Visual Tracking by Local Superpixel Matching with Markov Random Field

Heng Fan[1], Jinhai Xiang[2(✉)], and Zhongmin Chen[2]

[1] College of Engineering, Huazhong Agricultural University, Wuhan, China
[2] College of Informatics, Huazhong Agricultural University, Wuhan, China
jimmy_xiang@mail.hzau.edu.cn

Abstract. In this paper, we propose a novel method to track non-rigid and/or articulated objects using superpixel matching and markov random field (MRF). Our algorithm consists of three stages. First, a superpixel dataset is constructed by segmenting training frames into superpixels, and each superpixel is represented by multiple features. The appearance information of target is encoded in the superpixel database. Second, each new frame is segmented into superpixels and then its object-background confidence map is derived by comparing its superpixels with k-nearest neighbors in superpixel dataset. Taking context information into account, we utilize MRF to further improve the accuracy of confidence map. In addition, the local context information is incorporated through a feedback to refine superpixel matching. In the last stage, visual tracking is achieved via finding the best candidate by maximum a posterior estimate based on the confidence map. Experiments show that our method outperforms several state-of-the-art trackers.

Keywords: Visual tracking · Superpixel matching · Markov Random Field (MRF) · Local context information

1 Introduction

In computer vision field, object tracking plays a crucial role for its various applications, such as surveillance and robotics [14]. To develop a robust tracker, numerous algorithms have been proposed. Despite reasonable good results of these methods, visual tracking remains a challenge due to appearance variations caused by occlusion and deformation. To address these problems, a wide range of appearance models have been presented. In general, these appearance models can be categorized into two types: discriminative models [2,3,6,9,10,13,18,20] and generative models [1,5,7,8,15,16].

Discriminative algorithms focus on building online classifiers to distinguish the target from the background. These methods employ both the foreground and background information. In [2], an adaptive ensemble of classifier is trained to separate target pixels from background pixels. Kalal *et al.* [13] introduce a P-N learning algorithm for object tracking. However, this tracking method easily

E. Chen et al. (Eds.): PCM 2016, Part I, LNCS 9916, pp. 1–10, 2016.
DOI: 10.1007/978-3-319-48890-5_1

causes drift when object appearance varies. Babenko *et al.* [3] utilize the multiple instance learning (MIL) method for visual tracking, which can alleviate drift to some extent. Yang *et al.* [18] suggest a discriminative appearance model based on superpixels, which facilitates the tracking algorithm to distinguish the target from the background.

On the other hand, the generative models formulate tracking problem as searching for regions most similar to object. These methods are based on either subspace models or templates and update appearance model dynamically. In [16], the incremental visual tracking method suggests an online approach for efficiently learning and updating a low dimensional principal components analysis (PCA) subspace representation for the object. However, this representation scheme is sensitive to occlusion. Adam *et al.* [1] present a fragment-based template model for visual tracking. Mei and Ling [15] model the object appearance with sparse representation for visual tracking and achieve a good performance.

Though having achieved promising performance for object tracking, the aforementioned algorithms often suffer from drifting problems when substantial non-rigid and articulated motions are involved in the object.

To solve the problem of tracking non-rigid and/or articulated objects, we propose a novel tracking algorithm with local superpixel matching and markov random field (MRF). Our method mainly contains three stages. In the first stage, we construct a superpixel dataset by segmenting training frames into superpixels, and each superpixel in the dataset is represented with multiple features. Through this way, the appearance information of the object is encoded in the superpixel dataset. In the second stage, for each new frame, we represent it with its superpixels. We can compute its object-background confidence map by comparing its superpixels with their k-nearest neighbors in the superpixel dataset. In this process, the tracking task is treated as separating object pixels from background pixels. Taking the context information into consideration, we utilize MRF to further improve the accuracy of the confidence map. In addition, the local context information of each superpixel is incorporated through a feedback to refine superpixel matching. In the final stage, object tracking is achieved via searching the best candidate by maximum a posterior estimate based on the confidence map. When tracking is completed in each frame, we collect good tracking results to update the superpixel dataset. With the help of this update scheme, our tracker is able to adapt to the appearance changes of the target. Figure 1 illustrates the framework of the proposed method.

2 The Proposed Tracking Algorithm

2.1 Superpixel Dataset Construction

To build the superpixel dataset, we oversegment Q training frames to generate N superpixels by algorithm in [11]. For each superpixel s_i $(1 \leq i \leq N)$, we extract four kinds of features including SIFT histogram, RGB histogram, location histogram and PHOG histogram. These histograms are concatenated to represent a superpixel similar as in [19].

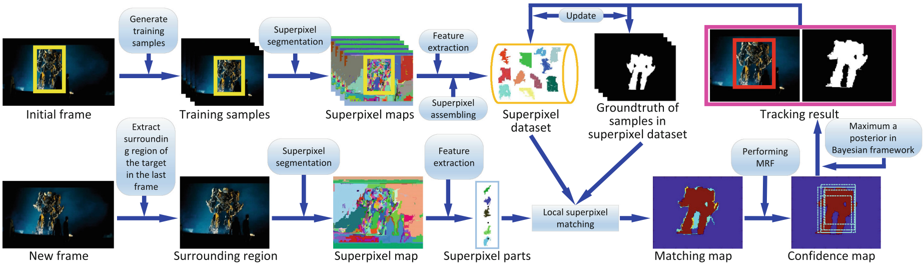


Fig. 1. Framework of the proposed method.

Let x_i denote the feature of s_i. We use y_i to represent its label, where $y_i \in \{0, 1\}$ (0 and 1 represent the background and object labels respectively) and is determined by

$$y_i = \begin{cases} 1, & a_i \geqslant 95\,\% \\ 0, & otherwise \end{cases} \tag{1}$$

where a_i represents target area ratio of s_i. We then collect all the training superpixels into a database and obtain the superpixel database $D = \{s_i, x_i, y_i\}_{i=1}^{N}$.

2.2 Object-Background Confidence Map

For each new frame, we firstly extract the surrounding region[1] of the target in the last and then segment this region into superpixels with the same method in [11]. Let M be the number of its superpixels. For the i^{th} superpixel s_j $(1 \leq j \leq M)$, we are able to calculate its label cost by comparing its k-nearest neighbor $\mathcal{N}_k(j)$ in superpixel dataset D as follows

$$U(y_j = c|s_j) = 1 - \frac{\sum_{i \in \mathcal{N}_k(j), y_i = c} \mathcal{K}(x_j, x_i)}{\sum_{i \in \mathcal{N}_k(j)} \mathcal{K}(x_j, x_i)} \tag{2}$$

where x_j denotes the feature of s_j, $c \in \{0, 1\}$ represents the label and $\mathcal{K}(x_j, x_i)$ is the intersection kernel between features x_j and x_i.

In this work, tracking is treated as separating object pixels from background pixels. In order to exploit the context relationship of object pixels and background pixels, we utilize MRF inference for contextual constraints. The energy function is given by

$$E(Y) = \sum_{p} U(y_p = c) + \lambda \sum_{pq} V(y_p = c, y_q = c') \tag{3}$$

[1] The surrounding region is a square area centered at the location of target X_t^c, and its side length is equal to $\lambda_s [X_s^t]^{\frac{1}{2}}$, where X_t^c represents the center location of target region X_t and X_t^s denotes its size. The parameter λ_s is a constant variable, which determines the size of this surrounding region.

where p, q are pixel indices, c, c' are candidate labels and λ is the weight of pairwise energy. The unary energy of one pixel is given by the superpixel it belongs to

$$U(y_p = c) = U(y_j = c|s_j),\ p \in s_j \tag{4}$$

The pairwise energy on edges is given by spatially variant label cost

$$V(y_p = c, y_q = c') = d(p, q) \cdot \mu(c, c') \tag{5}$$

where $d(p, q) = exp(-\|I(p) - I(q)\|^2/2\sigma^2)$ is the color dissimilarity between two adjacent pixels, and $\mu(c, c')$ is the penalty of assigning label c and c' to two adjacent pixels and defined by log-likelihood of label co-occurrence statistics

$$\mu(c, c') = -log[(P(c|c') + P(c'|c))/2] \times \sigma \tag{6}$$

Through this way, we can derive the labels of all pixels by performing MAP interference on $E(Y)$ with graph cut optimization in [4].

Taking into local context information of each superpixel into account, we adopt a simple yet effective feedback mechanism as in [19]. In the feedback process, we can obtain the pixel-wise classification likelihood of each pixel by

$$\ell(p, c) = \frac{1}{1 + exp(U(y_p = c))} \tag{7}$$

where $U(y_p = c)$ is the cost of assigning label c to pixel p in Eq. (4).

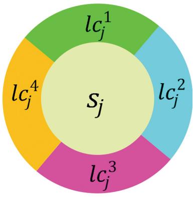


Fig. 2. Local context descriptor of superpixel.

For robust superpixel matching, we exploit the local context of each superpixel. For superpixel s_j, we divide its neighborhood into left, right, top, bottom four cells $\{lc_j^1, lc_j^2, lc_j^3, lc_j^4\}$ (see Fig. 2). For each cell lc_j^k $(1 \leq k \leq 4)$, we compute its sparse context $h_j^k = [h_{j1}^k, h_{j2}^k]$ by

$$h_{ic}^k = \max_{p \in lc_i^k} \ell(p, c) \tag{8}$$

where $\ell(p, c)$ represents the pixel-wise classification likelihood obtained by Eq. (7). For superpixel s_i, we can obtain spatial context descriptor $h_j = [h_j^1; h_j^2; h_j^3; h_j^4]$. Thus we can classify the superpixels of the new frame by Eq. (2) with new feature $[x_j; h_j]$.

Through the above process, we are able to obtain the matching score $Score(j)$ for superpixel s_j by

$$Score(j) = \frac{U(y_j = 1|s_j) - U(y_j = 0|s_j)}{U(y_j = 1|s_j) + U(y_j = 0|s_j)} \tag{9}$$

and the confidence map C for each pixel on the entire current frame as follows. We assign every pixel whose label is object with 1, and every pixel whose label is background or outside the surrounding region with -1. Figure 3 shows the matching maps, confidence maps and tracking results of the target in video Iceskater by our method.

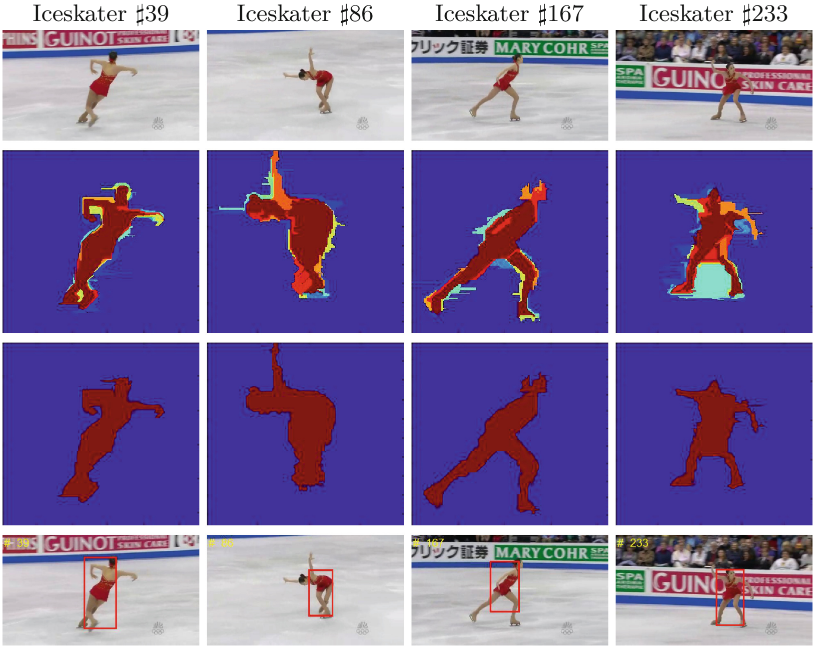


Fig. 3. Matching maps, confidence maps and tracking results on video Iceskater. First row: original images. Second row: matching maps of corresponding regions obtained by our local superpixel matching. Third row: confidence maps of corresponding regions derived by performing MRF on matching maps. Fourth row: the final tracking results of each frame (see details in Sect. 2.3).

2.3 Tracking Formulation

Our tracker is implemented within Bayesian framework. Given the observation set of targets $Z^t = \{z_1, z_2, \cdots, z_t\}$ up to the frame t, where z_τ $(\tau = 1, 2, \cdots, t)$

represents the observation of target in frame τ, we can obtain estimation $\widehat{X}_t$ by computing the maximum a posterior via

$$\widehat{X}_t = \underset{X_t^i}{\text{argmax}} \ \text{p}(X_t^i|Y^t) \tag{10}$$

where $\widehat{X}_t$ denotes the i^{th} sample at the state of X_t. The posterior probability $\text{p}(X_t^i|Z^t)$ can be obtained by the Bayesian theorem recursively via

$$\text{p}(X_t|Y^t) \propto \text{p}(z_t|X_t) \int \text{p}(X_t|X_{t-1})\text{p}(X_{t-1}|Z^{t-1})dX_{t-1} \tag{11}$$

where $\text{p}(X_t|X_{t-1})$ and $\text{p}(z_t|X_t)$ represent the dynamic model and observation model respectively.

The dynamic model indicates the temporal correlation of the target state between consecutive frames. We apply affine transformation to model the target motion between two consecutive frames within the particle filter framework. The state transition can be formulated as

$$\text{p}(X_t|X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \varPsi) \tag{12}$$

where $\varPsi$ is a diagonal covariance matrix whose elements are the variance of affine parameters. The observation model $p(z_t|X_t)$ represents the probability of the observation z_t at state X_t. In this paper, the observation for i^{th} sample at the state of X_t is designed as in [18] by

$$\text{p}(z_t|X_t^i) \propto \sum_{(w,v)\in C_t^i} v_t^i(w,v) \times [S(X_t^i)/S(X_{t-1})] \tag{13}$$

where C_t^i is the confidence map of the i^{th} candidate warped from confidence map of corresponding region, $v_t^i(w,v)$ denotes the confidence value of pixels at location (w,v), $S(X_t^i)$ represents the area size of the i^{th} candidate and $S(X_{t-1})$ is the area size of the object in last frame. Through Bayesian inference, we can determine the candidate sample with the maximum observation as the tracking result.

2.4 Online Update

Before tracking, the target in the initial frame is manually labeled. The Q training samples utilized for constructing database are the same with the first frame and stored in a set T with fixed length L. Our strategy is to choose the latest good tracking result, add it into set T and remove the oldest element in T if T is full. In this way, when superpixel matching starts, the superpixel dataset can be effectively updated to adapt to the object appearance changes by the new set T.

Every H frames, we compute the occlusion coefficient O_t of the latest tracking result X_t by

$$O_t = 1 - \frac{\sum_{(w,v)\in C_t} v_t(w,v)/S(X_t)}{\sum_{(w,v)\in C_{t-1}} v_{t-1}(w,v)S(X_{t-1})} \tag{14}$$

When heavy occlusion happens, the occlusion coefficient O_t will be large, and thus it is unnecessary to add the tracking result into T. We set a threshold θ to determine whether the tracking result is added into T. If $O_t > \theta$, we skip this frame to avoid introducing noise into T. Otherwise, we add the tracking result into T and remove the oldest element from T if the number of elements in T is larger than L.

3 Experiments

We evaluate our tracker on eight challenging image sequences and compare it with seven state-of-the-art tracking methods. These algorithms are SPT tracking [18], CT tracking [20], SCM tracking [23], STC tracking [21], ASLA tracking [12], PCOM tracking [17], MTT tracking [22]. The proposed algorithm is implemented

Table 1. Average center location error (CLE) in pixel. The best and the second best results are shown in **red** and **blue** fonts.

Sequence	ASLA	CT	MTT	PCOM	SCM	STC	SPT	Ours
Bikeshow	182.7	82.2	190.0	135.1	193.2	148.4	**22.3**	**6.8**
David3	104.5	89.8	307.0	100.0	67.6	**6.3**	8.3	**8.1**
Gymnastics	**15.9**	24.7	100.0	102.1	16.5	17.7	25.3	**11.9**
Iceskater	21.6	45.3	86.9	125.6	144.7	130.1	**17.6**	**12.7**
Mottocross2	**4.5**	18.8	32.5	86.8	10.9	**4.6**	19.2	10.0
Skater	9.3	14.3	**7.6**	88.5	12.6	15.5	14.3	**7.3**
Skater2	28.4	41.0	76.3	107.6	88.5	28.4	**24.2**	**9.1**
Transformer	43.1	52.8	42.9	104.6	37.5	44.2	**15.6**	**10.1**
Average	51.3	46.1	105.4	106.3	71.4	49.4	**18.4**	**9.5**

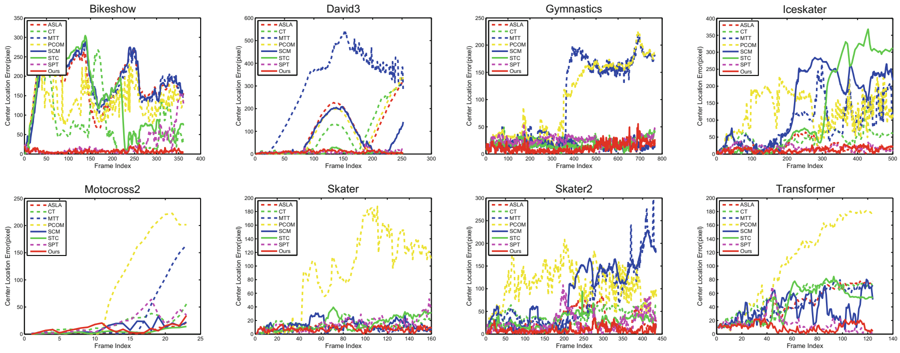


Fig. 4. Quantitative evaluation in terms of center location error in pixel. The proposed method is compared with seven state-of-the-art algorithms on eight challenging test sequences.

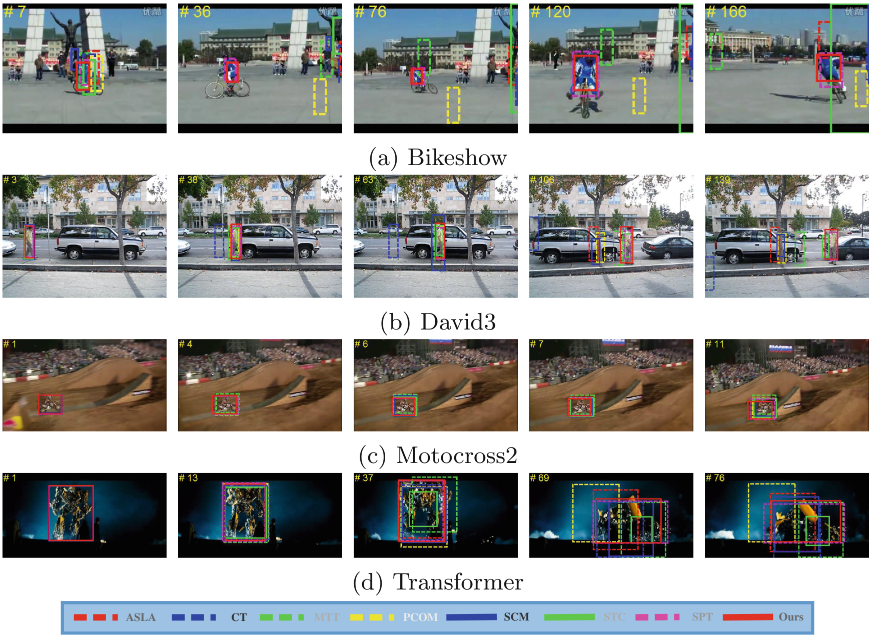
in MATLAB and runs at 1.5 frames per second on a 3.2 GHz Intel E3-1225 v3 Core PC with 8 GB memory. The parameters of the proposed tracker are fixed in all experiments. The number of neighbors k in Eq. (2) is set to 7. The number of particles in Bayesian framework is 300 to 600. The λ_s is set to 1.5. The number of initial training samples is 5. The length L of set T is fixed to 10, and H is set to 5. The threshold θ is 0.8.

3.1 Quantitative Comparison

We evaluate the above mentioned trackers by center location error (CLE) in pixels, and the comparing results are shown in Table 1. Figure 4 shows the center location error of utilized trackers on eight test sequences. Overall, the proposed tracker outperforms other state-of-the-art algorithms.

3.2 Qualitative Comparison

Deformation: Deformation is a disaster for a tracker because it is able to cause heavy appearance variations. Figure 5(a) and (d) demonstrate the tracking results in the presence of deformation. The proposed tracker is able to robustly locate the non-rigid object in these sequences we represent the object appearance with a robust superpixel database. With the help of update scheme, the



(a) Bikeshow

(b) David3

(c) Motocross2

(d) Transformer

Fig. 5. Screenshots of some sample tracking results.

superpixel database can be updated to adapt to object appearance changes, and thus our tracker is robust to deformation.

Occlusion: Occlusion is a common problem in visual tracking. Figure 5(b) shows the performance of our tracker in the presence of occlusion. When occlusion happens, object appearance will change because part of target is occluded. However, our tracker is still able to locate the object because our tracker can utilize the unoccluded part of the target for tracking with local superpixel matching.

Rotation: Figure 5(c) shows sampled experimental results of target with drastic rotation. In these sequence, the object suffers from not only rotation but also scale variation. Our methods demonstrates good performance to track the target owing to our appearance model. When rotation happens, the structure of object appearance will change. Nevertheless, our superpixel database can ignore this structure changes and distinguish the object superpixels from background superpixels via our matching method.

4 Conclusion

In this paper, we propose a novel method for object tracking, especially for the targets involved with non-rigid and articulated motions. This approach mainly consists of three stages. In the first stage, a superpixel database is constructed to represent the appearance of object. In the second stage, when a new frame arrives, it is firstly segmented into superpixels. Then we compute its confidence via superpixel matching and MRF. Taking context information into account, we utilize MRF to further improve the accuracy of confidence map. In addition, the local context information is incorporated through a feedback to refine superpixel matching. In the last stage, visual tracking is achieved through finding the best candidate by maximum a posterior estimate based on the confidence map. Experiments evidence the effectiveness of our method.

Acknowledgement. This work was primarily supported by Foundation Research Funds for the Central Universities (Program No. 2662016PY008 and Program No. 2662014PY052).

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments based tracking using the integral histogram. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 798–805 (2006)
2. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **29**(2), 261–271 (2007)
3. Babenko, B., Yang, M.-H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **33**(8), 1619–1632 (2011)
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **23**(11), 1222–1239 (2001)

5. Fan, H., Xiang, J.: Robust visual tracking with multitask joint dictionary learning. IEEE Trans. Circ. Syst. Video Technol. **PP**, 1 (2016)
6. Fan, H., Xiang, J., Zhao, L.: Robust visual tracking via bag of superpixels. Multimedia Tools Appl. **75**, 8781 (2015)
7. Fan, H., Xiang, J., Liao, H., Du, X.: Robust tracking based on local structural cell graph. J. Vis. Commun. Image Represent. **31**, 54–63 (2015)
8. Fan, H., Xiang, J., Ni, F.: Multilayer feature combination for visual tracking. In: Asian Conference on Pattern Recognition (ACPR), pp. 589–593 (2015)
9. Fan, H., Xiang, J.: Patch-based visual tracking with two-stage multiple Kernel learning. In: Zhang, Y.-J. (ed.) ICIG 2015. LNCS, vol. 9219, pp. 20–33. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21969-1_3
10. Fan, H., Xiang, J., Xu, J., Liao, H.: Part-based visual tracking via online weighted P-N learning. Sci. World J. (2014)
11. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph based image segmentation. Int. J. Comput. Vis. (IJCV) **59**(2), 167–181 (2004)
12. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829 (2012)
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **34**(7), 1409–1422 (2012)
14. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans. Image Process. (TIP) **24**(12), 5630–5644 (2015)
15. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **33**(11), 2259–2272 (2011)
16. Ross, D.A., Lim, J., Lin, R.S., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. (IJCV) **77**(1), 125–141 (2008)
17. Wang, D., Lu, H.: Visual tracking via probability continuous outlier model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3478–3485 (2014)
18. Yang, F., Lu, H., Yang, M.-H.: Robust superpixel tracking. IEEE Trans. Image Process. (TIP) **23**(4), 1639–1651 (2014)
19. Yang, J., Price, B., Cohen, S., Yang, M.-H.: Context driven scene parsing with attention to rare classes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3294–3301 (2014)
20. Zhang, K., Zhang, L., Yang, M.-H.: Fast compressive tracking. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **36**(10), 2002–2015 (2014)
21. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 127–141. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10602-1_9
22. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via structured multi-task sparse learning. Int. J. Comput. Vis. (IJCV) **101**, 367–383 (2013)
23. Zhong, W., Lu, H., Yang, M.-H.: Robust object tracking via sparsity-based collaborative model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1838–1845 (2012)