

# An Improved Fast Compressive Tracking Algorithm Based on Online Random Forest Classifier

Jintao Xiong, Pan Jiang, Jianyu Yang, Zhibin Zhong, Ran Zou and Baozhong Zhu

*University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, P. R. China*

**Abstract.** The fast compressive tracking (FCT) algorithm is a simple and efficient algorithm, which is proposed in recent years. But, it is difficult to deal with the factors such as occlusion, appearance changes, pose variation, etc in processing. The reasons are that, Firstly, even if the naive Bayes classifier is fast in training, it is not robust concerning the noise. Secondly, the parameters are required to vary with the unique environment for accurate tracking. In this paper, we propose an improved fast compressive tracking algorithm based on online random forest (FCT-ORF) for robust visual tracking. Firstly, we combine ideas with the adaptive compressive sensing theory regarding the weighted random projection to exploit both local and discriminative information of the object. The second reason is the online random forest classifier for online tracking which is demonstrated with more robust to the noise adaptively and high computational efficiency. The experimental results show that the algorithm we have proposed has a better performance in the field of occlusion, appearance changes, and pose variation than the fast compressive tracking algorithm's contribution.

**Index Terms**—fast compressive tracking, naive Byes classifier, online, random forest.

## 1 Introduction

In the current target tracking, the visual tracking algorithms are roughly divided into two categories, i.e. generative or discriminative. Most generative algorithms only take the holistic representation into consideration and do not make full use of the discriminative information between the target and the background. So these algorithms are difficult to represent the target correctly under occlusion and pose variation.

Besides, discriminative model regards the tracking problem as a binary classification problem by obtaining the decision boundary of target and background to distinguish them i.e. such as online boosting, MIL tracker, online Semi-Boost. In recent years, Kai Hua Zhang raised the ideas on generative and discriminative model and proposed the fast compressive tracking algorithm [1],

which is also demonstrated not efficient enough to deal with noise, thus suffering from drifting problem.

## 2 Compressive tracking

The compressive sensing theory states that the high-dimension of the feature space can be projected to low-dimension space which contains enough information to reconstruct the original high-dimension through random projection matrix.

$$v = Rx. \quad (1)$$

Where  $R \in \mathbb{R}^{n \times m}$  is a random projection matrix, and  $n \ll m$ . According to Johnson-Lindenstrauss [2] lemma, for the two K-sparse vectors, the following property can be obtained to ensure the integrity of important information.

$$(1 - \varepsilon) \|x_1 - x_2\|_{l_2}^2 \leq \|Rx_1 - Rx_2\|_{l_2}^2 \leq (1 + \varepsilon) \|x_1 - x_2\|_{l_2}^2 \quad (2)$$

$$r_{ij} = \sqrt{\rho} \times \begin{cases} 1, & \text{with Probability } \frac{1}{2\rho} \\ 0, & \text{with Probability } 1 - \frac{1}{\rho} \\ -1, & \text{with Probability } \frac{1}{2\rho} \end{cases} \quad (3)$$

To deal with the scale change problems, series of rectangle filters are used to convolve the original image, the rectangle filters are defined as:

$$h_{i,j}(x,y) = \begin{cases} 1, & 1 \leq x \leq i, 1 \leq y \leq j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Diaconis and Freedman [3] show that the low-dimension vectors which are projected from the high-dimension through random measure matrix are almost Gaussian.

$$p(v_i | y=1) \sim N(\mu_i^1, \sigma_i^1) \quad (5)$$

$$p(v_i | y=0) \sim N(\mu_i^0, \sigma_i^0) \quad (6)$$

Where  $\mu_i^1(\mu_i^0)$  and  $\sigma_i^1(\sigma_i^0)$  are mean and standard deviation of the positive (negative) class. Naive Bayes classifier is a generative model. We assume all the elements are independently distributed. We can estimate the unknown parameters by the maximum likelihood through:

$$L(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0) = \prod p(v_i | y) p(y) \quad (7)$$

$$\mu^1 = \frac{1}{n} \sum_k^{n-1} v_i(k) \quad (8)$$

$$\sigma^1 = \sqrt{\frac{1}{n} \sum_k^{n-1} (v_i(k) - \mu^1)^2} \quad (9)$$

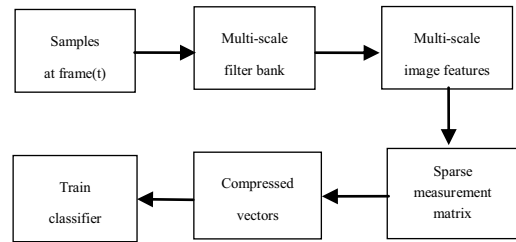
Taking the introduction of new samples into account in each new frame, the scale parameters are incrementally updated by:

$$\mu_i^1 = \lambda \mu_i^1 + (1 - \lambda) \mu_i^1 \quad (10)$$

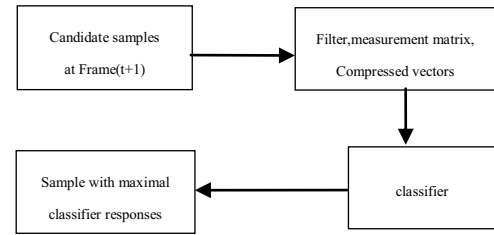
$$\sigma_i^1 = \sqrt{\lambda (\sigma_i^1)^2 + (1 - \lambda) (\sigma^1)^2 + \lambda (1 - \lambda) (\mu_i^1 - \mu^1)^2} \quad (11)$$

In the end, we can model with a naive Bayes classifier to predict the position from the candidate which is most likely to be the target sample. The main components of the proposed compressive tracking algorithm are shown by Figure. 1.

$$H(v) = \sum_{i=1}^n \log \left( \frac{p(v_i | y=1)}{p(v_i | y=0)} \right) \quad (12)$$



(a). Updating classifier at the t-th frame



(b). Tracking at the (t+1)-th frame

**Figure. 1.** Main components of the proposed compressive tracking algorithm

### 3 Online random forest

Integrated learning is a robust machine learning algorithms, especially random forest, which has been proved strong anti-interference ability, therefore, when the wrong obstruction sample is updated, it will not cause great impact. Online random forest is mainly with the idea of online learning and integrated learning.

#### 3.1 Random forest

Random forest is composed by a lot of classification and regression trees.

The randomness of the random forest is not only reflected in the choice of the samples, but also in the choice of the random tests. Random forest selects a series of tests randomly from the feature set, and then picking the best test among them according to the

maximum information gain. It is the reason why the random forest is not easy to over-fit.

Given the training set  $X$  and labels  $Y$ , and with an ensemble of trees,  $t_1(x) \dots t_k(x)$  we define the average number of votes at a sample  $(x, y)$ :

$$N_j = \frac{1}{K} \sum_{i=1}^K I(t_i(x) = j) \quad (13)$$

Where  $I$  is the indicator function and  $K$  is the number of the trees, we believe that the larger distance between the average number of the right class and any other class, the more confidence the classifier is. So we define the generalization error.

$$E = P_{X,Y}((N_Y - \max_{j \neq Y} N_j) < 0) \quad (14)$$

It also has been shown by Breiman [4] that as the number of trees increases, for almost surely all sequences  $\theta$ ,  $E$  converges to

$$E_1 = P_{\theta}(h(X, \theta) = Y) \quad (15)$$

$$E_2 = \max_{j \neq Y} P_{\theta}(h(X, \theta) = j) \quad (16)$$

$$P_{X,Y}((E_1 - E_2) < 0) \quad (17)$$

This result explains that the random forest do not over-fit as more trees as added. It has also been shown by Breiman [5] that this error has an upper bound

$$E \leq \rho \frac{1 - S^2}{S^2} \quad (18)$$

Where  $\rho$  is the mean correlation between pairs of trees in the forest and  $s$  is the strength of the ensemble.

### 3.2 Online learning

In the tracking process, the samples are updated with the image sequences continuously. We cannot get the entire training set in advance. What we need to do is to train the classifier with the samples which arrive sequentially, so that it would converge to the off-line model. Online bagging provides a good idea for the online updated.

Given a training set  $T$  of size  $N$ , the probability of

every sample which is selected randomly  $k$  times is

$$P(K = k) = C_N^k \left(\frac{1}{N}\right)^k \left(\frac{1}{N}\right)^{N-k} \quad (19)$$

When  $N \rightarrow \infty$ , the probability distribution is a Poisson distribution, so each classifier is updated by each sample  $k$  times where  $k$  is generated randomly from the Poisson distribution.

In the online random forest, there exist another important problem that we should tell the node when it is appropriate to perform a split. Firstly, the samples are gathered over times, so we should know whether there are enough samples for a node to perform a split. Secondly, the splits are whether good enough for classification purpose. It has been shown by Amir Saffari [6] that if a node splits when  $|R_j| > \alpha$  and  $\exists s \in S : \Delta L(R_j, S) > \beta$ , it will converge to the off-line model. Where  $\alpha$  is the minimum number of samples. The  $\beta$  is the minimum information gain and  $\Delta L(R_j, S)$  is representation of the information gain.

### 4 Improved method

The parameters of naive Bayes classifier will continue to be updated in the tracking process, However, when the target is in the event of partial occlusion or occlusion of all time, the noisy samples will be introduced into the model, and previous samples will be forgotten. The general solution is to judge when the target is under occlusion, so to design different update rates. But the finding of shelter materials will not only cause a large degree of complex algorithms, moreover, it does not guarantee the accuracy. However, it has been shown by Kai Hua Zhang [7] that the compressive tracking provides a good image representation way for the tracking problem. So we use it to make a discriminative representation of the samples in our algorithm.

The online random forest selects the partial samples and features from the total training set for the training of each trees, therefore, most of the trees will not be polluted by the noise. This is the most important reason why the online random forest is not straightforward to over-fit. Meanwhile, it has been proved theoretically in

(18) that the generalization error of random forest classifier has an upper bound.

Moreover, we can estimate the Out-Of-Bag-Error of the trees, through the Out-Of-Bag-Error, we propose to discard the tree which is polluted by the noisy samples, so it can reduce the influence of the factors such as occlusion, appearance changes, illumination change.

Eventually, when the target is completely blocked even severely blocked, there are no candidate positive samples in current frame. Therefore, it exists a situation that the output of each tree will be zero .the result can help us make a decision whether the target is completely blocked. What we need to do next is two things. Firstly, we should abandon the noisy samples we collect from the current frame. The second thing is to use the motion information of the target in the last several frames, including the target velocity, acceleration and other information to predict the target position.

The entire proposed algorithm is depicted in Algorithm.

Algorithm: improved fast compressive tracking based on online random forest

Require: the t-th and (t+1)-th image frame.

Require: the number of positive samples and negative samples in every frame N.

Require: the number of the trees in the forest M.

Require: the minimum number of the samples  $\alpha$ .

Require: the minimum gain  $\beta$ .

(1). According to the position of the target in the t-th frame, we collect positive and negative samples in

$$P^c = \{p \mid \|I(p) - I(t)\| < \gamma^c\}, N^{d,f} = \{p \mid \gamma^d < \|I(p) - I(t)\| < \gamma^f\}$$

with  $d < c < f$  and use a series of rectangle filters defined in (4) to convolve the original image and extract the compressed features trough the random projection matrix.

(2). Input the training samples into the random forest classifier.

For i=1: N

For j=1: M

K  $\leftarrow$  *possession*(1)

For k=1: K

If (M>0)

Find the node which the sample belongs to, and then update the node l=find leaf (i);

If (  $|R_j| > \alpha$  and  $\exists s \in S : \Delta L(R_j, S) > \beta$  )

Find the best test to create the left child and right child;

End

Else

Estimate the Out-Of-Bag-Error (OOB);

End

End

End

End

(3). Choose the candidate samples from the (t+1) frame in  $F^f = \{p \mid \|I(p) - I(t+1)\| < \gamma^f\}$  by shifting number of pixels  $\Delta_f$  and input them into the trained classifier.

For the candidate samples in the current frame, if all the responses of the trees are zero which means all the candidate samples are considered negative samples.

If (output [1...M] ==0)

Predict the position of the target and discard the samples collected from the current frame through.

$$I_{t+1} = 2 * I_t - I_{t-1}$$

Else

Find the tracking position  $I(t+1)^*$  in the current frame with the most votes. Choose the candidate samples from the (t+1) frame in  $F^g = \{p \mid \|I(p) - I(t+1)^*\| < \gamma^g\}$  by shifting number of pixels  $\Delta_g$  and input into the trained classifier. And find the tracking position  $I(t+1)^*$  in the current frame with the most votes.

End

(4). select the trees from the forest randomly F

If (F.OOB>rand ())

Discard (F);

New (F);

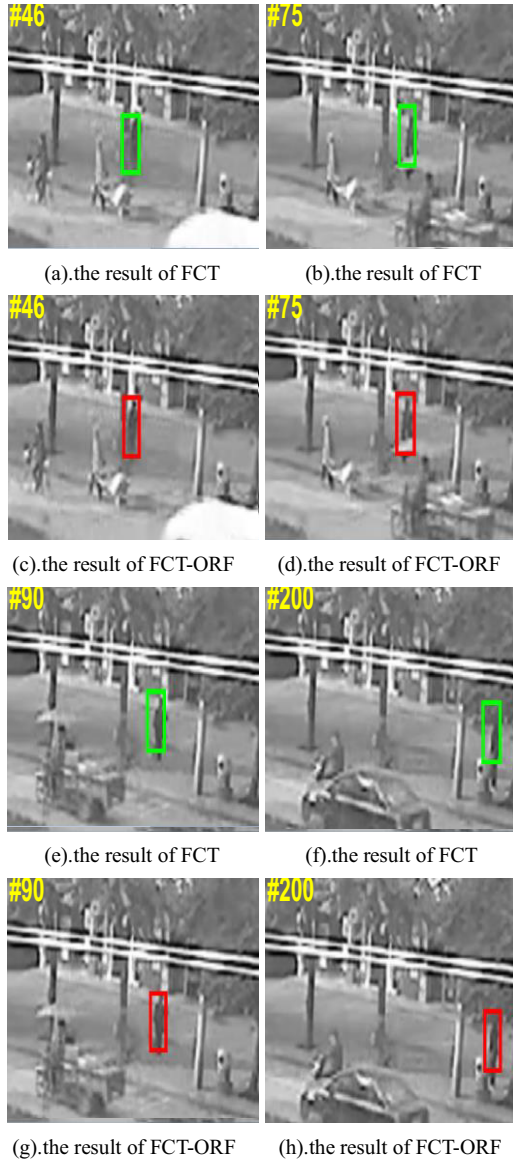
End

## 5 Experimental result

In the experiment, the dimension of the compressed feature is set to be 50. The number of the trees is 100. The maximum depth of the tree is 5. And choose 10 tests from the test set to split the node. The minimum number of the samples is 100.

The dataset consists of two sequences. It is a surveillance video from the CMBC, public sequence

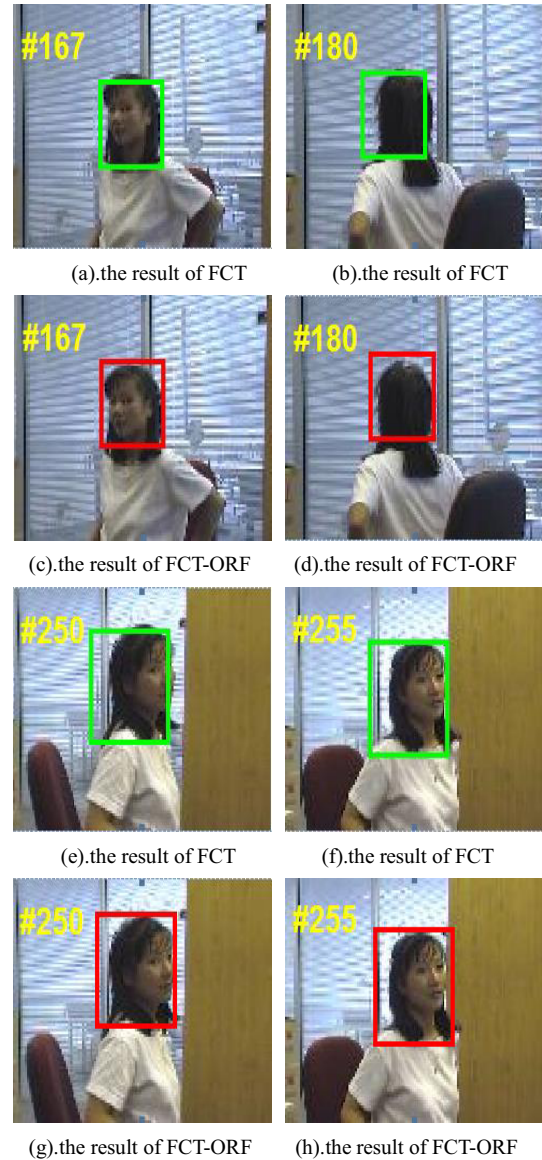
‘Rotating Girl’ respectively, which are presenting pose, scale variation and occlusion. As we can see from the result, our method tracks the target more accurately than FCT algorithm, when the target is under occlusion and pose variation. The results of the two sequences are showed in Figure 2 and Figure 3 respectively.



**Figure.2.**The results of the surveillance video from the CMBC.

In the sequence from the CMBC bank, the people we track is under the occlusion of the trees, the motorbike and the trash which means it will brow the noisy samples. If we do not discard the noisy samples and update the naïve Byes classifier with them. The result would be like this in the 46th 75th and 200th frame. However, Our method can handle with this problem well. Because even parts of the trees and the splitting of the nodes in the forest choose quite a lot of noise samples and the

polluted features for the training, leading to failure of prediction. Other trees updated by the clean samples and features will still maintain a good classification performance. Finally, their common voting will also be able to predict the correct position of the target.

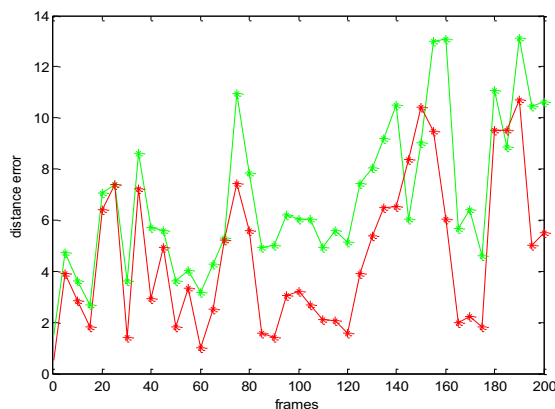


**Figure.3.**The results of the rotating girl.

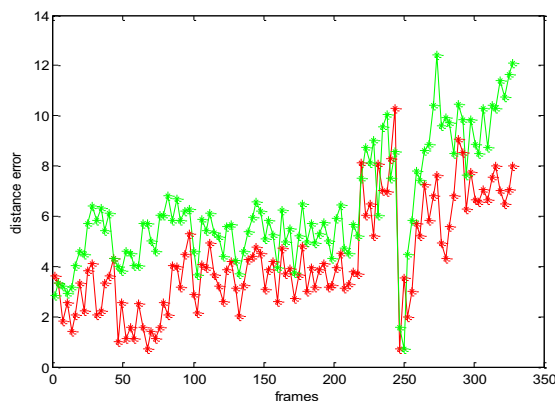
In the compressive tracking algorithm, when the girl turns the face quickly, we need a faster update rate to adapt to the changes in the appearance. However, with the high update rate, it is easy to introduce the noisy. Thus, leading to the drift in the tracking. As you can see the result in the 180th frame. Meanwhile, we use a smaller update rate to do the experiment, the result is showed in the 250th and 255th frames, the drift occurs for the sake of the update rate, which cannot adapt to the changes in the appearance. However, our method abandons the trees, which is updated by original samples

through estimating the Out-Of-Bag-Error adaptively. And the new trees are created with the current samples to accommodate with the changes in the appearance.

We evaluate the result with the center location error which is defined as the Euclidean distance between the central locations of the tracked objects and the manually labeled ground truth. The result of the sequences from the CMBC and Rotating girl are showed in Figure 4 and Figure 5 respectively.



**Figure.4.** the sequences from the ICBC — FCT — FCT-ORF



**Figure.5.** the rotating girl — FCT — FCT-ORF

## 6 Conclusion

In this paper, we proposed an improved compressive tracking algorithm based on online random forest. The simulation results show that our algorithm performs better than the original algorithms when the target is under occlusion, pose variation and appearance changes. The reasons are that we adopt the adaptive compressive sensing theory to make a discriminative representation of the target, moreover, the random forest classifier is robust to the noise for selecting the features and samples

to train the decision trees randomly. Finally, The Out-Of-Bag-Error can tell us whether a decision tree is polluted completely and then discard it.

## References

1. K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in Proc. Eur. Conf. Comput. Vis., 2012, pp. 864–877.
2. K. Zhang, L. Zhang, and M.-H. Yang, "Fast Compressive Tracking," in IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014)1631-1643.
3. P. Diaconis and D. Freedman, "Asymptotics of graphical projection pursuit," Ann. Statist., vol. 12, pp. 793–815, 1984.
4. L. Breiman. Out-of-bag estimates. Technical report, 1996.
5. L. Breiman. Random forests. Machine Learning, 45(1):5–32, October 2001.
6. A. Saffari, C. Leistner, J. Santner, M. Godec, H. Bischof, "On-line Random Forest," in International Conference Computer Vision, 2009, pp.1393-1400.
7. H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting" in Proc. British Mach. Vis. Conf, 2006, pp. 47–56.