DOI: 10.1051/matecconf/201712801009

# Adaptive Scale Compressive Tracking with Feature Integration

Mingqi Luo[1,a], Tuo Wang[2] and Bin Zhou[1]

[1]*The school of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shanxi, China*
[2]*Xi'an Jiaotong University Suzhou Academy, Suzhou, Jiangsu, China*

**Abstract.** Compressive tracking (CT) is utilized to cope with real-time tracking, which use a very sparse measurement matrix to compressive samples of targets and background, then a classifier is trained to distinguish foreground and background. However, this algorithm suffers from the drifting problem, and used the fixed size tracking box to detect, recognize, and update the samples and classifier. In order to solve these problems, we adopt a different way to extracted positive samples, and employ powerful features to exploit the advantages of feature fusion to describe target, a scale pyramid is used to realize adaptive scale tracking. Experimental results on various benchmark video sequences demonstrate the superior performance of our algorithm.

## 1 Introduction

Object tracking is one of the most important subjects in computer vision for its various applications in video surveillance, robotics, human computer interaction and driver-less vehicle. The job of tracking is to estimation the target states in subsequent frames under the condition that the initialized state of the target in the start frame has been know via a sequence of measurements made on the object. Despite significant progress has been made in recent years, the problem is still a tough difficult due to partial occlusion, geometric deformation, motion blur, illumination changes, background clutter and scale variations [1]. Tracking methods with fixed models of target usually fail because of the unavoidable changes of appearance. This is why the online learning methods are used to handle the variations in appearance. The learning methods that used by the online tracking can be divided into generative and discriminative methods. In generative online learning methods, the appearance of model for target is updated adaptively in response to appearance variations [2, 3, 4]. The discriminative methods utilize information about the target and the background simultaneously and tracking is treated as a classification problem [5, 6].

The sparse representation and compressed sensing techniques have attracted a great deal of attention in visual tracking. These algorithms represent target as a sparse combination of target templates. Zhang [7]proposed an effective tracking algorithm with a discriminative model, named compressive tracking, known as CT, adopts a very sparse measurement matrix to efficiently extract the features for the appearance model which makes is possible for real-time tracking and achieves relatively good results on some challenging video sequences. However, the algorithm suffers from the drifting problem because it just using a single feature to represent target, we can employ various powerful features to exploit the advantages of feature fusion. Secondly, the CT algorithm used the fixed size tracking box to detect, recognize and update the samples and classifier. Obviously, in practical, the target scale usually changed with the target moving. The fixed size tracking box easily caused the target lost when the target changed significantly in scale and it was not suitable in practical application. In [8], authors present a multi-scale compressive tracker. This tracker integrates an improved appearance model based on normalized rectangle features extracted in the adaptive compressive domain into the bootstrap filter. The compressive features of a particle in special scale were achieved by the projection via a modified adaptive random measurement matrix. This matrix is updated based on the initial matrix and the corresponding particle's current scale value. A 2-order transition model which considers two previous statuses has been used to estimate the current position and scale status for each particle. In [9], presents a novel enhanced compressive tracking. It not only utilizes the instances at the current frame, but takes the appearance of target in previous frames into account, experiments proved this way can obtain a much more effective classifier to detect object at each frame. In order to deal with the target appearance change, in [10], constructed a feature template using online vector boosting, because it filtering the noisy, this algorithm can deal with target appearance change. In [11], the motion estimator was introducing into appearance model in compressive space.

In this paper, we have proposed an effective and efficient tracking algorithm. The key contribution of this work can be summarized as follows. Firstly, we employ various powerful features to exploit the advantages of feature fusion. There are two types of features used in our

[a] Corresponding author: lmq232x@163.com

proposed tracker. Secondly, when the tracking window determined, a certain number of positive samples are extracted by the Gaussian distribution around the point of the canter of the tracking window. Thirdly, in order to estimation the scale accurately, a scale pyramid is used.

## 2 Compressive tracking

The compressive sensing (CS) theory states that an original high-dimensional sparse or compressible signal can be reconstructed from a low-dimensional signal, that is to say, 'random projection' and the 'random measurement matrix' are two important concepts involved with compressive sensing. From the compressed sensing theory[12], for a measurement matrix $R \in R^{n\times m}$ , $n << m$ if we want to use $R$ to compress high-dimensional signals (high-dimensional signals are compressible, such as image, video, audio, etc.) to low signals, while the low-dimensional signal can contain almost all the information, $R$ should satisfy the sparse and restricted isometry property(RIP). Baraniuk et al. [13] proved that the random matrix satisfying the Johnson-Lindenstrauss lemma also holds true for the restricted isometry property in compressive sensing. Therefore, if the random matrix in (1) satisfies $R$ the Johnson-Lindenstrauss lemma, it can reconstruct $x$ with minimum error from v with high probability if $x$ is compressive such as audio or image.

$$v = Rx \tag{1}$$

In CT, in order to tackle scale problems, a set of rectangle filters at multiple scales $\{h_{1,1},...,h_{w,h}\}$ , defined as

$$h_{i,j}(x,y) = \begin{cases} 1, 1 \le x \le i, 1 \le y \le j \\ 0, otherwise \end{cases} \tag{2}$$

Where $i$ and $j$ are the width and height of a rectangle filter, respectively, is used to convolve $z$ , $z \in R^{w\times h}$ . Thus, a high-dimensional multi-scale image feature vector $x = (x_1, x_2, ..., x_n) \in R^n$ , where $n = (wh)^2$ can be obtained. In [14], a random Gaussian matrix $R \in R^{n\times m}$ , which is a typical matrix satisfying restricted isometry property(RIP), is utilized to reduce the dimensionality. The entries in $R$ are defined as

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 \quad with \quad probability \dfrac{1}{2s} \\ 0 \quad with \quad probability \dfrac{1}{2s} \\ -1 \quad with \quad probability \dfrac{1}{2s} \end{cases} \tag{3}$$

In compressive tracking algorithm, the random measurement matrix is computed only once offline and remain fixed throughout the tracking process. Then the feature vector which is projected onto lower-dimensional space would be computed efficiently.

After calculating the features, updating the classifier with these features is followed. Diaconis and Freedman [15] showed that the random projections of high dimensional random vectors are almost always Gaussian. Therefore $\mu$ and $\sigma$ can be used to simulate the random features of rectangular distribution after projection using the equation (4), where $y = 1$ means positive samples, $y = 0$ means negative samples.

$$\begin{cases} p(v_i \mid y=1) \sim N(\mu_i^1, \sigma_i^1) \\ p(v_i \mid y=0) \sim N(\mu_i^0, \sigma_i^0) \end{cases} \tag{4}$$

The scale parameters in(4) are incrementally updated

$$\begin{cases} \mu_i^1 \leftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1 \\ \sigma_i^1 \leftarrow \sqrt{a} \end{cases} \tag{5}$$

Where

$$a = \lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2 \tag{6}$$

$$\begin{cases} \sigma^1 = \sqrt{\dfrac{1}{n} \sum\limits_{k=0|y=1}^{n-1} (v_i(k) - \mu^1)^2} \\ \mu^1 = \dfrac{1}{n} \sum\limits_{k=0|y=1}^{n-1} v_i(k) \end{cases} \tag{7}$$

$$H(v) = \log\left(\frac{\prod_{i=1}^{n} p(v_i \mid y=1) p(y=1)}{\prod_{i=1}^{n} p(v_i \mid y=0) p(y=0)}\right) = \sum_{i=1}^{n} \log\left(\frac{p(v_i \mid y=1)}{p(v_i \mid y=0)}\right) \tag{8}$$

By integrating the equation (5) and (7) to update the classifier, the $\lambda > 0$ is a learning parameter.

The process of searching the target is first to sample some samples around the target in the previous frame. The extracted sample features with different dimensions of the rectangle to convolution filter is every sample of high dimensional feature vector $x$ , and then use the sparse measurement matrix $R$ to do projection to the low dimensional vector $v$ on compressed domain. And $v$ can be calculated by generalized Haar-like features [16]. A naïve Bayes classifier is adopted to classify object as shown in Equation (8). The maximum value of the corresponding sample in $H(v)$ is the target of the next frame image.

## 3 The tracker

When a target is determined by a rectangle in the first frames, let point $X = (x_0, y_0)$ be the center of the rectangle, the number of positive samples center extracted around the center point with Gaussian distribution $N(\mu, \sigma)$ , all of these samples have same size of the current tracking rectangle window.

Color histogram[17] which is used in this paper has been widely used to represent, analyze, and characterize images. They allow for significant data reduction, and can be computed efficiently. Color histograms are robust to noise and local image transformation. In the target tracking domain, color histogram are a popular form of target representation, because of their independence from scaling and rotation, and robust to partial occlusions. Denote by $\{x_i\}_{i=1,\dots,n}$ the normalized pixels in target region. The probability of the feature $u(u=1,\dots,m)$ in the target model is computed as[18]:

$$\begin{cases} q=\{q_u\}_{u=1,2,\dots,m} \\ q_u=C_h\sum_{i=1}^{n}k(\left\|\frac{x_i}{h}\right\|^2)\delta(b(x_i)-u) \end{cases} \tag{9}$$

Where $q$ is the target model, $q_u$ is the probability of the $uth$ element of $q$ , $\delta$ is the Kronecker delta function, $b(x_i)$ associates the pixel $x_i$ to the histogram bin and $k(x)$ is an isotropic kernel profile. Constant $C_h$ is a normalization function defined by:

$$C_h=\frac{1}{\sum_{i=1}^{n_h}k(\left\|\frac{x_i}{h}\right\|^2)} \tag{10}$$

Similarly, the probability of the feature $u$ in the target candidate model from the candidate region centered at position $y$ is given by:

$$\begin{cases} p(y)=\{p_u(y)\}_{u=1,2,\dots,m} \\ p_u(y)=C_h\sum_{i=1}^{n}k(\left\|\frac{y-x_i}{h}\right\|^2)\delta(b(x_i)-u) \end{cases} \tag{11}$$

$$C_h=\frac{1}{\sum_{i=1}^{n_h}k(\left\|\frac{y-x_i}{h}\right\|^2)} \tag{12}$$

Where $p(y)$ is the target candidate model, $p_u(y)$ is the probability of the $uth$ element of $p(y)$ , $\{x_i\}_{i=1,\dots,n_h}$ are pixels in the target candidate region centred at $y$ , $h$ is the bandwidth and $C_h$ is a normalization function.

In order to calculate the likelihood of the target model and the candidate model, a metric based on the Bhattacharyya coefficient[19] is defined by using the two normalized histograms $q$ and $p(y)$ as follows

$$d(y)=\sqrt{1-\rho[\hat{p}(y),\hat{q}]} \tag{13}$$

Where

$$\rho(y)\equiv\rho[\hat{p}(y),\hat{q}]=\sum_{b=1}^{m}\hat{p}_b(y)\hat{q}_b \tag{14}$$

Histogram of oriented gradient (HOG) is one of the most popular visual features in vision community, since it is very effective in practical applications and can be computed efficiently. HOG feature was extracted on the distribution of the edge of targets in local region; it is a good representation of object shapes.

Because the HOG puts emphasis on the image gradient while color histogram focuses on the color information, that is to say, the two features are complementary to each other. The overall algorithm is summarized into Algorithm1.

Algorithm 1 ASFICT

Input t-th video frame

Step1. A certain number of positive samples whose center of those samples obeys the Gaussian distribution around the center point of the previous tracking window are extracted, randomly crop some negative samples far away from the target box, and extracted the features of color histogram and HOG feature.

Step 2. Use the Naïve classifier to each feature vector and find the tracking location with the maximal classifier response.

Step 3. Building the scale pyramid on the current location, extracted the color histogram and HOG feature on each scale pyramid, find the most similarity scale whose features have the most similarity with the previous features.

Step 4. Extracted samples as step1 to updated the classifier just as the CT algorithm.

Output: tracking location and the scale parameters.

## 4 Experiments

In this section, we implement our method on 7 challenging sequences, compared with the latest three algorithms (CT[7], WMIL[20], and FCT[14]). All the video clips are publicly available (all of the sequences can be download on visual tracking benchmark). For fair comparison, we use the source or the binary codes provided by the authors with tuned parameters for the best performance. Our MATLAB implementation currently runs on an Intel(R) 3.30GHz CPU with4.00GB RAM. All the challenging sequences were showed in Table 1.

| Sequence | Main challenge |
|---|---|
| David2 | scale, illumination and pose changes |
| Sylvester | pose changes |
| walking | self-occlusions |
| faceocc2 | particle occlusions |
| tiger1 | comprise illumination and pose variations |
| Tiger2 | motion blur and particle occlusion |
| Lemming | clutter |

**Table 1** The challenging sequences

We use two metrics to evaluate the performance of our tracking method, that is success rate (SR) and center location error (CLE) which are used in FCT tracking algorithm. The success rate is used in the PASCAL VOC challenge [21] defined as,

$$score=\frac{area(ROI_T\bigcap ROI_G)}{area(ROI_T\bigcup ROI_G)} \tag{15}$$

where $ROI_T$ is the tracking bounding box and $ROI_G$ is the ground truth bounding box. If the $score$ is larger than 0.5 in one frame, the tracking result is considered as success. In this paper, the $ROI_T$ and $ROI_G$ are present by its area.Table2 shows the tracking results in terms of success rate. The center location error is defined as the Euclidean distance between the central locations of the tracked objects and the manually labeled ground truth. Table 3 shows the average tracking errors of all methods.

| Sequence | TF | ASFICT -FF | CT-FF | WMI L-FF | FCT-FF |
|---|---|---|---|---|---|
| Tiger1 | 353 | 3 | 77 | 32 | 3 |
| Tiger2 | 364 | 20 | 48 | 28 | 18 |
| Walking | 411 | 80 | 68 | 50 | 116 |
| David2 | 536 | 40 | 10 | 3 | 58 |
| Sylvester | 1000 | 7 | 9 | 63 | 15 |
| Faceocc2 | 813 | 0 | 0 | 2 | 0 |
| Lemming | 299 | 20 | 29 | 0 | 21 |
| A-SR | -- | 24.2 | 34.4 | 25.4 | 33 |

**Table 2**

| Sequence | TF | ASFICT -FF | CT-FF | WMIL-FF | FCT-FF |
|---|---|---|---|---|---|
| Tiger1 | 353 | 22 | 37.0 | 24 | 22.4 |
| Tiger2 | 364 | 27 | 47.4 | 31 | 37.7 |
| Walking | 411 | 30.1 | 22.5 | 38 | 28.9 |
| David2 | 536 | 15.9 | 20 | 7.1 | 16.9 |
| Sylvester | 1000 | 8.6 | 9.0 | 9.1 | 8.5 |
| Faceocc2 | 813 | 16.3 | 7.3 | 16 | 16.3 |
| Lemming | 299 | 30.3 | 22 | 39 | 24.3 |
| A-SR | -- | 21.4 | 23.6 | 23.4 | 22.1 |

**Table 3**

Where TFrames is the total frames of single sequences, FF is the number of failed tracking frames, A-SR it the average success rate, A-CLE is the average CLE. We note that our method has lower fail tracking frames and less average CLE.

Fig1 shows the error plots of OUR algorithm, compressive tracking algorithm, fast compressive tracking algorithm, and weighted multiple instance learning tracking algorithm applied to the lemming, tiger2 and walking tested sequences.

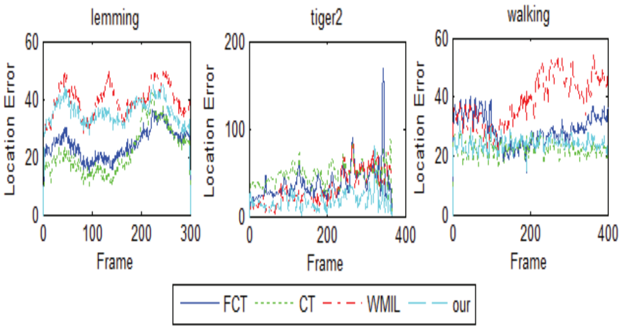

**Fig 1**

According to the experimental results showed by Table2, Table3, we can observe that our algorithm achieves the superior performance for most of test sequences in terms of the illumination, clutter and blurs motion. Compared with other state-of-art algorithm, ASFICT algorithm is more robust and effective.

## 5 Conclusions

In this paper, we proposed an effective and efficient tracking algorithm based on CT algorithm. In order to exploit the advantages of feature fusion, we choose the color histogram and HOG feature to descript target. Because in practical, the target scale usually changed with the target moving, then we use the scale pyramid to achieve adaptive compressive tracking. Experimental results on many challenge video clips, showed superior performance of our algorithm.

## Acknowledgment

Research supported by National Natural Science Foundation of China (No.61071217) and Jiangsu Science and Technology Program Foundation (No.BK20141216)

## References

1. Yilmaz,O.Javed,M.Shah, ACM, Computing Surveys **38**(2006).
2. Ross, D., Lim, J., Lin, R., Yang, M.-H. IJCV **77**(2006)
3. Li, H., Shen, C., Shi, Q.CVPR(2011)
4. Mei, X., Ling, H.,PAMI,33(2011)
5. Grabner, H., Grabner, M., Bischof,(2006)
6. Babenko, B., Yang, M.-H., ,PAMI**33**(2011)
7. K. Zhang, L. Zhang, and M. Yang,ECCV(2012)
8. Yunxia Wu · Ni Jia · Jiping Sun,31(2015)
9. Rui Xu, Xiaodong Gu,IAPR(2013)
10. Qingshan Liu, IEEE Trans Cybern,**1**(2016)
11. Eric Wang, Jorge Silva, 2009 IEEE/SP
12. D. Donoho. IEEE Trans. Information Teory,**4**(2006)
13. Diaconis, P., Ann. Stat.**12**(1984)
14. K. Zhang, L. Zhang, IEEE transactions on pattern analysis and machine intelligence.**36**(2014)
15. E.J. Candes and T. Tao. IEEE Transactions on Information Theory, 12(2005).
16. Viola P, Jones M,CVPR(2001)
17. P.Perez, C.Hue, J.Vermaak,ECCV**1**(2002)
18. D.Coumaniciu, V.Ramesh, IEEE Trans. on Pattern Analysis and Machine Intelligence, **25**(2003)
19. Kailath,T., IEEE Trans. Commun.Technol.,**1**(1967)
20. K. Zhang, H. Song. Pattern Recognition, **46**(2013)
21. Yi Wu , Jongwoo Lim. CVPR,(2013)