Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 1205210, 10 pages
https://doi.org/10.1155/2018/1205210

Research Article

Real-Time Inland CCTV Ship Tracking

Lei Xiao,[1,2] Minghai Xu,[2] and Zhongyi Hu[2]

[1]*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, Shanxi, China*
[2]*Intelligent Information Systems Institute, Wenzhou University, Wenzhou 325035, Zhejiang, China*

Correspondence should be addressed to Zhongyi Hu; hujunyi@163.com

Received 11 October 2017; Revised 27 March 2018; Accepted 29 April 2018; Published 12 June 2018

Academic Editor: Panos Liatsis

The predator algorithm is a representative pioneering work that achieves state-of-the-art performance on several popular visual tracking benchmarks and with great success when commercially applied to real-time face tracking in long-term unconstrained videos. However, there are two major drawbacks of predator algorithm when applied to inland CCTV (closed-circuit television) ship tracking. First, the LK short-term tracker within predator algorithm easily tends to drift if the target ship suffers partial or even full occlusion, mainly because the corner-points-like features employed by LK tracker are very sensitive to occlusion appearance change. Second, the cascaded detector within the predator algorithm searches for candidate objects in a predefined scale set, usually including 3-5 elements, which hampers the tracker to adapt to the potential diverse scale variations of the target ship. In this paper, we design a random projection based short-term tracker which can dramatically ease the tracking drift when the ship is under occlusion. Furthermore, a forward-backward feedback mechanism is proposed to estimate the scale variation between two consecutive frames. We prove that these two strategies gain significant improvements over the predator algorithm and also show that the proposed method outperforms several other state-of-the-art trackers.

1. Introduction

In the past decades, there are rapid progress in the domain of inland ship surveillance and management [1, 2]. Old systems often contain AIS (automatic identification system), GPS (global positioning system), VTS (vessel traffic service), and GIS (geographic information system), while newer systems usually incorporate CCTV system for its practicability and inexpensiveness [3–5]. CCTV system enables marine bureau to achieve long-term robust ship tracking from a pure visual perspective. However, this task is extremely challenging as the target ship may suffer diverse appearance changes caused by, for example, internal distractions from low video quality, rotation, scale variation, and external distractions from cluttered background, illumination change, and occlusion, just to name a few.

Two categories of deep-learning based visual trackers show dominated accuracy on the popular OTB-100 [6] and VOT2015 [7] visual tracking benchmarks. One is to employ the powerful CNN (convolutional neural network) as a feature extractor, mainly to get rid of the traditional hand-crafted low-level features. Therein, the extracted deep features from different convolutional layers have been effectively combined to get more discriminative representations [8]. The other is to pretrain the network via an end-to-end manner and fine-tune them with subsequent observations. However, despite their overwhelming accuracy toward the challenges, we aim to work at in this paper, it should be noted that these trackers are unable to work in common embedded systems like the ones in inland surveillance. In such systems, running time becomes a critical factor, as well as the RAM/ROM and peak memory consumption and so on. For example, typical deep-learning based visual trackers have 5-8G memory consumption and the specification is usually 1-2G, which obviously hinders their usability.

Fortunately, besides deep-learning based visual trackers, several other machine-learning based trackers have also shown promising performances on OTB-100 and VOT2015 benchmarks. In [9], the OAB (online boosting) feature extraction was proposed to construct the object appearance model. It utilized the only one positive sample, i.e., the tracking location in the current frame to update the model,

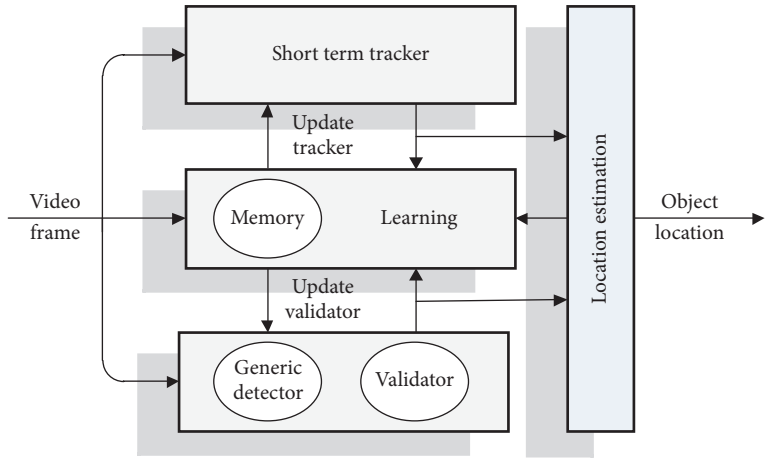


Figure 1: Framework of the predator system.

making the tracker easy to drift if the sample was misaligned. Then, the SemiB (semisupervised boosting) tracker was designed to alleviate the huge uncertainty of the "one sample update strategy". The basic rules therein were that only the initial samples were labelled and left the samples in later frames unlabelled. Then, the continuously updated boosting classifier decided which samples were positive or negative. To further void the tracking drift issue introduced by sample ambiguity in OAB tracker, the MIL (Multiple Instance Learning) [10, 11] tracker was proposed. In [10, 11] the training samples are in the form of bags. If a bag contained all negative examples, then it was negative; otherwise it was positive. Then the classifier was trained by optimizing the maximum likelihood probability, but as a function of the bags instead of the samples. The extensive results showed that MIL tracker could greatly ease the sample ambiguity problem with fewer parameter tweaks but turned to easily select indiscriminative features. Toward this issue, in [12], an EMIL (enhanced multiple instance learning) tracker was proposed. The point was to use a dynamic way to replace the traditional logistic function. These methods belong to the "tracking by detection" category, which transfer the object tracking challenge into object detection problem. There is no doubt that it is possible to get very fantastic tracking accuracy by borrowing some complicated techniques from the object detection domain. However, by doing so, the dynamic optimization between trackers and detectors is not established and we also sacrifice the desired real-time property for a practical system. Motivated by this insight, Kalal et al. [13] proposed a novel predator framework for long-term face tracking in unconfined video streams. This seminal work explicitly decomposed the tracker into three parts: tracking, learning, and detection. The complementarity of these three parts resulted in superior robustness to nonrigid deformation, rotation, and illumination change. However, when applied to inland CCTV ship tracking, predator algorithm has two main limitations. For one thing, when the target ship is partially or completely occluded by other objects, predator tracker is easily to drift to the covered object and cannot keep track of the target ship when the occlusion disappears. For another, the target ship may suffer potential diverse scale variations when it moves close or far away from the CCTV camera, which easily breaks the predator tracker's assumption of 3-5 elements in scale space.

In this paper, we aim to construct a practical inland CCTV ship tracking system. To achieve this goal, motivated by the discussions, we build our algorithm based on predator framework since it has demonstrated favourable accuracy with decent time consumption. This choice ensures our system to have a solid basis. On the other side, suitable strategies should be carefully designed when we attack the challenge of occlusion and scale distractors since predator framework has deficiency on them. Specifically, a random projection based short-term tracker is designed to ease the tracking drift issue when the target ship is under occlusion. Meanwhile, a forward-backward feedback strategy is designed to track the scale variation. The results demonstrate great improvements over the predator algorithm, as well as several other state-of-the-art trackers. We organize the rest paper as follows: Section 2 reviews the predator framework from a control system perspective. Section 3 details the proposed method. Section 4 evaluates our tracker on challenging sequences. Finally, Section 5 presents the concluding remarks.

2. Predator Framework

Predator framework is illustrated in Figure 1. Different from traditional "tracking by detection" trackers, predator tracker is designed as "tracking and detection". As shown in Figure 1, the system consists of a tracker, a learning module, and a detector. The short-term tracker is based on traditional LK algorithm [14] which recursively tracks the target object. Because the performance of LK algorithm heavily depends on the quality of the corner-points-like features, it is very likely that such short-term tracker drifts to the background if the target ship suffers partial or full occlusion. The detection part involves a cascade of classifiers including a variance filter, an ensemble classifier, and a nearest-neighbour classifier [15]. Each classifier is dedicated to reject different levels of candidate samples, from coarse to fine. For instance, the

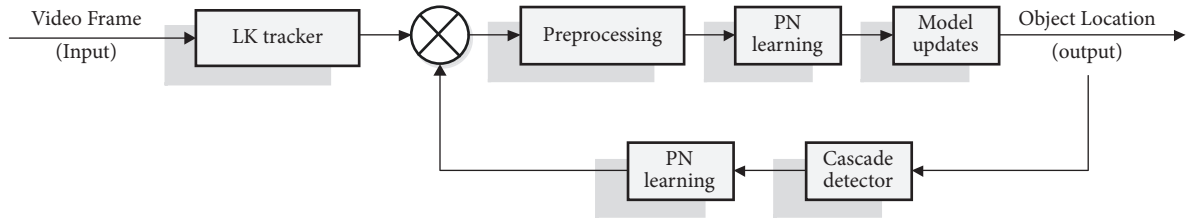


Figure 2: Predator closed-loop control system.

variance filter has the ability to reject more than 50% of the detected samples, then the ensemble classifier picks up the top 10 most likely samples from the rest half. At last, it is the nearest-neighbour classifier to decide the target object location from 1 out of 10 bounding boxes which pass the ensemble classifier. In Figure 1, notice that the tracker runs simultaneously with the detector and the final object location in current frame is determined by both the tracker and detector (typically as an ensemble). It is well noted that both the short-term tracker and cascade detector can make mistakes, and this is actually where the learning part comes into play. The learning, i.e., P-N online learning, employs the so called "P expert" and "N expert" to estimate the errors of the short-term tracker and cascade detector [13, 16]. After learning knowledge from the experts, it is expected to avoid the same errors happening in the future [16, 17].

A good way to understand predator is from a classical control theory viewpoint, by which the framework can be depicted as a closed-loop control system shown in Figure 2. The LK tracker acts as the open-loop controller and the cascade detector and PN learning are served as the feedback basis. These two signals are passed to compute the measured error, to formulate the objective the system should be optimized on. This insight makes predator apparently different from traditional open-loop tracker, e.g., mean-shift tracker and particle filter, which only depends on the tracker to make response to the input. The feedback branch observes and learns from the tracker if it keeps robust tracking. On the other hand, once the short-term tracker suffers drift or completely fails, the feedback branch will correct the tracker to reinitialize the tracking process. Benefitted from the feedback branch, predator tracker performs well in long-term robust tracking.

It is commonly known that an effective and efficient forward branch is of critical importance and the forward/feedback branches are expected to have similar power [18, 19]. However, the forward branch of predator is not as powerful as the feedback branch: corner-points-like features employed by LK tracker are very likely to drift due to occlusion appearance change; the tracker has no ability to adapt to the potential diverse scale variations of the target. As a result, the power of cascade detector is degraded, which has side effect on the forward branch or introduces new error. These discussions suggest that a more powerful forward branch has great potential to improve the performance of predator.

3. Short-Term Compressive Tracker

3.1. Strategy to Track under Occlusion. One of our main goals is to improve the performance when the target ship suffers occlusion distraction. In order to achieve this, we design a random projection based strategy illustrated in Figure 3.

In Figure 3(a), we perform classifier update at frame t. First, we sample positive and negative samples $z \epsilon R^{w*h}$ based on the object location $\mathrm{I_t}$. Specifically, we treat the samples as positive if $a < \|\mathrm{I_z} - \mathrm{I_t}\| < b$ and as negative if $c < \|\mathrm{I_z} - \mathrm{I_t}\|$ where a, b, c denote circle radius ($a < b < c$). Then, for each sample $\mathrm{I_z}$, we convolve it with a set of rectangle filters $\{h_{1,1}, \ldots, h_{w,h}\}$ defined as follows:

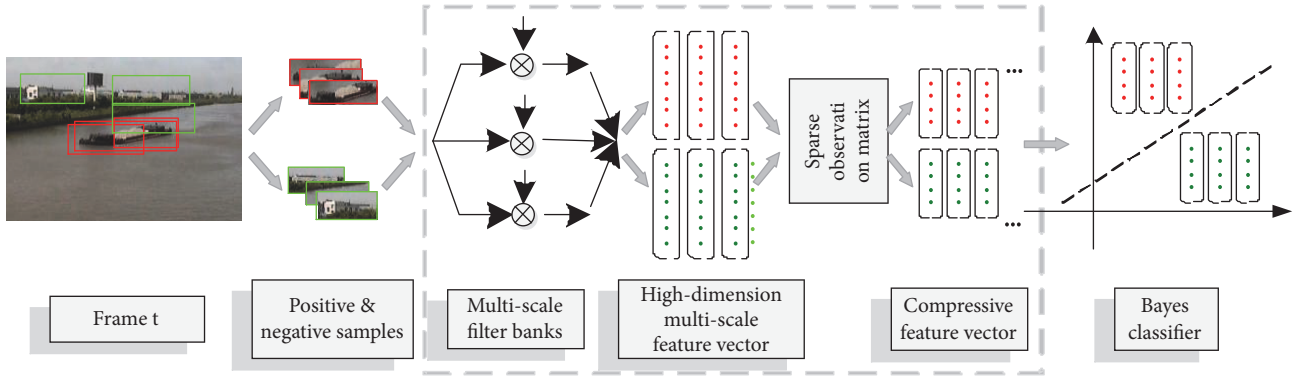
$$h_{p,q}(x, y) = \begin{cases} 1, & x_i \le x \le x_i + d,\ y_i \le y \le y_i + e \\ 0, & otherwise \end{cases} \quad (1)$$

where (x_i, y_i) denotes the upper left coordinate and $d \in [1, w], e \in [1, h]$ are the width and height of the rectangle filter, respectively. Each filtered feature map can be represented as a column vector in $\mathbb{R}^{w*h}$ and these vectors are sequentially concatenated to form a very high-dimensional image representation $x = (x_1, \ldots, x_m)^T \epsilon R^f$ *where* $f = (w * h)^2$). The dimension f is usually between 10^6 and 10^{10}. It should be noted that it is nontrivial to model on such heavy feature representations. Instead, compressive sensing theory said that a random projection matrix which satisfies the Johnson-Lindenstrauss lemma also holds true for the RIP (restricted isometric property) [20]. This property suggests that we can design such a random projection matrix to simplify the heavy feature representations while maintaining the discriminable information on a relatively low-dimensional space.

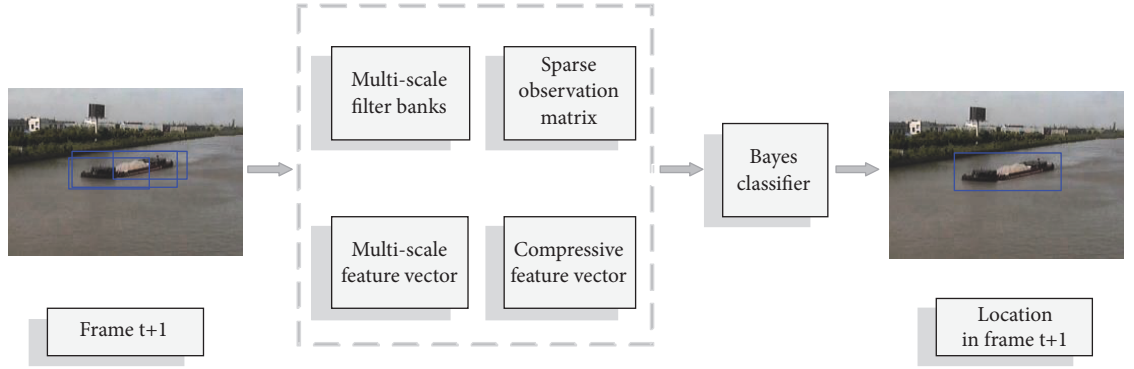
$$v = Rx \quad (2)$$

In (2), $x \epsilon R^m$ represents the high-dimensional space vector, $v \epsilon R^n$ represents the low-dimensional space vector, and $n \ll m$. In this paper, we adopt the group of sparse random projection matrices in [20, 21] to extract the compressive features for the target ship. The entry of the random projection matrix is defined as follows:

$$\mathrm{r_{i,j}} = \sqrt{s} * \begin{cases} 1 & \text{with probability } \dfrac{1}{2s} \\ 0 & \text{with probability } 1 - \dfrac{1}{s} \\ -1 & \text{with probability } \dfrac{1}{2s} \end{cases} \quad (3)$$



(a) Update classifier in frame t



(b) Target position in the current frame t+1

Figure 3: Strategy for tracking occlusion.

When $s = 2$ or $s = 3$, R satisfies the Johnson-Lindentrauss lemma. In order to produce a sparser observation matrix such that the projection of high-dimensional features onto compressive domain could remain sufficiently robust for occlusion, $s = 3$ is used in this paper. For this reason, it only needs to store 1/3 of the elements of the random projection matrix. Meanwhile, temporal complexity is $O(3 * n)$. In this way, the projection process is of low spatial complexity and highly effective.

In the next step, we assume that each feature in $v = (v_1, v_2, \ldots, v_m)^T$ is independent and develop an online updated naïve Bayes classifier:

$$H(v) = \sum_{i=1}^{n} \log\left(\frac{p(v_i \mid Y = 1)}{p(v_i \mid Y = 0)}\right) \quad (4)$$

where $Y \epsilon \{0, 1\}$ is the binary label. For simplicity, we assume uniform prior probabilities for class labels, i.e., $p(Y = 0) = p(Y = 1)$. Furthermore, the conditional distributions are modelled as Gaussian distribution, where $p(v_i \mid Y = 0) \sim N(\mu_i^0, \sigma_i^0)$, $pp(v_i \mid Y = 1) \sim N(\mu_i^1, \sigma_i^1)$. The recursive update equation can be deduced easily based on maximum likelihood estimation.

$$\mu_i^1 \longleftarrow \gamma\mu_i^1 + (1-\gamma)\mu^1 \quad (5)$$

$$\mu_i^0 \longleftarrow \gamma\mu_i^0 + (1-\gamma)\mu^0 \quad (6)$$

$$\sigma_i^1 \longleftarrow \sqrt{\gamma(\sigma_i^1)^2 + (1-\gamma)(\sigma^1)^2 + \gamma(1-\gamma)(\mu_i^1 - \mu^1)^2} \quad (7)$$

$$\sigma_i^0 \longleftarrow \sqrt{\gamma(\sigma_i^0)^2 + (1-\gamma)(\sigma^0)^2 + \gamma(1-\gamma)(\mu_i^0 - \mu^0)^2} \quad (8)$$

Here, $\gamma > 0$ is the learning rate, $\mu^1 = (1/n)\sum_{k=0|y=1}^{n-1} v_i(k)$ and $\mu^0 = (1/n)\sum_{k=0|y=0}^{n-1} v_i(k)$ represent initial mean value, and $\sigma^1 = \sqrt{\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$ and $\sigma^0 = \sqrt{\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^0)^2}$ represent initial standard deviation.

In Figure 3(b), we determine the target location at frame $t + 1$. We first collect potential samples Iz in a radius region s around the object location in the previous frame I_t, i.e., $\|I_z - I_t\| < s$. And then we use the same way to obtain the low-dimensional compressive feature for every potential sample. Finally, the sample with the maximum $H(v)$ in (4) is the target position in the current frame, and this process iterates to the whole tracking procedure.

3.1.1. Discussion. Compressive sensing theory has drawn the attention of researchers in various fields in recent years [22–24]. It is very appealing that signal measurement and coding could be conducted at a rate far lower than that of Nyquist sampling by taking advantage of the scarcity property of the signal. The main difference between compressive sensing and random projection is that compressive sensing needs

to further design a reconstruction algorithm from the low-dimensional compressive features v to the high-dimensional feature x and random projection does not need.

According to compression sensing theory, if we want to get rid of the redundancy of the feature representations x_1, x_2 while maintaining discriminative power, the random projection matrix R should meet the demand of RIP [20], i.e., $(1-\varepsilon)\|x_1-x_2\|_{l_2}^2 \leq \|Rx_1-Rx_2\|_{l_2}^2 \leq (1+\varepsilon)\|x_1+x_2\|_{l_2}^2$. The random Gaussian matrix $R \in \mathbb{R}^{n*m}(r_{ij} \sim N(0,1))$ is a typical projection matrix that satisfies the Johnson-Lindenstrauss lemma but we find that it is highly temporally and spatially complex. For this reason, we employ the sparse projection matrix R in (3). It has been proved that the theoretical dimensional bound [25] of random matrix R is as follows:

$$n \geq \frac{4+2\beta}{\epsilon^2/2-\epsilon^3/3}\ln(d) \tag{9}$$

where β controls the success probability, ϵ controls the accuracy level in projection, and d represents the quantity of input points in $\mathbb{R}^m$. But in practice, in the domain of inland CCTV ship tracking, we find that $n \geq 60$ is sufficient to obtain good tracking results.

Benefitted from the inherit power of compressive sensing, random projection based trackers have shown decent performance when the target object suffers partial or even full occlusion. The CT (compressive tracking) tracker was proposed in [25]. Therein, they utilized a very sparse random projection matrix to extract the features of the target object in a low-dimensional feature space. This algorithm demonstrated high accuracy in cluttered background, varying illumination, and occlusion, though being unable to track the scale variation. In [26], a MSRP (multi-scale random projections) method was proposed. In their method, they adopted a sparse projection matrix to extract the fern features in order to track in scenes with occlusion and illumination change. In [27], a MCFF (multiple compressed features fusion) method was proposed. Differently, they employed two kinds of adaptive random measurement matrices to extract two different features to track the target object. The experiments showed that MCFF achieved favourable performance when the target object was occluded and kept good track of the target when the occlusion disappeared. For our purpose, main limitation of the MCFF method is the heavy computational burden and the incapacity to track the scale variation.

3.2. Strategy to Track Scale Variation. As the target ship moves toward or far away from the CCTV camera, the scale state can dramatically change over time. Robust scale tracking is an important factor for inland surveillance system. In this paper, a self-adaptive strategy to track scale variation is proposed (see Figure 4 for illustration).

The procedures of the strategy are as follows:

(1) First, N random particles, $x_i \epsilon \mathscr{R}^d, (i = 1, \ldots, N)$ are uniformly sampled within the $j * k$ neighbourhood of target location X_t at current frame I_t. Here, $j = k = 10$, $N = 100$.

(2) Second, the mean-shift method [28] is used to estimate the particle position at frame I_{t+1}:

$$x_{i\epsilon(1,\ldots,N)}^{t+1} = \text{meanshift } x_{i\epsilon(1,\ldots,N)}^{t} \tag{10}$$

(3) Third, the mean-shift method is used inversely to estimate the position of $x_{i\epsilon(1,\ldots,N)}^{t+1}$ at frame I_t:

$$x_{i\epsilon(1,\ldots,N)}^{t'} = \text{meanshift } x_{i\epsilon(1,\ldots,N)}^{t+1} \tag{11}$$

(4) Then, the ratio of horizontal and vertical coordinates from the forward and backward path are computed. The target scale at frame I_{t+1} is the median of all particle ratios at frame I_t:

$$S_{t+1} = median\left(\frac{x_{i\epsilon(1,\ldots,N)}^{t'}}{x_{i\epsilon(1,\ldots,N)}^{t+1}}\right) \tag{12}$$

(5) Finally, the following scale update is used to improve the stability of scale adaptively:

$$S_{t+1} \longleftarrow \gamma S_t + (1-\gamma) S_{t+1} \tag{13}$$

where γ is the forgetting factor, set as 0.1 throughout the experiment.

3.2.1. Discussion. As discussed in Section 3.1, we obtain the high-dimensional image representation via convolutions with a group of rectangle filters at multiple scales. It is clear that the upper bound size of the filter bank is determined, which exactly equals the width and height of the target object we choose in the first frame. When the target ship moves toward to the CCTV camera, the filter bank has the suitable filter to account for scale decrease. However, when the target ship moves far away from the CCTV camera, the filter bank has no ability to account for scale increase. Meanwhile, both situations have the risk to collect dirty positive and negative samples, which may degrade the online updated classifier to cause drift.

The reason to introduce mean-shift method [28] into our scale tracking strategy is two unfolds. For one thing, this method computes the direction with fastest increasing probability density, a local extreme of probability density could be derived after several iterations. For another, the YCrCb features are a good complementary to the texture-like features exacted by random projection matrix. We use the mean-shift method twice, i.e., forward and backward, based on the observation that the trajectory should be identical if the movement is smooth. And if the target object suffers distractors, the trajectory will have deviations. In the experimental part, we notice (10-13) are good way to model the scale variation between consecutive frames, producing reasonably accurate and stable scale estimation.

4. Experiments

In this section, we conduct two experiments (indicated as experiment 1 and experiment 2, respectively) to demonstrate the favourable performance of the proposed algorithm.

4.1. Experimental Setup. In our experiment, a is set to 4, b is set to 8, c is set to 30, and m is set to 50. The environment

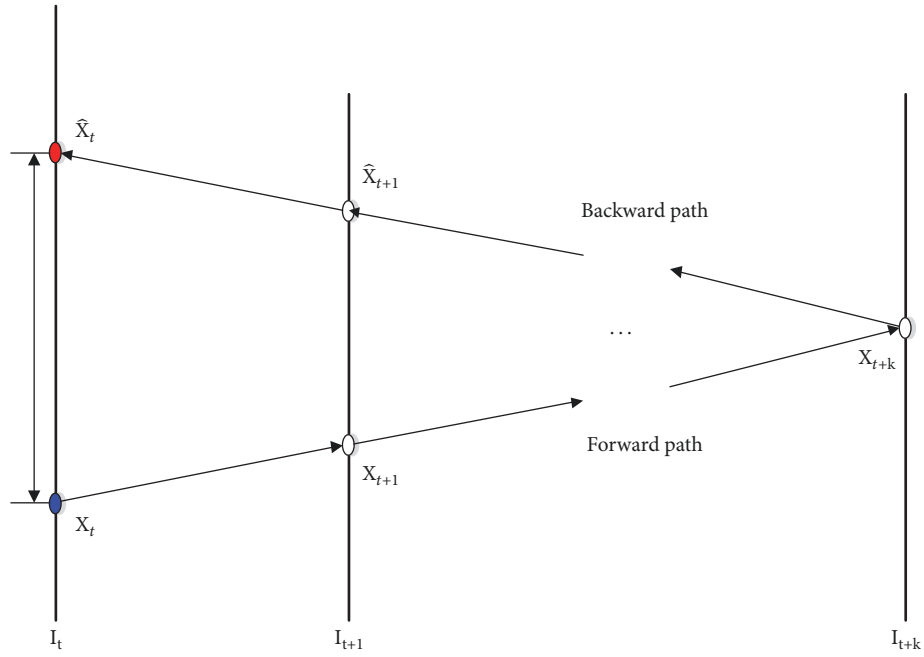


Figure 4: Strategy to track scale.

is an embedded machine of Intel (R) Core (TM) Dual CPU with 2.00 GHz dominant frequency and 1G RAM, which is a typical configuration in inland surveillance system.

4.2. Experiment 1. In this experiment, we evaluate our tracker on the popular visual tracking benchmark [6]. This benchmark contains 29 trackers and 100 challenging sequences. These sequences are very diverse and not limited to inland CCTV sequences. In [6], OPE (one-pass evaluation), TRE (temporal robustness evaluation), and SRE (spatial robustness evaluation) are designed to test the tracker's robustness to initialization. And in each initialization round, precision plot and successful plot are employed to show the ratio of successful frames at different centre location errors or overlap thresholds. For more information, we refer the readers to [6] for details. For our purpose, among the 29 public available trackers, we mainly compare those that achieve real-time performance. The qualitative results are shown in Figure 5 and the quantitative results are shown Figure 6. As shown in Figure 6, the proposed method gets all 6 first places among the evaluation metrics, often with a big margin over the second best tracker. For example, in the precision plot for OPE, our tracker gets a precision of 0.571, much higher than 0.477 from predator. The same phenomenon occurs in the success plot for OPE, where our tracker shows a 5.5% gain over the second best predator (0.458 compared to 0.403). Moreover, the precision/success plots of SRE/TRE show that the proposed method achieves overwhelming robustness to tracker's initialization.

4.3. Experiment 2. In this experiment, we evaluate our tracker on inland CCTV sequences. The main purpose of this experiment is to show the gained improvements over the predator algorithm with the designed strategies, especially when the target ship suffers occlusion and scale distractions. For the sake of simplicity, we adopt the common precision/recall [29, 30] and average centre location error evaluation metric and Table 1 shows a quantitate comparison of the two algorithms. For occlusion challenge, our tracker gets a precision of 0.93 compared to 0.82 from predator. More obviously, predator gets a low recall score, i.e., 0.27 in scale challenge, while our tracker gets a high score of 0.58. We can also notice that the centre location error for scale challenge is greater than occlusion challenge for both trackers and our tracker gets much lower centre location error in more decent real-time performance, which strongly suggests the effectiveness of the designed strategy.

CCTV1 sequence is collected from the Yangtze River marine bureau, which consists of 1260 frames in total. We choose the right-side ship to track in the first frame and we show several screenshots to detail the tracking process (see Figure 7). At frames 156 and 316, the target ship moves slowly toward the left side; both algorithms can keep good track of the target ship. At frames 622 and 897, the target ship is gradually occluded by another ship. The predator algorithm begins to drift the occluded object while the proposed method continues to work well. More obviously, in frame 1239, when the occlusion is alleviated, predator algorithm almost completely loses track of the target ship. On the contrary, the proposed method still keeps stable tracking. This result suggests that the compressive tracker is more powerful than LK short-term tracker and is less sensitive to the occlusion distractors. In fact, LK short-term tracker heavily relies on the assumption that neighbouring points on the same surface have similar movement patterns and their projections on image plane are also nearby. However, when the target object suffers occlusion, it is difficult to search for

Table 1: Evaluation on CCTV sequences.

	Sequence			
	CCTV 1		CCTV 2	
Challenge	occlusion		scale variation	
Precision	*0.93*	**0.82**	*0.84*	**0.75**
Recall	*0.65*	**0.42**	*0.58*	**0.27**
Centre location error (pixel)	*10.2*	**14.8**	*16.3*	**29.6**
Real-time performance (frames per second)	*6.7*	**4.3**	*6.8*	**4.3**

Note: italic font indicates the proposed algorithm; bold font indicates the predator algorithm.

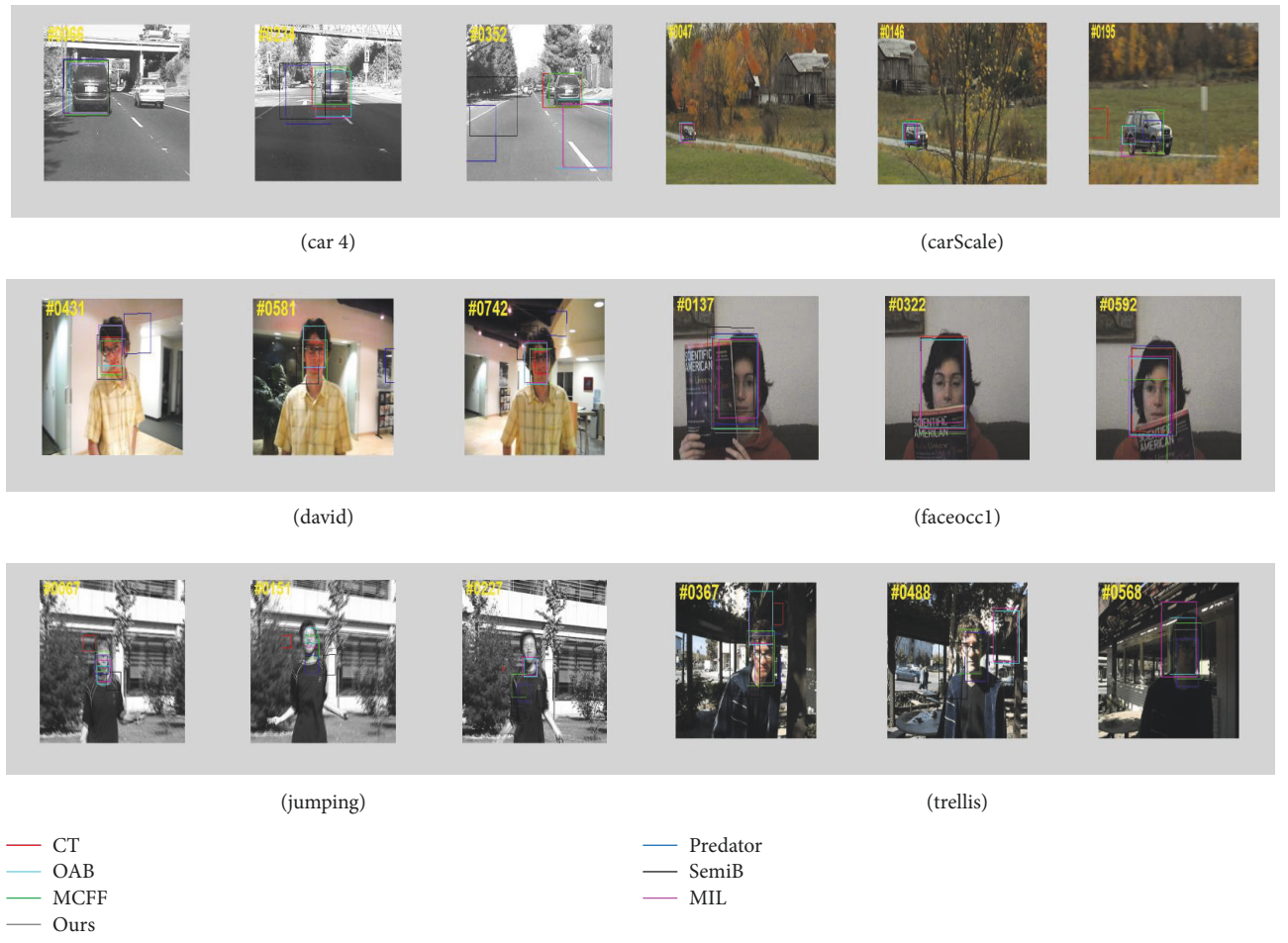


Figure 5: Sample tracking results.

those points with obvious gradients in both two directions for stable tracking. On the other hand, in the proposed method, the Johnson-Lindenstrauss lemma ensures that the designed random projection matrix can efficiently extract the low-dimensional features which have the inherit merit of insensitivity to occlusion. Then, the naïve Bayes classifier with online update facilitates the separation of the target ship from the background.

CCTV2 sequence comes from Wenzhou Navigation Department and it contains 836 frames. In the first frame, the right-side ship is selected as the object of interest. As the target ship moves close to the CCTV camera, the scale of the target dramatically changes. When the scale variation is not heavy, both methods track well (see frames 137, 216, and 304 in Figure 8). At frames 579 and 820, predator algorithm drifts to the background because it uses ambiguity samples to update the classifier. The proposed method performs well on scale distraction because the proposed scale strategy accurately estimates the scale variation. The design of forward-feedback mechanism is inspired by classical control theory. Mean-shift method has been proved effective when looking for local extremer of probability density and we also prove that the YCrCb feature shows relatively good discriminability.

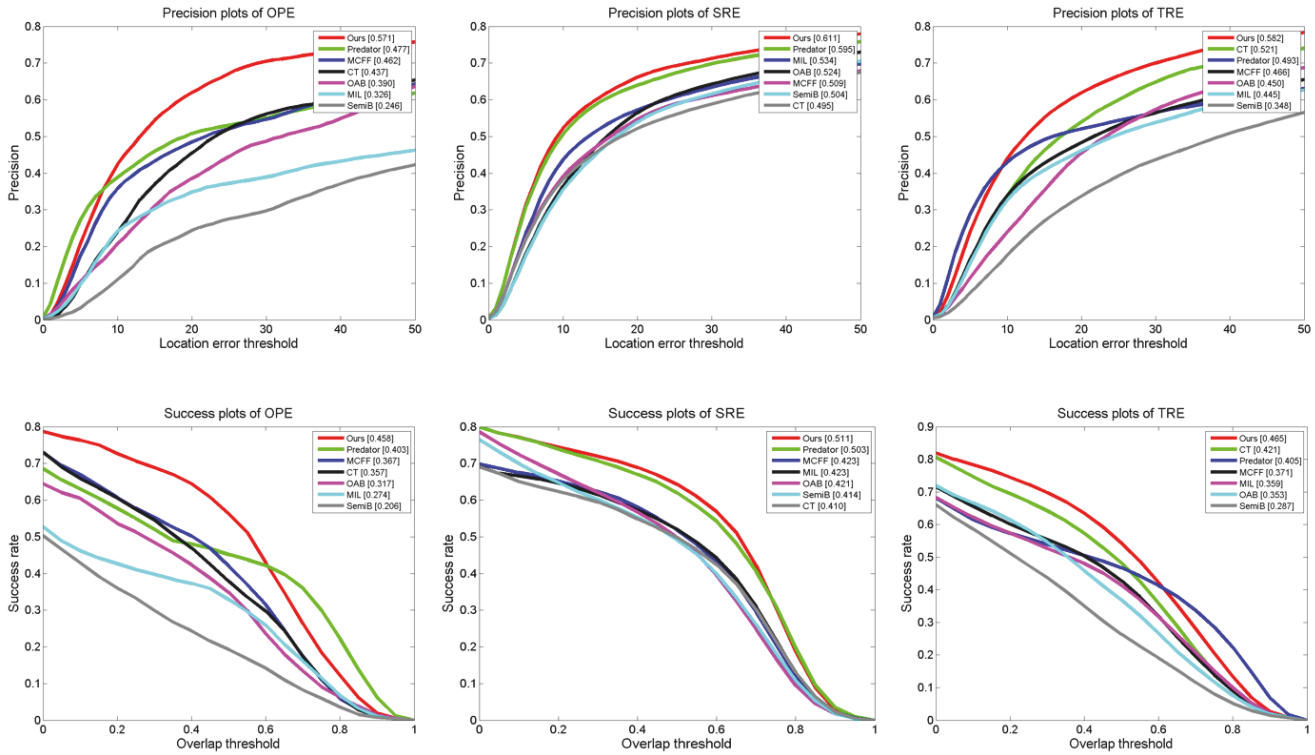


Figure 6: Success and precision plots.

Figure 7: Sample result to track the occlusion.

Figure 8: Sample result to track the scale.

5. Conclusions

This paper presents a practical inland CCTV ship tracking systems. First, we deeply analyse the predator framework from a classical control theory perspective. Second, we construct our system based on predator framework. Third, we design an occlusion and a scale strategy to ease the effect from these distractors. We show the two strategies gain significant improvements over the predator algorithm, and we also show the proposed method outperforms several other state-of-the-art trackers.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the financial support by Zhejiang Provincial Natural Science Foundation of China (Projects nos. LZ15F030002 and LY16F020022).

References

[1] F. Bi, J. Chen, Y. Zhuang, M. Bian, and Q. Zhang, "A Decision Mixture Model-Based Method for Inshore Ship Detection Using High-Resolution Remote Sensing Images," *Sensors*, vol. 17, no. 7, p. 1470, 2017.

[2] Y. Zhang, Q.-Z. Li, and F.-N. Zang, "Ship detection for visual maritime surveillance from non-stationary platforms," *Ocean Engineering*, vol. 141, pp. 53–63, 2017.

[3] Y. Wang, J. Zhang, X. Chen, X. Chu, and X. Yan, "A spatial-temporal forensic analysis for inland-water ship collisions using AIS data," *Safety Science*, vol. 57, pp. 187–202, 2013.

[4] Y. Rendao E, J. Huang, and S. O. Economics, "AIS Track Based on OD Segmentation," *Navigation of China*, vol. 2017, 2017.

[5] K. Honda, R. Shoji, and M. Inaishi, "A Study of Standard Ship Track with AIS Data," *The Journal of Japan Institute of Navigation*, vol. 137, no. 0, pp. 97–102, 2017.

[6] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, pp. 2411–2418, Portland, Ore, USA, June 2013.

[7] M. Kristan, R. Pflugfelder, J. Matas et al., "The Visual Object Tracking VOT2015 Challenge Results," in *Proceedings of the 2015 IEEE International Conference on Computer Vision: Workshop (ICCVW)*, pp. 564–586, Santiago, December 2015.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] H. Grabner, "Real-Time Tracking via On-line Boosting," in *Proceedings of the BMVC*, 2006.

[10] B. Babenko, S. Belongie, and M. Yang, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 983–990, Miami, Fla, USA, June 2009.

[11] B. Babenko, M. 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.

[12] F. Teng, Q. Liu, and J. Guo, "Real-time Ship Tracking via Enhanced MIL Tracker," in *Proceedings of the IETET Conference Publishing System*, pp. 399–404, 2013.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: tracking-learning-detection applied to faces," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 3789–3792, Hong Kong, September 2010.

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence (IJCA1 '81)*, vol. 2, pp. 674–679, 1981.

[15] 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, 2011.

[16] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: automatic detection of tracking failures," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 2756–2759, August 2010.

[17] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," in *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV '09)*, pp. 1417–1424, October 2009.

[18] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: a learning-based method for high-performance tracking control," *Control Systems IEEE*, vol. 26, no. 3, pp. 96–114, 2015.

[19] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: a learning-based method for high-performance tracking control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[20] D. Achlioptas, "Database-friendly random projections: johnson-lindenstrauss with binary coins," *Journal of Computer & System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[21] C. X. Xiao, "A joint mechanism of adaptive Bayesian compressed channel sensing based on optimized measurement matrix," *Journal of Electronics Information Technology*, vol. 34, no. 10, pp. 2299–2305, 2012.

[22] U. P. Shukla, N. B. Patel, and A. M. Joshi, "A survey on recent advances in speech compressive sensing," in *Proceedings of the 2013 IEEE International Multi Conference on Automation, Computing, Control, Communication and Compressed Sensing, iMac4s 2013*, pp. 276–280, ind, February 2013.

[23] S. Li, C. Ma, and Y. Li, "Survey on reconstruction algorithm based on compressive sensing," *Infrared Laser Engineering*, vol. 42, no. 1, pp. 225–232, 2013.

[24] H. Jiang, W. Deng, and Z. Shen, "Surveillance video processing using compressive sensing," *Inverse Problems and Imaging*, vol. 6, no. 2, pp. 201–214, 2017.

[25] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, pp. 864–877, 2012.

[26] F. Teng and Q. Liu, "Multi-scale ship tracking via random projections," *Signal, Image and Video Processing*, vol. 8, no. 6, pp. 1069–1076, 2014.

[27] F. Teng and Q. Liu, "Robust multi-scale ship tracking via multiple compressed features fusion," *Signal Processing: Image Communication*, vol. 31, pp. 76–85, 2015.

[28] C. Beyan and A. Temizel, "Adaptive mean-shift for automated multi object tracking," *IET Computer Vision*, vol. 6, no. 1, pp. 1–12, 2012.

[29] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I511–I518, December 2001.

[30] P. Viola, "Robust real time object detection," in *Proceedings of the the International Workshop on Statistical and Computational Theories of Vision—Modeling, Learning, Computing*, p. 87, 2001.