

# A Novel Two-Stage PCA Algorithm for Object Tracking

Yuanyuan Yang and Dong Hu

Education Ministry's Key Lab of Broadband Wireless Communication and Sensor Network Technology  
Education Ministry's Engineering Research Center of Ubiquitous Network and Health Service  
Jiangsu Province's Key Lab of Image Procession and Image Communications, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China

**Abstract**—In this paper, we propose a two-stage PCA algorithm to deal with the problem of target appearance changes in object tracking. Our method is based on particle filter framework and aims at building up a robust appearance model for the target. In the first-stage, PCA is applied on several templates collected before tracking to construct a low dimensional subspace for the targets. Correspondingly, in the later tracking, all particles will be projected into this subspace to calculate the weight of particles, as well as the location of the target by weighted sum of particles. In the second-stage, PCA is adopted to decompose the reconstruction error into orthogonal basis to find the basis with biggest variance which can best present the appearance changes. Correspondingly, we use this basis to update the old subspace. Besides, a threshold is set to decide when to update which greatly reduce the numbers of update. The two stages work together to establish a robust appearance model making our tracking algorithm more robust. Experimental results on public video sequence demonstrate the effectiveness of our proposed algorithm.

**Keywords**—particle filter; two-stage PCA; eigenspace; reconstruction error

## I. INTRODUCTION

Object tracking has wide application in video surveillance and man-machine interface. Many algorithms have been proposed up to now, but visual tracking still confronts challenges, such as appearance variation due to changes by illumination, view angle, shape, and occlusion as well.

Object tracking models consist of two modules, one is motion model and the other is appearance model. The motion model is aimed to predict the possible states of an object and the usual algorithm is mean-shift, optical flow and particle filter algorithms. The TLD [1] uses mean-shift as tracker, while our method is based on particle filtering because of its non-linear, non-Gaussian, nonparametric characteristics which make this algorithm more state.

Appearance model is used to represent the target object. The appearance can be modeled by many features, such as color, intensity, texture etc. The calculation of color histogram is quick and easy, but is too simple to represent targets and can be easily affected by background. Recently, many researchers

put their attention to intensity. An image can be totally described by intensity, but this kind of feature is redundant. So the biggest problem for the gray feature is to reduce the dimensions and remove redundancy. In [2], the author used sparse to reduce the dimensions of the image and at the same time hold the the useful information as much as possible. He constructed a sparse complete dictionary for the holistic and local feature to represent target. In [3], the author found a sparse matrix which satisfies Restricted Isometry Property (RIP) condition to decompose a image, then used the low dimensional positive and negative features to train a Bayes classifier. Except sparsification, David Ross[4] used PCA to construct a subspace for the target. The PCA consider a large number of redundant pixels as different dimensions and then find orthogonal basis which can eliminate correlation and at the same time highlight the differences between dimensions. In object tracking, this orthogonal basis can represent the target with the least error. David Ross also proposed an incremental learning to update the sample mean and eigenbasis to deal with pose changes in the tracking. In [5], the author also used PCA to model appearance in the particle filter framework, and new basis is added to subspace by incremental PCA, disabled basis is removed during the tracking. Appearance model based on PCA has many advantages, such as, the eigenspace representation see the pixels different dimensions connected to each other, while other features treat the target as a set of independent pixels. In addition, this kind of algorithm is robust to light changes and noise, but sensitive to pose changes, so we can use incremental PCA to capture these changes and add new basis into original eigenspace. So in our paper, we also use PCA to model appearance. But unlike [4],[5] which achieved good results at the price of massive computation because of incremental PCA, they ignored the fact that appearance changes do not happen at every frame and sometime in several continuous frame the original subspace is total capable of correctly tracking targets. And when adding new basis, we do not have to directly calculate the new data, we can only focus on the differences between templates and new data.

In this work, we use two-stage PCA to build appearance model. The first-stage apply PCA to establish subspace for the target and the second-stage update the subspace by using PCA to analysis reconstruction error. The rest of this paper is organized as follows: Section II introduce particle filter. Section III discuss the first-stage PCA and in the section IV

we give the details of the second-stage PCA. Experiment is given in section V.

## II. PARTICLE FILTER

Object tracking can be treated as a Bayesian filtering problem. The first step is to predict the state of current frame given all previous observations, this is a transfer of conditional probability based on Chapman-Kolmogorov equation. And the second step is to update the estimated state using the Bayes' rule. So Bayesian filtering equations can be described as follows:

$$p(x_t | y_{1:t-1}) = \int p(x_t | x_{t-1})p(x_{t-1} | y_{1:t-1})dx_{t-1} \quad (1)$$

$$p(x_t | y_{1:t}) = \frac{p(y_t | x_t)p(x_t | y_{1:t-1})}{\int p(y_t | x_t)p(x_t | y_{1:t-1})} dx_t \quad (2)$$

However the optimal Bayesian solution the posterior probability is very hard to calculate due to the recursion in the integral computation. So particle filter is proposed to approximately calculate the posterior probability whose core idea is replace integral operation with sample mean by randomly sampling posterior probability.

In particle filter in our paper, we use  $X_t = (x_t, y_t)$  (of course you can add the angle and size of target box if needed) to describe  $t^{th}$  frame state, which is the object position in the x-, y- coordinate(center position of the template), and assume that each parameter in  $X_t$  is independently by a Gaussian distribution around  $X_{t-1}$ . So we can calculate the probability of state transition by  $p(X_t | X_{t-1}) \sim N(X_t; X_{t-1}, \sigma)$  and  $\sigma = \text{diag}(\sigma_x^2, \sigma_y^2)$  which is variance of x and y. Particle filter algorithm calculate posterior probability by  $p(x_t | y_{1:t}) = \sum_{i=1}^{N_s} w_t^{(i)} x_t^{(i)}$ ,  $N_s$  is the number of particles,  $x_t^{(i)}$  represent the state of  $i^{th}$  particle,  $w_t^{(i)}$  is the weight of  $x_t^{(i)}$ , which is proportional to  $p(x_t | x_t^{(i)})$ . The calculation of weight is crucial to the estimation of state. In order to calculate weight, we need to construct robust appearance model for the target. In this paper, we choose gray feature and use PCA to reduce the dimensions.

## III. THE FIRST-STAGE PCA

In our tracking framework, we apply PCA to module object appearance. A image can be totally describe by gray features, but this kind of feature is high dimensional and full of redundancy. As PCA is a classical and popular tool for data analysis, we use PCA to removing redundancy and find several main directions of data changes. The idea of PCA is coming from Karhunen-Loeve transform, which prove that the vector x can be approximately represented by orthogonal basis

U. PCA find the subspace U by minimizing error  $e = \|X - UU^T X\|_2^2$ , where X is the training data.

In order to use the PCA, we change a image which is a matrix into a column vector  $X_i = [x_1, x_2, \dots, x_M]^T \in R^{M \times 1}$ , M is the dimension of  $i^{th}$  frame, and also is the number of pixels. In the particle filter framework, we can collect several targets in the initial frames as training data  $X = [X_1, X_2, \dots, X_N] \in R^{M \times N}$ , N is number of templates.

The algorithm of PCA can be stated as:

1. Compute the mean  $\bar{u}$  of data X.
2. Subtract the mean from X.
3. Calculate the covariance matrix R.
4. Calculate the eigenvalues and eigenvectors of R, choose the several eigenvectors with large eigenvalues.

in our paper, we use  $U = [u_1, u_2, u_3]$ , which is enough for the first several frames. In Figure 1. (a) is the template and (b) is the corresponding eigenspace we establish using PCA.

When tracking target, for the particle  $y \in R^{M \times 1}$ , we use Eq. (3) to calculate reconstruction error and Eq.4 to measure similarity between temple and particles which is weight of particle y.

$$e = \|(y - \bar{u}) - UU^T (y - \bar{u})\|_2^2 \quad (3)$$

$$w = \exp\left(\frac{-\|(y - \bar{u}) - UU^T (y - \bar{u})\|_2^2}{\sigma}\right) \quad (4)$$

Every particle gets its own weight by this way and finally we calculate weighted sum of all the particles and gain the location of object.



FIGURE 1. (A) IS THE TEMPLATE AND (B) IS CORRESPONDING EIGENVECTOR, WHICH MEANS WE CAN REPRESENT (A) USING (B) WITH LEAST ERROR

## IV. THE SECOND-STAGE PCA

The appearance of target is gradually changing as the tracking task progresses, therefore it is important to develop an efficient algorithm to update the eigenspace. The algorithms proposed in [5] need to calculate QR decomposition and SVD which lead to great computation and this algorithms need to be implemented at every several frames. Our algorithms aim to gradually construct a approximate complete orthogonal basis to represent the target by analyzing the reconstruction error to find the new basis.



(A) TRACK RESULT



(B) RECONSTRUCTION ERROR

FIGURE II. (A) IS THE TRACKING RESULT OF 5TH FRAME USING EIGENSAPCE ESTABLISHED BY PCA AND (B) IS THE CORRESPONDING RECONSTRUCTION ERROR

#### A. Reconstruction Error

By Eq. (3) we can see that, the reconstruction error is the information that can not be represented by the eigenspace, which indicates that the new information we need to add to the old subspace is also in the reconstruction error. The Figure II shows the track result and corresponding reconstruction error.

And when we project the reconstruction error into the subspace, we can prove that:

$$\begin{aligned}
 U^T \bar{e} &= U^T [(Z - \bar{u}) - UU^T (Z - \bar{u})] \\
 &= U^T (Z - \bar{u}) - U^T UU^T (Z - \bar{u}) \\
 &= U^T (Z - \bar{u}) - U^T (Z - \bar{u}) \\
 &= 0
 \end{aligned} \tag{5}$$

in where  $U^T U = I$  is unit matrix. Eq. (3) indicates that the residual  $\bar{e}$  is perpendicular to the subspace  $U$ , which motivates us that we can once again use PCA to find the eigenvectors of  $\bar{e}$ .

#### B. Apply PCA to Reconstruction Error

As the  $\bar{e}$  is perpendicular to the subspace  $U$ , so the eigenvector is also perpendicular to the subspace which can be added to the old subspace as the new basis. Our method use this way to process the new data and find the new basis. The the incremental PCA result of some frames can be shown in Figure 2. In the figure, we track the 9<sup>th</sup> frame, and the second row is the largest eigenvector of reconstruction error, which represents the direction of the appearance changes. So we can use this eigenvector to update the old subspace.

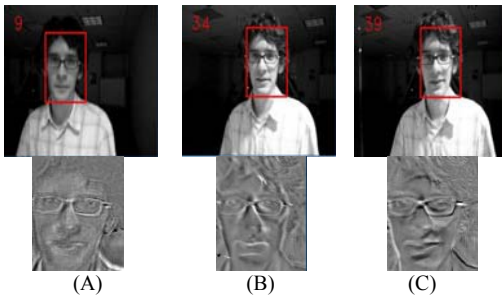


FIGURE III. PCA CONSTRUCT SUBSPACE U, AND CONTINUOUSLY UPDATE THE U USING OUR ALGORITHM. (A) INDICATES THAT THE TARGET CHANGES AND WE ADD A CORRESPONDING NEW BASIS TO THE OLD SUBSPACE AT THE 9TH FRAME. AND (B),(C) RESPECTIVELY THE INCREMENTAL PCA RESULT OF 34TH AND 39TH FRAME

#### C. When to Update

The value of  $\|\bar{e}\|_2^2$  decides when to update. When  $\|\bar{e}\|_2^2$  is small, the changes most come from background and a few changes is from target, the subspace can totally handle this. When  $\|\bar{e}\|_2^2$  start to increase, it indicates that target is changing. We set  $\alpha$  as the basic value when the tracking is stable. If  $\|\bar{e}\|_2^2$  is around  $\alpha$ , we consider that there is no need

to update, when  $\|\bar{e}\|_2^2$  start exceed  $\alpha$  and  $\frac{\|\bar{e}\|_2^2 - \alpha}{\alpha} > \varepsilon$ ,

$\varepsilon$  is the maximum tolerance value. we establish a buffer to collect the  $\bar{e}$  and when it reach t frames, we calculate the PCA of  $\bar{e}$ , and only add the first eigenvector.

In our paper, we set  $\alpha$  by averaging  $\|\bar{e}\|_2^2$  of previous 20 frames, of course, different sequences have different values which can get from experiment. Average is to seek the stability of track. The maximum tolerance  $\varepsilon$  is 0.2~0.3 according to different sequences. As for the capacity of buffer, we set 5. And after update, we empty buffer and wait for the next tracking result. When the buffer is not full but  $\|\bar{e}\|_2^2$  return to the value around  $\alpha$ , we also empty the buffer, because we see this situation as a emergency.

We can notice that our worst situation, updating every five frames, is the same as the algorithm in[5]. So our algorithm is fast than the traditional ways. As for the accuracy of tracking, we can see in the following experiment results

## V. EXPERIMENT RESULTS

This section demonstrates the effectiveness of the proposed visual tracker by comparing it with incremental visual tracker (IVT) in[5]. The experiment is implemented in Microsoft Visual Studio 2010 with opencv-2.4.11.

The first sequences is WOMEN, the results are shown in the Figure IV., and from the picture we can see that in these sequences, the background is a little simple, but changes abruptly and greatly, especially between the 21<sup>th</sup> and 44<sup>th</sup>, the background changes from white to gray. Our algorithm can still capture the target exactly with four times update. At meanwhile, the algorithm in [5] will automatically updates for about 15 times(update every five frame).

The second sequences is WALKING MAN IN THE FRONT, the results of tracking is shown in the Figure V, the background of this sequences is more complicated, and view is further then the first sequences, which make the tracking more difficult. And during the man walking, his pose changes randomly. Although that our algorithm can exactly track the target and most importantly during the whole tracking, only update form three to five times, while the algorithm in [5] update around 20 times, our method save much time and at the same time keep the accuracy. We can see that after 80<sup>th</sup> frame, the man starts to turn round, our algorithm capture this change and present it in the subspace.

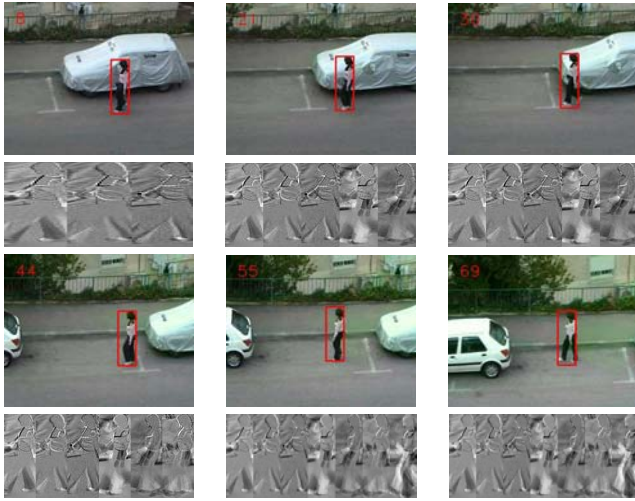


FIGURE IV. THE RESULTS OF TRACKING FOR WOMAN , THE IMAGE IN THE FIRST AND THREE IS THE RESULTS AND THE REST IS THE CORRESPONDING AND EIGENSPACE, WHICH WE CAN SEE THE NUMBER OF EIGENVECTORS IS INCREASING AS THE TRACKING GOING ON. AND ONLY UPDATE WHEN THE TARGET CHANGES HIS POSES

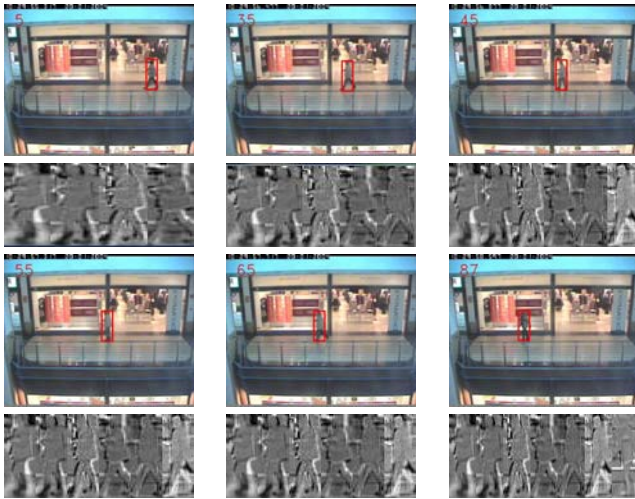


FIGURE V. THE RESULTS OF TRACKING FOR WALKING MAN, WITH COMPLICATED BACKGROUND

## VI. CONCLUSION

In this paper, we construct a object tracking model based on particle filter and apply PCA to model the appearance of target object and more importantly propose a novel incremental PCA which is simple and effective. This algorithm decomposes the residual data to find the new basis and only update at appropriate time. The experiment has showed the effectiveness of our method and far less update times compared to traditional incremental PCA.

The further study is about the  $\|\vec{e}\|_2^2$ . We will divide the situation when  $\frac{\|\vec{e}\|_2^2 - \alpha}{\alpha} > \alpha$  into two cases, one is  $\sigma < \frac{\|\vec{e}\|_2^2 - \alpha}{\alpha} < \tau$ , which is needed to update , the other is

$\frac{\|\vec{e}\|_2^2 - \alpha}{\alpha} > \varepsilon$ , which is considered losing the target and will need further research to find ways to fix.

## ACKNOWLEDGMENT

This work was supported by the Project of the Priority Academic Program Development of Jiangsu Higher Education Institutions: Information and Communication Engineering.

## REFERENCES

- [1] Kalal Z, Mikolajczyk K, Matas J. Tracking-Learning-Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011, 34(7):1409-1422
- [2] Zhong W, Lu H, Yang M H. Robust object tracking via sparse collaborative appearance model.[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2014, 23(5):2356-68.
- [3] Zhang K, Zhang L, Yang M H. Real-Time Compressive Tracking[C]// European Conference on Computer Vision. Springer-Verlag, 2012:864-877.
- [4] J.Lim, D.A. Ross, R.S.Lin, M.H. Yang, "Incremental learning for visual tracking, " Proc. Conference on Advances in Neural information Processing Systems, pp. 793-800, 2004.
- [5] Kim d Y, Jeon M. Spatio-temporal auxiliary particle filtering with l1-norm-based appearance model learning for robust visual tracking.[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2013, 22(2):511-22.
- [6] Wang D, Lu H, Yang M H. Online object tracking with sparse prototypes.[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2013, 22(1):314-25.