electronics

Article

The Kernel Based Multiple Instances Learning Algorithm for Object Tracking

Tiwen Han [†], **Lijia Wang** [*,†,‡] and **Binbin Wen** [†]

Department of Intelligent Manufacture, Hebei College of Industry and Technology, Shijiazhuang 050091, China; hantiwen@163.com (T.H.); ge_wenbinbin@163.com (B.W.)
* Correspondence: wanglijia1981@hotmail.com; Tel.: +86-186-109-05016
† These authors contributed equally to this work.
‡ Current address: Hebei College of Industry and Technology, Shijiazhuang 050091, China.

Received: 24 April 2018; Accepted: 13 June 2018; Published: 16 June 2018

Abstract: To realize real time object tracking in complex environments, a kernel based MIL (KMIL) algorithm is proposed. The KMIL employs the Gaussian kernel function to deal with the inner product used in the weighted MIL (WMIL) algorithm. The method avoids computing the pos-likely-hood and neg-likely-hood many times, which results in a much faster tracker. To track an object with different motion, the searching areas for cropping the instances are varied according to the object's size. Furthermore, an adaptive classifier updating strategy is presented to handle with the occlusion, pose variations and illumination changes. A similar score range is defined with respect to two given thresholds and a similar score from the second frame. Then, the learning rate will be set to be a small value when a similar score is out of the range. In contrast, a big learning rate is used. Finally, we compare its performance with that of the state-of-art algorithms on several classical videos. The experimental results show that the presented KMIL algorithm is faster and robust to the partial occlusion, pose variations and illumination changes.

Keywords: object tracking; kernel based MIL algorithm; Gaussian kernel; adaptive classifier updating

1. Introduction

Object tracking is a fundamental task in the fields of surveillance, robotics, human computer interaction, and so on. Recently, researchers have proposed many successful algorithms. However, the problem is still challenging due to factors such as occlusions, appearance variations, abrupt motion, and illumination changes [1].

The existing object tracking algorithms can be mainly classified into two groups: the generative method and the discriminative method [2]. The generative object tracking is one of the important problems, which learns an object model in the first frame and detects the area with the most similar appearance in the successive frames [3]. The MS tracker [4], IVT tracker [5], and VTD tracker [6] are the famous generative object tracking algorithms.

The discriminative tracking method learns and updates a binary classifier by using online training. The tracking-by-detection algorithm stems directly from the discriminative methods, which trains a classifier online and finds the most likely location from many candidate image patches as tracking evolves [7–11]. The Online-Boosting tracker updates a classifier by considering the tracked result as the positive sample [10]. However, it often fails when the tracked results drift from the real object location. Then, an improved algorithm (semi-supervised tracker) is proposed by Gabor [11]. The algorithm labels the positive samples in the first frame. However, the ambiguity problem exists in the tracking process. Zhang [3] describes a real-time CT tracker by utilizing the compressed features for training a Bayes classifier. In recent years, the researchers focus mainly on studying the real-time object tracking algorithms. The correlation filter-based tracking algorithms are proposed and have provided excellent performance. The algorithms include MOSSE [12], CSK [13], KCF [14], DCF [15], CN [16], DAT [17], Staple [18], and DSST [19]. These trackers have shown the advantage of being computationally efficient, which is especial useful for real-time applications. The KCF [14] tracker generates samples by applying a circulate matrix, which speeds up the computing of matrixes. As a result, the algorithm runs at hundreds of FPS [14]. However, it often suffers from drift problem due to the occlusion, illumination changes, and pose variations. The deep learning algorithm has been widely studied in the fields of computer vision including object tracking. The DeepSRDCF [20], GOTURN [21], C-COT [22], and ECO [23] benefiting from big data for learning a net model are proposed for object tracking. The experimental results have addressed that the obtained convolution features have a powerful ability of representation. However, the processing of training networks is time consuming due to the complexity network. Normally, to realize real-time object tracking, the deep learning algorithm runs on GPU and can achieve about 100 FPS (e.g., the GOTURN tracker). However, the GOTURN operates at 2.7 FPS for the only CPU [21]. To overcome the ambiguity problem in the online-boosting algorithm, the MIL tracker-related algorithms are proposed to learn a strong classifier from multiple instances in the positive and negative bags [24,25]. The WMIL tracker [25] runs at about 14FPS on a single CPU. Moreover, the WMIL tracker performs well over the CF related trackers in terms of occlusion, illumination variations and pose changes.

In this paper, we propose a Kernel based MIL (KMIL) object tracking algorithm. To further reduce the computational cost, the Gaussian kernel function is presented for resolving the inner product used in the WMIL algorithm. The WMIL algorithm often fails to track the object with different speed. To deal with the problem, the searching areas for cropping the instances are varied according to the object's size. Finally, an adaptive classifier updating strategy is presented. The similar score of the tracking result in the second frame is remembered as a reference. Then, two thresholds are defined and a range is obtained with respect to the reference. As tracking evolves, the updating rate of the classifier is adjusted to suit for the appearance changes when the maximum similar score of the sample is out of the range at the current frame. At last, the proposed algorithm is compared with the state-of-art algorithms.

The paper is organized as follows. The Section 1 is the introduction of the paper. The WMIL tracker is detailed in the Section 2. The Section 3 addresses the KMIL tracker. The experimental results are shown in the Section 4. The Section 5 summaries the paper.

2. The WMIL Tracker

Babenko [24] proposed an online MIL Boosting method for visual tracking. It detects an object by maximizing the bag likelihood function. Consequently, it suffers from being time consuming because the bag probability and instance probability are computed many times before selecting a most discriminative weak classifier. To deal with this problem, Zhang [25] proposed an efficient online approach (WMIL algorithm) to approximately maximize the bag likelihood function. In the algorithm, "positive" and "negative" bags are extracted for training classifiers. The positive bag is constructed by using the instances extracted from a circle centered at the object's location, while the negative bag is obtained by cropping the instances from an annular region around the object's location. Then, a strong classifier is trained in the Online Boosting frame by using the positive and negative bags. In the successive frame, candidate samples are cropped around the object's position in the previous frame. At last, the trained classifier detects the candidate sample with the maximum score as the final result. Furthermore, to deal with the problem of occlusion, illumination changes, and pose variations, the classifier is updated by using the positive and negative bags constructed by cropping instances according to the tracked result in the current frame.

It is assumed that the location of each instance x is denoted as $l_t(x)$ and the location of the object is denoted as l_t^* in the t^{th} frame. The instances for constructing a positive bag X^+ are cropped as: $X^+ = \{x : ||l_t(x) - l_t^*|| < r\}$, r is the searching radius centered as the location l_t^*, while the instances for constructing a negative bag X^- are cropped from an annular region $X^- = \{x : r < ||l(x) - l_t^*|| < \beta\}$, r and β $(r < \beta)$ are the radius of the annular region. In the Online Boosting frame, the algorithm trains K weak classifiers $\phi = \{h_1, h_2, \cdots, h_K\}$, which is defined as:

$$h_k(x) = In\frac{p(v_k(x)|y=1)}{p(v_k(x)|y=0)} \tag{1}$$

where $v(x) = (v_1(x), \cdots, v_k(x), \cdots, v_K(x))^T$ is a feature vector function of the instances. y_i is the label of the bag X_i. $y = 1$ means the bag is positive, while $y = 0$ means the bag is negative.

Then, M discriminative weak classifiers are selected to generate a strong classifier:

$$h(\cdot) = \sum_{k=1}^{M} h_k(\cdot) \tag{2}$$

The learned strong classifier detects an area with the maximum similar score as the final tracking location :

$$l_t^* = l(\arg\max_{x \in X^s} p(y|x)) \tag{3}$$

After detecting an object, the weak classifiers are updated according to the new location with a constant learning rate:

$$\begin{gathered} \mu_i^1 = \lambda\mu_i^1 + (1-\lambda)\bar{\mu} \\ \sigma_i^1 = \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)\frac{1}{N}\sum_{j=0|y_i=1}^{N-1}(v_k(x_{ij}) - \bar{\mu})^2 + \lambda(1-\lambda)(\mu_i^1 - \bar{\mu})^2} \end{gathered} \tag{4}$$

where λ is the learning rate. $\bar{\mu} = \frac{1}{N}\sum_{j=0|y_i=1}^{N-1}(v_k(x_{ij}))$ is the average of the kth feature $v_k(x_{ij})$ of the instances in the positive bag extracted around the tracking result at current frame.

3. The KMIL Object Tracking System

This section details the presented KMIL object tracking algorithm illustrated in Figure 1. Postive and negative bags are extracted around the object's location. Then, a strong classifier is learned from the first frame by using the online boosting WMIL algorithm [25], and the object image is saved as a reference. In the successive frame, the classifier is used to detect the most similar sample as the tracking result. Finally, the classifiers are updated according to the reference frame and current tracking result.

In the WMIL algorithm, the inner product is presented for selecting a weak classifier with the most discriminative ability, which reduces the computational time by avoiding computing the bag probability and instance probability many times [25]. However, the inner product is also computed M times, which is also time consuming. Recently, the kernel based approaches have been proposed for real time object tracking [26]. Inspired by the ideas in the WMIL [25] and DLSSVM [26] algorithms, we present a kernel based inner product method to select the most discriminative weak classifiers to further reduce the computational complexity.

As tracking evolves, the sample with the maximum similar score in the candidate area is detected by using the learned classifier. Normally, we assume that the object moves with the same speed and appearances around the object location from the previous frame. Therefore, the candidate samples are extracted in a fixed circle around the previous tracking location. To account for the size of an object, we change the circle adaptively. After tracking an object, the weak classifiers are updated with the new cropped samples for handling the appearance changes. Normally, a constant learning rate is used, which may lead to "over-updating" or "less-updating". To further handle these problems, an adaptive weak classifiers strategy are presented.

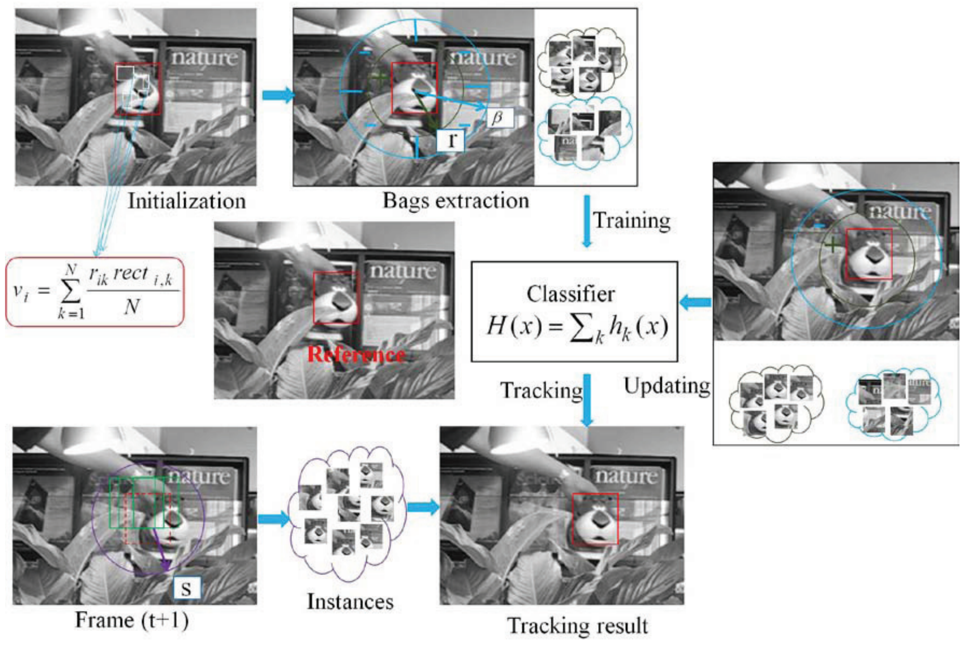

Figure 1. The flow chart of the Kernel based MIL algorithm.

3.1. The Kernel Based MIL Tracker

We assume that there are N instances in the positive bag and L instances in the negative bag. The tracking location at current frame is denoted as l_{10}. The positive bag and negative bag $\{X^+, X^-\}$ are considered as the training data. Different from the method of computing bag probability used in the WMIL algorithm, we compute the bag probability with respect to the included instances' probability.

$$p(y = 1|X^+) = \frac{1}{N} \sum_{j=0}^{N-1} p(y = 1|x_{1j}) \tag{5}$$

The method means that each instance contributes equally to the bag probability according to the including instances probability. Then, the bag probability mainly depends on the instances with higher probability. Especially in the case of tracking drift, the instances near the real object but far away from the center of the previous location contribute more to the bag probability.

Similar to the positive bag, the probability of the negative bag is computed with respect to the including instances equally.

$$p(y = 1|X^-) = \frac{1}{L} \sum_{j=N}^{N+L-1} p(y = 0|x_{0j}) \tag{6}$$

Similar to the WMIL tracker, our KMIL tracker trains weak classifiers $\phi = \{h_1, h_2, \cdots, h_K\}$, and selects the most discriminative weak classifiers by using an efficient criterion [25].

$$h_k = \underset{h \in \phi}{\arg\max} < h, \nabla \eta(H) > |_{H=H_{k-1}} \tag{7}$$

where h is the weak classifier in the classifier pool and H is the learned strong classifier constructed by $K - 1$ selected weak classifiers.

The inner product is computed as:

$$< h, \nabla \eta(H) >= \frac{1}{N+L} \sum_{j=0}^{N+L-1} h(x_{ij} \nabla \eta(H)(x_{ij})). \tag{8}$$

$$\eta = \sum_{s=0}^{1}(y_s log(p(y=1|X^+)) + (1-y_s)log(p(y=0|X^-))) \tag{9}$$

$$\eta(H) = \sum_{s=0}^{1}(y_s log(\sum_{j=0}^{N-1} p(y=1|x_{1j})) + (1-y_s)log(\sum_{j=N}^{N+L-1}(1-p(y=1|x_{0j})))) \tag{10}$$

$$\begin{aligned}\eta(H)(x_{ij}) &= y_i \frac{w_{j0}\sigma(H(x_{ij}))(1-\sigma(H(x_{ij}))}{\sum\limits_{m=0}^{N-1} w_{j0}\sigma(H(x_{im}))} \\ &= -(1-y_i)\frac{\sigma(H(x_{ij}))(1-\sigma(H(x_{ij}))}{\sum\limits_{m=N}^{N+L-1}(1-\sigma(H(x_{im}))}\end{aligned} \tag{11}$$

where $\sigma(z) = \frac{1}{(1+e^{-z})}$ is the sigmoid function and $\sigma(H(x_{ij}))$ is the jth instance probability in the ith bag.

The results from the WMIL tracker [25] shown that the criterion is efficient because it can avoid computing the bag probability and instance probability K times before selecting a weak classifier. Therefore, it is more efficient than the log-likelihood function used in the MIL tracker [25].

However, there is higher dimension computing in the inner product $\frac{1}{N+L}\sum\limits_{j=0}^{N+L-1} h(x_{ij}\nabla\eta(H)(x_{ij}))$, which is also time consuming. To handle with the problem, we use the kernel function. The inputs h and $\nabla\eta(H)$ are mapped to the feature space by using $\phi(h)$ and $\phi(\nabla\eta(H))$, where $\phi(\cdot)$ is the Hilbert mapping of the inputs. The kernel is defined as:

$$k(h, \nabla\eta(H)) = \phi(h)^T\phi(\nabla\eta(H)) \tag{12}$$

In practice, we choose to use the Gaussian kernel [26]:

$$k(h, \nabla\eta(H)) = exp(\frac{-||h-\nabla\eta(H)||^2}{\rho^2}) \tag{13}$$

where ρ is the bandwithd of the Gaussian function.

As new frames come, candidate samples are extracted around the tracking result: $X^s = \{x : \|l_{t+1}(x) - l_t^*\| < s\}$, where $s(r < s < \beta)$ is the radius.Then,the learned strong classifier detects the sample with the maximum similar score as tracking result. In the WMIL, MIL, and CT trackers, it is assumed that the object moves around the previous tracking location. And the candidate samples are cropped in a fixed circle. These trackers have shown continuous performance in term of accuracy on tracking large object with continuous motion. However, they often fail to track a small object because of their fast motion. To deal with the problem, we present a method to vary the radius for extracting the candidate sample with respect to the target's size. The radius is set to be 25 if the object is big. On the contrary, the radius is 35 for tracking the small object.

3.2. The Classifiers Update Strategy

After detecting an object, the classifier is updated to deal with the problems of occlusion, pose variations, and illumination changes. Normally, a learning rate is set to make a balance between the previous frame and the current frame [25]. We have tried the updating method with a fixed learning rate and found that the experimental results are unstable when there are appearance variations. With a small learning rate, the parameters of the classifier will be updated mainly with the mean and variance of the new tracked location's features. As a result, the interference from background will be introduced to update the classifier, which results in "over-updating". If the learning rate is too large, the new tracked area will affect the parameters of the classifier rarely. Then, the classifier will be "less-updating" and can't deal with the illumination changes and pose variations.

To address the problem mentioned above, we present an adaptive classifier updating strategy. From experimental results of the WMIL tracker, we found that the similar scores of the tracking results vary frequently from the tracked locations in the beginning frames. Therefore, we define the similar score S_0 of the tracking location in the second frame as a reference. Two thresholds H_1 and H_2 are also defined. Then, a similar score range $(S_0 - H_1, S_0 + H_2)$ is obtained with respect to the two thresholds and the reference. As tracking evolves, we consider the tracking results with the similar score outside the defined range as the appearance changes case. Then, a large learning rate is defined for updating the parameters of the classifier. If the similar score of tracking results are within the defined range, we use a small learning rate to update the classifier. The learning rate is defined as follows:

$$r = \begin{cases} 0.25 \text{ if } s \in (S_0 - H_1, S_0 + H_2) \\ 0.85 \text{ if } s \notin (S_0 - H_1, S_0 + H_2) \end{cases} \tag{14}$$

4. Experiments

We compared KMIL tracker with the state-of-art object tracking algorithms, such as MIL [7], CT [3], WMIL [25], KCF [14], and DSST [19]. The binary code released by the authors are used for testing these trackers. The videos "Tiger2", "Lemming", "Shaking", "Deer", "Sylvester", "Faceocc1", "Tiger1", and "Football1" from the OTB25 database are downloaded from the Network for testing these trackers. There are illumination changes and pose variations in the "Tiger1", "Tiger2", "Sylvester" and "Lemming" sequences. The serious occlusions exist in the sequences "Occluded face", "Football1", "Shaking" and "Lemming" videos. The objects move fast and vary their pose frequently in the "Tiger2", "Tiger1", and "Lemming" sequences. All the algorithms are implemented in the MATLAB and run on a core 2 CPU, 2.33GHz and 2GB RAM computer.

4.1. Parameters Setting

In these algorithms, the parameters r, α, and β determine the instances in the positive and negative bags. The s determines the area for cropping the candidate samples. The algorithms with bigger r, α, β, s can extract more instances which make the algorithms perform well to track an object, but result in time consuming. The K is the number of the weak classifiers, while M is that of the selected discriminative weak classifier for constructing a strong classifier. The classifier with bigger K and M can discriminate an object easily, which also lead to computing complexity. The parameters are set to be the same as the presented papers [3,7,19,25], which are illustrated in Table 1. The results of these algorithms have shown that these algorithms perform the best with the parameters. In the KMIL tracker, the radius for extracting candidate samples is set to be 25. In contrast, the radius is 35. Different from the fixed learning rate of the CT, MIL, WMIL trackers, the learning rate of the KMIL tracker is adapted according to the maximum score of the candidate sample at current frame. When the maximum score is in a given range, the learning rate is set to be 0.25, or else it is 0.85. The number of weak classifiers is 150, while that of the selected discriminative weak classifiers is 50. The KCF tracker employs the HOG feature and Gaussian kernel. In the first frame, a model is learned with the image patch centered at the initial position. In the successive frames, the position with the maximum value is detected over the candidate patch. Finally, a new model is trained at the new tracking position [14].

The CT, MIL, WMIL, KMIL, KCF, DSST trackers are implemented on the mentioned videos. After training the classifiers, the area with the maximum similar score is considered as the tracking result. Then, the classifiers are updated to overcome the drawbacks of occlusion, pose variations, and illumination variations.

Table 1. The parameters for the MIL, CT, WMIL and KMIL trackers. r is the radius for positive bag, α and β are the radius for the negative bag, s is for the searching area, K is the number of the weak classifiers, M is the number of the selected most discriminative weak classifiers. λ is the learning rate.

Parameters	r	α	β	s	K	M	λ
CT	4	8	30	20	/	/	0.85
MIL	4	/	50	35	250	50	0.85
WMIL	4	$\alpha = 2r$	$\beta = 1.5s$	25	150	15	0.85
KMIL(big object)	4	$\alpha = 2r$	$\beta = 1.5s$	25	150	15	0.25/0.85
KMIL(small object)	4	$\alpha = 2\mathrm{r}$	$\beta = 1.5s$	35	150	50	0.25/0.85

4.2. Tracking Object Location

Here, we detail the tracking object locations of the above trackers evaluated on the eight classical videos. The results are shown in Figure 2. The MIL tracker is time consuming because of its classifier selecting strategy. As a result, it often suffers from failure for the long time object tracking. The KCF tracker uses the circulate matrix and kernel function for completing real time object tracking. However, its searching area and learning rate are constant. Therefore, it often fails to track the object in the complex environment, especially when the object is small and moves fast. The WMIL and CT trackers update the classifier with a constant learning rate. Therefore, they result in tracking drift in the complex environment. Furthermore, the WMIL tracker computes the bag probability according to instances' distance. Consequently, the tracking drift will be aggregated. Benefiting from the constant learning rate, adaptive searching radius and the bag probability, the KMIL tracker can deal with the drift problem in the complex environment. The tracking object locations in Figure 2 demonstrate that the KMIL tracker performs well over the CT, MIL, WMIL, KCF, DSST trackers.

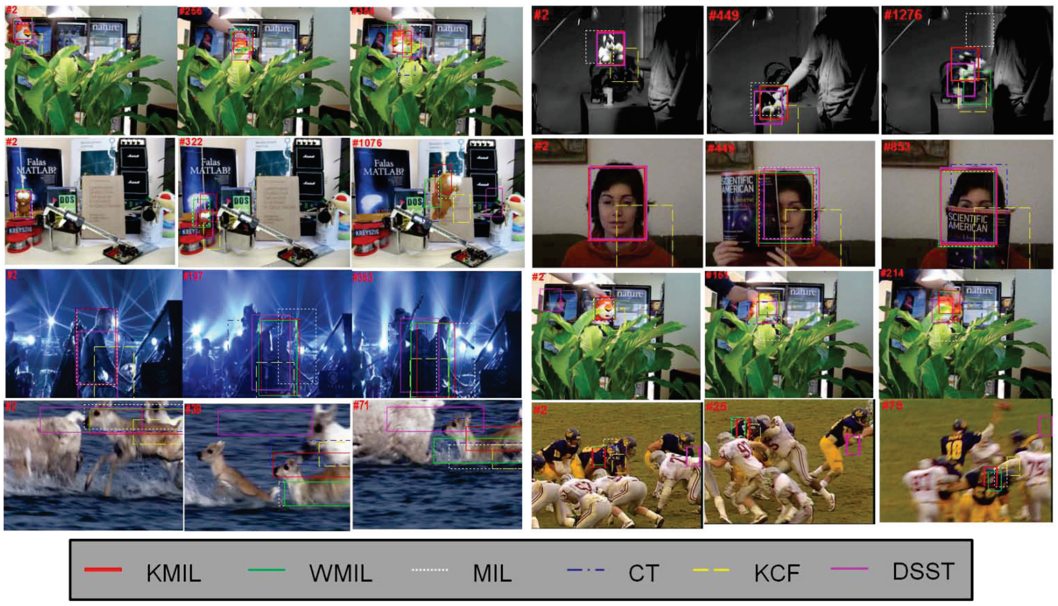

Figure 2. The illustration of the tracking locations on the sequences: "Deer", "Tiger2", "Faceocc2", "Sylvester", "Football1", "Shaking", "Tiger1", "Lemming".

4.3. Quantitative Analysis

We use the precision curves to evaluate the performance of the proposed algorithm. The precision curves illustrate the percentage of correctly tracked frames for a range of distance thresholds [14]. The correctly tracked frame is the one with target center within a distance threshold of the ground truth. The tracker with a higher precision at low threshold is more accurate. Similar to the previous works [7,13,27], we choose the 20 pixels as the threshold. The experimental results are shown in Figure 3. From the experimental results, we found that the precision of the KMIL tracker is higher than the other algorithms at 20 pixels for the "Tiger2", "Lemming", "Deer", "Slyvery", and "Faceocc1" sequences. For the "Tiger1" sequence, the KMIL tracker achieves the second higher precision at the 20 pixels. The MIL tracker has the highest precision for tracking the "Football man" shown in the Figure 3h at 20 pixels. All of the algorithms can achieve 1 precision at 35 pixels. The experimental results have shown the effectiveness of the proposed KMIL tracker. Especially, the KMIL tracker is more efficient in term of precision than the popular kernel based trackers KCF and DSST.

Another performance criteria we chose for our evaluation is the success plot. In the tracking process, the tracking result is denoted as a bounding box (called a) and the real position is the ground truth (called b). Then, an overlap score is defined with respect to the two regions.

$$OS = \left| \frac{a \bigcap b}{a \bigcup b} \right| \tag{15}$$

where $\bigcap$ and $\bigcup$ mean the intersection and union, respectively. $|\cdot|$ counts the number of the pixels. The tracked targets with the overlap score larger than the given threshold (0.5 is used) are considered as the successful results. The success curve is the ratios of the success frames to the whole frames. The experimental results are shown in Figure 4. The higher the success curve is, the stronger tracking ability the tracker has. It shows that the "red" line obtained by using the KMIL tracker on the sequences: "Tiger2", "Lemming", "Shaking", "Deer", "Sylvester", "Faceocc1", and "Tiger1" is higher than other line. In the Figure 4h, the higher success curve is obtained by using the MIL tracker on the "Football1" sequences. However, it is time consuming, which will be detailed in the next section. Furthermore, the well known fast KCF and DSST trackers run at a low success rate especially for tracking the small object (e.g., on the "Tiger1" sequence).

4.4. Computational Cost

This section details the computational cost of the above trackers. The average computing time processing an image is defined as: $t_{avr} = \frac{t_{all}}{N_{fra}}$. t_{all} is the total computing time processing all of the images in the whole video sequence. N_{fra} is the total number of frames. t_{avr} is the obtained average computing time. There are four factors lead to computational cost in the CT, MIL, WMIL, and KMIL trackers. The first factor is the total number of weak classifiers in the classifier pool. The number of the selected discriminative weak classifiers is the second factor influencing the computational time. The number of the instances due to big searching area also results in computational complexity. At last, the method for selecting discriminative weak classifiers which computes high dimension matrix is also time consuming. We did experiments by using the parameters in the Table 1. The Frames Per Second (FPS) of all the trackers are illustrated in Table 2. The KCF and DSST trackers run faster than the MIL, CT, WMIL, and KMIL trackers. However, it has low precision and success rate. Benefiting from the kernel function, the KMIL tracker avoids computing the $(h\left(x_{ij}\right) \bigtriangledown \eta\left(H\right)\left(x_{ij}\right))M$ times. As a result, it is with lower computational time than the MIL, WMIL trackers.

Table 2. The FPS (Frames Per Second) for different algorithms conducted on sequences "Tiger2", "Lemming", "Shaking", "Deer", "Sylvester", "Faceocc1", "Tiger1", "Football1".

Video Clip	MIL	CT	WMIL	KCF	KMIL	DSST
Tiger2	3.52	10.14	7.34	54.33	9.96	260
Lemming	3.13	9.41	8.96	35.73	9.98	103
Shaking	3.38	13.03	14.69	30.12	18.14	279
Animal	3.52	11.40	8.87	28.76	10.25	479
Sylvester	3.65	13.32	7.98	42.31	14.37	137
Faceocc2	3.46	13.21	13.85	38.28	17.70	260
Tiger1	3.02	10.4	7.93	10.94	9.22	265
Football1	3.73	13.59	8.77	224	13.04	500

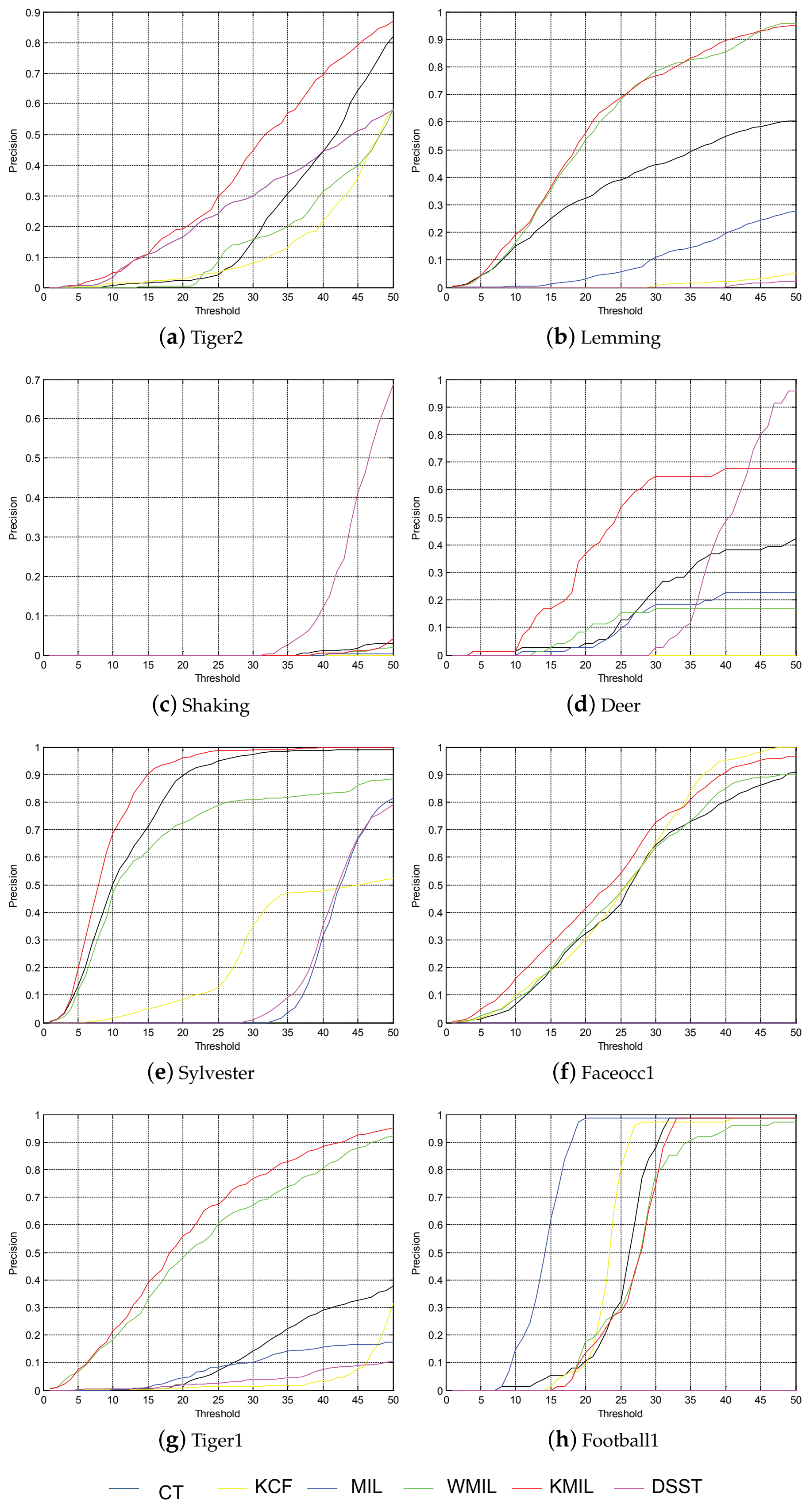

Figure 3. Precision plot for sequences: "Tiger2", "Lemming", "Shaking", "Deer", "Sylvester", "Faceocc1", "Tiger1", "Football1".

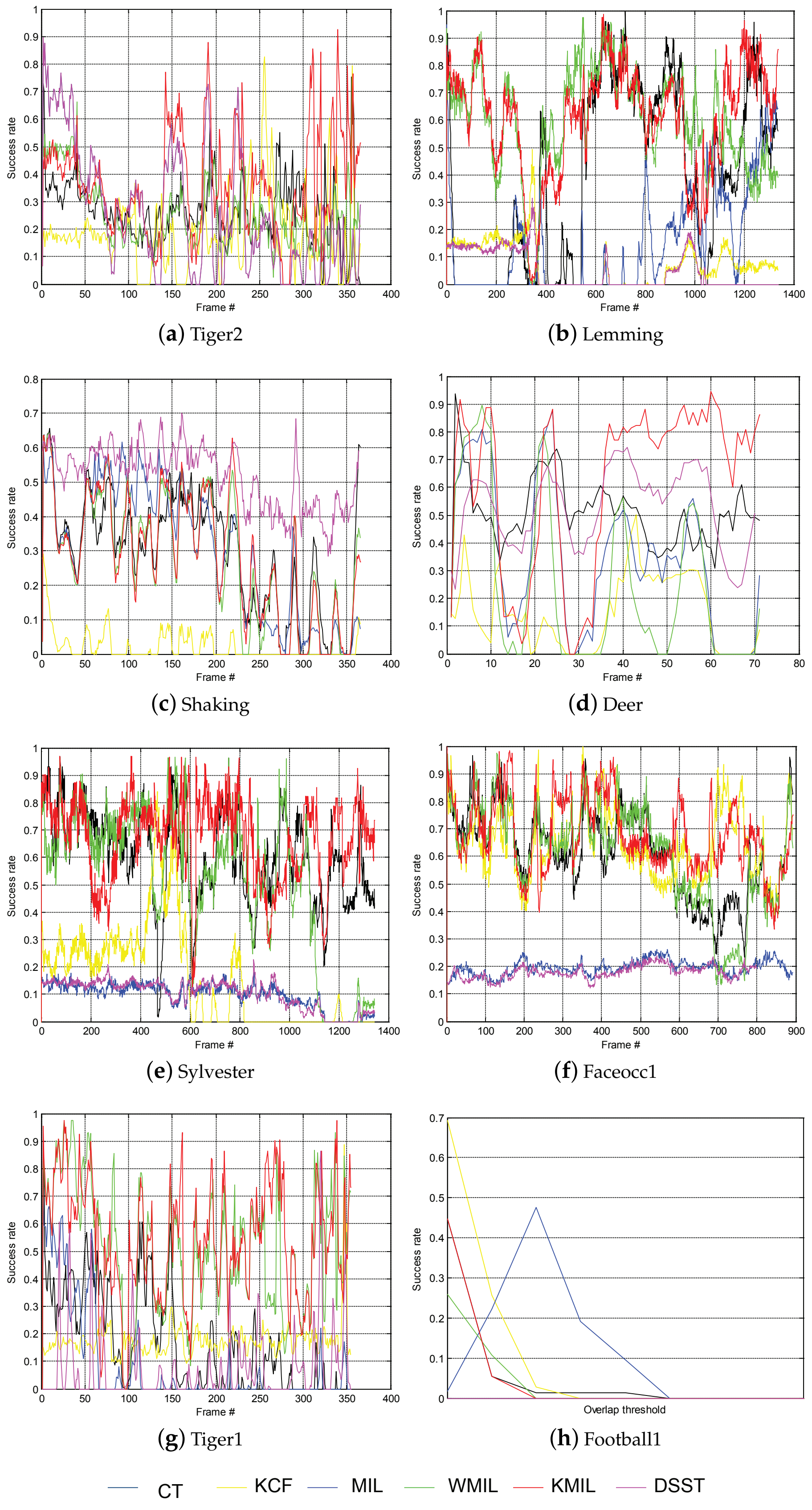

Figure 4. Success plot for sequences: "Deer", "Tiger2", "Faceocc2", "Sylvester", "Football1", "Shaking", "Tiger1", "Lemming".

5. Conclusions

In this paper, we revisit the core of the WMIL formulation to counter the issues of computation complexity and drift problem. We introduce a Gaussian kernel based multiple instance learning algorithm for real-time vision applications. We also suggest a simple yet effective searching circle update strategy that is especially suitable for small but moving fast objects. Lastly, we also present a classifier update method for handling the appearance changes with respect to two thresholds and reference similar score. The experiments conducted on several classical videos demonstrated that the KMIL tracker was efficient in terms of time-consumption and robustness.

Author Contributions: Conceptualization, T.H., L.W.; Methodology, L.W., T.H.; Software, B.W.; Validation, L.W., B.W.; Formal Analysis, B.W.; Investigation, B.W.; Resources, B.W.; Data Curation, B.W.; Preparation, L.W., B.W.; Writing, T.H., B.W.; Visualization, L.W.; Supervision, T.H.; Funding Acquisition, T.H.

Funding: This research was funded by [the key research and development program of the Hebei province science] grant number [18210329D]; [the Natural Science Foundation of Hebei Province] grand number [F2018205178]; [the Hebei College of Industry and Technology Natural Science Program] grand number [ZRY2017004].

Acknowledgments: Visual Tracker Benchmark http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html.

Conflicts of Interest: The authors declare no conflict of interest.The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MS	Mean Shift
IVT	Incremental Visual Tracking
VTD	Visual Tracking Decomposition
MOSSE	Minimum Output Sum of Squared Errors
CSK	Circulant Structure with Kernels
CN	Color Names
DCF	Discriminative Correlation Filter
DAT	Distractor Aware Tracking
C-COT	Continuous Convolution Operator
DeepSRDCF	Deep Spatially Regularized Discriminative Correlation Filter
DLSSVM	Dual Linear Structure Support Vector Machine
ECO	Efficient Convolution Operators
CT	Compressive Tracking
WMIL	Weighted Multiple Instance Learning
KMIL	Kernel based Weighted Multiple Instances Learning
KCF	Kernelized Correlation Filter
DSST	Discriminative Scale Space Tracking

References

1. Yilmaz, A. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 13. [CrossRef]
2. Gao, J.; Ling, H.; Hu, W.; Xing, J. Transfer Learning Based Visual Tracking with Gaussian Processes Regression. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 188–203. [CrossRef]
3. Zhang, K.; Zhang, L.; Yang, M.H. Real-time compressive tracking. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 864–877. [CrossRef]
4. Dadgostar, F.; Sarrafzadeh, A.; Overmyer, S.P. Face Tracking Using Mean-Shift Algorithm: A Fuzzy Approach for Boundary Detection. In Proceedings of the Affective Computing and Intelligent Interaction, First International Conference, (ACII 2005), Beijing, China, 22–24 October 2005; pp. 56–63. [CrossRef]

5. Ross, D A.; Lim, J.; Lin, R.S. Incremental Learning for Robust Visual Tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [CrossRef]
6. Kwon, J.; Lee, K.M. Visual tracking decomposition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1269–1276. [CrossRef]
7. Babenko, B.; Yang, M.H.; Belongie, S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632. [CrossRef] [PubMed]
8. Hare, S.; Saffari, A.; Torr, P.H.S. Struck: Structured output tracking with kernels. In Proceedings of the 2011 IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 263–270. [CrossRef]
9. Avidan, S. Support Vector Tracking. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, USA, 8–14 December 2001; pp. 1184–1191. [CrossRef]
10. Grabner, H.; Grabner, M.; Bischof, H. Real-Time Tracking via On-line Boosting. In Proceedings of the British Machine Vision Conference 2006, Edinburgh, UK, 4–7 September 2006; pp. 47–56. [CrossRef]
11. Grabner, H.; Leistner, C.; Bischof, H. Semi-supervised On-Line Boosting for Robust Tracking. In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 234–247. [CrossRef]
12. Bolme, D.S.; Beveridge, J.R.; Draper, B.A. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
13. Rui, C.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 702–715. [CrossRef]
14. Henriques, J.F.; Rui, C.; Martins, P. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
15. Danelljan, M.; Hager, G.; Khan, F.S. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 Decenber 2015; pp. 4310–4318. [CrossRef]
16. Danelljan, M.; Khan, F.S.; Felsberg, M. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097. [CrossRef]
17. Possegger, H.; Mauthner, T.; Bischof, H. In defense of color-based model-free tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 2113–2120. [CrossRef]
18. Bertinetto, L.; Valmadre, J.; Golodetz, S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; Volume 38, pp. 1401–1409. [CrossRef]
19. Danelljan, M.; Hager, G.; Khan, F.S. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef] [PubMed]
20. Danelljan, M.; Hager, G.; Khan, F.S. Convolutional Features for Correlation filter-based Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, 7–13 December 2015; pp. 621–629. [CrossRef]
21. Held, D.; Thrun, S.; Savarese, S. Learning to Track at 100 FPS with Deep Regression Networks. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765. [CrossRef]
22. Danelljan, M.; Robinson, A.; Khan, F.S. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488. [CrossRef]
23. Danelljan, M.; Bhat, G.; Khan, F. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939. [CrossRef]

24. Babenko, B.; Yang, M.-H.; Belongi, S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transa. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632. [CrossRef] [PubMed]
25. Zhang, K.; Song, H. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognit.* **2013**, *46*, 397–411. [CrossRef]
26. Ning, J.; Yang, J.; Jiang, S.; Zhang, L.; Yang, M.-H. Object Tracking via Dual Linear Structured SVM and Explicit Feature Map. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4266–4274. [CrossRef]
27. Wu, Y.; Lim, J.; Yang, M.-H. Online Object Tracking: A Benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418. [CrossRef]