Journal of Informatics and Mathematical Sciences
Vol. 8, No. 3, pp. 201–208, 2016
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications

http://www.rgnpublications.com

Robust Multiple Instance Learning Fast Compressive Tracking

Research Article

Li Hong Wang*, Rui Min Wu and Jin Lin Gao

Longqiao College of Lanzhou University of Finance and Economics, Lanzhou 730101, China
***Corresponding author:** lmq232x@163.com

Abstract. Fast compressive tracking algorithm performs more effective and robust than some other state-of-art tracking algorithm, it crop samples from the current frame, all these samples have the same weighted in learning procedure, in order to integrates the sample importance into the learning procedure, motived by the weighted multiple instance learning algorithm, we present a novel enhanced fast compressive tracking, which integrates the samples importance into learning procedure. Experimental results on various benchmark video sequences demonstrate the superior performance of our algorithm.

Keywords. Object tracking; Fast compressive tracking; Multiple instance learning

MSC. 05C50; 05C90; 94C15

Received: September 22, 2016 **Accepted:** October 5, 2016

1. Introduction

Object tracking is a well-studied problem in computer vision and has many practical applications, e.g., automated surveillance, video indexing, traffic monitoring, human computer interaction, and vehicle navigation, etc. During past decades, various algorithms have been proposed to tackle this problem, but it still has many challenging factors, such as illumination changes, occlusions, pose, motion, appearance changes, and real time processing requirements, may affect the performance of these methods [15]. An effective appearance model is very important for a successful tracking algorithm. Tracking algorithm can be generally categorized as either generative [9–11] or discriminative [2, 7] based on their appearance models.

Generative tracking algorithms learn a template to represent the target object, then use it to search for the smallest area of the image reconstruction error. The IVT algorithm [11] utilizes an incremental subspace model to adapt appearance changes. Sparse representation also has been used in the ℓ_1-tracker where an object is modeled by a sparse linear combination of target and trivial templates [10]. Li *et al.* [9] further extend the ℓ_1-tracker by using the orthogonal matching pursuit algorithm for solving the optimization problems efficiently. There are still some problems that remain to be solved, although these generative tracking algorithms have made great progress. First, these generative tracking algorithms do not use the background information which is likely to improve tracking stability and accuracy. On the other hand, the drift problems is likely to occur if the appearance model of the target has large number of changes during continuous frames at the outset. Because it is often assumed that the target is not changed much during that time.

Discriminative models, that are also called tracking-by-detection methods, pose the tracking problem as a binary classification task in order to find the decision boundary for separating the target object from the background. In [1], S. Avidan utilized boosting methods to train a strong classifier and then used mean shift method to find the location with maximum classifier score at each frame. Motivated by using Haar features to represent faces andutilizing Adaboost methods to train classifiers in [13], H. Grabner *et al.* presented OAB [7] and SemiT [8] algorithms based on the online features selection. In [2], an online *Multiple Instance Learning* (MIL) algorithm is proposed, which uses multiple positive samples and negative ones to train classifiers. *Weighted Multiple Instance Learning* (WMIL), as improved MIL tracker, is introduced by [16]. *Compressive Tracking* (CT) [17] finds a sparse measurement matrix to extract features of positive samples and negative ones, after dimensionality reduction, a binary classifier is trained to distinguish foreground and background. K. Zhang [18] propose an effective and efficient tracking algorithm with an appearance model based on features extracted in the compressed domain. They use a very sparse measurement matrix that satisfies the *Restricted Isometry Property* (RIP) in compressive sensing theory, there by facilitating efficient projection from the image feature space to a low-dimensional compressed subspace. For tracking, the positive and negative samples are projected with the same sparse measurement matrix and discriminated by a simple naive Bayes classifier learned online. This algorithm is also named fast compressive tracking, that is FCT.

Although FCT performs more effective and robust than some other state-of-art tracking algorithm, it still utilizes positive samples of current frame to train classifier. However, these positive samples do not discriminatively consider the importance in its learning procedure. Motivated by the idea of WMIL, in this paper, we proposed a weighted instances learning fast compressive tracking algorithm. The paper is organized as follows: in Section 2, we introduce the related works of WMIL and FCT algorithm. Section 3 describe our tracking systems. Section 4 gives the detailed experiment setup and results. Finally, Section 5 concludes the paper.

2. Preliminaries

The basic flow of MILTrack algorithm is as follows. When a new frames comes, a set of image patches that are close to the location of tracking object in the last frame are cropped out. A greedy strategy is used to find a best image patch as tracker location from the set with the MIL classifier. After locating the object, weak classifiers are updated by training data in the form of bags. Then, appearance model is updated with MILBoost algorithm.

Unlike traditional supervised learning algorithm, Multiple Instance Learning represents the training examples in bags $\{(X_1,Y_1),(X_2,Y_2),\dots,(X_n,Y_n)\}$, where $X_i = \{(x_{i1},x_{i2},\dots,x_{im})\}$, is a bag and y_i is the label of X_i, x_{ij} is an instance in X_i. If a bag contains at least one positive instance, it is labeled positive, otherwise labeled negative. MILBoost [12] is used to solve MIL problem in MILTrack algorithm. MILTrack uses randomly generated Haar-like features to represent image patch, which can be computed fast with integral image trick, and weak classifiers are based on Bayes model. In order to integrates the samples importance into the learning procedure, a new bag probability function that combines the weighted instance probability: the weight for the instance near the object location is larger than that far from the object location which means the instance near the object location contributes larger to the bag probability is proposed, which named *Weighted Multiple Instance Learning* (WMIL). From the compressed sensing theory [5], for a measurement matrix $R \in R^{n\times m}$ $(n \ll m)$, if we want to use R to compress high-dimensional signals x (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 in the x, Rshould satisfy the sparse and *Restricted Isometry Property* (RIP). Baraniuk *et al.* [4] 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 R in (2.1) satisfies 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.

$$v = Rx. \tag{2.1}$$

For each sample $Z \in R^{w\times h}$, its multiscale representation is constructed Z by convolving with a set of rectangle filters at multiple scales $\{h_{1,1},\dots,h_{w,h}\}$, defined as

$$h_{i,j}(x,y) = \begin{cases} 1, & 1 \le x \le i,\ 1 \le y \le j, \\ 0, & \text{otherwise}, \end{cases} \tag{2.2}$$

where i and j are the width and height of a rectangle filter, respectively. In [18], 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

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 & \text{with probability } \frac{1}{2s}, \\ 0 & \text{with probability } 1-\frac{1}{s}(3), \\ -1 & \text{with probability } \frac{1}{2s}. \end{cases} \tag{2.3}$$

The random measurement matrix R used to be computed only once offline and remains fixed throughout the tracking process. After calculating the features, updating the classifier with these features is followed. Diaconis and Freedman [3] showed that the random projections of high dimensional random vectors are almost always Gaussian. Therefore, μ and σ can be used to simulate the random features of rectangular distribution after projection using the equation (2.5), where $y=1$ means positive samples, $y=0$ means negative samples.

$$\begin{cases} p(v_i \mid y=1) \sim N(\mu_i^1, \sigma_i^1), \\ p(v_i \mid y=0) \sim N(\mu_i^0, \sigma_i^0). \end{cases} \tag{2.4}$$

The scale parameters in (2.3) are incrementally updated

$$\begin{cases} \mu_i^1 \leftarrow \lambda \mu_i^1 + (1-\lambda)\mu^1, \\ \sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2}\ , \end{cases} \tag{2.5}$$

$$\begin{cases} \sigma^1 = \sqrt{\dfrac{1}{n} \sum\limits_{k=0, y=1}^{n-1} (v_i(k) - \mu^1)^2}\ , \\ \mu^1 = \dfrac{1}{n} \sum\limits_{k=0, y=1}^{n-1} v_i(k), \end{cases} \tag{2.6}$$

$$H(v) = \log\left(\frac{\prod\limits_{i=1}^{n} p(v_i \mid y=1) p(y=1)}{\prod\limits_{i=1}^{n} p(v_i \mid y=0) p(y=0)} \right) = \sum_{i=1}^{n} \log\left(\frac{p(v_i \mid y=1)}{p(v_i \mid y=0)} \right). \tag{2.7}$$

By integrating the equation (2.5) and (2.6) to update the classifier, the $\lambda > 0$ is a learning parameter. A coarse-to-fine search strategy is adopted to further reduce the computational complexity in the detection procedure.

3. Proposed Algorithm

We assume that the tracking window in the first frame has been determined. At each frame, we sample some positive samples near the current target location and negative samples far away from the object center to update the classifier. To predict the object location in the next frame, we draw some samples around the current target location and determine the one with the maximal classification score. Obviously, the FCT tracking algorithm do not considered the sample importance into the learning procedure, motived by the idea of WMIL, we combination the sample importance into FCT tracking algorithm based on the fact that the weight for instance near the object location is larger than that far from the object location. The main steps of our algorithm are summarized in Algorithm 1. Let $\ell_t(x) \in R^2$ denote the location of sample x at the t-th frame.

Algorithm 1 (The proposed algorithm).

Input: t-th video frame

Output: Tracking location $\ell_t(x^*)$ and classifier parameters

1. Coarsely sample a set of image patches that is positive samples in $D^{\gamma_c} = \{Z \mid \|\ell(Z) - \ell_{t-1}\| < \gamma_c\}$, where ℓ_{t-1} is the tracking location at the $(t-1)$th frame by shifting a number of pixels Δc.
2. Randomly crop some negative samples from set $X^{\zeta,\beta} = \{x \mid \zeta < \|\ell_t(x) - \ell_t(x_0)\| < \beta\}$, where $\gamma_c < \zeta < \beta$.
3. Extract the feature vector $v(Z)$ with low-dimensionality.
4. Use the classifier H in (2.7) to each feature vector $v(Z)$ and find the tracking location ℓ'_t with the maximal classifier response.
5. Finely sample a set of image patches in $D^{\gamma_f} = \{Z \mid \|\ell(Z) - \ell'_t\| < \gamma_f\}$ by shifting a number of pixels Δf, and extract the feature vector $v(Z)$ with low dimensionality.
6. Use the classifier H in (2.7) to each feature vector $v(Z)$ and find the tracking location ℓ_t with the maximal classifier response.
7. Sample two sets of image patches $D^{\alpha} = \{Z \mid \|\ell(Z) - \ell_t\| < \alpha\}$ and $D^{\zeta,\beta} = \{Z \mid \zeta < \|\ell_t(x) - \ell_t(x^*)\| < \beta\}$, with $\alpha < \zeta < \beta$ and extract the feature vector $v(Z)$ with these two sets of samples.
8. Update the classifier parameters according to (2.5).

4. Experiments

4.1 Parameters Setting

We assume that the tracking window in the first frame has been determined. Given the target location at the current frame, the search radius for drawing positive samples α is set to 4 which generate 45 positive samples. The inner ζ and out radii β for $D^{\zeta,\beta}$ the set that generate negative samples are set to 8 and 30, respectively. In addition, 50 negative samples are randomly selected from the set $D^{\zeta,\beta}$. The search radius γ_c for the set D^{γ_c} to coarsely detect the object location is 25 and the shifting step Δc is 4. The radius γ_f for the set D^{γ_f} to fine-grained search is set to 10 and the shifting step Δf is set to 1. The dimensionality of projected space n is set to 100, and the learning parameter λ is set to 0.85. All of the parameters are set the same as the FCT tracking algorithm.

4.2 Experimental Results

All the video clips are publicly available (all of the sequences can be download on [14]). David2 sequence and Sylvester sequence are used to test the scale, illumination and pose changes, walking sequence and Faceocc2 sequence are for self-occlusions or particle occlusions, Tiger1 and Tiger2 sequences are used for comprise illumination, pose variations, motion blur, and particle occlusion at the same time which make them very challenging. Lemming sequence is used for test the clutter. For fair comparison, we use the source or the binary codes provided by the authors with tuned parameters for the best performance. 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 [6] defined as, score = $\frac{\text{area}(ROI_T \cap ROI_G)}{\text{area}(ROI_T \cup ROI_G)}$, 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. Table 1 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 2 shows the average tracking errors of all methods.

Table 1. The success rate

Sequence	TFrames	Ours-FF	CT-FF	WMIL-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. Center location error

Sequence	TFrames	Ours	CT	WMIL	FCT
Tiger1	353	22.0	37.0	24.0	22.4
Tiger2	364	27.0	47.4	31.0	37.7
Walking	411	30.1	22.5	38.0	28.9
David2	536	15.9	20.0	7.1	16.9
Sylvester	1000	8.6	9.0	9.1	8.5
Faceocc2	813	16.3	7.3	16.0	16.3
Lemming	299	30.3	22.0	39.0	24.3
A-CLE	–	21.4	23.6	23.4	22.1

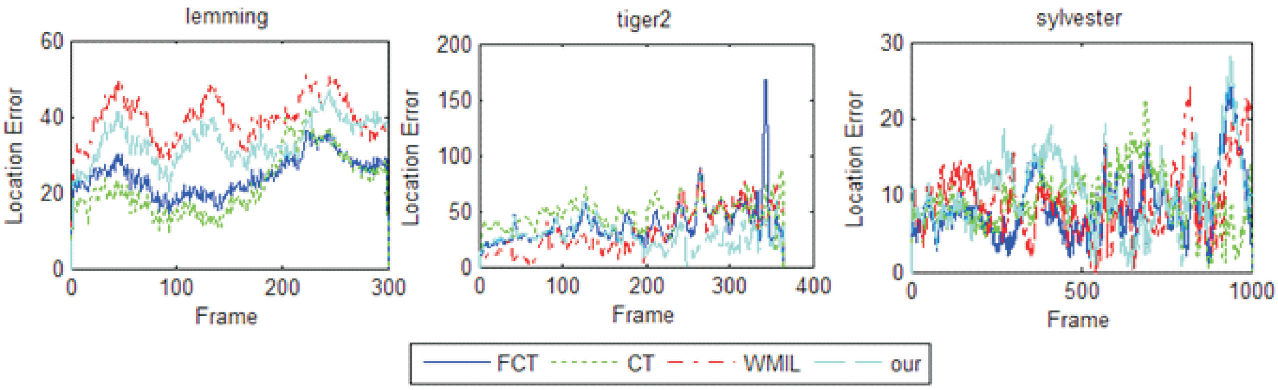

Figure 1

Figure 1 shows the error plots of OUR algorithm, CT algorithm, FCT algorithm, and WMIL algorithm applied to the lemming, Tiger2 and Sylvester tested sequences.

According to the experimental results showed by Table 1, Table 2 and Figure 1, 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, our algorithm is more robust and effective.

5. Conclusion

In this paper, motivated by the weighted multiple instance learning tracking algorithm, in order to integrates the samples importance into the learning procedure, we fusion the bag probability function that combination the weighted instance probability into fast compressive tracking algorithm to improve the tracking performance. Experimental results on many challenge video clips, compared with fast compressive tracking algorithm, compressive tracking algorithm and the weighted multiple instance tracking algorithm which are more robust and effective than other state-of-art tracking algorithms, showed that our method is robust and effective to handle clutter, blue motion, and easier to achieve stable and accurate tracking results.

References

[1] S. Avidan, Ensemble Tracking, *IEEE Trans. on Pattern Analysis* **29** (2) (2008), 261–271.

[2] B. Babenko, M.-H. Yang and S. Belongie, Robust object tracking with online multipleinstance learning, *IEEE Trans. on Pattern Analysis* **33** (2011), 1619–1632.

[3] E.J. Candes and T. Tao, Decoding by linear programming, *IEEE Transactions on Information Theory* **51** (12) (2015), 4203–4215.

[4] P. Diaconis and D. Freedman, Asymptotics of graphical projection pursuit, *Ann. Stat.* **12** (1984), 228–235.

[5] D. Donoho, Compressed sensing, *IEEE Trans. Information Teory* **52** (4) (2006), 1289–1306.

[6] M. Everingham, L. Gool, C. Williams, J. Winn and A. Zisserman, The pascal visual object class (voc)challenge, *International Journal of Computer Vision* **88** (2) (2010), 303–338.

[7] H. Grabner, M. Grabner and H. Bischof, Real-time tracking via online boosting, *The British Machine Vision Conference* (2006), 47–56.

[8] H. Grabner, C. Leistner and H. Bischof, Semi-supervised onlineboosting for robust tracking, *The European Conference on Computer Vision* (2008), 1–10.

[9] H. Li, C. Shen and Q. Shi, Real-time visual tracking using compressive sensing, *IEEE Conference on Computer Vision and Pattern Recognition* (2011), 1305–1312.

[10] X. Mei and H. Ling, Robust visual tracking and vehicle classification via sparse representation, *IEEE Trans. on Pattern Analysis* **33** (2011), 2259–2272.

[11] D. Ross, J. Lim, R. Lin and M.-H. Yang, Incremental learning for robust visualtracking, *International Journal of Computer Vision* **77** (2008), 125–141.

[12] P. Viola, J. Platt and C. Zhang, Multiple InstanceBoosting for Object Detection, *Advances in Neural Information Processing Systems* **18** (2006), 1417–1424.

[13] P. Viola and M. Jones, Rapid object detection using a boosted cascade ofsimple features, *IEEE Conference on Computer Vision and Pattern Recognition* (2001), 1–9.

[14] Y. Wu, J.W. Lim and M.-H. Yang, Online object tracking: a benchmark, *Computer Vision and Pattern Recognition* (2013), 2411–2418.

[15] A. Yilmaz, O. Javed and M. Shah, Objecttracking: a survey, *ACM Computing Surveys* **38** (2006), 1–45.

[16] K. Zhang and H. Song, Real timevisual tracking via online weightedmultiple instance learning, *Pattern Recognition* **46** (2013), 397–411.

[17] K. Zhang, L. Zhang and M. Yang, Real time compressive tracking, *The European Conference on Computer Vision* (2012), 866–879.

[18] K. Zhang, L. Zhang and M. Yang, Fast compressive tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36** (2014), 2002–2015.