac{\Gamma\left(n+1\right)}{\prod\limits_{j=1}^{d}\Gamma\left(X_i^{(j)}+1\right)}\frac{\Gamma\left(\sum_{j=1}^{K}\lambda_j'\right)}{\prod\limits_{j=1}^{K}\Gamma\left(\lambda_j'\right)}\frac{\prod\limits_{j=1}^{K}\Gamma\left(X_i^{(j)}+\lambda_j'\right)}{\Gamma\left(n+\sum_{j=1}^{K}\lambda_j'\right)} \tag{6}$$

Algorithm 1. DPMM algorithm.

Set multiple appearance model assignments $z = z_{t-1}$ of observation at last iteration $t-1$. For each $i \in \{1, ..., N\}$ frame targets, resample and rebuild each appearance model z_i as follows:

repeat

Step1. Remove a data item X_i from its appearance model k and update its model parameter θ_{k}.

Step2. For each of the K existing clusters, determine the predictive likelihood using Eq. 6, and then determine the likelihood of a potential new appearance cluster $\bar{k}$ via Eq. 7.

Step3.Sample cluster assignment z_i from the $(K+1)$ - dim multinomial Eq. 4.

Step4. Update appearance model parameter θ_k to reflect the assignment of x_i to cluster z_i. If $z_i = \bar{k}$, create a new cluster and increment cluster or appearance number K.

Step5. Repeat the steps shown above until convergence.

until $i = k$;

where $\lambda^{'}$ is the posterior of λ and Γ is the gamma function. Similarly, new clusters $\bar{k}$ are based upon the predictive likelihood implied by the prior hyper parameters λ:

$$p\left(X_i \mid z_i = \bar{k}, z_{-i}, X_{-i}, \lambda\right) = p\left(X_i \mid \lambda\right) = \frac{\Gamma\left(A\right)}{\Gamma\left(N + A\right)} \prod_{k=1}^{K} \frac{\Gamma\left(n_k + \lambda_k\right)}{\Gamma\left(\lambda_k\right)} \tag{7}$$

where $A = \sum_k \lambda_k$ and $N = \sum_k n_k$, and where n_k = number of x_i's with value k. Combining these expressions, we employed Gibbs sampler at Algorithm 1.

3.3 AMAM Tracking

Given the observation set of the target$X_{1:t} = [X_1, ..., X_t]$, where each X_t represents a quantized HOG feature, up to time t, the tracking result s_t can be determined by the Maximum A Posteriori (MAP) estimation, $\hat{s_t} = argmaxp\left(s_t|X_{1:t}\right)$, where $p\left(s_t|X_{1:t}\right)$ is inferred by the Bayes theorem recursively with

$$p\left(s_t \mid X_{1:t}\right) \propto p\left(X_t \mid s_t\right) \int p\left(s_t \mid s_{t-1}\right) p\left(s_{t-1} \mid X_{1:t-1}\right) ds_{t-1}. \tag{8}$$

Let $s_t = [l_x, l_y, \theta, s, \alpha, \phi]$, where $l_x, l_y, \theta, s, \alpha, \phi$ denote x, y translations, rotation angle, scale, aspect ratio, and skew respectively.We apply the affine transformation with those six parameters to model the target motion between two consecutive frames. The state transition is formulated as $p\left(s_t|s_{t-1}\right) = N\left(s_t; s_{t-1} \sum\right)$, where $\sum$ is the covariance matrix of six affine parameters.

The observation model $p\left(X_t|s_t\right)$ denotes the likelihood of the observation X_t at state s_t. The Noisy-OR (NOR) model is adopted for doing this:

Algorithm 2. AMAM Tracking Algorithm.

For each$i \in \{1, ..., N\}$, resample z_i as follows:
repeat
Step1. $L_t(X)$ denote the location of sample X_t at the t-th frame. We have the object location $L_t(X)$ where we assume the corresponding sample is X_t representing the quantized HOG feature.
Step2. We apply the affine transformation to $L_t(X)$ with six affine parameters to product candidate samples s_{t+1}.
Step3. For each candidate samples s_{t+1}, we extract quantized HOG feature X_{t+1}, then use NOR model of Eq. 9 and each of the multiple appearance models H^k to compute the likelihood of X_{t+1}.
Step4. We select the state s_{t+1} which has maximum probability of X_{t+1}
Step5. Let $X_{1:t+1} = [X_1, ..., X_{t+1}]$ which represents the quantized HOG features set of the target up to time $t + 1$, and then use Algorithm 1 to recreate multiple appearance models.
until $i = N$;

$$p\left(X_t \mid s_t\right) = 1 - \prod\nolimits_k \left(1 - p\left(X_t \mid s_t, H^k\right)\right) \tag{9}$$

where $H_k, k \in (1, 2, ..., K)$ represents the multiple appearance models learned from Algorithm 1.

The equation above has the desired property that if one of the appearance models has a high probability, the resulting probability will be high as well.

Algorithm 2 illustrates the basic flow of our algorithm.

4 Experiments

In our experiments, we employ 10 challenging public tracking datasets selected from [2] and using the same evaluation methods, the center location error and success rate, to verify the performance of our algorithm. The proposed approach is compared with ten state-of-the-art tracking methods. Table 1 shows all the tracking methods we need to evaluate. In addition, we evaluate the proposed tracker against those methods using the source codes provided by the authors and each tracker is running with adjusted parameters for fair evaluation.

4.1 The AMAM Modeling

Figure 2 shows how the AMAM working. These small face images under the main frame shows the appearance instance belong to each appearance model and the historical instances while tracking. The red rectangle in main frame is the tracking result based on the model in red, and the green one is the ground truth. The instances of each appearance models increasing while long term tracking, and the number of appearance models increasing while the inner and inter distance changing based on the DPMM Algorithm 1.

Table 1. Compare trackers and their representations in our experiment [2]

Trackers	Representation	Trackers	Representation
LOT [17]	L, Color	IVT [18]	H, PCA
ASLA [19]	L, SR, GM	L1ANG [20]	H, SR, GM
MTT [21]	H, SR, SM	VTD [22]	H, SPCA, GM
OAB [23]	H, Haar, DM	MIL [5]	H, Haar, DM
TLD [21]	L, BP, DM	Struck [25]	H, Haar, DM
Ours	H, DPMM		

4.2 Tracking System

Figure 3 shows how the AMAM tracking results based on 2. Bounding boxes in red are our results. We can find that our tracker can track the target very well while most of the other tracker are drifting.

In order to measure the performance of tracking result, we employed two traditional measurement operator. One is the center error (CE), and the other is coverage rate (CR).

The center error is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths, for calculating precision plot. In a general way, the overall performance of one sequence depends on the average center location error over all the frames of one sequence, but when the tracker loses the target, we will only get the random

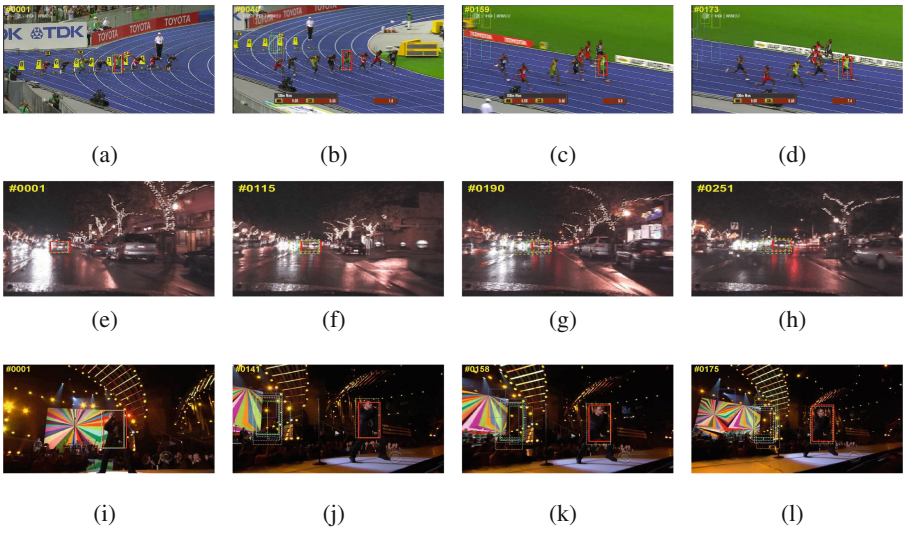
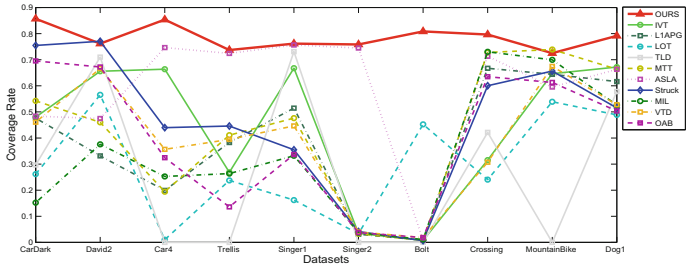
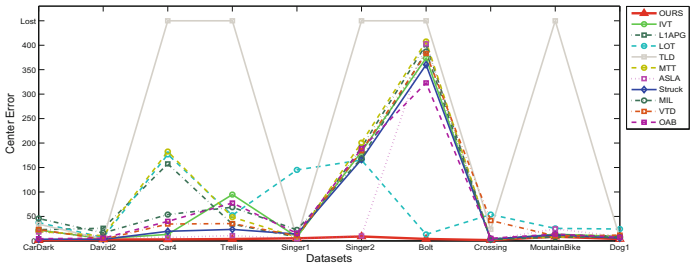


Fig. 3. Three tracking video sequences with all tracking result from the all trackers. The bounding boxes in red are our results.



(a)



(b)

Fig. 4. Tracking result compare with the recent appeared 10 trackers listed in Table 1 by coverage rate (a) and center error (b) and measurement operators in public datasets.

output location and the average error value which may not measure the tracking performance correctly [5]. Therefore, we use the precision plot to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. Figure 4(b) shows result in our experiment. Since the smaller is better, our AMAM tracker performs good in those public testing videos.

The coverage rate is defined as the bounding box overlapping rate between the tracking target and the ground truth. A higher score means the tracking result is closer to ground truth. The formula of calculating score is $score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)}$ while the formula of calculating average score is $avrScore = \frac{\sum_{1}^{frameLength} score}{frameLength}$.

In Table 2, we compare the performance of trackers in each testing dataset with the same testing result shown in Fig. 4. We selected the best performed tracker and the second tracker in each testing data both based on CR and CE operators. We also calculate the differences between them for tracking accuracy measuring. In Table 3, we also import the variation and average CR to measure the robustness and accuracy. Since the inhumations, backgrounds and targets are different at all, if the tracker performing stable with low variation, the tracker can be considered robustness.

Table 2. Compare of all trackers in all datasets by converge rate (CR) and center error (CE).

Dataset	Our CE	Other best CE	CE differences	Our CR	Best CR of others	CR differences
carDark	1.24	3.42	2.18	0.857	0.7549	0.1021
david2	2.89	3.07	0.18	0.7619	0.7708	-0.0089
car4	3.01	6.8	3.79	0.8538	0.7466	0.1072
trellis	3.84	10.86	7.02	0.7372	0.7243	0.0129
singer1	5.22	4.22	-1	0.7617	0.7566	0.0051
singer2	8.97	9.96	0.99	0.7585	0.7448	0.0137
bolt	3.91	13.18	9.27	0.8085	0.4524	0.3561
crossing	1.58	2.51	0.93	0.7967	0.7304	0.0663
mountainBike	8.61	7.75	-0.86	0.7253	0.7391	-0.0138
dog1	3.59	5.48	1.89	0.7916	0.6719	0.1197

Table 3. Compare of trackers by variance and average coverage rate (ACR) in performance.

Our variance	Other's min variance	Mean variance	Our ACR	Best other ACR
0.002031	0.041081246	0.06492	0.78522	0.59107

From the Table 2, we find that our AMAM Tracker is outperform in 8 testing videos. The differences between best and our tracker in CR and CE are less than 1.4 % and 1 pixel in the rest 2 testing videos.

From the Table 3, the variation of our tracker in all videos is 0.002, extremely lower than others both in average and individual.The ACR of our AMAM tracker in all testing videos is 19 % higher than others. That means our tracker can perform more robust and more accurate than others.

5 Conclusion

This paper tackled the drifting problem in tracking and proposed an Adaptive Multiple Appearance Model framework for long term robust tracking. We simply employed HOG to build the basic appearance representation of the tracking target in experiment but all efficient representation of visual objects could be joint in our algorithm framework. Historical appearance descriptions could be employed and grouped unsupervised and modeled by Dirichlet Process Mixture Model automatically. And tracking result can be selected from candidate targets predicted by trackers based on those appearance models by voting and confidence map. Experiment in several public datasets shows that, our tracker has low variation (less than 0.002) and high tracking performance (19 % better than other 10 trackers in average) when compared with the state-of-the-art methods.

Acknowledgments. This material is based upon work supported by the Key Technologies Research and Development Program of China Foundation under Grants No. 2012BAH38F01-5. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Key Technologies Research and Development Program of China Foundation.

References

1. Li, X., Hu, W., Shen, C., et al.: A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. (TIST) **4**(4), 58:1–58:48 (2013)
2. Wu, Y., Lim, J., Yang, M.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2411–2418 (2013)
3. Smeulders, A., Chu, D., Cucchiara, R.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **36**(7), 1442–1468 (2014)
4. Wang, H., Suter, D., Schindler, K.: Effective appearance model and similarity measure for particle filtering and visual tracking. In: Proceedings of European Conference Computer Vision (ECCV) (2006)
5. Babenko, B., Yang, M., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **33**(8), 1619–1632 (2011)
6. Wang, N., Yeung, D.: Learning a deep compact image representation for visual tracking. In: Proceedings of the NIPS, (5192), pp. 809–817 (2013)
7. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. J. Comput. Graph. Stat. **9**, 249–265 (2000)
8. Ferguson, T.: A Bayesian analysis of some nonparametric problems. Ann. Stat. **1**(2), 209–230 (1973)
9. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007). IEEE society
10. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proceedings of Computer Vision and Pattern Recognition, **1**, pp. 260–267 (2006)
11. Stenger, B., Woodley, T., Cipolla, R.: Learning to track with multiple observers. In: Proceedings of Computer Vision and Pattern Recognition, pp. 2647–2654 (2009)
12. Yu, Q., Dinh, T.B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: ECCV (2008)
13. Gao, Y., Ji, R., Zhang, L., Hauptmann, A.: Symbiotic tracker ensemble toward a unified tracking framework. IEEE Trans. Circuits Syst. Video Technol. (TCSVT) **24**(7), 1122–1131 (2014)
14. Zhang, L., Gao, Y., Hauptmann, A., Ji, R., Ding, G., Super, B.: Symbiotic black-box tracker. In: Proceedings of the Advances on Multimedia modeling (MMM), pp. 126–137 (2012)
15. Wang, N., Wang, J., Yeung, D.: Online robust non-negative dictionary learning for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV 2013), pp. 657–664 (2013)
16. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Proceedings of the CVPR (2006)
17. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: CVPR (2012)

18. Ross, D., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. IJCV **77**(1), 125–141 (2008)
19. Jia, X., Lu, H., Yang, M.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR (2012)
20. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust L1 tracker using accelerated proximal gradient approach. In: CVPR (2012)
21. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: CVPR (2012)
22. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)
23. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via online boosting. In: BMVC (2006)
24. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N learning: bootstrapping binary classifiers by structural constraints. In: CVPR (2010)
25. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. In: ICCV(2011)