Visual Object Tracking Using Pairwise Pixel Descriptor

Anonymous CVPR submission

Paper ID 2924

Abstract

In this paper, we propose a novel algorithm for fast generic objects tracking. First, a simple yet effective feature, which is a tuple consisting of two colors and called pairwise pixel descriptor, is proposed to describe an edge pixel. The pairwise pixel descriptor simultaneously encodes the gradient and color information, which enables our tracker to distinguish the target object from background image regions with the similar gradients or colors. Second, based on the generalized Hough transform, we train a Hough model by using the PPDs of all object's edge pixels for fast detection of the target object. Before that, we first train a Bayes classification model to distinguish the object's edge pixels from the background's edge pixels. Both models are combined to track the target object to improve the robustness of the proposed tracker. Furthermore, we show that both models can be efficiently implemented by a two-dimensional look-up table, making the proposed tracker fast. The experimental results on a dataset of 77 sequences demonstrate that the proposed method is comparable to state-of-the-art trackers.

1. Introduction

Visual object tracking is one of the most fundamental tasks in computer vision and has a wide range of applications such as video surveillance, intelligent robot and human-computer interaction. Although much progress has been made during the past decades, building a generic object tracking algorithm remains a big challenge due to the following two aspects:

First, it's hard to model the variations of an object's appearance, which is related to the question of how to represent an object without any prior knowledge of the object [6]. Indeed, a well-designed feature or representation scheme can dramatically improve the tracking performance [22]. However, developing a proper and effective feature or representation scheme for tracking is still an open question.

Second, the movement of an object in a video is usually unpredictable. To "guess" where the object will appear in a new frame, the uniform or the Gaussian motion model is often employed [20]. However, both motion models are usually time-consuming due to the repeated calculations (e.g. repeated feature extractions) on many overlapped image regions. Hence, these motion models are the bottleneck of a tracker for real-time applications.

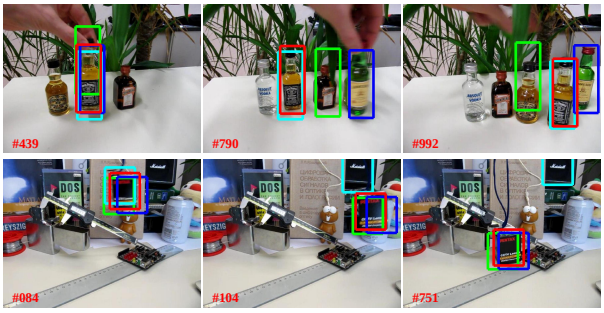

Figure 1. Illustration of the importance of the representation scheme. In the first row, trackers (e.g., KCF and STRUCK) using gradients alone may drift to background image regions with the similar gradients to the target object. Similarly, in the second row, tracker (e.g., DAT) using colors alone may drift to background image regions with the similar colors to the target object. While our tracker successfully tracks the target object in both cases using the proposed feature.

In this paper, we try to address these challenges by proposing a novel tracking algorithm using pairwise pixel descriptor (PPD). First, we argue that it is difficult to discriminate one non-edge pixel from another. Therefore, instead of using all pixels in an image region (e.g. [3, 6, 17]), we only consider edge pixels to track an arbitrary object in a video. More specifically, we propose a simple yet effective feature called pairwise pixel descriptor (PPD) to describe the edge pixel. Both the gradient direction of the edge pixel and the corresponding colors of the pairwise neighborhood pixels are simultaneously encoded in the proposed feature, as enables our tracker to distinguish an object from background image regions with the similar gradients or colors (as illustrated in Figure 1). Second, based on the generalized Hough transform [5], we train a Hough model by using

the PPDs of all object's edge pixels for fast detection of the target object. Before that, we first need to distinguish object's edge pixels from background's edge pixels. Here, we train a Bayes classification model to implement this task. Both models are combined to track the target object, and are updated continuously to adapt to the changes of the object appearance. Moreover, we implement both the models with a two-dimensional look-up table to make our tracker fast. Finally, we evaluate our tracker on a dataset of 77 sequences to demonstrate its favorable performance compared to a variety of state-of-the-art trackers.

2. Related work

In order to track a generic object without any prior knowledge, a number of part-based tracking approaches have been proposed. Adam *et al*. [2] proposed to divide the target object into several non-overlapped parts. Each part is tracked independently by comparing its histogram with the corresponding image patch histogram. Only those reliable matched parts are used to infer the final position of the target. Though this approach is originally designed for the purpose of addressing partial occlusions, it was demonstrated to be effective for generic object tracking as well. Kwon *et al*. [13] proposed an approach to track non-rigid objects that are composed by a dynamic set of image patches, and they used the Basin Hopping Monte Carlo sampling to speed up the calculations. Some approaches [7, 16, 18, 23] integrated segmentation algorithms into their tracking framework to track an arbitrary object. For example, Wang *et al*. [23] use the SLIC algorithm [1] to extract superpixels, and then used mean shift clustering to separate superpixels from the target object and background into different clusters. Godec *et al*. [7] proposed a patch-based voting algorithm, which is based on the generalized Hough transform. More specifically, they utilized a graph-cut algorithm to segment foreground from background, and used the segmentation result to update the Hough forest to adapt to the deformations of the target object.

Generally, the part-based methods represent the target object using image patchs, which are inevitably include more or less background information. Thereby, they will suffer from the drifting especially when the target object undergoes non-rigid deformations. To address this problem, some tracking methods utilized pixel-based features or descriptors to describe the target object. Avidan *et al*. [3] proposed an ensemble tracking method by training an online Adaboost algorithm to label each pixel as foreground or background. Horst *et al*. [17] proposed a tracking algorithm based on the color of each pixel. By using an adaptive object model, which can suppress nearby regions with a similar appearance, this approach is simple and achieves state-of-the-art performance in some cases. Nebehay *et al*. [15] proposed a keypoint-based approach for deformable object tracking by only considers keypoints instead of all pixels to represent the target object. The approach runs fast by using BRISK [14], which is a fast keypoint detector and descriptor.

The work of Duffner *et al*. [6] is somewhat similar to ours. The authors proposed a pixel-based Hough tracker that obtains a favorable performance on non-rigid object tracking and is faster than the patch-based Hough tracker [7]. However, it may fail when the background image regions contain amounts of pixels similar to the target object in terms of gradients and colors. We address this problem by proposing a simple yet effective feature to describe edge pixels. Our motivation comes from such an observation that edge pixels usually contain more discriminative information than non-edge pixels and thus should be more useful to infer the object center position.

3. Pairwise Pixel Descriptor

Given an image I of which each pixel at position (x, y) is denoted as $I(x, y)$, and the horizontal and the vertical gradient image of I are denoted as I_H and I_V, respectively. We consider the pixel $I_H(x, y)$ as a horizontal edge pixel if its magnitude is larger than a predefined threshold δ, and the pixel $I_V(x, y)$ a vertical edge pixel if its magnitude larger than δ.

The pairwise pixel descriptor (PPD) of a horizontal or a vertical edge pixel at position (x, y) is defined as a tuple consisting of two color indices, namely:

$$ppd_H(x, y) = (C_{I(x,y-r)}, C_{I(x,y+r)}) \quad (1)$$

$$ppd_V(x, y) = (C_{I(x-r,y)}, C_{I(x+r,y)}) \quad (2)$$

where $r > 0$ is a small displacement value, and C_I denotes the color index of the corresponding pixel in the image I, e.g. $C_{I(x,y-r)}$ is the color index of the pixel $I(x, y - r)$. Here, the color index is computed in the HSV color space that has $N_h \times N_s \times N_v$ quantized bins. In this paper, we set $N_h = 16$, $N_s = 4$, and $N_v = 4$. Hence, the color index ranges from 0 to 255.

PPD has three desirable properties. First, PPD is an informative local image descriptor defined on an edge pixel: the color-pair causing the edge pixel and the gradient direction (either horizontal or vertical) of the edge pixel are incorporated in PPD. Such information enables us to build a robust detector or classifier of the tracked object using PPD. Second, PPD is robust against deformations. We find that, when the tracked object suffers from rigid or non-rigid deformations between two consecutive frames, only the shape of its edges changes accordingly. In other words, only positions of its edge pixels change, but the color-pairs and the gradient directions of the corresponding edge pixels mostly remain the same. This property makes PPD very effective to represent the non-rigid objects in video sequences. Third,

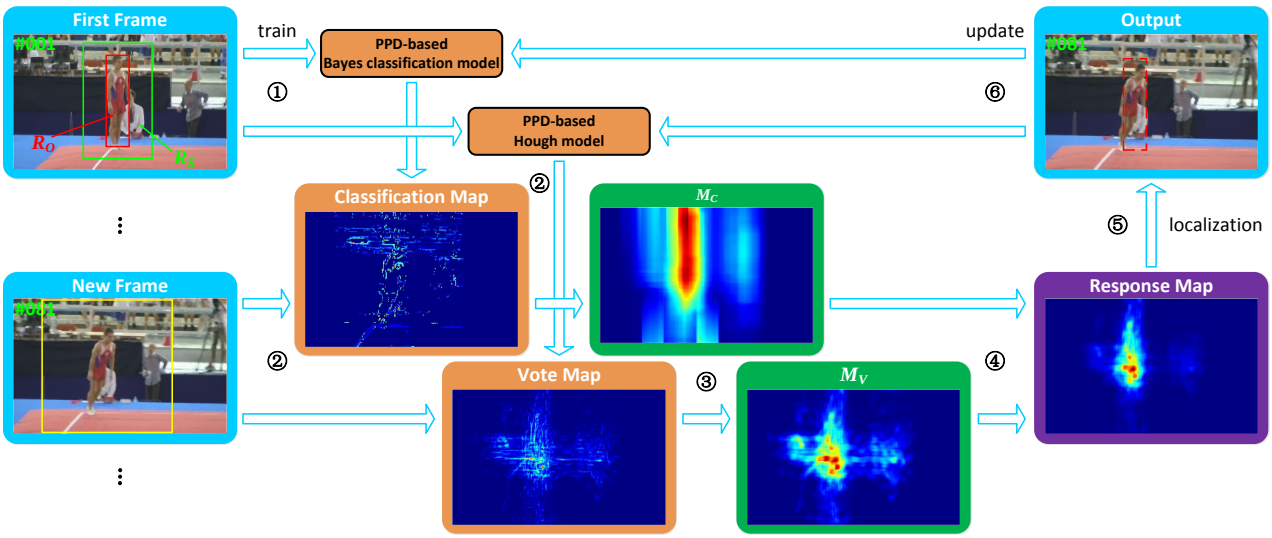

Figure 2. A flowchart of the proposed tracking approach. We build a Bayes classification model and a Hough model using the proposed feature in the first frame. In a new frame, both models are combined to track the target object and then updated using the tracked result. For more details, please refer to Section 4.

the procedure to extract PPDs from an image is simple and fast, as is crucial for many on-line vision applications.

4. Overall Tracking Approach

Based on PPDs, we build a Bayes classification model and a Hough model to track the target object. The overall tracking procedure is illustrated in Figure 2. Given the first frame of a video sequence and an initial bounding box (solid red rectangle in Figure 2), we first build a Bayes classification model and a Hough model using PPDs extracted from the region inside an enlarged bounding box (solid green rectangle in Figure 2). In particular, the Bayes classification model is trained, using color-pairs of the corresponding edge pixels as features, to classify object's and background's edge pixels (see Section 4.1). The Hough model is an object detector relying on the generalized Hough transform [5] and use edge pixels as "voters" (see Section 4.2). In a new frame, edge pixels are detected, and the corresponding PPDs are extracted inside a search window (solid yellow rectangle in Figure 2). Then both models are applied to every edge pixel to produce a classification map and a vote map, respectively. The final response map is a combination of the classification map and the vote map. The position of the maximum in the response map determines the new position of the target object (see Section 4.3). Finally, both the Bayes classification model and the Hough model are updated continuously using the tracked result to adapt to the changes of the object appearance (see Section 4.4).

4.1. Bayes Classification Model

Our goal is to compute the probability of an edge pixel belonging to the target object. To this end, we train a Bayes classification model using color-pairs of both horizontal edge pixels and vertical edge pixels as features.

Let us denote p as an edge pixel and $c_p \in \Re^2$ as the corresponding color-pair of p. Given the initial bounding box in the first frame, we consider regions inside and outside the bounding box as object region R_O and surrounding background region R_S, respectively (see Figure 2). Similar to [17], we apply the Bayes rule to obtain the probability of an edge pixel belonging to the target object O:

$$P(p \in O|c_p) \approx \frac{P(c_p|p \in R_O)P(p \in R_O)}{\sum_{\Omega \in \{O,S\}} P(c_p|p \in R_\Omega)P(p \in R_\Omega)} \quad (3)$$

In practice, we can estimate the likelihood terms and the prior probability as:

$$P(c_p|p \in R_\Omega) \approx \frac{|c_p \in R_\Omega|}{|p \in R_\Omega|} \quad (4)$$

$$P(p \in R_\Omega) \approx \frac{|p \in R_\Omega|}{|p \in R_O| + |p \in R_S|} \quad (5)$$

where $|\cdot|$ denotes the cardinality. Then, Equation (3) can be approximated as:

$$P(p \in O|c_p) \approx \frac{|c_p \in R_O|}{|c_p \in R_O| + |c_p \in R_S|} \quad (6)$$

Here, an edge pixel p is regarded as an object's edge pixel if $P(p \in O|c_p) > 0.5$.

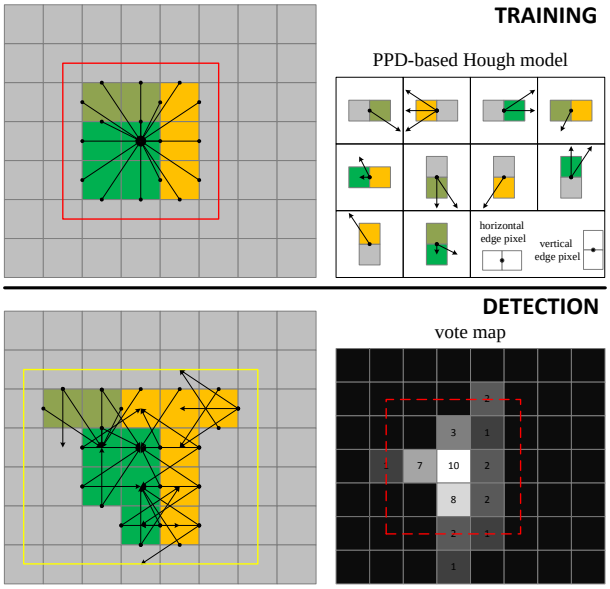

Figure 3. Training and detection with the PPD-based Hough model. Numbers in the vote map denote the number of supported voters at the corresponding position.

4.2. PPD-based Hough Model

We build an object detector, which is called PPD-based Hough model, relying on the generalized Hough transform [5]. In contrast to the existing Hough based detectors that use small image patches [7] or pixels [6] as "voters", our method uses edge pixels instead.

The proposed PPD-based Hough model is illustrated in Figure 3. In the training phase, we detect both horizontal edge pixels and vertical edge pixels in the object region (solid red rectangle in Figure 3) as "voters". Each voter stores a vote vector (black arrow in Figure 3) pointing to the center of the target object. We gather votes of all voters into bins according to their corresponding PPDs (i.e. color-pairs and gradient directions). Finally, the trained PPD-based Hough model is composed by a set of PPDs and related vote vectors.

In the detection phase, we first detect edge pixels inside the search window (solid yellow rectangle in Figure 3). Each edge pixel votes according to the related vote vectors in the trained Hough model. The votes of all edge pixels are accumulated in a vote map, and the position with the highest sum of votes is the most likely position of the object's center. Please note that, we ignore the edge pixels whose PPDs are not in the trained Hough model.

4.3. Tracking

For a new coming video frame, we first detect edge pixels inside the search window. Then, the Bayes classification model classifies the edge pixels into either object's edge pixels or background's edge pixels to produce a classification map. After that, the Hough model is applied to object's edge pixels. Votes of all object's edge pixels are accumulated to produce a vote map. Although the position of the object can be estimated by using the vote map alone, the estimated position may "diffuse" due to the deformations of the object. To address this problem, we utilize both the classification map and the vote map to track the object. More specifically, we first perform the mean filtering on both maps. For the classification map, the size of the filter is the same size of the initial bounding box. For the vote map, the size of the filter is 7×7 for all video sequences. We denote the filtered classification map and the filtered vote map as M_C and M_V, respectively. Then, we compute a final response map M_R by multiplying M_C and M_V:

$$M_R = M_C \odot M_V \tag{7}$$

where $\odot$ denotes the dot product operation. Then, the new object center is set to be the position of the maximum value in the final response map M_R. Note that, we do not yet address the scale estimation in the current implementation, and thus the tracked result has the same width and height with the initial bounding box.

4.4. Model Adaptation

Both the Bayes classification model and the Hough model are updated at each frame. As for the Bayes classification model, we utilize a linear update scheme to update the probability of an edge pixel p belonging to the target object:

$$P_{t+1}(p \in O) := \begin{cases} \eta P'_{t+1}(p \in O) + (1-\eta) P_t(p \in O) & \text{if } P_t(p \in O) > 0 \\ P'_{t+1}(p \in O) & \text{otherwise} \end{cases} \tag{8}$$

where $\eta = 0.01$ is the learning rate, $P'_t(p \in O)$ is the probability that evaluated by Equation (6) in the new object regions. The subscript t or $t-1$ denotes the frame number.

As for the Hough model, we update the vote vectors of each PPD. Here, we denotes R_N as a 7×7 region centers at the new object center. If the end point of a vote vector lies in the region R_N, we say that this vote vector contributes to the new object center. We remain those vote vectors that contribute to the new object center and delete others. Besides, we add new vote vectors of all object's edge pixels in the new frame into the corresponding PPDs. Note that, all background's edge pixels are not used to update the Hough model.

5. Implementations

For part-based or pixel-based tracking methods, one time-consuming step may be the matching of pairwise fea-

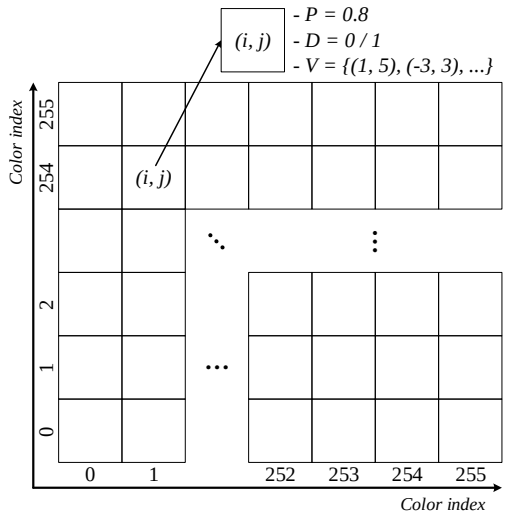

Figure 4. A two-dimensional look-up table $T_{256\times256}$, in which an element $T_{i,j}$ corresponds to a PPD of color pair (i, j). (P: the probability of color pair (i, j) belonging to the target object. D: the gradient direction, $D = 0$ indicates a horizontal gradient and $D = 1$ indicates a vertical gradient. V: a set of vote vectors).

tures, which will slow down the trackers especially when the feature has a high dimension or the matching function is complex.

Owe to the simplicity of the proposed feature, we can speed up the matching step by using a look-up table to implement both the Bayes classification model and the Hough model. In particular, we design a two-dimensional table $T_{N\times N}$, in which an element $T_{i,j}$ corresponds to PPD of the color pair (i, j). Here, $N = 256$ is the total number of the color index in the HSV color space. We further design a data structure to store the corresponding probability P, gradient direction D, and vote vectors V into $T_{i,j}$ (See Figure 4).

6. Experiments

In our experiments, we compute the horizontal and the vertical gradient images by using the following two filters:

$$F_H = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, F_V = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (9)$$

We set a big threshold for δ, i.e. $\delta = 50$ (in Section 3) to suppress weak edge pixels. Due to the low resolution of some video sequences, pixels close to the edge is usually blurred. Thus, we set $r = 3$ (in Equation (1) and (2)) to avoid the choosing the blurred pixels. In Figure 2, the sizes of the enlarged bounding box (solid green rectangle) and the search window (solid yellow rectangle) are $2W \times 2H$ and $(W + 80) \times (H + 80)$ respectively, where W and H denote the width and the height of the initial bounding box (solid red rectangle) respectively.

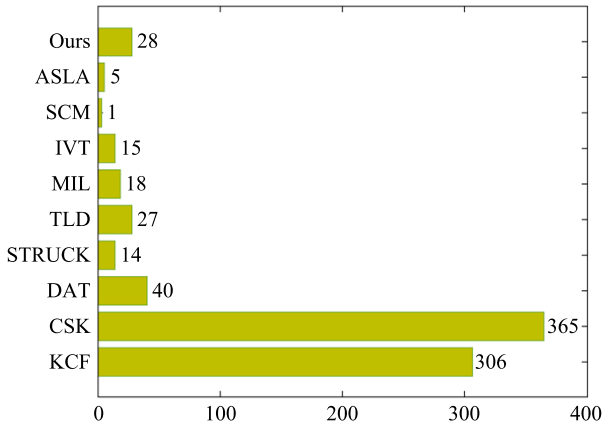

Figure 5. Comparison on running speed on the Vojir dataset. Our algorithm compares favorably to the state-of-the-art trackers.

6.1. Evaluation Protocol

Dataset. We evaluate the proposed method by comparing it with other state-of-the-art trackers on the tracking dataset[1] of Vojir et al. [21]. This dataset contains 77 test sequences, which have been extensively used by previous work. In this dataset, most of the objects of interest are non-rigid, making it suitable for evaluating the proposed method. Moreover, the dataset contains various situations like illumination variation, occlusion, fast movement, etc., which make it a challenging dataset to evaluate tracking methods.

Evaluation Metrics. We adopt the evaluation metrics in [24], that is, the precision plot and success plot. The precision plot shows the percentage of successfully tracked frames whose location are within a given threshold distance of the ground truth, and the representative precision score at threshold=20 is usually used to rank the trackers. The success plot, which measures an overlap score (ranges from 0 to 1) between the tracked bounding box and the ground truth bounding box at each frame, shows the percentage of successfully tracked frames. The area under curve (AUC) of each success plot is used to rank the tracking algorithms. We run the one-pass evaluation (OPE) on the Vojir dataset for all trackers.

Compared Algorithms. We run seven representative state-of-the-art trackers, including STRUCK [8], SCM [25], ASLA [11], TLD [12], CSK [9], MIL [4] and IVT [19], and report their results on the Vojir dataset. We also run two recent trackers in this experiment, that is, DAT [17] and KCF [10].

[1] Available at http://cmp.felk.cvut.cz/ vojirtom/dataset/

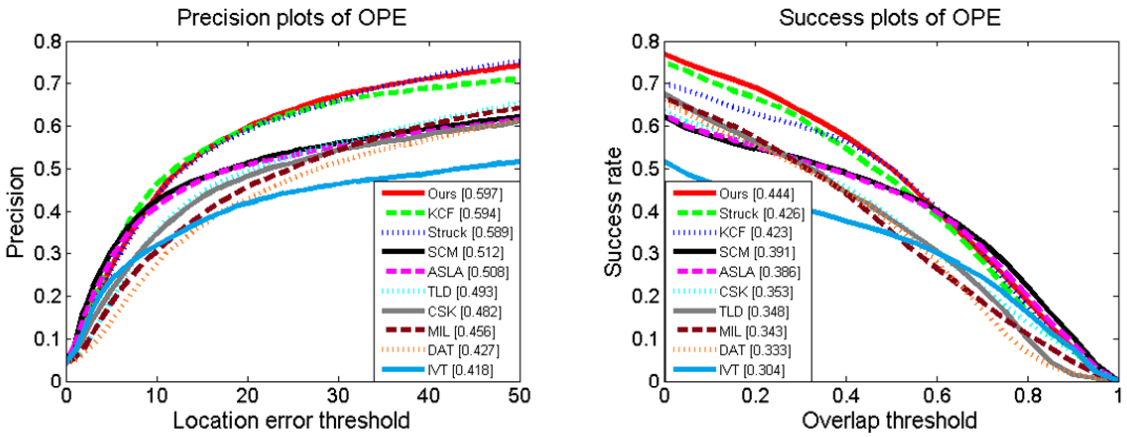

Figure 6. Tracking results of our method and the state-of-the-art trackers on the Vojir dataset. Left: Success plot of the location error. Right: Success plot of the overlap measure. Our method obtains the best performance in both evaluations.

6.2. Results

Comparison to State-of-the-Art Trackers. For a fair evaluation, we run the source codes provided by the authors on the Vojir dataset. The one-pass evaluation for all trackers is shown in Figure 6. Both the precision plots and success plots demonstrate that our algorithm outperforms most of the state-of-the-art trackers except KCF and STRUCK. We also show qualitative results on some selected challenging sequences in Figure 7. For example, results on sequences *Lemming* and *Tiger1* show that our algorithm is robust to partial occlusion, as owes to the PPD-based Hough model that utilizes the visible edge pixels as voters to find the center of the target object. Results on sequences *Asada* and *Figure skating* show that our algorithm is also robust to large deformations, which benefits from the proposed pixel-based feature.

Comparison on Speed. We run all the trackers on a same computer to make a fair comparison on the speed. The average frame rate of all trackers over the whole dataset is reported in Figure 5. While the correlation based trackers like CSK and KCF achieve superior results, our method obtains a result close to TLD. Note that, our algorithm and is implemented in Matlab, while some compared trackers are implemented in C/C++ (e.g., MIL and STRUCK) or a mixture of Matlab and C/C++ (e.g., TLD, CSK and KCF).

7. Conclusion

In this paper, we propose a novel algorithm for generic objects tracking. To represent an object without any prior knowledge, we propose a simple yet effective feature, called pairwise pixel descriptor (PPD), to describe the object at (edge) pixel level. When the background image regions have similar gradients or colors with the target object, the proposed PPD feature, which simultaneously encodes both gradient and color informationcan make our tracker avoid the drifting. Moreover, PPD is robust to deformations because colors close to an edge change slowly in a video sequence. Using PPD, we build a Bayes classification model and a Hough model to track the target object. Both models can be implemented with a two-dimensional look-up table to speed up the proposed algorithm. We test our method on the Vojir dataset that contains 77 sequences, and the experimental results demonstrate our method is comparable to other state-of-the-art trackers.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. Technical report, 2010. 2

[2] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805. IEEE, 2006. 2

[3] S. Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007. 1, 2

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009. 5

[5] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981. 1, 3, 4

[6] S. Duffner and C. Garcia. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In *Proceedings of the IEEE international conference on computer vision*, pages 2480–2487, 2013. 1, 2, 4

[7] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10):1245–1256, 2013. 2, 4

[8] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011*

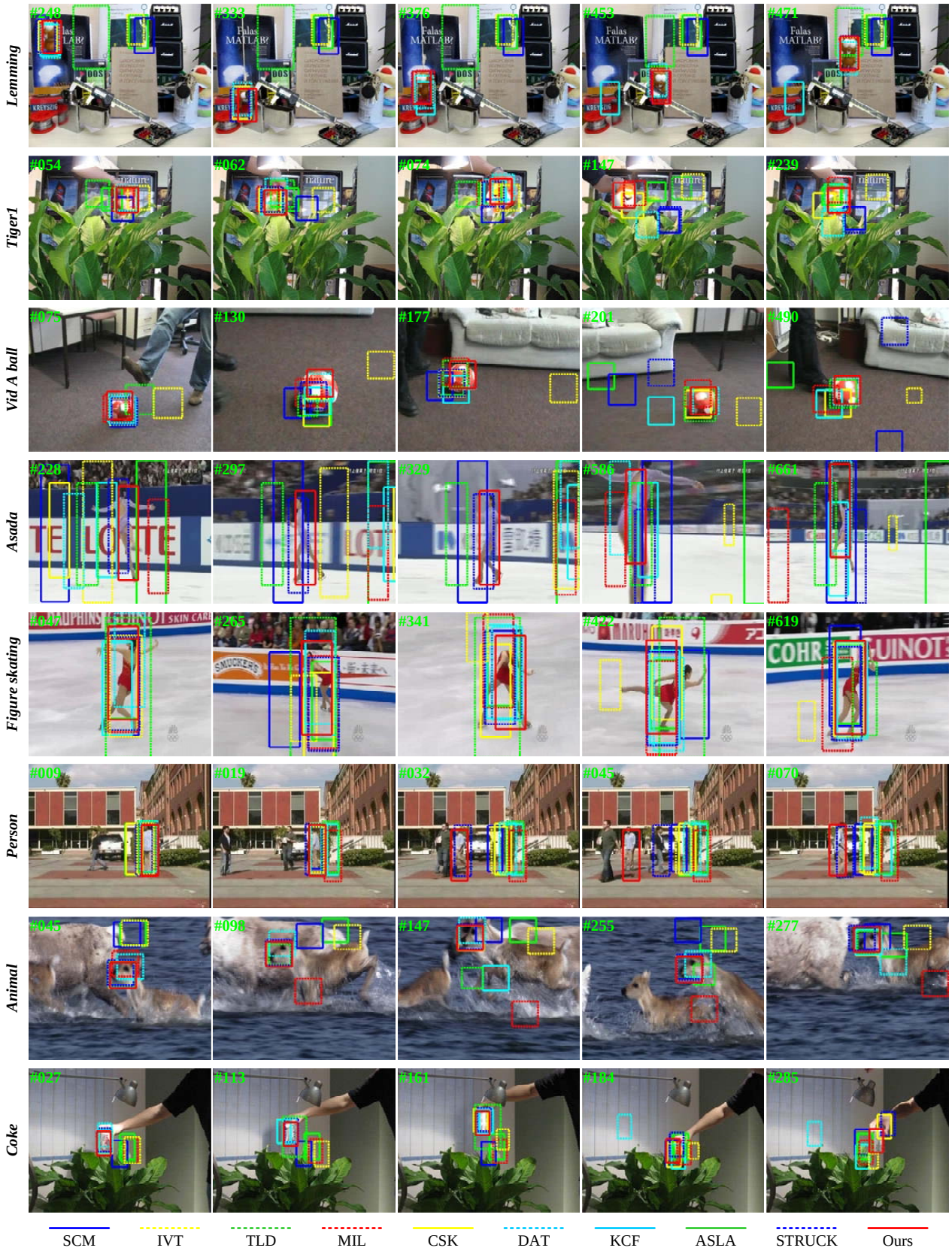

Figure 7. Comparisons between our tracker with other state-of-the-art trackers in challenging situations, which demonstrates our tracker is robust to partial occlusion and deformations.

IEEE International Conference on, pages 263–270. IEEE, 2011. 5

[9] 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*, pages 702–715. Springer, 2012. 5

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):583–596, 2015. 5

[11] 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*, pages 1822–1829. IEEE, 2012. 5

[12] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56. IEEE, 2010. 5

[13] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1208–1215. IEEE, 2009. 2

[14] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. IEEE, 2011. 2

[15] G. Nebehay and R. Pflugfelder. Clustering of static-adaptive correspondences for deformable object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2784–2791, 2015. 2

[16] S. S. Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2

[17] H. Possegger, T. Mauthner, and H. Bischof. In defense of color-based model-free tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2113–2120, 2015. 1, 2, 3, 5

[18] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[19] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 5

[20] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014. 1

[21] T. Vojíř and J. Matas. The enhanced flock of trackers. In *Registration and Recognition in Images and Videos*, pages 113–136. Springer, 2014. 5

[22] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia. Understanding and diagnosing visual tracking systems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3101–3109, 2015. 1

[23] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *2011 International Conference on Computer Vision*, pages 1323–1330. IEEE, 2011. 2

[24] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 5

[25] 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*, pages 1838–1845. IEEE, 2012. 5