Object Tracking Using On-line Distance Metric Learning

Dongcheng Chen
Jiangsu Automation Research Institute
Lianyungang, 222000, China
E-mail: `chendongcheng8710@163.com`

Jingying Hu
Jiangsu Automation Research Institute
Lianyungang, 222000, China
E-mail: `540543133@qq.com`

Weiguo Lyu
Jiangsu Automation Research Institute
Lianyungang, 222000, China
E-mail:`tiane1028@163.com`

In order to improve the real-time quality and precision of object tracking, an algrithm using distance metric learning is studied. First, instances are selected around the objects, and features vectors are extracted by using the compress sensing theory. Second, distance metric is trained according to the random projection theory. Finally, the Mahalanobis distance of target object and possible instance are calculated, and the instance which has the smallest distance from object is the tracking object. Experiments on three videos show that the caculating load of the compressed features is ¾, which is less than the one using the uncompressed ones. Using the trained distance metric to calculate the location of target improves the tracking precision.

CENet2017
22-23 July 2017
Shanghai, China

1. Introduction

In recent years, machine learning has been widely used in object tracking and recognition. Different from traditional object tracking, using tracking machine regards the object tracking as an object classification problem. Using some algorithms to classify the object and the background in the frame, and the object which has the highest probability will be regarded as the tracked object. The greatest advantage of machine learning is that it makes the computer learning like human. When using machine learning to track object, the computer can learn the shape, the position and scaling changing of the object, so it can track the object in complex surroundings.

In recent years, algorithms on object tracking have been proposed. For example, the multiple instance learning object tracking algorithm was proposed by Babenko[1]. In the multiple instance learning algorithm, the object classifier is trained by using the multiple instance and the tracker works well when tracking. Grabner proposed the On-line Boosting tracking algorithm, which trains the classifier using multiple instance with different strategy from MIL, but still works well[2]. The P-N learning tracking algorithm proposed by Kalal trains the random fern classifiers using positive and negative instance, which can track object in very complex surroundings[3]. Donoho proposed the compressed sensing theory, the theory shows that object can be expressed with much less data than the original[4]. It can reduce much computing load when tracking or recognizing. Zhang proposed a new tracking algorithm using compressed sensing and multiple instance learning theory, which can track object in real time[5]. Davis used the distance metric learning to train the classifiers, and it worked better than traditional classification algorithm, but the theory was not used in instance classification in image processing[6].

For the good performance of distance metric learning(DML) algorithm in mode classification, we proposed a new algorithm using DML and compressed sensing theory in this paper. First, we compressed the instance features by using compressed sensing theory, which compressed the dimension of the original instances. The tracked instances is searched by using the MIL strategy, and we trained the distance metric by using the algorithm[6]. Then we found the object by classifying the object and background by using the trained metric matrix. The experiment showed that the proposed algorithm can track the object efficiently, and it is adaptive to the changing of the object and the similar objects.

2.Positive andNegative Instances Selecting

In traditional object tracking, only one positive instance is selected. When the object shape is changed or partial occluded, the tracking rectangle will shift away. In this paper, multiple instance will be used when tracking. As Fig.1 shows, the tracked object is called positive instance, while the background is negative instance. For instance, at time t, the position of instance x in current frame is l_t, the category is $y\in\{0,1\}$. When y=1, the instance x is the tracked object, otherwise, it's background. At time t-1, the position of the tracked object is l_{t-1}^*, then, we have the instance set X_r at time t.

$$X_r=\{x \mid \| l(x)- l_{t-1}^* \| < r\} \tag{2.1}$$

Where l(x) is the position of instance to be classified at time t, r is the searching radius.

In order to ensure the position l_t^* of the object at time t, the probability p(y=1|x) of all the possible instances x will be calculated. If in the circle of radius r, each position will have the same possibility to be the object position, as (2.2)

$$p(l_t^*|l_{t-1}^*)\propto \begin{matrix} 1 & f\,\|l(x)-l_{t-1}^*\|<r \\ 0 & otherwise \end{matrix} \qquad (2.2)$$

Then the new position of the object is as (2.3) shows.

$$l_t^*=(\underset{x\in X^s}{argmax})\,p(y=1|x) \qquad (2.3)$$

When the new position l_t^*being found, new positive and negative instances should be updated. As showed in Figure 1, the N positive instances are in the circle of radius of a as (2.4) shows.

$$X^1=\{x_{1i}\|l(x)-l_t^*\|<a\} \qquad (2.4)$$

Similarly, the L negative instances are in the cirque of radius from b to c as (2.5) shows.

$$X^0=\{x_{0i}\|l(x)-l_t^*\|<c\} \qquad (2.5)$$

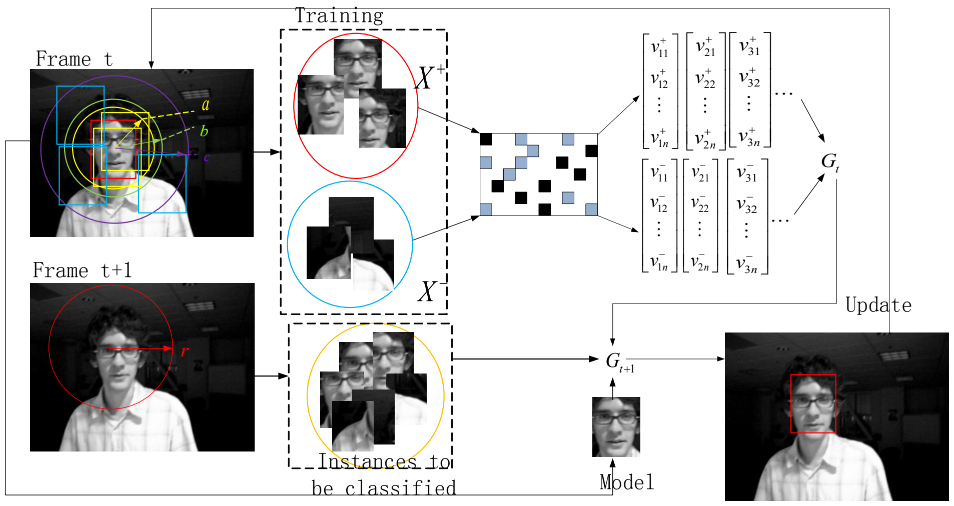


Figure 1: Basic flow of the tracking system

3.Feature Compressing

3.1Random Projection

A random matrix $R\in R_{n\times m}$ can be used to project high-dimensional data space $x\in R_m$ to low-dimensional space $v\in R_n$, just as (3.1).

$$v=Rx \qquad (3.1)$$

where n<<m. As Johnson-Lindenstrauss lemma stated, with high probability the distances between the points in a vector space are preserved if they are projected onto a random subspace with high dimensions[7]. Baraniuk et al. proved that when the random matrix satisfies the Johnson-Lindenstrauss lemma, it will still be true for the restricted isometry property in compressive sensing[8]. Thus, for the compressive high-dimensional image feature x, a random matrix satisfying Johnson-Lindenstrauss lemma can be used to project x to v of the low-dimensional subspace. As (3.1) shows. Where R is a very sparse matrix which satisfies the Johnson-Lindenstrauss lemma. The compressed vector v includes most information of image feature vector x.

3.2Random Measurement Matrix

One of the typical measurement matrix satisfying the restricted isometry property is random Gaussian matrix $R\in R_{n\times m}$ where r_{ij}~N(0,1). When number m is large, R is dense matrix and computing load will be very large. Here we adopt a very sparse random measurement matrix as (3.2) shows.

$$r_{ij}=\sqrt{s}\begin{array}{c}1,\frac{1}{2s}\\ 0,1-\frac{1}{s}\\ -1,\frac{1}{2s}\end{array} \tag{3.2}$$

Achioptas proved when s=2 or 3, the random measurement matrix adapts Johnson-Lindenstrauss lemma. When s=3, the matrix R is a very sparse one, and the computation load is only 1/3. Li et al proved that when s=m/log(m), if r_{ij}~N(0,1), the accuracyof random projection is similar to the conventional random projections. Zhang et al used s=m/4, then the random measurement projection matrix R is a very sparse random matrix. For each row of R, only d, d≤4, entries need to be computed. The computing complexity is only O(dn). Here we use s=m/4 as Zhang et al did [6].

3.3Feature Extracting

The haar-like feature of image is easy computed and consumes little time . It is widely used in object detection. Here haar-like feature is used to represent instances. When using haar-like feature to represent image, the feature will be high-dimensional, which will lead to a very heavy computing load. The previous random measurement matrix theory is used to compress the high-dimension haar-like feature. As Figure 2 shows, R is a very sparse matrix. In the matrix R, the dark rectangle is 1, the gray ones is -1, and the others are 0. The vector x is original haar-like feature vector, which has not been compressed. The vector v is the compressed haar-like feature. In Fig.2, the random measurement matrix R is an m×n matrix, where n<<m. The high-dimension vector x is projected to the low-dimension v through matrix R. According to the compress sensing theory, the extracted low-dimension feature contains almost all the original information. Here we use the compressed feature to represent object, which can distinguish object from background [9].

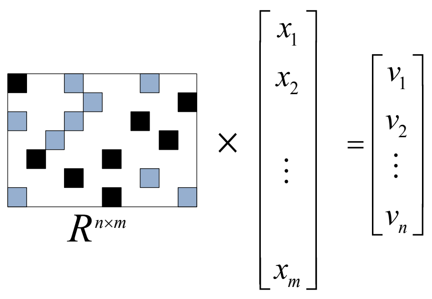


Figure 2: The feature compressing principle

4.Distance Metric Learning

4.1Mahalanobis Distance

The most popular distance measurement is Euclidean distance. It is widely used in pattern recognition, but the classification accuracy is not very high. Blitzer et al demonstrated that Mahalanobis distance can classify pattern better than the Euclidean distance. In fact, Mahalanobis distance is a special Euclidean distance. The two vectors are projected to another subspace, in which the Euclidean distance is measured [10].

In pattern classification, Euclidean distance or inner product is used to compute the distance between two vectors. Blitzer et al learned about pattern classification and concluded that distance metric learning can improve classification rate significantly. The aim of on-line DML is to train a metric matrix which can be used to classify positive and negative instances, and make the classification result better. The Euclidean distance and Mahalanobis distance of vector x and y are as (4.1) and (4.2) show.

$$d(x,y)=(x-y)^T(x-y) \tag{4.1}$$

$$d_G(x,y)=(x-y)^T G(x-y) \tag{4.2}$$

Where G is distance metric matrix, and it is a positive semidefinite matrix. It can be divided as $G=L^TL$. Then (4.2) can be rewritten as (4.3).

$$d_G(x,y)=(L(x-y))^T(L(x-y)) \tag{4.3}$$

From (4.3) we can see that the Mahalanobis distance of x and y is the Euclidean distance of x and y in L subspace. If G is an identity matrix, the Mahalanobis distance is same as the Euclidean distance. Figure 3 is a pattern classification problem. In the figure, the circles represent the classifying threshold. As is shown in the left in Fig.3, it is impossible to classify the three instances by using traditional Euclidean distance. All the instances will be classified into positive set if using a large distance, otherwise some instances will be misclassified. If projecting the instances to another subspace, it's easy to find a classifying circle to classify the instances correctly. Then a distance metric matrix G is needed, which is a positive semidefinite matrix. When classifying the instances, all instances are projected to L subspace. The distance metric matrix G can be trained by using some known distribution. Using Mahalanobis distance to classify instances can overcome the disadvantage of Euclidean distance, and it can improve the precision of classification [11].

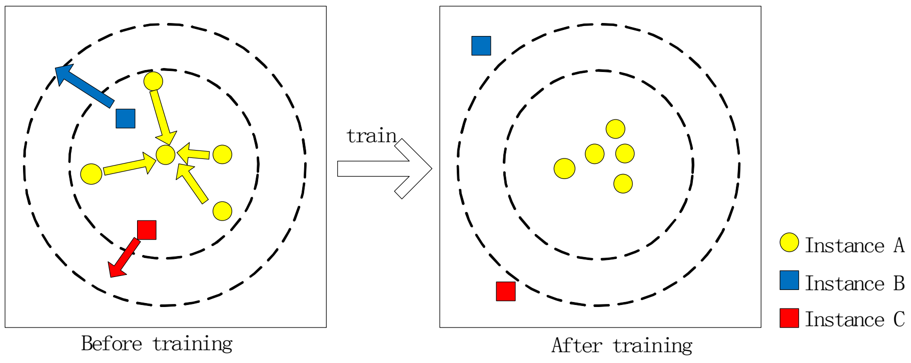


Figure 3: Difference of before and after metric training

4.2Training of Metric Matrix

The function φ is a convex function in the convex function set S, then the Bregman divergence of φ is as (4.4).

$$D_\varphi(x,y)=\varphi(x)-\varphi(y)-(x-y)^T\nabla\varphi(y) \tag{4.4}$$

When $\varphi(x)=x^Tx$, the Bregman divergence is , which is the Euclidean distance of vector x and y. When $\varphi(x)=x^TGx$,then Bregman divergence is $D_\varphi(x,y)=(x-y)^TG(x-y)$, which is the

Mahalanobis distance of vector x and y. Kulis et al expand the vector Bregman divergence to matrix Bregman divergence, as (4.5) shows.

$$D_{\varphi}(X,Y)=\varphi(X)-\varphi(Y)-tr((\nabla\varphi(Y))^T(X-Y)) \tag{4.5}$$

Kulis use the matrix Bregman divergence to find the optimal metric matrix G of the Mahalanobis distance, where φ(X)=-log detX_0. LogDet divergence is one kind of Bregman matrix divergence, it is often used to measure the similarity of metric matrix G and G_0. As (4.6) shows.[11]

$$D_{ld}(G,G_0)=tr(GG_0^{-1})-logdet(GG_0^{-1})-n \tag{4.6}$$

(4.7) is used to solve the optimal metric matrix G.

$$\begin{aligned}&\min_{A>0,\xi} D_{ld}(G,G_0)+\gamma D_{ld}(diag(\xi),diag(\xi_0))\\ &s.t.\ tr(G(x_i-x_j)(x_i-x_j)^T)\leqslant\xi_{c(i,j)}(i,j)\in S\\ &tr(G(x_i-x_j)(x_i-x_j)^T)>\xi_{c(i,j)}(i,j)\in D\end{aligned} \tag{4.7}$$

Where c(i,j) is constraint of number G(i,j), ξ is slack parameter, γ is constraint parameter. In order to solve (4.7) to get the optimal matrix G, algorithm as Table1 is used.

```
Input：X：feature matrix, including set of positive and negative instances；$S$：positive instance set；$D$：negative instance set；$G_0$: initial distance metric ；$\gamma$:constraint parameter；$\xi$:slack parameter
    $G=G_0$, $\lambda_{ij}=0$, $\xi=1$, $\gamma=0.15$
    For $i$ = 1…frame
    For $j$ = 1…$d$
    $p\leftarrow(x_i-x_j)^TG(x_i-x_j)$
            $\delta\leftarrow1$ if$(i,j)\in S$;-1otherwise
            $\alpha\leftarrow min(\lambda_{ij}\frac{\delta}{2}(\frac{1}{p}-\frac{\gamma}{\xi}))$
            $\beta\leftarrow\frac{\delta\alpha}{1-\delta\alpha p}$
            $\xi\leftarrow\frac{\gamma\xi}{\gamma+\delta\alpha\xi}$
            $\lambda_{ij}\leftarrow\lambda_{ij}-\alpha$
            $G=G+\beta G(x_i-x_j)(x_i-x_j)^TG$
            End for
    $x_{i+1}\leftarrow\underset{x\in X}{argmin}(x_i-x_j)^TG(x_i-x_j)$
        Output：position of the target
    End for
```

Table 1: On-line learning algorithm of distance metric

5.Experiments

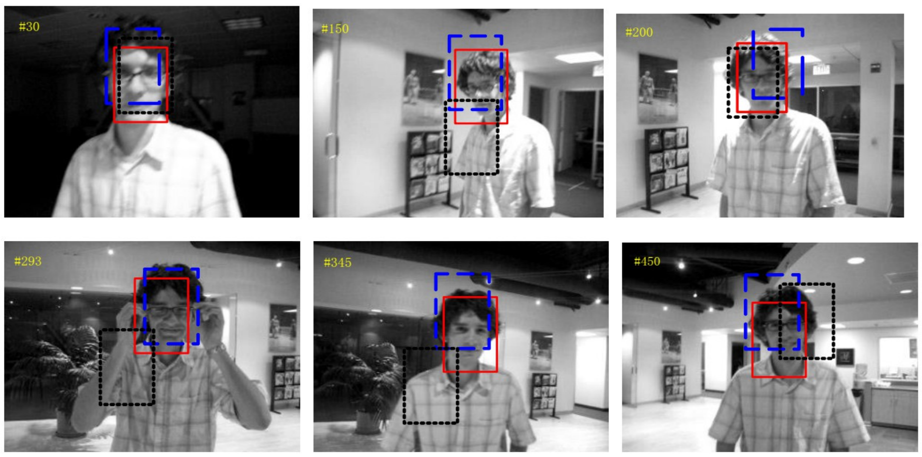
To evaluate our approach, we perform comprehensive experiments on three benchmark databases. In all the experiments, parameters remain unchanged. The positive instances are selected as (2.4) shows, where the parameter a=4, 45 instances are selected to form the positive instance set X^+. The negative instances are selected as (2.5) shows,where b = 2*a and c = 1.5*b, also, 45 negative instances are ramdomly selected from the negative pool to form the negative set X$^-$. Generally, objects will not move more than 15 pixels between frames in most videos. So here, 25 pixels is the searching radius. Negative instances in the cirque will be much more than 45, only 45 instances are random selected from the the negative pool [12][13].

Experiment is adopted, three videos including david in door, running car and car meeting are used. MIL and Boosting Tracker is used as the comparison algorithm. The tracking location error is calculated using (5.1) as follows.

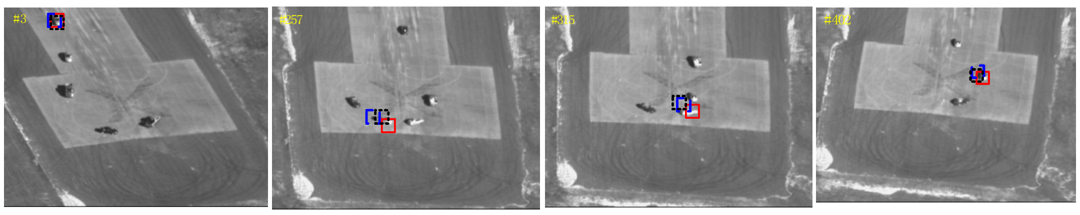
$$LocationError = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{5.1}$$

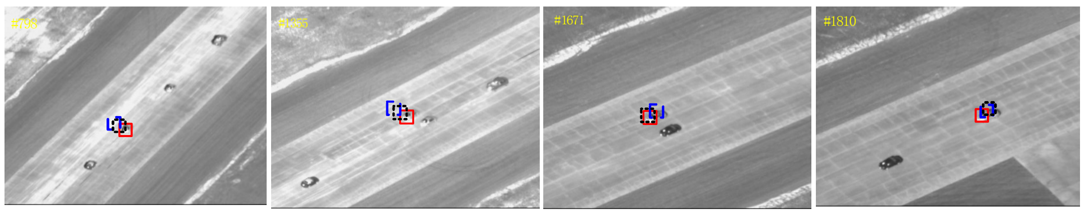
In (5.1), (x_i,y_i) is the position of the object calculated by using the algrithm, and (x_c,y_c) is the real position of the object.

In Figure 5,(a), is the result of the video David in door. There are 462 frames in the video, the object rectangle is 60 pixle×88 pixel. As Figure 4 (a) shows, in the video, the object David walk from dim place to bright room, and turns his face when walking, takes off his glasses, which may influent the tracking algrithm. The MIL tracker works well in the video sequence. The processing rate of our algrithm is 15 fps, while the processing rate of Boosting Tracker and MIL tracker is 10 fps and 12fps. Our algrithm works faster and tracks much more accurately than the two comparisons. In the second experiment, as Figure 4 (b) and Figure 5 (b) show, there are 1821 frames in the Runing car sequence. There are five cars in the video, and three of them seem similar in shape and color. The object rectangle is 24 pixle×24 pixel in the frame. From the error curve, it is obvious that our algrithm wave strongly. The third experiment is performed in the car meeting video sequence, and there are 1300 frames in the sequence. The object rectangle is 24 pixle×24 pixel. Our algrithm always works well when meeting car or turing, which indicate that our algrithm can perform well on similar object disturbing. In the experiment 2 and 3, the processing rate of MIL tracker and Boositng Tracker is 15 fps and 12 fps, while our algrithm can track the object at 18 fps.

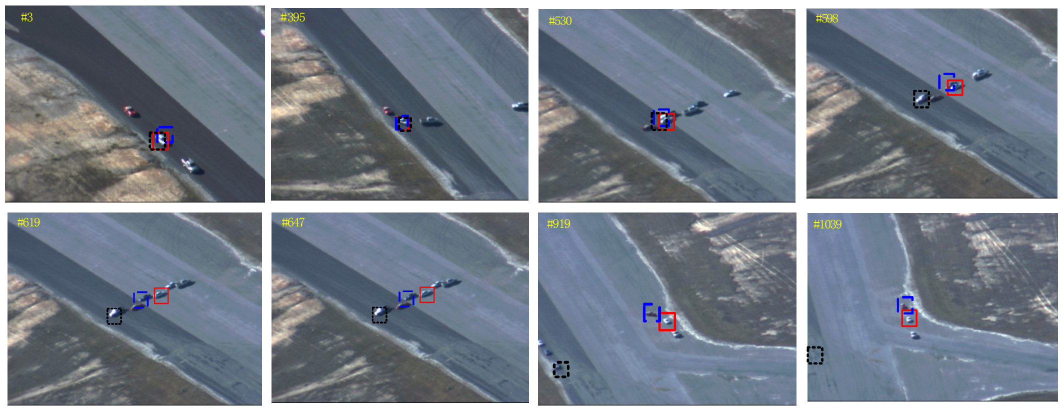


(a)David in Door Sequence





(b)Running car sequence



(c)Car Meeting Sequence

Figure 4: Tracking result

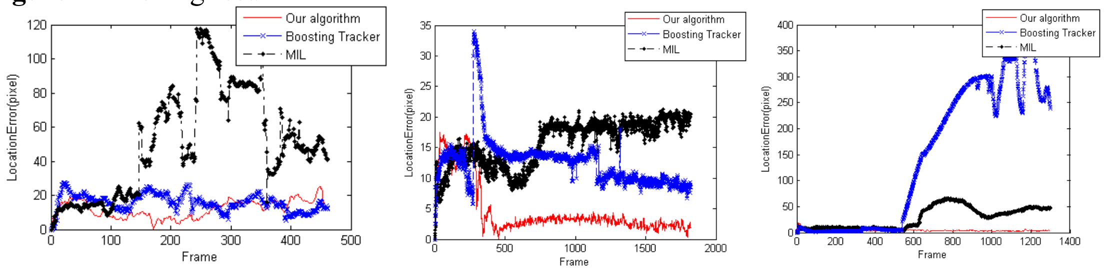


Figure 5: Tracking error curve of three sequences

From the experiments above, for not converging of the distance metric, it is obvious that our distance metric learing tracker wave strongly in the begining frames. As tracking goes on, less training times is needed, which can increase the realtime quality of the tracking algrithm.

5.Conclusion

In this paper, the distance metric learning in pattern classification is adopted on object tracking. The computing load decrease a lot when the feature is compressed by using the compressed sensing theory. The distance metric is trained by using the positive and negative instances from every frame. The trained metric is used to measure the Mahalanobis distance between object and possible instances, which can make the tracking stable. The tracking real-time quality is improved by using the feature compressing method.

References

[1]BABENKO B, YANG M, BELONGIE S, *Robust object tracking with online multiple instance learning*[J]. Pattern Analysis and Machine Intelligence ,IEEE Transactions on, 33 (2011) 1619–1632.

[2]Grabner H, Bischof H. *On-line boosting and vision*[C]. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 2006, 1: 260-267.

[3]Kalal Z, Mikolajczyk K, Matas J. *Tracking-learning-detection*[J]. Pattern Analysis and Machine Intelligence , IEEE Transactions on, 2012, 34(7): 1409-1422.

[4]Donoho D L. *Compressed sensing*[J]. Information Theory, IEEE Transactions on, 2006, 52(4): 1289-1306.

[5]Zhang K, Zhang L, Yang M H. *Real-time compressive tracking*[M].Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012: 864-877.

[6]Davis J V, Kulis B, Jain P, et al. *Information-theoretic metric learning*[C].Proceedings of the 24th international conference on Machine learning. ACM, 2007: 209-216.

[7]Achlioptas D. *Database-friendly random projections: Johnson-Lindenstrauss with binary coins*[J]. Journal of computer and System Sciences, 2003, 66(4): 671-687.

[8]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.

[9]Li P, Hastie T J, Church K W. *Very sparse random projections*[C].Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006: 287-296.

[10]L. Čehovin, A. Leonardis, M. Kristan. *Visual Object Tracking Performance Measures Revisited*[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2016, 25(3):1261-1274.

[11]Kulis B, Sustik M, Dhillon I. *Learning low-rank kernel matrices*[C].Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 505-512.

[12]Blitzer J, Weinberger K Q, Saul L K. *Distance metric learning for large margin nearest neighbor classification*[C].Advances in neural information processing systems. 2005: 1473-1480.

[13] Xiang Y, Alahi A, Savarese S. *Learning to Track: Online Multi-object Tracking by Decision Making*[C]. Computer Vision ,IEEE International Conference on, 2015:4705-4713.