

Visual Target Tracking Based on Compressive Feature Weighting

Yinzhe Zhang^{1, a}, Guixi Liu^{1, b}, Hongyan Duan¹

¹School of Mechano-Electronic Engineering, Xidian University, Xi'an, 710071, China

^aemail: zhangyinzhe0810@126.com, ^bemail: gxliu@xidian.edu.cn

Keywords: Visual Target Tracking; Compressive Features; Feature Weighting

Abstract. At present, most of the features used for tracking are high in dimension, and each dimension of the feature have the same weight, which will increase the amount of computation during tracking and may lead to a poor tracking performance. In this paper, we propose a visual target tracking algorithm based on compressive feature weighting. Unlike the existing methods, we extract compressive features of the target to reduce the amount of calculation, and each dimension of the compressive features is given a different weight by feature weighting method, then the weighted compressive features are used in a Bayes classifier to track the target. In addition, to reduce information loss, multiple compressive features are extracted, and a Kalman Filter is also used to improve the tracking performance. Experimental results demonstrate that the proposed tracking algorithm can achieve a very good performance in some challenging environments, even when the camera is moving.

Introduction

In computer vision field, with the progress of research, the types of target tracking algorithm are also gradually increasing. Some tracking algorithms learning prior knowledge from a small set of training samples to get the initial classifier, then gradually corrected the classifier based on the probability distribution of the test samples, and finally the classification results are used for tracking[1]. In [2] a new tracker called Median-Flow is given. Based on the Median-Flow tracker and P-N learning method, an effective tracking framework named TLD (Tracking-Learning-Detection) is put forward in [3]. In [4], compressive features used for tracking are extracted using a very sparse measurement matrix.

In this paper, we propose a visual target tracking algorithm based on compressive feature weighting. Compressive features in our method are extracted based on compressive sensing theory. Multiple compressive features are extracted to reduce information loss, and each dimension of the compressive features is given a different weight by feature weighting method, then the weighted compressive features are used in a Bayes classifier to track the target. To improve the tracking performance, a Kalman Filter is used while tracking.

The paper is organized as follows. Section 2 describes the method to extract compressive features. Section 3 introduces the tracking method based on compressive feature weighting. Section 4 presents experimental results of the proposed method. Finally, section 5 gives the conclusion of this paper.

Compressive Features Extraction

The Johnson–Lindenstrauss lemma [5] states that a small set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. A typical projection matrix satisfies the Johnson–Lindenstrauss lemma is Gaussian random matrix, whose elements obey the Standard Normal Distribution. However, if the Gaussian random matrix has a large dimension, it will be a dense one, and its operation and storage consumption are large. [5] proposed using the random projection matrix defined as (1):

$$p_{ij} = \sqrt{s} \times \begin{cases} 1 & \text{with probability } 1/2s \\ 0 & \text{with probability } 1-1/s \\ -1 & \text{with probability } 1/2s \end{cases} \quad (1)$$

When $s=2$ or 3 , the matrix will satisfy the Johnson–Lindenstrauss lemma. A compressive feature can be extracted by (2):

$$\mathbf{v} = P\mathbf{x} \quad (2)$$

Where $\mathbf{x} \in \mathbb{R}^N$ is the original feature vector, $\mathbf{v} \in \mathbb{R}^K$ is the compressive feature and $P \in \mathbb{R}^{K \times N}$, $N \gg K$ is the projection matrix.

In fact, we can use a much sparser projection matrix to compress features. [6] points out that one can use $s \gg 3$ (e.g., $s = \sqrt{N}$, or even $s = N / \log N$) to significantly speed up the computation.

As described above, we can use a very sparse projection matrix to get the compressive features of the target. However, some useful information may be lost during this process because the projection matrix is a very sparse one. To reduce information loss, we extract multiple compressive features of the target. To do this, we will use several different projection matrixes to extract multiple compressive features, as shown in Figure 1.

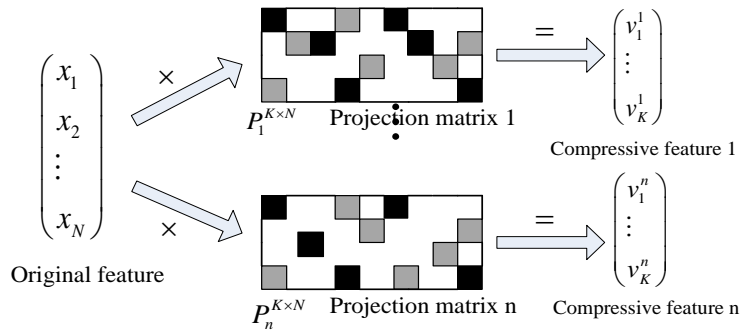


Fig.1. Multiple compressive features extraction

Various features can be used in our method. Without loss of generality, we use Haar-like features which have been widely used for target detection in this paper. The dimensionality of the original feature vector \mathbf{x} is in the order of 10^6 to 10^{10} , as is shown in [4]. So we can use $s = \sqrt{N}$ and $s = N / 6$ to generate two different projection matrixes and thus two kinds of compressive features are obtained. Actually, you can use different values of s to get other projection matrixes and extract different compressive features.

Tracking Based on Compressive Feature Weighting

Usually, after we extract some features of the target, each dimension's weight default to 1, treated equally. However, not each dimension of the features is strongly associated with the classification results, in fact, there are some dimensions of weak correlation affect the classification results. To this end, we use the feature weighting method to improve the classification accuracy.

In this paper, we use the standard deviation of each dimension to determine the weight coefficients. Small standard deviation indicates that the feature is more stable, and this dimension can have a greater weight.

Assume that $\mathbf{v}_i = (v_i^1, v_i^2, \dots, v_i^K)^T$ are the compressive features extracted in the previous section, we use $\mathbf{s} = (s^1, s^2, \dots, s^K)^T$ to represent their standard deviation vector, where

$s^k = \sqrt{\frac{1}{Z-1} \sum_i (v_i^k - \bar{v}^k)^2}$ is the standard deviation of the k th dimension, $\bar{v}^k = \frac{1}{Z} \sum_i v_i^k$ is the average value of the k th dimension, and Z is the total number of the compressive features.

According to standard deviation vector \mathbf{s} , we can define the weight coefficients of each dimension as $W^k = \frac{1}{1 + s^k}$. As a result, the weighted compressive features can be represented

as $\mathbf{v}_i^W = (W^1 v_i^1, W^2 v_i^2, \dots, W^k v_i^k, \dots, W^K v_i^K)^T$.

A Bayes classifier is used to complete the detection and tracking task. One of the typical Bayes classifier [4] is defined as (3):

$$H(\mathbf{v}^W) = \log \frac{\prod_{k=1}^K p(W^k v^k | y=1) p(y=1)}{\prod_{k=1}^K p(W^k v^k | y=0) p(y=0)} = \sum_{i=1}^n \log \left(\frac{p(W^k v^k | y=1)}{p(W^k v^k | y=0)} \right) \quad (3)$$

Where $p(y=1)$ represents the probability of positive samples, $p(y=0)$ represents the probability of negative ones. We assume that $p(y=1)=p(y=0)=0.5$.

From [4] we know that $p(W^k v^k | y=1)$ and $p(W^k v^k | y=0)$ are all obey the Normal Distribution. That is:

$$p(W^k v^k | y=1) \square N(\mu_k^1, \sigma_k^1), p(W^k v^k | y=0) \square N(\mu_k^0, \sigma_k^0) \quad (4)$$

We can update the parameters of the classifier using (5):

$$\begin{aligned} \mu_k^1 &\leftarrow \lambda \mu_k^1 + (1-\lambda) \mu^1 \\ \sigma_k^1 &\leftarrow \sqrt{\lambda (\sigma_k^1)^2 + (1-\lambda) (\sigma^1)^2 + \lambda (1-\lambda) (\mu_k^1 - \mu^1)^2} \end{aligned} \quad (5)$$

Where $\mu^1 = \frac{1}{I} \sum_{i=0|y=1}^{I-1} W^k v^k(i)$, $\sigma^1 = \sqrt{\frac{1}{I} \sum_{i=0|y=1}^{I-1} (W^k v^k(i) - \mu^1)^2}$, and $\lambda > 0$ is a learning parameter.

We can use a Kalman filter to predict the states of the target, which can improve tracking accuracy. To do this, we use a constant velocity model shown in (6):

$$\begin{bmatrix} x(k) \\ \Delta x(k) \\ y(k) \\ \Delta y(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-I) \\ \Delta x(k-I) \\ y(k-I) \\ \Delta y(k-I) \end{bmatrix}, \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \Delta x(k) \\ y(k) \\ \Delta y(k) \end{bmatrix} + V \quad (6)$$

Where $x(k)$ is x position, $y(k)$ is y position, $\Delta x(k)$ is the incremental in x direction, $\Delta y(k)$ is the incremental in y direction and V is the measurement noise. The results of the Bayes classifier can be used as the measurement values of $x(k)$ and $y(k)$. With the states prediction of the target, we can detect the target around the predictive position (a small region), which will simultaneously improve the tracking speed and accuracy. The overall framework of our method is shown in Figure 2.

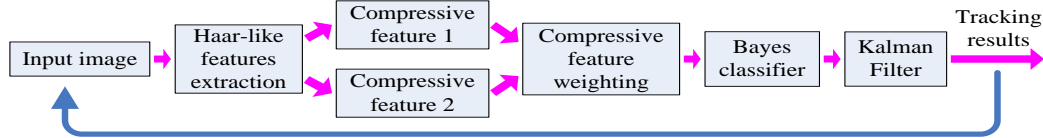


Fig.2. The overall framework of our method

Experimental results

To test the effect of the proposed tracking algorithm, we have conducted some experiments on several complex video sequences, the tracking targets including human faces and ground moving vehicles. The detailed information of the experiment videos is shown in Table 1.

Tab.1. Detailed information of the experiment videos

Detailed information	The targets		
	Vehicles	Faces	Vehicles
Scene	Outdoor	Indoor	Outdoor
Length(frame)	309	487	387
Size(pixel)	640×360	720×576	480×360
Camera status	Moving	Fixed	Moving

Several experimental results of our method are shown in Figure 3. In the experiments, multi-target tracking task is accomplished using multi-threading method.

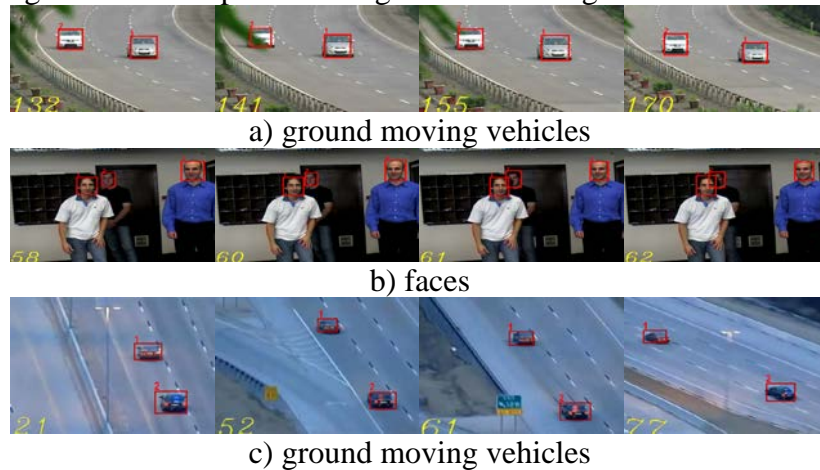


Fig.3. Experimental results of the proposed method

In the 141st frame of the video (a), the target 2 is occluded by leaves, and in the 62nd frame of the video (b), target 2 is occluded by another target. As can be seen from the experimental results, the tracking method proposed in this paper can achieve a good performance even when partial occlusion occurs. Furthermore, video (a) and video (c) are generated by moving cameras, that is to say, the proposed tracking method can be used on a moving platform.

Conclusion

In this paper, we extract compressive features of the target to reduce the amount of calculation, and each dimension of the compressive features is given a different weight by feature weighting method, then the weighted compressive features are used in a Bayes classifier to track the targets. To improve the tracking performance, a Kalman Filter is used while tracking. Experiments show that the proposed tracking method can stability track the targets in some cluttered environments.

Acknowledgement

This work was partially supported by National Ministries Foundation of China under grants #Y42013040181, National Ministries Research of Twelfth Five projects under grants #Y31011040315, and Fundamental Research Funds for the Central Universities under grant #NSIY191414.

References

- [1] Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking. In Computer Vision–ECCV, pp. 234-247, 2008.
- [2] Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: Automatic detection of tracking failures. In Pattern Recognition (ICPR), pp. 2756-2759, 2010.
- [3] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(7), 1409-1422, 2012.
- [4] Zhang K, Zhang L, Yang M H. Real-time compressive tracking. In Computer Vision–ECCV, pp. 864-877, 2012.
- [5] Achlioptas D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. Journal of computer and System Sciences, 66(4), 671-687, 2003.
- [6] Li P, Hastie T J, Church K W. Very sparse random projections. In: KDD, pp. 287–296, 2006.