VISUAL TRACKING OF GROUND AND AIR TARGETS

Roman Zakharov
Samara State Aerospace University
Ph.D. student
34, Moskovskoe shosse, Samara, 443086, Russian Federation
E-mail: roman.zakharov@yandex.ru
Salimzhan Gafurov (Samara State Aerospace University, Lappeenranta University of Technology, Lappeenranta, Finland), Vera Volkova (Samara State Aerospace University)

ABSTRACT

We consider the problem of visual tracking. In recent years, many approaches have been proposed in the field of visual tracking of different objects. In this paper presented the method of visual tracking of various objects. The method is used to track ground and air targets. Extensive experiments demonstrate that the proposed tracking framework outperforms the state-of-the-art methods in challenging scenarios, especially when the illumination changes dramatically.

1 INTRODUCTION

Object tracking is one of the most important component in a wide range of machine vision applications, such as building surveillance systems for unmanned systems, human computer interaction for control unmanned vehicles, tracking and object recognition, tracking and landing runways, fire detection, object tracking enemy.

There are a lot of detection systems developed for unmanned aerial vehicles (UAV) and they are as a rule based on various sensors. Technologies of visual control are believed to be the most promising because of video cameras low cost, compact size and easiness of replacement in case of breakdown.

One the main function of UAV is objects tracking. Visual tracking is one of the most active areas of research in computer vision. Despite the considerable progress in machine vision has been made in recent years, the answer for a question about the most effective method for object tracking still remains unknown.

In this paper the tracking method is proposed. Some theoretical results of its implementation are shown here. The theoretical investigations were carried out in online mode by means of developed scheme model updates. The method does not allow system to retrain tracking by another object. It improves the accuracy of proposed method even in poor lighting.

2 RELATED WORK

In this section, we review recent algorithms for object tracking in terms of several main modules: target representation scheme, search mechanism, and model update.

Representation scheme. Object representation is one of the major components in every visual tracker. Since the pioneering work of Lucas and Kanade [16, 17], holistic templates (raw intensity values) have being widely used for tracking [18, 19, 20].

Furthermore, Mei and Ling [23] proposed a tracking approach based on sparse representation to handle the corrupted appearance and recently it has been further improved [21, 22, 24, 25]. In addition to holistic template, many other visual features have been adopted in tracking algorithms such as color histograms [26], histograms of oriented gradients (HOG) [9, 10], covariance region descriptor [27, 28] and Haar-like features [12]. Recently the discriminative model has been widely adopted in tracking [29].

Numerous learning methods have been adapted to the tracking problem, such as SVM [30], structured output SVM [32], ranking SVM [31], boosting [12], semiboosting [33] and multi-instance boosting [34]. An object can be represented by parts where each part is represented by descriptors or histograms to make trackers more robust to pose variation and partial occlusion.

Search Mechanism. Deterministic or stochastic methods have been used to estimate the state of the target objects. Here the tracking problem is posed within an optimization framework, assuming the objective function is differentiable with respect to the motion parameters. Gradient descent methods can be used to locate the target efficiently [35, 36, 37, 38].

Stochastic search algorithms such as particle filters [39, 40] have being widely used as they are relatively insensitive to local minima and computationally efficient [41, 42, 43].

Model Update. It is crucial to update the target representation. Effective update algorithms have also been proposed via online mixture model [44], online boosting [12], and incremental subspace update [43]. For discriminative models, the main issue has been improving the sample collection part to make the online-trained classifier more robust [33, 34, 7, 32].

2. DESCRIPTION OF THE PROPOSED TRACKING METHOD

The proposed visual tracking method consists of three main stages (**figure 1**):

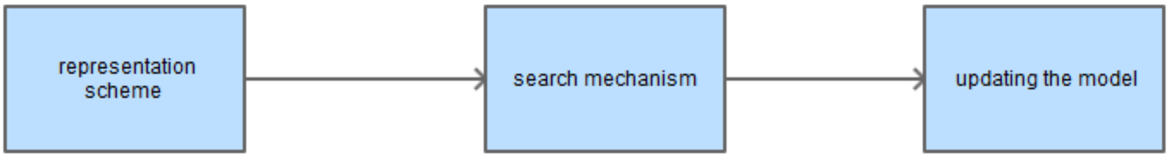

Figure 1. The main stages of the visual tracking. For each new frame all steps are performed.

2.1 Representation Scheme

In out method the histogram of local sensitivity (LSH) [5] was used for the objects representing. The conventional image histogram is a 1D array. Each of its values is usually an integer indicating the frequency of occurrence of a particular intensity value. Let matrix I denotes an image. The corresponding image histogram H is a B-dimensional vector defined as:

$$H(b) = \sum_{q=1}^{W} Q(I_q, b), b = 1 \ldots B \qquad (1)$$

,

where W - the number of pixels,
B - the total number of bins,
$Q(I_q,b)$ – pixel location. This term is equal to zero except when intensity value I_q (at pixel location q) belongs to bin b.

Equation (1) gives the linear computational complexity in the number of bins at each pixel location – $O(b)$. As a matter of fact, in practice the computational complexity can be reduced to $O(1)$ because the addition operation in Eq. 1 can be ignored when $Q(Iq,b) = 0$.

The computational complexity of the brute-force implementation of the local histograms is linear in space of the nearest neighborhoods. Nevertheless, this dependence can be changed by using integral histogram, which reduces the computational complexity to $O(B)$ at each pixel location.

Let H_p^I denotes the integral histogram computed at pixel p. It can be calculated based on the previous integral histogram computed in its turn at pixel p-1 in a way similar to the integral image:

$$H_p^I(b) = Q(I_p, b) + H_{p-1}^I(b),\ b=1..B \qquad (2)$$

For simplicity, let I denotes a 1D image. H_p^I contains all the pixels contributions to the left side of pixel p. Then the local histogram between pixel p and another pixel q on the left of p is computed as

$$H_p^I(b) - H_q^I(b)\ \text{for}\ b = 1 \ldots B \qquad (3)$$

For the local histogram, pixels inside a local neighborhood have equal contribution. As far as pixels offsetted from the target center, they should have less weight due to they are believed to content the background information or not relevant objects. As a result, their contribution to the histogram should be minimized. We used a local sensitive histogram algorithm to avoid this problem.

Let H_p^E denotes the local sensitive histogram computed at pixel p. It can be written as:

$$H_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|} * Q(I_q, b), b = 1 \dots B, \tag{4}$$

where $\alpha \in$ *(0; 1)* - a parameter regulating the weight reduction of the pixel due to increasing the distance between it and the center of the target.

2.2 Search Mechanism

We used two following methods for search mechanism organization:

- location of the object based on Part Based Detector (PBD) [6]. The objects localization based on the PBD is required for initializing objects tracking, when the tracking objects are lost in search area to improve the tracking stability;
- finding the offset of the object between two cades in the search area. Search object offsetting is based on the knowledge of the object location in the previous time frame.

A core component of the PBD model is templates or filters captured the appearance of object parts based on local image features. Filters define scores for placing parts at different image positions and scales. These scores are combined using a deformation model that scores the arrangements of parts based on geometric relationships (figure 3). Models are built from linear filters that are applied to dense feature maps. A linear filter is defined by a *w × h* array of *d*-dimensional weight vector. Intuitively, a filter is a template that is tuned to respond to an iconic arrangement of image features. Filters are typically much smaller than feature maps and can be applied at different locations within a feature map.

A dense feature map is an array whose entries are d-dimensional feature vectors computed on a dense grid of image locations (e.g., every 8 × 8 pixels). Each feature vector describes a small image path and in such a manner results some invariants.

The framework of this search mechanism is independent of the specific choice of features. In fact, we use a low-dimensional variation of the histogram of oriented gradient (HOG) [9]. HOG features introduce invariances to photometric transformations and small image deformations.

A filter is a rectangular template defined by an array of *d*-dimensional weight vectors. The response, or score, of a filter *F* at a position *(x; y)* in a feature map *G* is the "dot product" of the filter and a subwindow of the feature map with top-left corner at *(x; y)*:

$$\sum_{x'y'} F[x',y'] * G[x+x',y+y'] \tag{5}$$

We define a score at different positions and scales in an image. This is done using a feature pyramid which specifies a feature map for a finite number of scales in a fixed range. In practice, we compute feature pyramids by computing a standard image pyramid via repeated smoothing and subsampling, and then computing a feature map from each level of the image pyramid. **Figure 2** illustrates the construction.

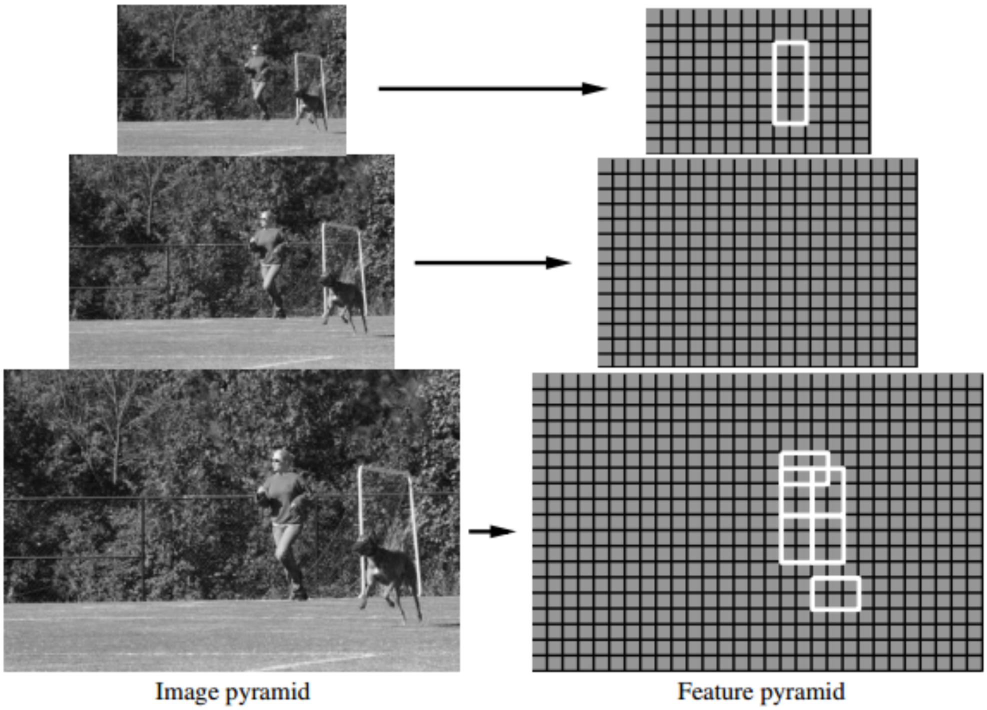

Figure 2. A feature pyramid and an instantiation of a person model within that pyramid. The part filters are placed at twice the spatial resolution of the placement of the root.

The scale sampling in a feature pyramid is determined by a parameter *λ* defining the number of levels in an octave. That is the number *λ* of levels we need to go down in the pyramid to get a feature map which resolution is computed as twice resolution of another level. In practice, we have used *λ*= 5 in training and *λ* = 10 at test time. Fine sampling of scale space is important for obtaining high performance of models.

Let *F* is a *w* ×*h* filter. Let *H* is a feature pyramid and *p = (x, y, l)* specify a position *(x, y)* on the lth level of the pyramid. Let *(H, p, w, h)* denote the vector obtained by concatenating the feature vectors in the *w*×*h* subwindow of *H* with top-left corner at *p* in row-major order.

The score of *F* at *p* is *F′* ×*(H, p, w, h)*, where *F′* is the vector obtained by concatenating the weight vectors in *F* in row-major order.

The estimation of a filter *F* at a particular feature map location is obtained by taking the dot product of *F′s* array of weight vectors. The product of this operation is, concatenated into a single long vector associated with the feature vectors extracted from a *w* × *h* window of the feature map. We apply the same filter to multiple feature maps, each computed from a rescaled version of the original image because objects can appear at a wide range of scales.

Figure 3 shows some examples of filters, feature maps, and filter responses.

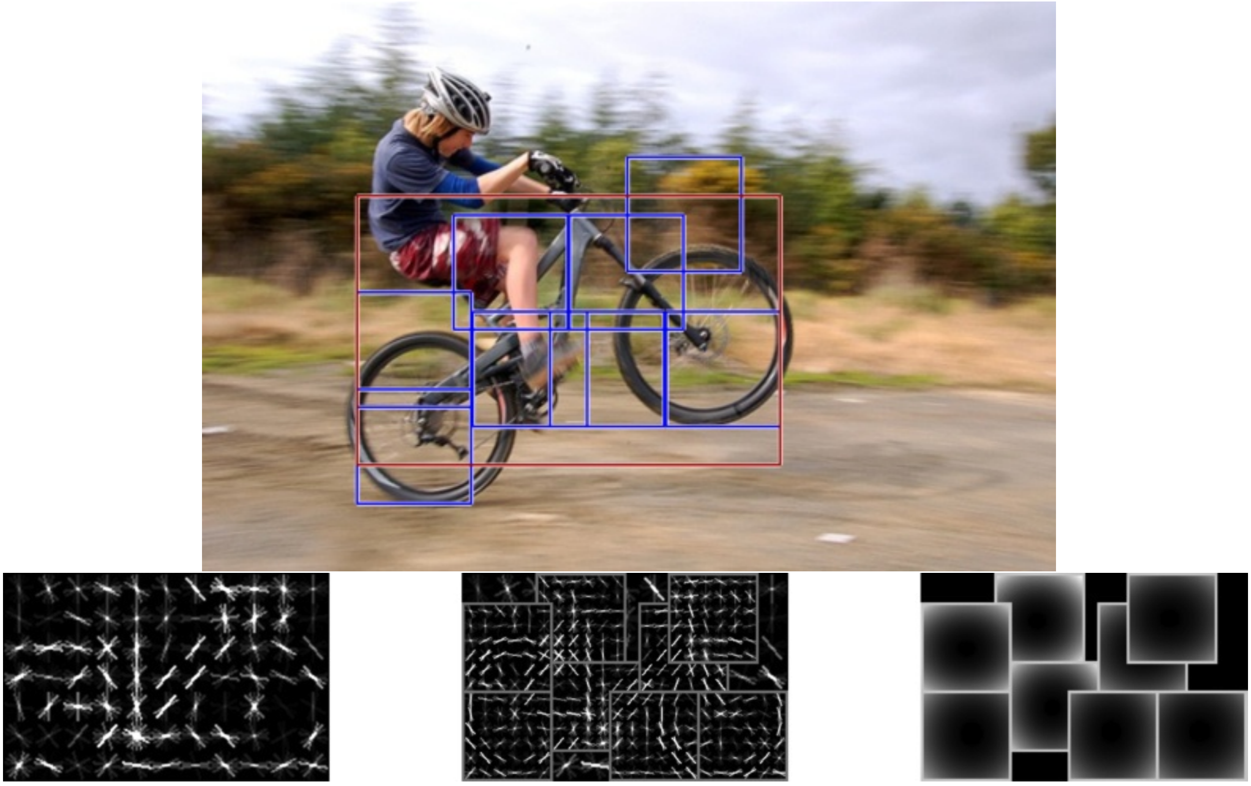

Figure 3. The model is defined by a coarse root filter (a), several higher resolution part filters (b), and a spatial model for the location of each part relative to the root (c). The filters specify weights for histogram of oriented gradients features. Their visualization shows the positive weights at different orientations. The visualization of the spatial models reflects the "cost" of placing the center of a part at different locations relative to the root.

3 MODEL UPDATING

The object is represented by a local sensitivities histogram. Tracking features may vary in time that leads to significantly deterioration of objects visual tracking. For this reason a method of information updating of the local sensitivity histogram is proposed (**Figure 4**) to improve the reliability of the proposed tracking method:

1. Find all the various regions of the histogram between the template object and the histogram of the object at the current frame.
2. If the number of differ regions is smaller T1 (lower threshold is chosen experimentally), then go to item 7.
3. If the number of differ regions is more T2 (upper threshold is chosen experimentally), then the object is considered lost and run PBD.
4. If the number of differ regions is over T1 and less T2 then run the update method of the histogram information
5. Nearest clusters computed from the training sample in the histogram on the current frame to update the histogram. Random forest model is selected as a classifier model.
6. Updating the template histogram is performed using regions derived from a histogram of the current frame and the regions selected from the nearest cluster on the previous step.
7. Get a new frame.

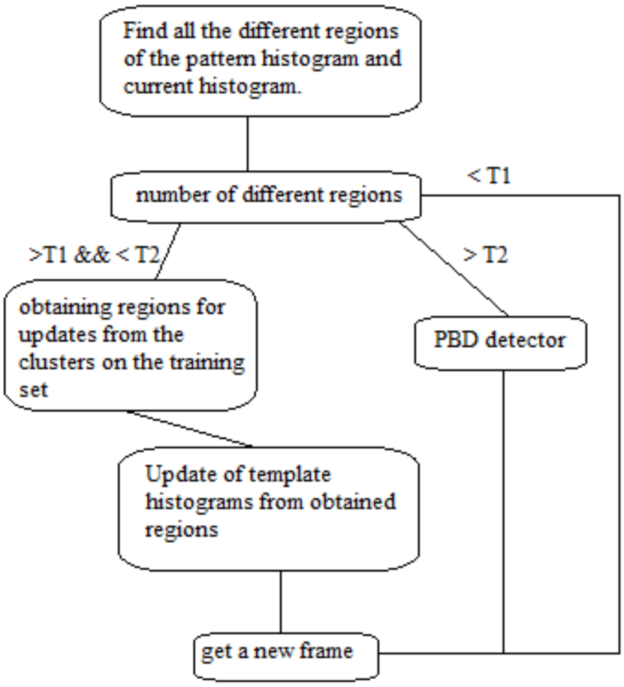
Find all the different regions of the pattern histogram and current histogram.
number of different regions
< T1
>T1 && < T2
> T2
obtaining regions for updates from the clusters on the training set
PBD detector
Update of template histograms from obtained regions
get a new frame

Figure 4. Shows a method of updating the model

Due to the training sample clustering proposed tracker cannot retrained and track an object of another class. Crossing paths problems is known to appear in multiple. Our method uses an optimization algorithm (Hungarian algorithm) solving the assignments problem to solve this problem. This algorithm makes it possible to separate the two close trajectories.

4 EXPERIMENTS

This section evaluates the effectiveness of the proposed tracking method. We have compared the proposed tracker with 7 state-of-the-art trackers (the implementations provided by the authors were used for fair comparisons). The following methods were used tracking, the real time L1 tracker (L1T) [46], the real-time compressive tracker (CT) [47], the multiple instance learning tracker (MIL) [48], the structured output tracker (Struck) [33], the visual tracking decomposition method (VTD) [49], the TLD tracker [1] and the multi-task sparse learning tracker (MTT) [25].

To evaluate methods of tracking in our experiments the following criteria: tracking success rate, computed manually labeled ground truth.

Let term $S = \frac{\mathbf{area(B_T \cap B_G)}}{\mathbf{area(B_T \cap B_G)}}$ denotes the overlap ratio, where B_T and B_G are the bounding boxes of the tracker and of the ground-truth, respectively. When the overlap ratio is larger than 0.5, the tracking result of the current frame is considered as a success.

We used 15 standard video sequences to compare the proposed tracker with another wel-known trackers (**Table 1**). These sequences were: Biker, Car, David indoor, Man, Motor rolling, Women, Basketball, Box, Occluded face 2, Surfer, Board, Bird, Coupon, Crowds, Trellis.

Table 1 is presented the results of proposed method testing in comparison with 7 state-of-the-art trackers. The first column is the name of a video sequence, the last line - is the averaged value for the tracker as a whole.

Table 1 - The comparison results

Sequence	*S* (%)							
	Our	TLD	L1T	CT	MIL	Struck	VTD	MTT
Biker	**59**	38	23	34	40	49	45	44
Car	**85**	58	43	43	38	59	44	49
David indoor	**92**	90	41	46	24	67	32	**92**
Man	**99**	98	98	60	21	**99**	31	**99**
Motor rolling	**75**	14	5	11	9	11	6	5
Women	**77**	30	8	6	6	87	5	8
Basketball	**85**	1	75	32	27	2	**96**	3
Box	**83**	60	4	33	18	**90**	34	25
Occluded face 2	**99**	76	60	**100**	94	79	77	82
Surfer	**75**	**86**	1	3	2	67	2	3
Board	**93**	16	3	73	76	71	13	63
Bird	**95**	12	44	53	58	48	81	13
Coupon	**99**	98	24	58	77	99	38	**100**
Crowds	**75**	16	59	9	4	82	8	9
Trellis	**90**	31	67	35	34	70	54	34
Average	**85.4**	48.2	37	39.7	35.2	65.3	37.7	41.9

Figure 5 presents a work of the proposed tracker.

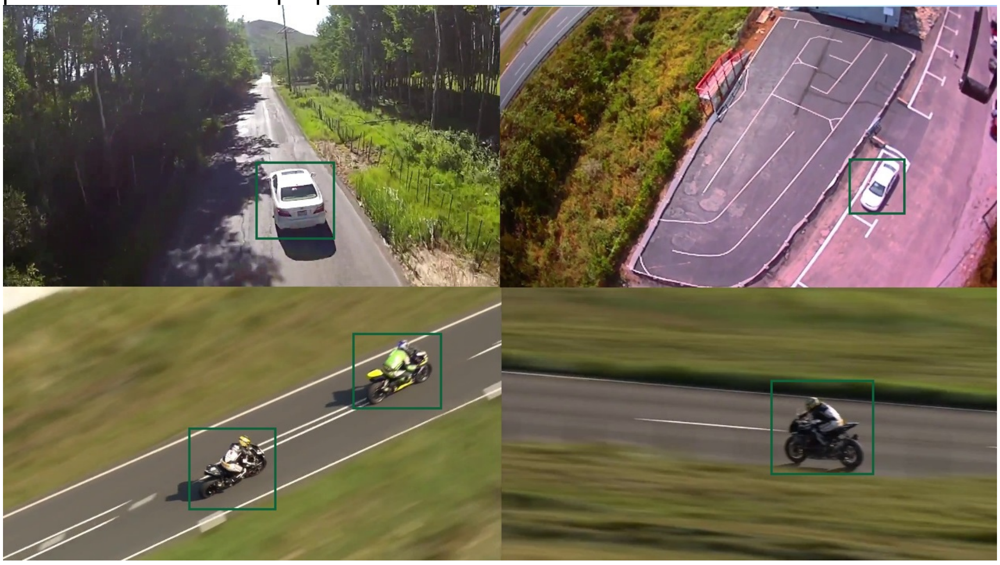

Figure 5. The working process of the proposed tracker. Testing was performed based on PASCAL VOC-2007 images databases

5 CONCLUSION

In this paper, we propose an effective method tracking. Experimental results show good results of the proposed method in accuracy and robust of the tracking in comparison to state-of-the-art methods. This method can be used for tracking various objects. Moreover, the proposed method shows very good results of tracking in case with poor visibility. Our method has shown the best result in average value during the conducted tests.

REFERENCES

1. Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. PAMI, 34:1409–1422, 2012.
2. L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In CVPR, pages 1910–1917, 2012.
3. T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In CVPR, pages 2042 –2049, 2012.

4. VOC-2007 URL: http:// pascallin.ecs.soton.ac.uk/challenges/VOC/
5. Shengfeng He, Qingxiong Yang, Rynson W.H. Lau, Jiang Wang, Ming-Hsuan Yang. Visual Tracking via Locality Sensitive Histograms. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013), pp. 2427-2434, Portland, June, 2013.
6. Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D. Object Detection with Discriminatively Trained Part Based Models // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. V. 32. №9. P. 1627–1645.
7. Z. Kalal, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, Forward-Backward Error: Automatic Detection of Tracking Failures," ICPR, 2010.
8. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. PAMI, 25(5):564–577, 2003.
9. N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, 2005.
10. F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-Tracking Using Semi-Supervised Support Vector Machines. CVPR, 2007.
11. P. Viola and M. J. Jones. Robust Real-Time Face Detection. IJCV, 57(2):137–154, 2004.
12. H. Grabner, M. Grabner, and H. Bischof. Real-Time Tracking via On-line Boosting. In BMVC, 2006.
13. J. Fan, Y. Wu, and S. Dai. Discriminative Spatial Attention for Robust Tracking. In ECCV, 2010.
14. B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In CVPR, 2009.
15. R. Zakharov. Multi-Target Pedestrian Tracking Algorithm. Supplementary Proceedings of the 3rd International Conference on Analysis of Images, Social Networks and Texts (AIST-SUP 2014), Yekaterinburg, Russia, April 10-12, 2014.
16. B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with An Application to Stereo Vision. In IJCAI, 1981.
17. S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. IJCV, 56(3):221–255, 2004.
18. N. Alt, S. Hinterstoisser, and N. Navab. Rapid Selection of Reliable Templates for Visual Tracking. In CVPR, 2010.
19. G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. PAMI, 20(10):1025–1039, 1998.
20. I. Matthews, T. Ishikawa, and S. Baker. The Template Update Problem. PAMI, 26(6):810–815, 2004.
21.
22. X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection. In CVPR, 2011.
23. Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred Target Tracking by Blur-driven Tracker. In ICCV, 2011.
24. X. Mei and H. Ling. Robust Visual Tracking using L1 Minimization. In ICCV, 2009.
25. T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust Visual Tracking via Multi-task Sparse Learning. In CVPR, 2012.
26. D. Wang, H. Lu, and M.-H. Yang. Online Object Tracking with Sparse Prototypes. TIP, 22(1):314–325, 2013.
27. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. PAMI, 25(5):564–577, 2003.
28. O. Tuzel, F. Porikli, and P. Meer. Region Covariance: A Fast Descriptor for Detection and Classification. In ECCV, 2006.
29. Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai. Real-time Probabilistic Covariance Tracking with Efficient Model Update. TIP, 21(5):2824–2837, 2012.
30. R. T. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. PAMI, 27(10):1631–1643, 2005.
31. S. Avidan. Support Vector Tracking. PAMI, 26(8):1064–1072, 2004.
32. Y. Bai and M. Tang. Robust Tracking via Weakly Supervised Ranking SVM. In CVPR, 2012.
33. S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured Output Tracking with Kernels. In ICCV, 2011.
34. H. Grabner, C. Leistner, and H. Bischof. Semi-supervised On-Line Boosting for Robust Tracking. In ECCV, 2008.
35. B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In CVPR, 2009.
36. B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with An Application to Stereo Vision. In IJCAI, 1981.
37. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking PAMI, 25(5):564–577, 2003.

38. J. Fan, Y. Wu, and S. Dai. Discriminative Spatial Attention for Robust Tracking. In ECCV, 2010.
39. L. Sevilla-Lara and E. Learned-Miller. Distribution Fields for Tracking. In CVPR, 2012
40. P. P´erez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In ECCV, 2002.
41. M. Isard and A. Blake. CONDENSATION–Conditional Density Propagation for Visual Tracking. IJCV, 29(1):5–28, 1998.
42. X. Jia, H. Lu, and M.-H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. In CVPR, 2012.
43. X. Mei and H. Ling. Robust Visual Tracking using L1 Minimization. In ICCV, 2009.
44. D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. IJCV, 77(1):125–141, 2008.
45. A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust Online Appearance Models for Visual Tracking. PAMI, 25(10):1296–1311, 2003.
46. C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In CVPR, pages 1830–1837, 2012.
47. K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In ECCV, 2012.
48. B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. PAMI, 33(8):1619 –1632, aug. 2011.
49. J. Kwon and K. M. Lee. Visual tracking decomposition. In CVPR, pages 1269–1276, 2010.