Visual Tracking via Random Partition Image Hashing

Guan, Mingyang; Wen, Changyun; Lim, Kwang-Yong; Shan, Mao; Tan, Paul; Ng, Cheng-Leong; Zou, Ying

2016

Guan, M., Wen, C., Lim, K.-Y., Shan, M., Tan, P., Ng, C.-L., et al. (2016). Visual tracking via random partition image hashing. 2016 14th International Conference on Control, Automation, Robotics & Vision (ICARCV), 1-6.

<https://hdl.handle.net/10356/82416>

<https://doi.org/10.1109/ICARCV.2016.7838673>

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://doi.org/10.1109/ICARCV.2016.7838673>].

Downloaded on 26 Apr 2023 23:37:24 SGT

Visual Tracking via Random Partition Image Hashing

Mingyang Guan, Changyun Wen, Kwang-Yong Lim, Mao Shan, Paul Tan, Cheng-Leong Ng, Ying Zou

ST Engineering-NTU Corporate Laboratory

School of Electrical and Electronic Engineering

Nanyang Technological University, Singapore 639798

Email: {myguan, ecywen}@ntu.edu.sg, limkwangyong@stengg.com

shanmao@ntu.edu.sg, {paultan, ngclg}@stengg.com, zouy0011@e.ntu.edu.sg

Abstract—In this paper, we propose a discriminative and robust appearance model based on features extracted from a random partition image hashing algorithm to account for severe occlusion and disappearance. We divide the original image into multiple sub-blocks with random positions and scales. Hash functions are used to map blocks into compact binary codes, with which more effective target matching can be achieved. The tracking task is then formulated by producing a confidence map for the target and background, and obtaining the best samples using maximum *a posteriori* estimate. Experimental results demonstrate that our tracker can achieve more accurate tracking results in situations of occlusion, out-of-view, and violent motion blur when compared with most of state-of-the-art competing algorithms. Besides, the proposed tracking algorithm is able to run in real time.

I. INTRODUCTION

Given a bounding box defining the object of interest in the first frame, the aim of target tracking is to automatically indicate the presence of the target and its location if positive. The visual object tracking is an important problem in computer vision analysis, and it remains a challenging problem due to a variety of reasons, such as change of illumination, motion blur, partial occlusion, target disappearance, requirement of real-time processing, to name a few. An effective appearance model is of great importance to a visual tracking algorithm, and for this reason has garnered much attention of researchers in recent years [1]–[7].

In general, tracking algorithms can be categorized as generative [1]–[3] and discriminative approaches [6], [7]. Discriminative algorithms treat the tracking problem as a binary classification task to distinguish the target from background. Collins *et al.* [8] demonstrate that most of discriminative features can be learned on-line to handle the background/foreground classification problem. Tang *et al.* [9] propose an on-line semi-supervised learning framework to separate the target object from background by support vector machine. Babenko *et al.* [10] combine the idea of boosting with multiple instance learning to alleviate ambiguity among training samples. Kalal *et al.* [11] develop a semi-supervised learning algorithm in which negative and positive samples are selected from an on-line classifier with structural constraints.

Generative tracking algorithms typically learn the target appearance model and search for an image region that best

matches it with the minimal reconstructed error. Mei *et al.* [12] use sparse representation in the ℓ_1 -tracker, in which each target candidate is sparsely represented by target and trivial templates. Kwon and Lee [13] decompose the observation into multiple basic models that are constructed by sparse principal component analysis, to deal with appearance and motion change. Jia *et al.* [14] propose a novel alignment-pooling method which integrates local structural information to make the tracker more robust for partial occlusion. Recently, Henriques *et al.* [15] propose a novel tracking algorithm which uses a circulant structure to reduce the samples and make the tracker run at hundreds of frames-per-second.

Despite numerous visual tracking algorithms proposed in the literature, several problems remain unsolved. One of them happens in a long-term tracking, when the target disappears from the camera view, the tracking algorithm is required to restore target tracking once the target reappears at a later time. Unfortunately, most of tracking algorithms would fail and suffer from drifting problem in this situation. They either focus on tracking in local regions, or lack an efficient and/or effective feature extracting algorithm.

The paper focuses on investigation of the aforementioned temporary target disappearance problem. We propose an effective yet fast tracking algorithm with an appearance model based on features extracted from a random partition hash image. In order to extract random partition hash image features, we first construct multiple rectangle filters with random locations and scales, followed by projection of all samples into the discriminative hash binary space. Meanwhile, we assign an impact factor for each rectangle filter, which is used to guide the appearance model to focus on discriminative and significant parts of the target. To take advantage of motion information, the optical flow tracker in [16] is introduced to estimate the object motion, based on which a confidence map is built for samples. At last, we use the confidence map to separate the target from the surrounding background via a naive Bayes classifier. Experimental results show that the proposed visual tracking algorithm runs in real time, and its accuracy outperforms that of most state-of-the-art visual trackers in scenarios where the target temporarily leaves the camera view and/or severe occlusion happens.

The remainder of the paper is organized as follows: In Section II, we present our proposed feature extracting algo-

rithm and tracking algorithm. In Section III, qualitative and quantitative evaluations of our proposed tracker on a number of challenging videos and some discussions on the experimental results are made. Finally, Section IV concludes our work.

II. THE PROPOSED ALGORITHM

In this section, we first introduce the Bayesian estimation model adopted in the proposed tracking algorithm. Next, a random partition image hashing algorithm is presented to extract features from the image and compact them into binary codes for object matching. Finally, the prior probability model update scheme to change the searching strategy during the tracking process is described.

A. Bayesian Estimation Model

The goal of target tracking algorithm is to determine 1) the location of the object if it is present, or 2) indicate that the object is absent in the current frame. In this paper, object tracking is carried out within the Bayesian estimation framework [17]. Given the observation set $Y_{1:t} = \{Y_1, \dots, Y_t\}$ up to the t^{th} frame, the target state X_t can be obtained by the maximum *a posteriori* estimate which is defined as

$$\hat{X}_t = \arg \max_{X_t^i} P(X_t^i | Y_{1:t}) \quad \forall i = 1, 2, \dots, N \quad (1)$$

where X_t^i denotes the state of the i^{th} sample $\{S_t^i\}_{i=1}^N$ at time t . The posterior probability $P(X_t | Y_{1:t})$ can be inferred by the Bayesian theorem recursively,

$$P(X_t | Y_{1:t}) \propto P(Y_t | X_t) \int P(X_t | X_{t-1}) P(X_{t-1} | Y_{1:t-1}) dX_{t-1} \quad (2)$$

In this work, X_t can be further decomposed to $X_t = (X_t^c, X_t^s)$, where X_t^c denotes the center position of the target and X_t^s is the scale; $P(X_t | X_{t-1})$ and $P(Y_t | X_t)$ are the dynamic model and observation model, respectively. The dynamic model $P(X_t | X_{t-1})$ describes the temporal correlation of the target states between consecutive frames and the observation model $P(Y_t | X_t)$ denotes the likelihood of the observation Y_t at state X_t .

In the proposed approach, we introduce a random partition image hashing algorithm to extract hash features for each sample. Specifically, it is employed to compute the similarity score between the candidate area and the target. In our method, we use the proposed feature extracting algorithm to construct the observation model $P(Y_t | X_t)$. Details on the extraction of random partition image hashing features are presented in Section II-B.

A confidence map is built according to an optical flow tracker to serve as *a priori* information for each sample, which denotes the prediction $P(X_t | X_{t-1})$ of target state in the next frame. Details about the confidence map can be found in Section II-C. The main steps of the proposed algorithm are summarized in Table I.

TABLE I
PROPOSED VISUAL TRACKING ALGORITHM

Initialization:
for $j = 1$ to M (the count of partitions)
1. produce random values between 0 and 1 for x_j, y_j, w_j, h_j
2. generate a random rectangle filter $f_{x_j, y_j, w_j, h_j}^j(u, v)$
end for
Tracking:
for every frame in the sequence
1. Sample the current frame I_t into N slide windows $\{S_t^i\}_{i=1}^N$
2. Use the mean shift optical flow algorithm to predict target state
3. Build the confidence map $C(X_t^i)$ using Eq. 6), 7) and 8)
4. Extract random partition hash features for target and each sample
5. Calculate the likelihood $D(S_t^i)$ for each sample by Eq. 5)
6. Calculate the maximum <i>a posteriori</i> estimate by Eq. 2) and 1)
7. Update the confidence map by Eq. 9)
end for

B. Random Partition Image Hashing

Before the hashing process takes place, a low-pass Gaussian filter is applied on the input image, which is then scaled into a low dimensional space. The low-pass filtering is an effective preprocessing step to enhance the robustness of image process [18]. To extract random partition image hashing features, we first randomly produce a set of M rectangle filters $\{f_{x_j, y_j, w_j, h_j}^j(u, v)\}_{j=1}^M$ at multiple scales and different locations. The j^{th} filter defined as

$$f_{x_j, y_j, w_j, h_j}^j(u, v) = \begin{cases} 1, & x_j \leq u < x_j + w_j, y_j \leq v < y_j + h_j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where x_j, y_j, w_j and h_j are respectively the start point, the width and height of the defined rectangle filter. These filters are then convolved with i^{th} sample S_t^i , so as to obtain a set of partitions, each of which represents a local part of the original image.

Random partition image hashing is applied to extract features for each candidate target. Each partition is represented as $U_j(x, y, w, h)$, where (x, y) is the center point of rectangle, w and h denote the width and height, respectively. Perceptual hashing algorithm is applied to turn each partition U_j into hash binary in the same way. Firstly, U_j is resized to 8×8 . The motivation of this step is to reduce computation and make the obtained hash codes resistant to various resolutions. However, it can reserve most of discriminative information. Then, mean value μ_j of each partition is computed to normalize pixel value $I(k)$ of image into a binary code by

$$h(k) = \begin{cases} 1, & I(k) < \mu_j \\ 0, & \text{otherwise} \end{cases} \quad 1 \leq k \leq 64 \quad (4)$$

Finally, we combine the binary vector of each partition together as a feature vector to represent partition U_j .

Similarity measurement is required to decide the confidence for each of samples $\{S_t^i\}_{i=1}^N$. In the proposed algorithm,

hamming distance is employed to obtain the likelihood of each sample at time t , compared with the target based on the appearance model. It is computed as

$$D(S_t^i) = \sum_{j=1}^M F_U^{i,j} \oplus F_T^j \quad (5)$$

where F_T^j and $F_U^{i,j}$ denote the j^{th} hash features of the target object and the i^{th} sample, respectively. In the proposed algorithm, we consider the observation $P(Y_t|X_t)$ to be proportional to the likelihood $D(S_t^i)$ of each sample.

C. Confidence Map

For samples $\{S_t^i\}_{i=1}^N$ in the current frame I_t , a confidence map is built on the basis of the optical flow information and the target state estimate in the previous frame I_{t-1} .

The optical flow is used to compute the motion between consecutive frames of a sequence. Horn and Schunk [19] consider the optical flow problem as the following energy function

$$e = \int_{\Omega} |\nabla_{g_1}|^2 + |\nabla_{g_2}|^2 d\Omega + \lambda \int_{\Omega} I_t(x + \mathbf{g}(x)) - I_{t-1}(x)^2 d\Omega \quad (6)$$

where I_{t-1} and I_t represent the consecutive frames, $\mathbf{g} = (g_1, g_2)^T$ is the two dimensional flow field. The first term in Equation 6 is regularization term $\int_{\Omega} |\nabla_{g_1}|^2 + |\nabla_{g_2}|^2 d\Omega$ smoothing the flow field, and second term is a data term $\int_{\Omega} (I_t(x + \mathbf{g}(x)) - I_{t-1}(x))^2 d\Omega$. λ is a coefficient balancing the data term and regulation term. However, by minimizing the Eq. 6) can get the best state estimate from previous frame.

The mean shift optical flow algorithm in [16] is adopted in the paper, since it adapts to appearance changes quickly and accurately, and also offers high robustness when dealing with occlusions [20].

With the optical flow information, we are able to obtain the motion model and predict the target state in the current frame based on the estimate from the previous frame. Nevertheless, the optical flow based motion prediction may contain random estimation errors due to visual occlusion, target disappearance, etc. To deal with this issue, a Gaussian function $f_g(\nabla x, \nabla y)$ is introduced to probabilistically model the estimation uncertainty for every sample. It is defined as

$$f_g(\nabla x, \nabla y) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{(\nabla x - \mu_1)^2}{2\sigma_1^2} - \frac{(\nabla y - \mu_2)^2}{2\sigma_2^2}} \quad (7)$$

where (μ_1, σ_1) and (μ_2, σ_2) represent the mean and standard deviation in x and y directions, respectively. $(\nabla x, \nabla y)$ is the difference between center points of sample bounding box and optical flow estimation. This generates confidence $C(X_t^i)$ for the i^{th} sample bounding box written as

$$C(X_t^i) = \begin{cases} f_g, & \text{if } f_g > P_g \\ P_t, & \text{otherwise} \end{cases} \quad (8)$$

where $P_g > 0$ serves as a small probability threshold adopted to increase the tracker's robustness against erroneous motion prediction.

D. Online Update

As previously discussed in Section II-A, the Bayesian tracking model is applied in the visual tracking process. However, the maximum *a posteriori* estimate does not contain the probability distribution information of the estimation result, which may result in inconsistency problem especially when the uncertainty is large in some scenarios like target disappearance or occlusion.

To overcome this limitation, an on-line update scheme for the standard deviations σ_1 and σ_2 used in constructing the confidence map is adopted in the tracking algorithm. They are updated at each time step based on the probability P_{t-1} at the maximum *a posteriori* estimate at the previous time step. It is given as

$$\sigma = \begin{cases} \frac{\alpha}{P_{t-1}}, & P_{t-1} > P_{\sigma} \\ \infty, & \text{otherwise} \end{cases} \quad (9)$$

where $P_{t-1} = P(\hat{X}_{t-1}|Y_{1:t-1})$, and $\alpha > 0$ is a constant. Essentially, in Equation 9, the σ has a positive correlation with the uncertainty degree of the posterior distribution $P(X_{t-1}|Y_{1:t-1})$. When P_{t-1} is smaller than a pre-set threshold P_{σ} , then $\sigma = \infty$, which means the confidence map becomes a uniform distribution. In cases where target disappearance or occlusion happens, the on-line update scheme enables quick and correct recapture of the target when it appears in the scene again.

III. EXPERIMENTS

A. Experimental setup

The performance of the proposed visual tracking approach has been evaluated by using 12 publicly available video sequences¹. Captured in different environments, these testing sequences contain a variety of events, such as fast movement, motion blur, illumination change, occlusion, out-of-view, shape deformation, etc. The performance of the proposed approach is compared with that of 6 state-of-the-art visual trackers, namely the TLD tracker [11], the compressive tracker (CT) [21], the L1 tracker (L1APG) [22], the distribution fields tracker (DFT) [23], the incremental learning tracker (IVT) [2], and the circulant structure tracker (CSK) [15]. The average running time of the Matlab implementation of the proposed approach is about 15 ms per frame on a workstation with an Intel Xeon E5-1630 3.70 GHz processor and 8.0 GB RAM. In the experiments, we set the quantity of partitions $M = 20$ for the proposed tracker. Besides, P_g , α , and P_{σ} are set to 0.1, 1, and 0.08, respectively.

B. Experimental Results

Some of the tracking results in the experiments are illustrated in Figure 1, which intuitively compares the performance of proposed algorithm and 6 other competing trackers in 12 video clips. The result of quantitative and qualitative are recorded for each trackers on different sequences.

¹ Available at http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html

Fig. 1. Comparison of the proposed approach (red) and 6 other trackers, which are TLD [11] (blue), CT [21] (green), LIAPG [22] (black), DFT [23] (cyan), IVT [2] (yellow), and CSK [15] (magenta). The result of our tracker is marked as red. The proposed tracking algorithm can deal well with severe occlusion, plane rotation, illuminate changing and fast movement

TABLE II

SUCCESS RATE (%) OF TRACKING. THE RED, BLUE, AND GREEN COLORS INDICATE THE BEST, THE SECOND, AND THE THIRD BEST PERFORMANCE RESPECTIVELY. IN TOTAL 9757 EVALUATED FRAMES

Sequence (frames)	TLD	CT	LIAPG	DFT	IVT	CSK	Proposed
<i>BlurBody</i> (334)	17	29	54	11	7	59	89
<i>Box</i> (1161)	70	41	37	60	33	9	72
<i>Dancer2</i> (150)	68	96	71	99	99	100	100
<i>FaceOcc1</i> (892)	92	93	98	80	92	98	96
<i>Girl2</i> (1500)	28	25	16	18	20	10	78
<i>Human2</i> (1128)	43	32	18	23	29	18	70
<i>BlurFace</i> (493)	100	20	56	29	14	100	100
<i>Lemming</i> (1336)	34	34	17	48	17	43	80
<i>Jumping</i> (313)	70	7	32	42	18	9	100
<i>Skater</i> (160)	39	88	77	50	52	56	80
<i>Suv</i> (945)	84	21	56	42	45	57	93
<i>Sylvester</i> (1345)	80	71	43	41	67	71	88
Average SR	59	46	47	50	41	52	87
Average FPS	13	87	14	17	10	181	62

Two popular metrics are also adopted to quantitatively evaluate these trackers. The first metric is the success rate based on the score in each frame, which is computed as

$$s = \frac{\text{area}(R_t \cap R_g)}{\text{area}(R_t \cup R_g)}$$

where R_t is the tracking result and R_g is the ground truth bounding box. For each frame, if the score s is larger than 0.5, it is considered success tracking for this frame. Table II summarises the success rate results, obtained given a fixed set of algorithm parameters for each testing video sequence. It can be concluded that the proposed tracking algorithm ranks the first or the second in all of testing sequences. Table II also shows the mean FPS for each tracker. Although not the fastest, the proposed tracker achieves an average FPS of 62, sufficiently fast for on-line running.

The other metric is the centre location error (CLE), which measures the centre error of the target tracking result with respect to the ground truth. Figure 2 illustrates the quantitative evaluation results based on the CLE metric. It is clearly shown that overall the proposed tracking algorithm obtains the smallest centre errors among all tested trackers in the most of video sequences.

The detailed performance comparison is discussed from different perspectives as follows.

Partial/Severe Occlusion and Out-of-View: The *Human2* and *FaceOcc1* sequences contain segments of part occlusion. Besides, the targets in *lemming*, *girl2*, and *suv* sequences experience severe occlusion even out-of-view for a few times. The proposed tracking approach is however not sensitive to partial occlusion, as its tracking model takes into account random partitions with different locations and scales for each sample. In the meanwhile, the confidence map and on-line update scheme enable adaptive searching strategies based on learning of appearance change from previous tracking results. As the quantitative results show in Figure 2, the proposed tracker manages to detect and handle all severe occlusion cases in the aforementioned sequences. Only some of the other trackers can correct drifts caused by severe occlusion, though not successful each time, and the rest fail to handle the large appearance change and recover from the drifts. The *girl2* sequence serves as an example that contains multiple times of out-of-view. Clearly, it is seen that the proposed tracker can quickly recover from drifts caused by out-of-view, while all other trackers are not able to deal with the out-of-view situations properly, which result in substantially higher CLEs than that of the proposed tracker.

Fast Movement and Plane Rotation: The object in the *BlurFace* sequence undergoes fast movement. Overall, the proposed approach achieves the best performance in this situation. TLD and CSK are also able to produce well tracking results. The other trackers, on the other hand, can not correct the drifts due to fast movements, while some of the them, e.g. CT and IVT, completely lose the target since the first several frames. In the *Jumping* sequence, significant motion blur happens due to a combination of severe target movement. Only TLD and the proposed method can track the target accurately in this sequence. When the deformation and fast movement occur in *BlurBody* sequence, the proposed tracker outperforms the other competing trackers in terms of accuracy and success rate. The proposed tracker also performs the best in the *Skater*, *Box* and *Sylvester* sequences, which contain plane rotation.

Illumination Change: In the *Sylvester* sequence, the background light blurs the target from time to time. According to Table II, the proposed tracker obtains the highest success rate, i.e., 88%, with the aid of the random partition image hashing mechanism. The object appearance varies in the *Human2* sequence, under the influence of changing background light. In the meanwhile, frequent pose variations happen to the target. In this case, the proposed algorithm obtains the success rate as high as 70%, which is much higher than other 6 trackers.

Fig. 2. Quantitative evaluation in terms of CLE (in pixel). The proposed tracking method is compared with 6 state-of-the-art trackers on 12 challenging video clips.

IV. CONCLUSION

This paper presents a simple, fast, and robust visual tracking approach with an effective appearance detection model. An efficient hash feature extracting method is used to convert the image patch into hash code vector. The tracking task is formulated as a binary similarity calculating problem with on-line update the searching strategy. Our tracker also incorporates the Bayesian estimation model to evaluate the possibility for each image patch region. Experimental evaluation with publicly available sequences highlights the high speed and success rate, high accuracy and robustness of the proposed visual tracking algorithm.

REFERENCES

- [1] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.
- [2] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.
- [3] D. Wang and H. Lu, "Visual tracking via probability continuous outlier model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3478–3485.
- [4] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *IEEE transactions on image processing*, vol. 22, no. 1, pp. 314–325, 2013.
- [5] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1838–1845.
- [6] S. Avidan, "Support vector tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.
- [7] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1639–1651, 2014.
- [8] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.
- [9] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.
- [10] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.
- [11] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [12] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.
- [13] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1269–1276.
- [14] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1822–1829.
- [15] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*. Springer, 2012, pp. 702–715.
- [16] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust

- object detectors during unstable tracking,” in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1417–1424.
- [17] M. Isard and A. Blake, “Condensationconditional density propagation for visual tracking,” *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.
 - [18] A. Swaminathan, Y. Mao, and M. Wu, “Robust and secure image hashing,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215–230, 2006.
 - [19] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
 - [20] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, “Prost: Parallel robust online simple tracking,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 723–730.
 - [21] K. Zhang, L. Zhang, and M.-H. Yang, “Real-time compressive tracking,” in *European Conference on Computer Vision*. Springer, 2012, pp. 864–877.
 - [22] C. Bao, Y. Wu, H. Ling, and H. Ji, “Real time robust l1 tracker using accelerated proximal gradient approach,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1830–1837.
 - [23] L. Sevilla-Lara and E. Learned-Miller, “Distribution fields for tracking,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1910–1917.