

A fast on-line boosting tracking algorithm based on cascade filter of multi-features

HU Song, SUN Shui-Fa^{*1}, MA Xian-Bing, QIN Yin-Shi, LEI Bang-Jun

Abstract. Exhaustive search in local area around the object is used for tracking in the conventional Haar-like feature based on-line boosting tracking methods (HBT). Tracking is frequently failed and can hardly find the object again when the object moves fast or occluded completely. In addition, on-line tracking will generate cumulative errors during updating when the state of the object and the environment changes. In this paper, we propose a fast on-line boosting tracking algorithm based on cascade filters of multi-features. Algorithm generates detection windows with various positions and sizes during initialization. Cascade detection screenings possible object windows by using multi-features: the manner of object motion, variance of detection windows, Object Similarity Statistic (OSS) model, color distribution and overlap ratio in turn. The left candidate object windows are passed to the strong classifier of on-line boosting and the object is located finally. In order to exert the advantage of boosting, naive Bayes classifier is used to replace the original simple threshold classifier in HBT. Experiment results on various videos show that the robustness, accuracy and real-time ability are improved.

Keywords: object tracking; on-line boosting; cascade detection; OSS model; color histogram

1 Introduction

Object tracking has been applied to various fields of production and living widely. A tracking system should realize real-time update of object states for a long-term tracking system. When object is completely occluded and appears again after a period of time, algorithm should have ability to detect object again [1-3]. Haar-like feature based on-line boosting tracking (HBT) [4] algorithm adjusts the structure of feature pool according to the performance of features during tracking. But algorithm efficiency will decrease and there will be over-fitting classification error [5].

¹ *Corresponding Author: SUN Shui-Fa(✉), e-mail: watersun1977@hotmail.com

HU Song, SUN Shui-Fa, MA Xian-Bing, QIN Yin-Shi, LEI Bang-Jun
Institute of Intelligent Vision and Image Information, China Three Gorges University, China

Conventional HBT algorithm [3, 6] only search object within a small region around last tracked windows. If object moves fast or is completely occluded, the tracking will be failed and it is difficult to detect object again.

During tracking, object state is confronted with various changes, such as changes in scale and shape, illumination change in the background, etc. Because fixed window size is used, when object's size becomes smaller, tracking window contains part background information and when object's size becomes larger, object's part information is missed. So cumulative error increases in learning and leads to a failed tracking eventually. Many tracking methods have taken scale factor into account [2, 7].

In this paper, we propose an on-line boosting tracking algorithm based on cascade filter of multi-features. Detection windows with different size and location get fewer and fewer by passing through a cascade of detection modules. The remaining detection windows are passed to boosting classifier to identify object finally. Features used include object motion mode, patch variance, Object Similarity Statistic (OSS) model, Weighted Color Histogram (WCH) model, etc. In addition, our method adopts Gentle AdaBoost to replace the original Discrete AdaBoost and uses weighted Bayes classifier to calculate results of weak classifiers.

2 Cascade detection and tracking

The flow diagram of the proposed algorithm based on detecting and filtering of detection windows is illustrated in figure 1. In the initialization, algorithm generates detection windows with various dimension scales referencing object size with scale factor taking into account. The advantages are that all detection windows are only generated once. Then a series of detecting and filtering are carried out to filter out large numbers of non-object windows. Finally, the improved HBT searches object from the remaining candidate windows. The concrete steps is listed as follows:

a. Filtering out detection windows that don't meet the size requirement according to object motion mode. Object that moves parallel to camera view usually has a small change in size and scale $S_{current} \in (0.8S_{last}, 1.2S_{last})$; object that moves vertically to camera view change greatly in size and scale $S_{current} \in (0.5S_{last}, 1.5S_{last})$.

b. Comparing variance of detection windows with variance of last tracked object and reserving windows whose variance change in the allowable range.

c. Comparing detection windows with Object Similarity Statistic (OSS) model of each pixel of last tracked object. When more than 50% pixels in detection window are considered to be object pixels, the detection window is reserved.

d. Comparing detection windows with Weighted Color Histogram (WCH) model of last object and when similarity is above the given threshold ($Threshold=0.90$), the detection window is reserved.

e. Calculating overlap rate between detection window and last object window and reserving windows that meet requirement ($Overlap > 50\%$) as the displacement of object between two neighbouring frames is not too big. If there doesn't exist windows that meet requirement, the remaining detection windows of previous step are passed to next stage.

f. Finding out object through strong classifier of boosting and updating variance threshold, OSS model, WCH model, detection windows' overlap rate with current object and strong classifier. If object is not found, every models and parameters are not to be updated. Partial object features and models involved in algorithm are detailed as follows.

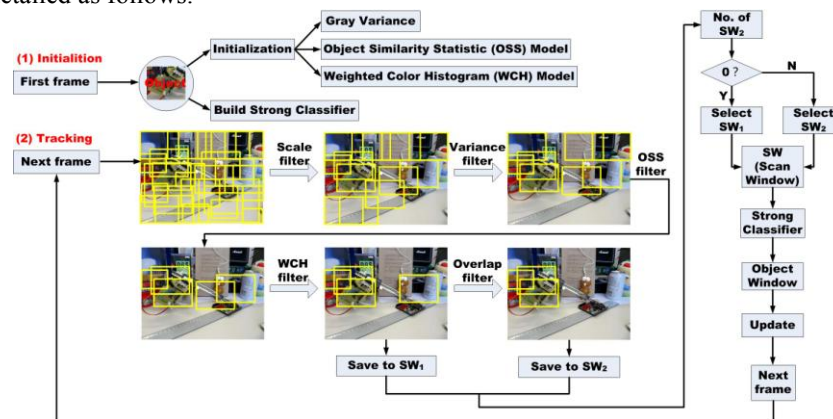


Fig. 1 Flow diagram of whole algorithm

2.1 Variance of detection window

Variance filter is the second stage of cascade. This stage rejects all detection windows, whose gray-value variance is smaller than 80% or bigger than 120% of tracked window. The stage exploits the fact that gray-value variance σ of a detection window p can be expressed as equation (1).

$$\sigma = E(p^2) - E^2(p) \quad (1)$$

The complexity of computation of the expected values $E(p^2)$ and $E(p)$ is reduced greatly by using integral image [8] because integral image is also used in calculating Haar-like feature value.

2.2 Object Similarity Statistic (OSS) model

Considering morphologic change of moving object and enhancing the robustness of window selecting, we propose Object Similarity Statistic (OSS) model and use it to filter windows. This idea comes from the ViBe background modeling method [9]. However, object pixel is modelled instead of the background pixel. Formally,

let $v(x)$ the value in a given Euclidean color space at x in the image, and v_i a object sample value with an index i . Each object pixel x is modelled by a collection of N object sample values. The OSS model $M_{OSS}(x)$ is illustrated in figure 2.

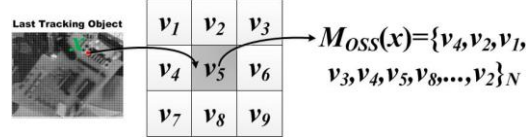


Fig. 2 OSS model of pixel

Algorithm classifies each pixel value $v(x)$ in detection window according to their corresponding OSS model and thus classifies detection window as candidate object window or not. In this paper, $N=20$, gray level difference threshold $R=20$, match sample threshold $C_{min}=2$, the number of match object pixels threshold $N_{min}=50\%N_{all}$. N_{all} is the total number of pixels in every detection windows. For more detail see ref. [9].

2.3 Weighted Color Histogram (WCH) model

In this paper, we use color feature to filter detection windows. We model object using Weighted Color Histogram (WCH) model [7]. Let $\mathbf{p}=(p_1, \dots, p_N)^T$ the WCH model of tracked object previously and $\mathbf{q}=(q_1, \dots, q_N)^T$ the candidate object. The equations (2) to (4) illustrate the process of update. The Bhattacharyya distance is calculated as Eq. (2), and is transformed to weight p_w with Eq. (3). During update, WCH model is updated only if the weight is above given threshold, as Eq. (4).

$$\rho = \sum_{i=1}^N \sqrt{p_i \cdot q_i} \quad (2)$$

$$p_w = e^{\frac{-(1-\rho)}{2 \cdot \sigma_0^2}} \quad (3)$$

$$\mathbf{p}_{t+1} = \begin{cases} (1-\lambda) \cdot \mathbf{p}_t + \lambda \cdot \mathbf{q}_t & \text{if } p_w > thr \\ \mathbf{p}_t & \text{else} \end{cases} \quad (4)$$

where σ_0 the scale parameter, λ update coefficient and thr update threshold. In this paper, $\sigma_0=0.1$, $\lambda=0.2$, $thr=0.9$.

2.4 Filtering detection window using overlap rate

Generally, there is overlapping portion between the last tracked object and the current object. Figure 3 shows the relationship between the moving speed and overlap rate. In order to take advantage of this, overlap rate detection module is added at the end of cascade detection and reserve detection windows that have an overlap rate above 50% with last tracked object window. If there is no detection window

that meets the requirement, then the moving speed of object is very fast and candidate windows of last stage are passed to the final on-line boosting module.

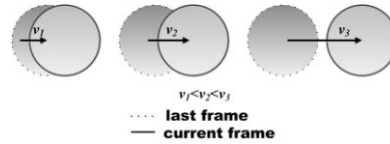


Fig. 3 Overlap between two sequential frames

3 Classifier building and updating

Boosting transforms a weak learning algorithm into a strong one. This is done by combining weighted weak classifiers which have been generated by repeating training with different subsets of training data. AdaBoost is a member of boosting family and was originally proposed by Freund and Schapire [10], namely Discrete AdaBoost. The output of Discrete AdaBoost's weak classifiers are $\{-1,+1\}$. There are several other AdaBoost algorithms, such as Real AdaBoost, Logit Boost, Gentle AdaBoost, etc. Gentle AdaBoost that is proved to be the best one among the above ones by Lienhart R. et al [11] is adopted in the proposed system to replaced the Discrete AdaBoost.

4 Experiments and analysis

The best module combination is found via comparative experiments. As computational complexity of WCH model is higher than OSS model, WCH module is after the stage of OSS module. Experiments with following combination of the cascade detections are carried out: OSS+WCH+Discrete (A), WCH+ WeightedBayes (B), OSS+WeightedBayes (C), OSS+WCH+WeightedBayes (D). A comparison with conventional HBT is also implemented. We benchmark our approach on the following sequences: *Board*, *Box*, *Lemming*, *Liquor* [12] and *BoBoT-A~F* [13]. These challenging video sequences include many main challenges, such as rotation, partial and full occlusion, scale changes, moving cam, etc.

By interpreting a frame as true positive when overlap rate between tracked object and ground truth exceeds 0.5. Table 1 shows the tracking results of A, B, C, D and HBT. Bold font means the best.

Table 1 Number of successfully tracked frames

Sequence	Size	TotalFrames	HBT	A	B	C	D
<i>Board</i>	640×480	697	187	381	645	548	649
<i>Box</i>	640×480	1160	431	184	586	829	930
<i>Lemming</i>	640×480	1335	666	1321	1328	505	1328
<i>Liquor</i>	640×480	1740	896	1094	1112	460	1278
<i>BoBoT-A</i>	320×240	601	601	417	568	186	601
<i>BoBoT-B</i>	320×240	628	628	623	628	463	628
<i>BoBoT-C</i>	320×240	403	403	394	199	403	403
<i>BoBoT-D</i>	320×240	946	946	946	946	451	946
<i>BoBoT-E</i>	320×240	304	304	304	304	304	304
<i>BoBoT-F</i>	320×240	452	270	396	320	150	402

D performs best in table 1 and HBT is not so good as D. The main reason is that HBT adopts local search strategy. Once tracking failed, object can hardly be detected again, as shown in Figure 4(a).



(a) Comparative result of HBT and D on sequence *BoBoT-F*



(b) Comparative result of A and D on sequence *Board*



(c) Comparative result of B and D on sequence *Box*



(d) Comparative result of C and D on sequence *BoBoT-B*

Fig. 4 Comparison of tracking performance (Green boxes represent D and blue boxes represent the other methods)

A performs not as good as weighted Bayes in most sequences when compared with D. It proves that the original outputs ($\{-1,+1\}$) of weak classifiers lead to accumulative error, as shown in Figure 4(b).

Performance of B is not as good as D in most sequences. Without the OSS detection module, number of detection windows passed to WCH module is very

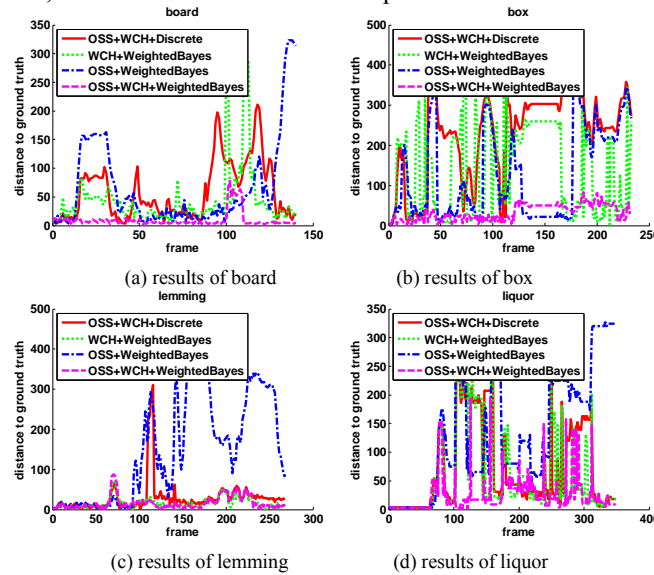


Fig. 5 Comparison of tracking errors

large. So background windows of similar color with object are easily misclassified. OSS model contains space information of pixels and can filter out detection windows which have big differences in spatial organization but have similar color statistics, as shown in Figure 4(c).

C performs not so good as D in most sequences. Without the WCH detection module, number of remaining windows passed by OSS module is still very large. The value of a two-rectangle Haar-like feature used by HBT is the difference between the sum of the pixels within two rectangular regions. So space information of each pixel is missed and background that has a similar pixel-block distribution with object is selected wrongly, as shown in Figure 4(d).

Figure 5(a)(b)(c)(d) are error graphs of sequence *Board*, *Box*, *Lemming* and *Liquor* tracked by approach A, B, C and D respectively. Error is the distance between center of tracked window and ground truth of object.

5 Conclusion and future work

In this paper, we improve the conventional HBT based on object detection. Algorithm filters out non-object detection windows through a cascade detection system and passes the remaining detection windows to boosting to decide the final object window. Cascade filters windows rough at first and then uses effective models in sequential stages to achieve real-time. During the process of window filter, scale,

variance, space distribution of gray level, color and overlap are all considered. It can detect object again after a period of occlusion.

Video surveillance is a very promising field today. Excellent tracking algorithms should be applied to real-life applications. Future work is to further improve robustness when there exists long time occlusion, complex background and similar interference.

Acknowledgments This study is funded by NSFC (No.61102155, No.61272237, No.61272236), OYMI Research Team Plan of Hubei Province of China (No.T201002).

References

1. Babenko B, Ming-Hsuan Y, Belongie S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, 33(8):1619-1632.
2. Kalal Z, Mikolajczyk K, Matas J. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, 34(7):1409-1422.
3. Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang. Real-Time Compressive Tracking. *Proceedings of the 12th European Conference on Computer Vision*, 2012:864-877.
4. Grabner H, Bischof H. On-line boosting and vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006:260-267.
5. Robert E. Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *The Annals of Statistics*, 1998, 26(5):1651-1686.
6. Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking. *Proceedings of the 10th European Conference on Computer Vision*, 2008:234-247.
7. Katja Nummiaro, Esther Koller-Meier, Luc Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 2003, 21(1):99-110.
8. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001:1511-1518.
9. Olivier Barnich, Marc Van Droogenbroeck. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. *IEEE Transactions on Image Processing*, 2011, 20(6):1709-1724.
10. Freund Y, Schapire R. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 1997, 55(1):119-139.
11. Lienhart R, Kuranov A, Pisarevsky V. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *Proceedings of the 25th DAGM Symposium*, 2003:297-304.
12. Santner J, Leistner C, Saffari A, et al. PROST:Parallel Robust Online Simple Tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010:723-730.
13. Dominik A. Klein, Dirk Schulz, Simone Frintrop, Armin B. Cremers. Adaptive Real-Time Video-Tracking for Arbitrary Objects. In *Proc. IROS*, 2010: 772-777.