Robust and fast Tracking-Learning-Detection

Shuai Cheng, Guangwen Liu
School of Electronic Information Engineering
Changchun University of Science and Technology
Changchun, China
lgwen_2003@126.com

Junxi Sun
School of Computer Science and information Technology
Northeast Normal University
Changchun, China

***Abstract*—To improve the robust and processing speed of the Tracking-Learning-Detection(TLD), the robust and fast TLD tracker is proposed. Replacing with the forward-backward error predictor, The two powerful failure predictors, including the neighbourhood consistency predictor and the markov predictor in the tracker, are used to reduce the computational cost and improve the precise. The RANSAC algorithm is added to estimate the global motion model and improve the success rate of tracking. Replacing with P-N learning in sample learning procedure, We use a novel online weighted P-N learning which integrates the sample importance into an efficient online learning procedure to alleviate drift to some extent. Experimental results on various benchmark video sequences demonstrate the superior performance of the proposed algorithm to state-of-the-art tracking algorithms in robustness, stability and efficiency.**

***Keywords*—Tracking-Learning-Detection; Failure predictors; RANSAC; weighted P-N learning**

I. INTRODUCTION

Object tracking is one of the most important components in a wide range of applications in computer vision, such as surveillance, behavioral recognition. Object tracking remains a very challenging problem. Numerous factors affect the performance of a tracking algorithm, such as appearance change, illumination variation, occlusion, as well as background clutters[1].

Numerous algorithms have been proposed with focus on effective appearance models, which can be categorized into generative algorithm[2-3] and discriminative algorithm[4-6]. A generative tracking method learns an appearance model to represent the target and search for image regions with best matching scores as the results. Discriminative methods treat tracking as a binary classification problem which learns to explicitly distinguish the object being tracked from its background.

Kalal et al. design a novel Tracking-Learning-Detection(TLD) [7] framework that decomposes the long-term tracking task into three sub-tasks: tracking, learning and detection. The flock of tracker(FoT) follows the object from frame to frame. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The P-N learning estimates detector's errors by two types of "experts" and updates it to avoid these errors in the future.

However, the performance of the predictor based on NCC and FB[8] is not robustness and high computational cost for FoT. The P-N learning does not take into account any information about the importance of the training samples that determinate the label of sample, therefore, may lead to wrong label. Such classifier may be inaccurate when the training samples are imprecise which causes drift.

In this paper, we propose a novel improved TLD tracker (ITLD) to improve the performance of TLD. Combined with NCC predictor, We add two new predictors of local tracker failure-the neighbourhood consistency predictor and the markov predictor to replace with the FB predictor to increase the robustness and reduce the computational cost. The RANSAC algorithm estimates the global motion model for local trackers to alleviate drift to some extent. The improved TLD tracker integrates the sample weight into the classification procedure to decide the label of sample. The method improves the accuracy of classification and the performance of the classifier trained by the correct label training sample.

II. ITLD

The ITLD framework is designed and added some new character based on TLD framework. The components of the framework are characterized as follows. Its block diagram is shown in Fig. 1. Red font models is different with ones of TLD.

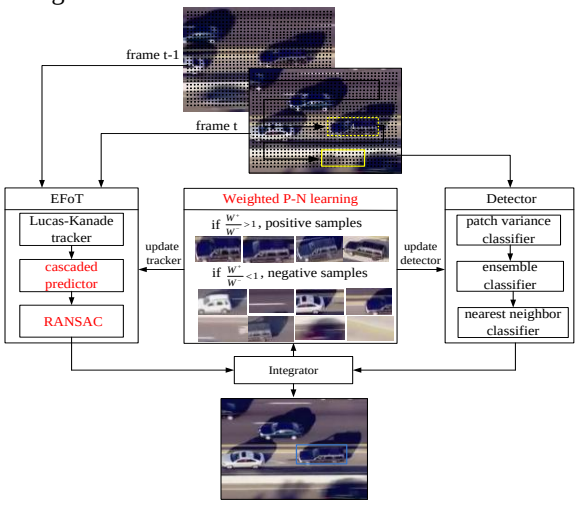


Fig. 1. Block diagram of ITLD.

Enhanced Flock of Tracker(EFoT): the tracking component is based on Median-Flow tracker extended with failure detection[8]. Combined with NCC predictor, we add the neighbourhood consistency predictor and the markov predictor.

The three predictors compose of the cascaded predictor. We introduce the RANSAC algorithm to estimator the object motion.

Detector: the detector scans the input image by a scanning-window and each patch decides about presence or absence of the object by cascaded classifier. We structure the cascaded classifier into three stages: patch variance, ensemble classifier and nearest neighbor classifier.

Weighted P-N learning: the task of the learning component is to initialize the object detector in the first frame and update the detector in run-time using the P-expert and the N-expert[7]. We assigns weights to each sample in training sample set to improve discriminative power of classifier by decreasing classification errors and increasing the accuracy of tracker.

Integrator: it combines the bounding box of the tracker and the bounding boxes of the detector into a single bounding box output. If neither the tracker not the detector output a bounding box, the object is declared as not visible. Otherwise the integrator outputs the maximally confident bounding box.

III. EFoT

Based on FoT, the EFoT adds two new model: the cascaded predictor, and the RANSAC. In this section, we present our design for describe detail each module.

A. The Neighbourhood Consistency Predictor

The neighbourhood consistency predictor[9] assumes that the motion of neighbouring local trackers is often very similar, whereas a failing local tracker returns a random displacement. The neighbourhood consistency predictor is implemented as follows in Fig. 2.

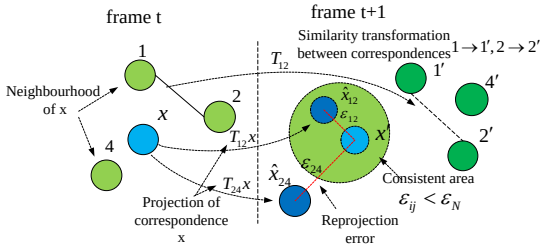


Fig. 2. Neighbourhood score computation for two pairs of correspondences.

Each unique pair of correspondences (green) $i, j \in 1,2,3,4$ generate a similarity transformation T_{ij}. The tested (blue) correspondence x is transform by the estimated similarities and the reprojection error $\varepsilon_{ij} = \left\| \hat{x}_{ij} - x' \right\|^2$ is computed. The final score is the number of $\varepsilon_{ij} < \varepsilon_N$ (number of $\hat{x}_{ij}$ points inside green circle around x'). A set of neighbouring local trackers N is defined for x.

We definite the neighbourhood consistency scoring functions given in (1). A local tracker is defined to be consistent if $S^N \geq \theta$, where θ is a threshold for this predictor.

$$S^N = \frac{1}{Z} \sum_{j,i \in N} \left[\left\| T_{ji} x - x' \right\|^2 < \varepsilon_N \right] \qquad (1)$$

Where, $[\cdot]$ is Indicator function.

B. The Markov Predictor

The markov predictor[9] is based on the model of the past performance of a local tracker bound to a region specified by object coordinate frame. The model is in the form of a markov chain with two states, $s_t \in \{0,1\}$, depicted in Fig. 3.

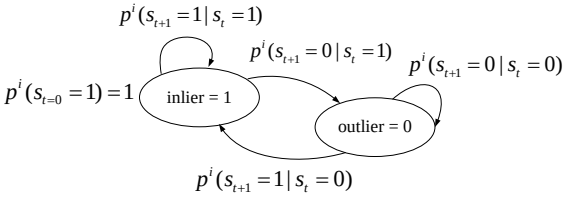


Fig. 3. The state diagram of the Markov chain for the local tracker.

This is the state diagram of the markov chain in the form of a two-state probabilistic automaton with transition probabilities p^i, where i identifies the local tracker and initial state $s_{t=0}=1$.

The prediction that certain local tracker would be an tentative inlier (or an outlier) is done according to (2).

$$\begin{bmatrix} p^i(s_{t+1}=1) \\ p^i(s_{t+1}=0) \end{bmatrix} = T_t^i \cdot \begin{bmatrix} p^i(s_t=1) \\ p^i(s_t=0) \end{bmatrix} \qquad (2)$$

Where, transition matrix T_t^i described in (3), $p^i(s_t=1) \in \{0,1\}$ is binary and depends on the previous state (inlier/outlier) of the i th local tracker. The left side of (2) are then probabilities that next state would be inlier (outlier).

$$T_t^i = \begin{bmatrix} p^i(s_{t+1}=1 \mid s_t=1) & p^i(s_{t+1}=1 \mid s_t=0) \\ p^i(s_{t+1}=0 \mid s_t=1) & p^i(s_{t+1}=0 \mid s_t=0) \end{bmatrix} \qquad (3)$$

C. RANSAC

We use RANSAC algorithm[10] for model estimation to solve the drifting of the tracker. The RANSAC select the hypothesis set most likely to lead to a good model at each time. Therefore, to improve the precision of the tracker and solve the problem of drift, the RANSAC estimates the object motion model of the local trackers got by the cascaded predictor with negligible extra computational cost.

IV. Weighted P-N Learning

P-N learning[7] is a semi-supervised online learning algorithm. During training process, the classifier may identify the unlabeled data with wrong labels. It can degrade discriminative performance of classifier and therefore lower the accuracy of tracking. To improve the accuracy and robustness of the classifier, we use a weighted P-N learning[11] by assigning weight to each sample in training set. Each sample in training set has two categories of weight which are termed P-weight W^+ and N-weight W^-. P-weight represents the probability of being a positive sample, and N-weight represents the probability of being a negative sample. The W^+ and W^- of sample can be then obtained by the following formulation:

$$W^+ = W_i^+ + W_c^+$$
$$W^- = W_i^- + W_c^- \quad (4)$$

Where, W_i define weight in the iteration process. Besides, the probability of the sample in training set obtained by classifier is defined as the classification weights W_c .

In iteration process, a sample from training set is represented as a positive sample for C^+ times and as a negative sample for C^- times. The positive weight W_i^+ and the negative weight W_i^- are determined by the following formulation:

$$W_i^+ = C^+ / C^+ + C^-$$
$$W_i^- = C^- / C^+ + C^- \quad (5)$$

Based on scanning-window, each input sub-window is represented by the feature vector x . The randomized forest classifier is adopted to obtain the posterior probability $P(y=1|x)$. The following formulation is defined as the classification weights:

$$W_c^+ = P(y=1|x)$$
$$W_c^- = 1 - P(y=1|x) \quad (6)$$

At last, the sample is determined to be either positive or negative via the following formulation:

$$W^+ / W^- \geq 1, \quad positive\ sample$$
$$W^+ / W^- < 1, \quad negative\ sample \quad (7)$$

V. Experimental Results

This section reports on a set of quantitative and qualitative experiments comparing ITLD with relevant algorithms, the first two experiments evaluate our short-time tracker and learning compared with original TLD algorithm. ITLD is compared with results which reports on performance of 4 trackers on the challenging dataset.

A. Comparsion 1: FoT

This experiment compares various short-time trackers with the different predictor and the motion estimation on 4 sequences. For every sequence, we compare the T_{FB+NCC} tracker (the tracker of original TLD), T_C tracker (the tracker with cascaded predictor) and the T_R tracker (the tracker with cascaded predictor and motion estimation). The performance was accessed using the number of successfully tracked frames. The number of frames where overlap with a ground truth bounding box is larger than 50%. Frames where the object was occluded were not counted.

Table Ⅰ shows the results. Bold font means the best score. The scores of the T_{FB+NCC} tracker are shown in the 3rd column. In failure detection, Let d_i denote the displacement of a single point and d_m be the median displacement. A residual of a single displacement is then defined as $|d_i - d_m|$. A failure of the tracker is declared if median $|d_i - d_m| > 10\,pixels$. The score is lowest than other tracker, because the failure detection is not accurate.

The scores of the T_C tracker are displayed in the 4th column. The number of successfully tracked frames of the tracker was significantly increased for all sequences. This demonstrates the improvement of the tracker using cascaded predictor. The cascaded predictor detects the failure of local tracker using spatio-temporal information to improve the accurate.

The last columns of Table Ⅰ report the performance of T_R tracker. The score is higher than T_C. This demonstrates the improvement of the tracker using RANSAC. The cascaded predictor removes the local trackers that do not complete the correct tracking effectively to reduce the error of estimation for the global model. The RANSAC estimate the global model of tracker accurately to solve the problem of drift during tracking.

TABLE I. Number of successfully tracked frames

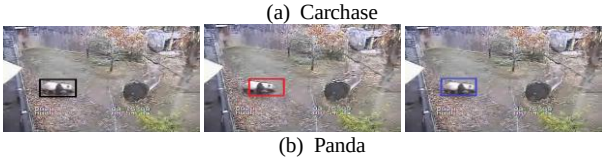
Sequence	T_{FB+NCC}	T_C	T_R
David	761	761	761
Jumping	40	81	124
Carchase	182	231	301
Panda	70	95	124
T(ms)	7.64	4.52	5.03

The speed was measured as the average time needed for frame-to-frame tracking. Table Ⅰ shows the results in the last row. For T_{FB+NCC}, the forward-backward procedure slows down tracking approximately by a factor of two, since the most time consuming part of the process, the Lucas-Kanade local optimization is run twice. With the added new failure predictors, T_C run the Lucas-Kanade local optimization once. T_C with much higher robustness to local tracker problems is achieved with negligible extra computational cost. Compared with T_C, T_R introduces the RANSAC algorithm extra, so the cost time is a little higher.

B. Comparsion 2: P-N learning

This experiment qualitatively evaluates weighted P-N learning and compares it with P-N learning. Fig. 4 shows the achieved results.

(a) Carchase



(b) Panda

Fig. 4. The tracking results with weighted P-N learning.

In Fig. 4, the left columns are the ground truth. The middle columns report tracking results with P-N learning which causes drift to some extent. Because it does not discriminatively consider the sample importance in its learning procedure. The right columns of tracking results are displayed based on weighted P-N learning, which integrates the sample importance into an efficient online learning procedure by positive weight

and negative weight is known when training the classifier. The weighted P-N learning improves the accuracy of classification and alleviates the drift.

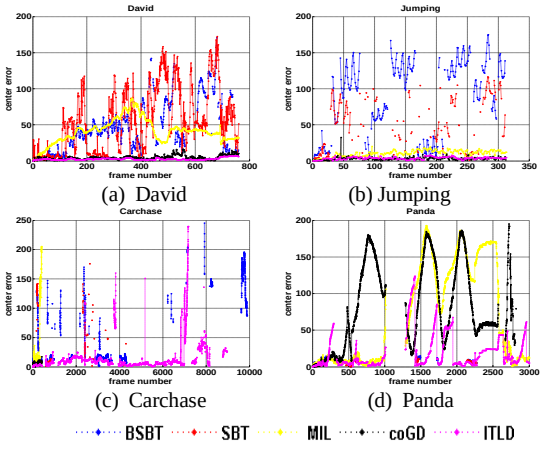
C. Comparsion 3: other trackers

We compare ITLD with some state-of-the-art trackers in this section. These trackers are: BSBT[4], SBT[5], MIL[6], CoGD[12]. The performance is evaluated using precision P. P is the number of true positives divided by number of all responses, A tracking was considered to be correct if its overlap with ground truth bounding box was larger than 50%. Table Ⅱ shows the achieved results. Bold numbers indicate the best score. ITLD scored best in 3/4 sequences.

TABLE II. AVERAGE PRECISION.

Sequence	BSBT	SBT	MIL	CoGD	ITLD
David	0.16	0.27	0.12	0.99	**1.00**
Jumping	0.16	0.14	0.37	**1.00**	**1.00**
Carchase	0.38	0.79	0.49	**0.92**	0.88
Panda	0.44	0.58	0.14	0.12	**0.59**
mean	0.39	0.69	0.38	0.71	**0.89**

Fig. 5 contains an illustration showing the achieved performance evaluated by central-pixel error(in pixel). In most cases, ITLD outperforms other tracker. Overall, it is the most stable tracker.



(a) David (b) Jumping (c) Carchase (d) Panda

Fig. 5. The comparison of central-pixel error

VI. CONCLUSION

This paper presents an improved TLD tracker for the short-time tracker and learning. The short-time tracker based on the cascaded predictor and the RANSAC algorithm outperforms ones based on the FB+NCC predictor such as speed, the number of correctly tracked sequences. An online weighted P-N learning is used to assign weight to each sample in training sample set, which can decrease classification errors and can improve the discriminative power of classifier to alleviate drift on all sequences. The improved TLD tracker was compared with state-of-the-art tracking algorithms and surpassed them in terms of precision and central-pixel error overall. Deep learning architectures is used to learn richer invariant features via multiple nonlinear transformations to obtain the image representations expressively to extend this work in the future.

ACKNOWLEDGMENT

This research has been supported by the National Natural Science Foundation of China (Grant No. 61172111), and Science and Technology Development of Jilin province(20090512, 20100312).

REFERENCES

[1] Y. Wu, J. Lim, and H. M. Yang, “Online Object Tracking: a Benchmark,” IEEE Conference on Computer Vision and Pattern Recognition, USA, 2013, pp. 2411-2418.

[2] D. Ross, J. Lim, R. Lin, and M. H. Yang, “Incremental Learning for Robust Visual Tracking,” International Journal of Computer Vision, vol. 77, no. 1, pp. 125–141, 2008.

[3] L. Sevilla-Lara and E. Learned-Miller, “Distribution Fields for Tracking,” IEEE Conference on Computer Vision and Pattern Recognition, USA, 2012, pp. 1910–1917.

[4] S. Stalder, H. Grabner, and L. V. Gool, “Beyond Semi-supervised Tracking: Tracking should be as simple as detection, but not simpler than recognition,” IEEE 12th International Conference on Computer Vision Workshops, Kyoto, 2009, pp. 1409–1416.

[5] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised On-line Boosting for Robust Tracking,” 10th European Conference on Computer Vision, France ,2008, pp. 234–247.

[6] 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.

[7] 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.

[8] Z. Kalal, K. Mikolajczyk, and J. Matas, “Forward-Backward Error: Automatic Detection of Tracking Failures,” 20th International Conference on Pattern Recognition, UK, 2010, pp. 23–26.

[9] T Vojir and J Matas, “The Enhanced Flock of Trackers,” Registration and Recognition in Images and Videos, vol.532, 2014, pp.113-136.

[10] M.A. Fischler and R.C. Bolles. “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.” Communications of the ACM, vol.24, no.6, pp.381–395, 1981.

[11] H F, J H Xiang, J Xu, and H H Liao. “Part-Based Visual Tracking via Online Weighted P-N Learning,” the Scientific World Journal, vol.5, no.1, pp. 24-37, 2014.

[12] Q. Yu, T. B. Dinh, and G. Medioni, “Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers,” European Conference on Computer Vision, France, 2008, pp. 678-691.