Research on target tracking algorithm based on information entropy feature selection and example weighting

Huang Chao Bing, *Li Ting Hui
College of Electronic Engineering, Guangxi Normal University, Guilin, Guangxi, 541004, China
*Corresponding author’s e-mail: tinghuili@gxnu.edu.cn

Abstract. An improved algorithm based on information entropy feature selection and sample weighting is proposed to solve the drift problem when the multi-instance learning (MIL) target tracking algorithm is updating classifier. Algorithm based on Boosting framework, adopt the method of MIL as sample selection, the positive and negative samples collected around the target area, to extract the image feature is compressed and weak classifier. Based on maximum entropy principle from a number of weak classifier to select several optimal classifier, the resulting strong classifier can be used to track the location of the next frame target. The results show that the algorithm is robust to target occlusion and fast motion.

1. Introduction
Target tracking is a popular research field of computer vision today, and it has a wide range of applications: such as video surveillance, human-computer interaction, drone surveys, autonomous driving[1]. Although target tracking has been successfully applied in some systems, there are still many challenges: such as tracking target occlusion, motion blur, scale transformation, and background similar interference.

The concept of strong and weak learning originated from the PAC learning model proposed by Valiant and Kearns in 1984 [2]. In 1990, Schapire proposed Boosting. The main idea is to linearly combine weak classifiers into strong classifiers. In 1995, Freund and Schapire made a difference. The algorithm's classifier was improved, and an adaptive algorithm AdaBoost [3] was proposed. In 2001, Viola et al. applied AdaBoost to face detection [4], and in 2000, NCOza proposed online AdaBoost [5], adding a feature selector. MIL is a concept proposed in the mid to late 1990s when studying the problem of drug activity prediction. In 2005, Viola et al. proposed target detection based on MILBoost [6], and in 2008, Grabner et al. proposed a tracking algorithm based on semi-supervised online Boosting [7]. The MIL target tracking algorithm uses multiple examples to represent the collected target samples, and puts several examples in different "packages", which is different from using an image example as a training sample in the past, but in the form of a "package". The training sample is more robust, but it will produce information redundancy, resulting in a large amount of calculation, and during the update process, positive examples with less information content and weak classifiers with poor performance may be selected. As a result, the model does not better adapt to the changes in the appearance of the tracking target and causes drift problems.

In response to the above problems, this article proposes the following improved methods: 1. Compress and reduce the dimension of haar-like feature vectors, reduce the amount of calculation, and reduce the calculation time; 2. Apply different weights to positive and negative examples to select...
information content for the classifier. More positive examples; 3. Select the optimal weak classifier based on the maximum entropy as the objective function.

2. Algorithm description

2.1. Motion model

The first frame of the video used as the training sample is marked, and the subsequent frames are not marked, and the target position of the next frame is obtained by training with the sample of the first frame, and so on. $x$ is an example of any image block obtained around the target area, $y$ represents the classification result, $y=0$ is a negative example, $y=1$ is a positive example, a positive example includes the image block of the area where the target is located, the following is the MIL target tracking algorithm the basic steps:

1. At any time $t$, the location of the target is recorded as $l^*_t$, the function $l(x)$ represents the location of the image block $x$, with $l^*_t$ as the center, at the current target location, $s$ is the search radius to obtain a set of image blocks and calculate image characteristics:

$$X^s = \{x : \|l(x) - l^*_{t-1}\| < s\}$$ (1)

$l^*_{t-1}$ is the location of the target at time $t-1$ in the previous frame, that is, the image block around the target area of the previous frame is used as the training sample, and the location of the target at time $t$ is likely to be searched at time $t-1$ appears in the radius:

$$p(l^*_t|l^*_{t-1}) \propto \begin{cases} 1 & \text{if} \|l^*_t - l^*_{t-1}\| < s \\ 0 & \text{otherwise} \end{cases}$$ (2)

2. Calculate $p(y=1|x)$ for all collected image examples $x \in X^s$, that is, calculate the probability that any image example $x$ collected around the target area is a positive sample, using the classifier of the appearance model to calculate, and then use the greedy strategy to update the target position of the current frame:

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

3. After obtaining the target position of the current frame, collect positive and negative example samples around the target area of the current frame, put all the positive examples in the positive package, and the negative examples in the negative package, use these samples to update the classifier, and the positive and negative the sample collection range is as follows:

$$X^+ = \{x : \|l_t(x) - l^*_t\| < \alpha\}$$
$$X^- = \{x : \alpha < \|l_t(x) - l^*_t\| < \beta\}$$ (4) (5)

$X^+$ is the search range of positive examples, $X^-$ is the search range of negative examples. $\alpha$ and $\beta$ correspond to the search radius of positive and negative examples respectively. Use the collected positive and negative packages to update the classifier of the MIL appearance model, and loop until the end of the video.

2.2. Appearance model

The MIL algorithm collects multiple image block examples around the target in any frame, and puts them into different "packages". The collection of packages is used as the training set. Now suppose there are $n$ packages, each package has a label, but the example in the package has no label, then $\{(X_1, y_1), ..., (X_n, y_n)\}$, $y_i \in (0,1)$, $y_i = 0$ means the package is a negative package, and $y_i = 1$ is a positive package; there are $m$ examples in each package, then $X_i = \{x_{i1}, x_{i2}, ..., x_{im}\}$, $i = 1, 2, ..., n$, through training set training classifier.

As can be seen from 2. in section 2.1, the focus of the appearance model lies in the classifier, which is to identify the background and the target through the classifier. The output of the model algorithm is a strong classifier, which is composed of weak classifiers, $H(x) = \sum_k h_k(x)$, and these weak classifiers are the optimal weak classifiers selected from the weak classification pool by maximizing the likelihood function, that is, $k$ optimal weak classifiers are selected from $M$ weak classifiers calculator: $h_k = \arg \max_{h \in \{h_1, ..., h_m\}} L(H_{k-1} + h)$, where $L(\cdot)$ represents the likelihood function:
\[ L = \sum_{i \in (0,1)} \left( y_i \log p(y_i = 1|X_i) + (1 - y_i) \log (1 - p(y_i = 1|X_i)) \right) \]  

(6)

Where \( H_{k-1} \) represents the strong classifier composed of the first \( k - 1 \) weak classifiers, and \( p(y_i = 1|X_i) \) represents the probability that the training sample package is a positive package:

\[ p(y_i = 1|X_i) = 1 - \prod \left(1 - p(y_{ij} = 1|x_{ij})\right) \]  

(7)

\( p(y_{ij} = 1|x_{ij}) \) is the probability of a positive example, \( p(y_{ij} = 1|x_{ij}) = \sigma(H(x)).\sigma(x) = \frac{1}{1+e^{-x}}. \)

2.3. Algorithm improvement

2.3.1. Haar-like feature compression

It can be seen from Section 2.1 that the collected image samples need to calculate their feature values, and use Haar-like for feature extraction. Haar-like was originally proposed by Papageorigiou [8] and other scholars, and subsequently used for face detection, and found that AdaBoost combination has a very good effect, so the weak classifier mentioned above is actually composed of Haar-like features, common feature styles are as follows:

Figure 1. Haar-like feature basic style

They can be used as sliding windows to traverse the entire picture sample to obtain image features. The feature value is to subtract the black part from the white part. The integral graph is usually used to quickly calculate the feature value.

Suppose the original image signal is \( N \)-dimensional \( S \), the observation matrix is \( M \times N \)'s \( O \), and \( M \) is much smaller than \( N \), and the compressed feature vector is \( M \)-dimensional \( C = O \times S \). First, the image signal is originally sparse, which satisfies compression for signal requirements, the observation matrix \( O \) satisfies the RIP (finite equidistant property) criterion, and the random matrix that satisfies the JL[9] theorem can satisfy the RIP criterion. The definition of sparse matrix elements constructed by image samples is as follows:

\[ r_{ij} = \sqrt{b} \times \begin{cases} 
1 & p = \frac{1}{2b} \\
0 & p = 1 - \frac{1}{b} \\
-1 & p = \frac{1}{2b} 
\end{cases} \]  

(8)

\( p \) is the probability, \( b \in [2,4] \) is a randomly generated integer. When \( b = 3 \), most of the elements in the matrix are 0, and the amount of calculation is reduced.

2.3.2. Example weighting

It can be seen from Section 2.1 that the original calculation of the probability of the correct packet is:

\[ p (y_i = 1|X_i) = 1 - \prod \left(1 - p(y_{ij} = 1|x_{ij})\right), \]  

in order to strengthen the correct packet and The importance of positive examples, weighting the examples, now change the calculation of the positive packet probability to:

\[ p (y_i = 1|X_i) = \sum_{j=1}^{N} w_j p (y_{ij} = 1|x_{ij}), w_j = \frac{1}{c} e^{-||l(x_i) - l_i^*||} \]  

(9)

c is the normalization constant, \( l(x_i) \) is the position of a certain example \( x_i \), and \( l_i^* \) is the target position.
2.3.3. Information entropy

It can be seen from Section 2.2 that the MIL target tracking algorithm uses the maximum likelihood function to obtain the optimal weak classifier. In order to obtain better performance, the principle of maximum entropy is used to obtain the optimal weak classifier. Information entropy is related to thermodynamics. It was proposed by Shannon [10] in 1948. The emergence of information entropy is to solve the problem of quantifying information. The information entropy of discrete random variable \( X \) is:

\[
H(X) = -\sum_{x \in X} p(x) \log p(x)
\]

According to the principle of maximum entropy:

\[
L = -\sum_{i=1}^{M} \left( p_i y_i \log(p_i) + (1 - p_i)(1 - y_i) \log(1 - p_i) \right) \tag{10}
\]

\( p_i \) represents the probability of the sample packet.

The comprehensive improvement part proposes the following target tracking algorithm framework:

(1) Update of appearance model classifier

\[
\begin{align*}
\text{Input:} & \quad \text{image training set} \{(X_1, y_1), \ldots, (X_n, y_n)\}, \quad y_i \in \{0, 1\}, \quad X_i = \{x_{i1}, x_{i2}, \ldots\} \\
1. & \quad \text{Update} M \text{ weak classifiers in the initial image sample set, and initialize the strong classifier } H_{ij} = 0 \\
2. & \quad \text{for} \quad k = 1, \ldots, K \\
3. & \quad \text{for} \quad m = 1, \ldots, M \\
4. & \quad p^m = \sigma \left( H_{ij} + h_m(x_{ij}) \right) \ast w_{ij} \text{ weight the examples according to section 2.2.2} \\
5. & \quad p^m = 1 - \prod_{j}(1 - p^m_j) \text{ use the Noisy-OR [11] model to calculate the probability of the sample package} \\
6. & \quad L = -\sum_{i=1}^{M} \left( p_i y_i \log(p_i) + (1 - p_i)(1 - y_i) \log(1 - p_i) \right) \\
7. & \quad \text{end} \\
8. & \quad m^* = \arg\max_{m} L^m \text{ get the best weak classifier} \\
9. & \quad h_k(x) \leftarrow h_m^* \cdot (x) \\
10. & \quad H_k = h_{k-1} + h_k \\
11. & \quad \text{end}
\end{align*}
\]

Output: \( H(x) = \sum_{k} h_k(x) \) strong classifier

(2) Motion model of target tracking

\[
\begin{align*}
\text{Input:} & \quad \text{the target position } l_{t-1}^* \text{ at time } t - 1 \text{ in the previous frame, and the image at time } t \text{ in the current frame} \\
1. & \quad \text{According to } X^s = \{x: \|l(x) - l_{t-1}^*\| < s\}, \text{ collect a set of image samples around the target position at time } t - 1, \text{ calculate its feature vector, and according to 2.2.1 Section compresses the vector} \\
2. & \quad \text{Use (1) to calculate } p(y = 1|x), \quad x \in X^s \\
3. & \quad \text{According to } l_t^* = \arg\max_{x \in X^s} p(y|x) \text{ to obtain the target position at time } t \text{ of the current frame} \\
4. & \quad \text{Get the positive and negative sample package near the current target location } l_t^*: X^+ = \{x: \|l_t(x) - l_t^*\| < \alpha\}, X^- = \{x: \alpha < \|l_t(x) - l_t^*\| < \beta\} \\
5. & \quad \text{Update the classifier in (1) and keep looping until the end of the video}
\end{align*}
\]

3. Experiment and analysis

Experimental environment: windows10 operating system, Intel(R) Core(TM) i3-6100 processor, 3.7GHz, running memory 8G, python3.6 programming platform.
Experimental parameter settings: search radius $s = 25$, positive packet sample generation range $\alpha \in (0,4)$, negative packet sample generation range $\beta \in (6,50)$, weak classification pool $M = 250$, select $k = 53$ optimal Weak classifier, the learning rate $\eta = 0.87$.

Experimental evaluation index: center position error $CLE = \sqrt{(x_i - x)^2 + (y_i - y)^2}$, $x_i$ and $y_i$ represent the predicted target center coordinate value, $x$ and $y$ represent the actual coordinate value, the more the value is The smaller the better; the overlap ratio score $\frac{\text{area}(ROI_t \cap ROI_g)}{\text{area}(ROI_t \cup ROI_g)}$, $ROI_t$ represents the predicted target area, $ROI_g$ is the actual area, score $> 0.5$ indicates successful tracking, the larger the value, the better.

The experimental data set is OTB100, five video sequences are selected for experimental comparison, and the comparison algorithm is CT[12], MIL[13], OAB[14].

![Figure 2. 5 kinds of video sequence tracking center position error waveforms](image_url)

First row: Trellis, Tiger2; Second row: Tiger1, Faceoc1; Third row: David2

- **Ours**
- **CT**
- **MIL**
- **OAB**
Table 1. Features of five video sequences

| Video sequence | Features                                      |
|---------------|-----------------------------------------------|
| Trellis       | Light changes, blur, background changes, human faces |
| Tiger2        | Fast movement, motion blur, occlusion         |
| Tiger1        | Fast movement, motion blur, occlusion         |
| Faceooc1      | Block up and down, block left and right, face  |
| David2        | Blurred, similar faces, background targets    |

Experimental results: Ours represents the algorithm of this article. Make the following analysis based on the error waveform and video characteristics:

Trellis: Ours is relatively stable in the first 200 frames, while OAB has completely deviated from the target position at about 111 frames. Basically, the performance of the entire video is not good. The target slowly turns brighter from gray and dim light at 213 frames, and MIL starts to deviate from the target, 241 frames It has completely deviated. At this time, the Ours tracking frame is still in the target range. At 341 frames, the target almost shifts from the dark to the bright. The 4 algorithm tracking frames almost completely deviate until the end of the video.

Tiger2: Ours is stable overall. The OAB completely deviates from the target position at about 112 frames until the end of the video. The target at frame 273 is blocked and moving quickly. The MIL and CT tracking frames start to deviate slightly from the target, and the error of Ours decreases.

Tiger1: Combining the previous Tiger2 and Tiger1, Ours has good performance in fast motion, motion blur and partial occlusion, overall stability, while the other three algorithms have large overall fluctuations. As can be seen from the waveform diagram, CT performs relatively well, followed by OAB, and MIL performed poorly in this video sequence.

Faceooc1: When the face target is half-occluded at 190 frames, the performance of the four algorithms is not bad. At 250 frames, the book is slowly removed. Both MIL and Ours have mistakenly used the book as the target. The tracking frame moves down, CT Compared with OAB, the performance is good, 800 frames are occluded, Ours improves the tracking ability, and the errors of the other three algorithms increase slightly.

David2: From the waveform diagram, we can see that Ours has a stable overall performance. CT deviated from the target from the very beginning. MIL and OAB started to deviate slightly from the target at around 169 frames. The head of the target moved up and down quickly at frame 367. The OAB completely deviated from the target and returned to the target at about frame 417. The overall performance of MIL and Ours was similar.

Table 2. Average overlap rate of five video sequences

| Video sequence | CT  | MIL | OAB | Ours |
|---------------|-----|-----|-----|------|
| Trellis       | 0.37| 0.28| 0.15| 0.42 |
| Tiger2        | 0.6 | 0.61| 0.07| 0.79 |
| Tiger1        | 0.31| 0.07| 0.08| 0.73 |
| Faceooc1      | 0.75| 0.72| 0.77| 0.81 |
| David2        | 0.0 | 0.6 | 0.4 | 0.75 |

Table 3. Average center position error of five video sequences

| Video sequence | CT  | MIL | OAB | Ours |
|---------------|-----|-----|-----|------|
| Trellis       | 41.69| 71.47| 98.33| 41.63|
| Tiger2        | 28.19| 27.17| 252.66| 17.26|
| Tiger1        | 53.7 | 115.61| 98.39| 13.83|
| Faceooc1      | 25.82| 29.86| 24.66| 20.33|
| David2        | 76.7 | 10.93| 33.91| 6.41 |
The previous experimental evaluation indicators have explained that the tracking is successful if score $\geq 0.5$, but Table 2 shows that the average overlap rate of the four algorithms does not exceed 0.5 in the Trellis video sequence, so it is classified as a tracking failure. The highest overlap rate is Ours. It can be seen from Table 3 that the performance of MIL in Trellis and Tiger1 is poor. Combined with Table 1, it can be seen that MIL has poor robustness under conditions of fast movement, occlusion, and light changes. The CT algorithm performs the most stable, while Ours It shows better robustness in fast moving and occluded situations, and the performance in Faceooc1 and David2 is very similar to MIL, that is, the performance of the two is similar in face detection.

4. Conclusions
This article mainly aims to improve the classifier update of the MIL appearance model. The tracker mainly obtains the tracking results based on the classification of the target and the background. Therefore, it is particularly important to select a good classifier. This article chooses to weight the examples, then uses the principle of maximum entropy to select the optimal weak classifier, and finally aggregates into a strong classifier. The experimental results show that the algorithm in this paper has good robustness in motion blur, occlusion, and background interference, but it does not perform well under lighting changes, and the video sequence needs more diversity to verify the robustness of the algorithm. It is the focus of follow-up research.

Acknowledgments
National Natural Science Foundation of China (61964004)

References
[1] MENG L, YANG X. Overview of target tracking algorithms [J]. Acta Automatica Sinica, 2019, 45(7):1244-1260.
[2] Valiant L G. A theory of the learnable [J]. Communications of Acm, 1984, 27(11): 1134-1142.
[3] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[C]// II European Conference on Computational Learning Theory. Springer, Berlin, Heidelberg, 1995:119-139.
[4] Viola P A, Jones M J. Rapid Object Detection using a Boosted Cascade of Simple Features[C]// Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001.
[5] Oza N C. Online Ensemble Learning[C]// Seventeenth National Conference on Artificial Intelligence & Twelfth Conference on on Innovative Applications of Artificial Intelligence. DBLP, 2000.
[6] Viola P A, Platt J C, Zhang C. Multiple Instance Boosting for Object Detection[C]// Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]. 2005.
[7] Grabner H, Leistner C, Bischof H. Semi-supervised On-Line Boosting for Robust Tracking[C]// European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2008.
[8] Papageorgiou C P, Oren M, Poggio T. General framework for object detection[C]// Computer Vision, 1998. Sixth International Conference on. IEEE, 1998.
[9] Baraniuk R, Davenport M, Devore R, et al. A Simple Proof of the Restricted Isometry Property for Random Matrices[J]. Constructive Approximation, 2008, 28(3):253-263.
[10] Shannon C E. The mathematical theory of communication[J]. Bell Labs Technical Journal, 1950, 3(9):31-32.
[11] Wu Y, Lim J, Yang M H. Online Object Tracking: A Benchmark[C]// Computer Vision & Pattern Recognition. IEEE, 2013.
[12] Zhang K, Zhang L, Yang M. Fast Compressive Tracking[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(10):2002-2015.
[13] Babenko B, Yang M H, Belongie S. Robust Object Tracking with Online Multiple Instance
[13] Yang F, Yang J W, Yang Y. Learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1619-1632.

[14] Grabner H, Grabner M, Bischof H. Real-Time Tracking via Online Boosting[C]. British Machine Vision Conference, 2006, 1(5): 6-9.