A Novel Face Detection and Tracking Method Based on Feature Weighting

Yehong Chen[1], Pil Seong Park[2]

[1.] School of Information, Qilu University of Technology, Jinan 250353, China
[2.] Department of Computer Science, University of Suwon, Gyunggi-do 445-743, Korea
chenyh@spu.edu.cn

Abstract: We propose a tracking algorithm based on image classification involving online feature weighting. The algorithm uses automatically produced general Haar-like features through feature extraction and feature selection using an online-built object model, and combines Principal Component Analysis (PCA, a generative method) and Fisher´s Discriminative Analysis (FDA, a discriminative method). That is, we first train the Fisher classifier to distinguish the foreground candidates from background. Then target matching is performed based on similarity with PCA codes of the candidates in feature vectors. The discriminating function of Fisher classifier is a linear combination of the weighted feature values. We also propose a feature discriminative power evaluation equation based on multi-class FDA which gives more discriminative results between the foreground and background. Both the PCA and FDA are online updated to adapt to variation in the images of the tracked object over time, e.g., by noise, occlusion, or a cluttered background. Experimental results show that the proposed method improves detection accuracy when compared with some competitive algorithms.

[Chen Y, Park PS. **A Novel Face Detection and Tracking Method Based on Feature Weighting.** *Life Sci J* 2014;11(9):23-28]. (ISSN:1097-8135). http://www.lifesciencesite.com. 4

Keywords: FDA; PCA; Discriminative Power; Feature Weight; Online Update;

1. Introduction

Visual object tracking is one of the most important research areas of computer vision, and is widely used in many scenarios, e.g., security surveillance, traffic management, etc. It can be defined as the problem of generating the trajectory of an object over time by recognizing it in every frame and linking to the same object from one frame to the next. Most important issues related to tracking include the use of appropriate image features for appearance representation, selection of motion models, and detection of objects. The main challenge in designing a robust visual tracking algorithm is the inevitable variation in the images of the tracked object over time, e.g., by noise, occlusion, or a cluttered background. Numerous approaches for object tracking have been proposed (e.g., Qiang and Zhao (2006), Yang et al. (2011), Yilmaz et al. (2006)), but no tracking algorithm ever performs perfectly in all such conditions.

All object tracking methods can be classified into two categories: generative models and discriminative models. In generative models, algorithms represent the target object in a particular feature space, and then search for the best matching score within the image region. In discriminative models, algorithms treat visual tracking as a binary classification problem to define the boundary between the target image patch and the background. A major shortcoming of discriminative methods is their noise sensitivity, while generative ones would easily fail within cluttered background. In fact, how to combine generative methods and discriminative methods into a coherent framework is a classic question and needs more research (Zhao et al., 1998).

Feature weighting (or feature selection) plays an important role in the performance of the classification (Fisher, 1936). Until now, a wide range of automatic feature selection algorithms has been investigated (Chen and Park (2013), Chen et al. (2013), Dollar et al .(2007), Lin et al. (2004), Wang et al. (2005)). AdaBoost, for example, has been proven to be a powerful tool (Viola and Jones, 2001), but it needs offline-training, huge training data, long time, and all information known ahead of time. Because the appearance of the object and its surrounding background will change during tracking, how to achieve a better balance between adaptivity and stability when using online learning methods is still an open problem (Qiang and Zhao (2006), Yang et al. (2011), Yilmaz et al. (2006)).

In this paper, we propose a tracking algorithm based on image classification involving online feature weighting. Our work aims to solve three problems: how to combine generative models and discriminative models, how to perform feature weighting based on multi-class background FDA, and how to implement smoothly online model updating to deal with the drift problem .

This paper is organized as follows: In Sec. 2, we introduce some closely related works. In Sec. 3, we introduce the fundamental principles from which our method is deduced. In Sec. 4, experimental results are presented, then followed by conclusions and future works.

2. Related works

In object representation, dimensionality reduction is vital. The sparse representation obtained by L1-norm minimization is aimed to get rid of the curse of dimensionality (Bao et al. (2012), Huang and Aviyente (2006)). Recently a compressive sensing method has been used in tracking, which uses the sparse representation of the target object in a feature space and achieved good results (Zhang et al. (2012)).

Subspace analysis is a basic approach to achieve dimensionality reduction to find a transformation basis matrix that projects the object from a higher dimensional space onto a lower dimensional subspace (Abdi and Williams (2010), Fisher (1936), Li et al. (2008), Lin et al. (2004), Welling (2005), Zhao et al. (1998)). PCA (principal component analysis) and LDA (linear discriminative analysis) are two famous methods.

The classical Fisher linear discriminant analysis computes the optimal projection direction *W* by maximizing the following objective function (Welling, 2005):

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|}$$

$$\text{where} \quad S_B = \sum_{i=1}^{n} N_i (m_i - m)(m_i - m)^T$$

$$S_W = \sum_{i=1}^{n} \sum_{x \in \chi_i} (x - m_i)(x - m_i)^T .$$

S_B and S_W are the "between" and "within" class scatter matrices, where m_i is the mean of the class *i*, N_i is the number of samples in class *i*, and *m* is the mean of overall samples. Computation of *W* is an optimization problem, and we can use an off-the-shelf optimal algorithm to calculate it.

FDA is thought of as to require a weighted linear combination for a classification problem (Fisher, 1936). The weights can be obtained from the training data set directly.

Suppose two classes of observations have means $\vec{u}_{y=0}, \vec{u}_{y=1}$ and covariances $\Sigma_{y=0}, \Sigma_{y=1}$. Then the linear combination of features $\vec{w} \cdot \vec{x}$ will have means $\vec{w} \cdot \vec{u}_{y=i}$ and covariances $\vec{w}^T \Sigma_{y=i} \vec{w}$ for *i*=0,1. Fisher defined the separation score between these two distributions as the ratio of the variance between the classes to the variance within the classes:

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}} = \frac{(\vec{w} \cdot (\vec{u}_{y=1} - \vec{u}_{y=0}))^2}{\vec{w}^T (\Sigma_{y=0} + \Sigma_{y=1}) \vec{w}}$$

It can be shown that the maximum separation occurs when

$$\vec{w} \propto (\Sigma_{y=0} + \Sigma_{y=1})^{-1} (\vec{u}_{y=1} - \vec{u}_{y=0}) \qquad (2\text{-}1)$$

Lin et al. (2004) introduced a formulation of multi-class FDA. In an extreme case, they assum each background sample is generated from a non-target class and formulated the object/background distinction as an FDA problem, in which each background sample is regarded as a separate class. Our work follows this direction and further improves the formula for feature weighting which takes into account multi-class background.

PCA is a generative method, which searchs for the projection direction that gives the largest variance. PCA can be done by eigen-decomposition of the covariance matrix of the training data (Abdi and Williams, 2010). A major shortcoming of discriminative methods is noise sensitivity. However the use of a PCA method which is effective in removing noise information for matching target candidates will supplement the shortcoming of FDA classification (Zhao et al., 1998). Current popular L1 trackers for which sparse representation is obtained by L1-norm minimization is also based on PCA (Bao et al., 2012).

3. Proposed Algorithm

3.1 Feature space and feature extraction

To generate feature templates automatically, we use general Haar-like features. Feature extraction can be done by introducing a sparse random Gaussian measurement matrix $R \in \mathbb{R}^{n \times m}$ to project a high dimensional image vector $\overline{x} \in \mathbb{R}^m$ onto a lower dimensional vector $\overline{v} \in \mathbb{R}^n$ by $\overline{v} = R\overline{x}$. *R* is a sparse random measurement matrix. For details about *R*, see Zhang et al. (2012). Both generative and discriminative models of object appearance representation are built on this compressed feature space.

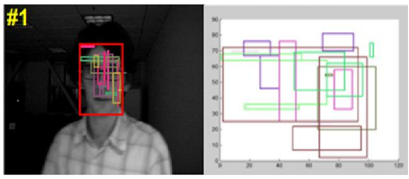
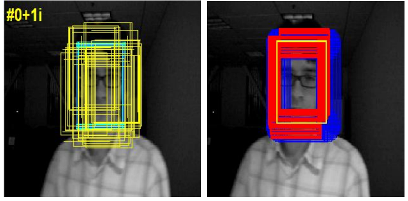


Figure 1. Extracted features within a sample. Every group of rectangles in the same color corresponds to a general Haar-like feature. The right figure is an example of a general template for feature extraction. The movie is from http://www4.comp.polyu. edu. hk/ ~cslzhang/CT/CT.htm.

3.2 A two-stage framework

The training data is generated automatically around the target at the previous position detected by the second stage (however, the target has to be labeled manually at the beginning).

In the learning stage, given the captured target by the previous tracking result, the algorithm crops training samples (both target samples and background samples) around the target in the predefined region, which are then used for learning/updating an FDA classifier. In the next frame, the detector samples patches as input to an FDA classifier to distinguish and collect target candidates from the background, and finds the best position of the target through a PCA matching process. The entire two-stage tracking process is repeated at every frame.



(a) (b)

Figure 2. The two-stages framework. (a) The algorithm takes both positive (in cyan) and negative (in yellow) samples to train the FDA classifier. (b) In the next frame, the FDA classifier picks the target candidates (in red) out of the detected samples (in blue), and finds MAP (maximum a posteriori, in yellow) using the PCA code.

3.3 Parameters

Let X^0 and X^1 be the background and foreground sample sets, resprectively. In each set, a sample is represented as a vector in a feature space, and let v_i be the i-th feature value. For simplicity, we use y=0 for a background sample in X^0 and y=1 for a foreground sample in X^1. We assume the conditional distributions $p(v_i \mid y=1)$ and $p(v_i \mid y=0)$ are Gaussian distributed, with four parameters (μ_i^1,σ_i^1, μ_i^0,σ_i^0), where μ_i^1 and σ_i^1 are the mean and standard deviation of the positive sample set X^1 in the i-th feature, and μ_i^0 and σ_i^0 are the mean and standard deviation of the negative sample set X^0 in the i-th feature.

3.4 FDA classifier

In our work, FDA is thought of as a way to get a linear combination of weighted features for classification, and we can directly get weights from the training data (Welling, 2005). Similarly to Viola and Jones (2001), we propose a multi-class FDA classifier where X^1 has just one class but X^0 is a multi-class set. Since every sample in X^0 is marked as one class, the mean and the variance of every background class is just the sample value and 0, respectively. Similarly to (2-1), we use the following equations under multi-class background situation to calculate the vector W_{fda} as a feature weight score:

$$w_{\text{fda}}(i) = \{\sum_{v_i \in Y} (\mu_i^1 - v_i)^2 + (\sigma_i^0)^2\} / (\sigma_i^1)^2, \qquad (3-1)$$

$$W_{\text{fda}}(i) = \text{normalize}(w_{\text{fda}}(i)),$$

$$f(v) = \sum_{i=1}^{n} W_{\text{fda}}(i)\ v_i,$$

$$\theta = W_{\text{fda}} \cdot u^1,$$

$$h = \sqrt{(W_{\text{fda}}^T \Sigma^1 W_{\text{fda}})}$$

$$F(v) = \begin{cases} 1, & \text{if } |f(v) - \theta| \le 2h \\ 0, & \text{if } |f(v) - \theta| > 2h, \end{cases}$$

where u^1 is the mean vector of all positive samples, Σ^1 is the diagonal matrix with diagonal entries $(\sigma_i^1)^2$, and n is the number of features. This FDA classifier is used to test samples and pick out target candidates from a group of detected samples. The result is a group of target candidates.

3.5 Matching the best target

We store the current target template T for matching. After getting target candidates from the FDA classifier, all candidate vectors are projected by the PCA transformation matrix W_{pca} to get PCA codes. Then we calculate the similarity between the current target template T and all target candidates.

Let V_T be the template vector, and v_j be some candidate vector. Let W_{pca} be the PCA transformation matrix, and d_n be the number of candidate samples. The current detected target response will be taken as the one with maximal value of similarity s_j:

$$s_j = \frac{(W_{\text{pca}} V_T) \cdot (W_{\text{pca}} v_j)}{|W_{\text{pca}} V_T| \ |W_{\text{pca}} v_j|}$$

$$J = \arg\max_j s_j, \qquad j = 1, \ldots, d_n,$$

where J is the index of the best matched image patch.

3.6 Online update of models

3.6.1 Online update of a generative model

In order to learn different views of an object, it is best to store a complete set of target templates. However it is impossible before all frames are viewed. Hence we use a cyclic array *TA* as the training data for PCA, which has a head index *T_head* and a tail index *T_tail,* and *T_total* is the length of the array. Our solution is to pick up a number (*T_total*) of the most recent typical object views stored in the template set, and drop out-of-date templates in time. In every frame, the template set is refreshed, and we apply the PCA process on all templates in *TA*, and obtain a new PCA transform matrix W_{pca}.

When a new tracking result is captured, it is not always suitable as a new target template because the result may have been ruined by occlusion, view change or inaccuracy. In order to keep the templates more reasonable, we take the linear combination of the captured target T_1 and the previous target template T_0, and T is updated by

$$T = \lambda T_1 + (1 - \lambda) T_0, \qquad 1 \ge \lambda \ge 0,$$

where λ is the learning parameter of the template.

3.6.2 Online feature weighting update

The FDA classifier has to be refreshed at every frame for background change and target view variation. The algorithm collects the current positive sample set and the negative sample set, and uses them as training data to update the FDA classifier for getting a new W_{fda}. We also take a linear combination of the previous weight vectors, i.e., W_{fda} is updated by

$$W_{\text{fda}} = \eta W_1 + (1-\eta)W_0 \qquad 1 \geq \eta \geq 0$$

where W_1 is the currently learned W_{fda} whereas W_0 is the previous W_{fda} before refreshing. η is the learning parameter of feature weighting.

3.7 Our tracking algorithm

We summarize our tracking algorithm in Algorithms 1~3.

Algorithm 1. The proposed tracking algorithm

Input: a video stream, and the initial target position x^1.
Output: tracking result at every frame

```
Initialize TA with sampling targets in the first frame
Repeat from the first frame to the last frame
    1.Execute Algorithm 2(online learning and updating)
    2.Execute Algorithm 3(detection).
```

Algorithm 2. Online learning and updating

Input: t-th video frame, the tracked location $\boldsymbol{x}^t$, and the template set ***TA***
Output: Feature weight vector $\boldsymbol{W}^t_{\text{fda}}$**,** Fisher classifier $\boldsymbol{F}^t(\,)$, and PCA transformation matrix $\boldsymbol{W}^t_{\text{pca}}$

1. Sample a positive sample set $\boldsymbol{X}^1$ and a negative sample set $\boldsymbol{X}^0$ around $\boldsymbol{x}^t$ and extract feature vectors for every set.
2. Learn the parameters $\boldsymbol{p}^t_i=(\mu^1_i, \sigma^1_i, \mu^0_i, \sigma^0_i)$.
3. Learn and update feature weight vector $\boldsymbol{W}^t_{\text{fda}}$ and **FDA** classifier $\boldsymbol{F}^t(\,)$.
4. **S**ample around the location $\boldsymbol{x}^t$**,** extract feature vector v as a new target template, and refresh TA with λ .
5. Apply PCA and update the transformation matrix $\boldsymbol{W}^t_{\text{pca}}$ with η .

Algorithm 3. Detection

Input: (t+1)-th frame, the feature weight vector $\boldsymbol{W}^t_{\text{fda}}$, PCA transformation matrix W^t_{pca}, Fisher classifier $\boldsymbol{F}^t(\,)$.
Output: detection result $\boldsymbol{x}^{t+1}$**.**

1. Sample a set of image patches around the previous tracked location $\boldsymbol{x}^t$ at the (t+1)-th frame, and extract features V with low dimensionality.
2. Apply Fisher classification $\boldsymbol{F}^t(\,)$ to each sample vector V and find the target candidate set ***C***
3. On the set **C**, use the transformation matrix $\boldsymbol{W}^t_{\text{pca}}$ to project samples to get the PCA code for every candidate.
4. Calculate the similarity of PCA codes between candidates and the current target templates stored in ***TA*****(T_head)**.
5. The candidate with the maximal similarity is chosen as the current tracking result x^{t+1}.

4. Experiments and Results

Using MATLAB, we compared the result of our proposed algorithm (we name it FP) with those by CT in Zhang et al. (2012) and by SIGKK in Chen and Park (2013) which was proposed in our previous work. The source code of FP was created by modifying the source code of Zhang et al. (2012) to implement our proposed strategies. We used the same parameters except for the number of features, which we increased from 50 to 100. Both the template update parameter λ and the weighting learning parameter η in our experiment were set to 0.85.

We used the public dataset 'David indoor' (in gray scale) from http://www4.comp.polyu. edu. hk/ ~cslzhang/CT/CT.htm, which has 462 frames with resolution 320*240. We ran each algorithm 10 times and computed the average results.

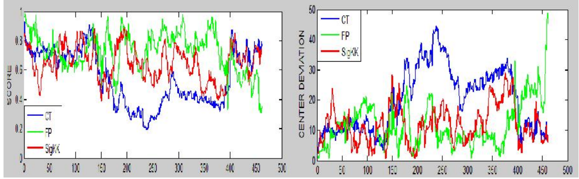
Our tracker ran at 20 fps (frames per second) on the PC with Pentium Dual-Core 2.80 GHz CPU and 4 GB RAM.

4.1 Accuracy comparison

We used two metrics to evaluate the three algorithms. The first metric is the success rate defined by

$$score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)},$$

where ROI_T is the tracking bounding box and ROI_G is the ground truth bounding box. If the score is higher than 0.5 in some frame, the tracking result in that frame is considered as a good result. The other metric is the center location error in pixel distance from the ground truth. We manually measured three times, and use their average.



(a) (b)

Figure 3. Comparison of the three algorithms FP, CT, and SigKK by the two metrics. (a) shows the *score* variations (in time) of the three algorithms. (b) shows the variation of center location error in pixel distance.

Table 1. Accuracy comparison by the two metrics

Algorithms	Metrics	
	Successes (# of frames)	Center deviation (in pixels)
CT	238	19.90
FP	441	11.77
SigKK	403	11.96

Figure 3 and Table 1 show the comparison using the two metrics. In general, the proposed FP algorithm is better than SigKK, which is again better than CT.

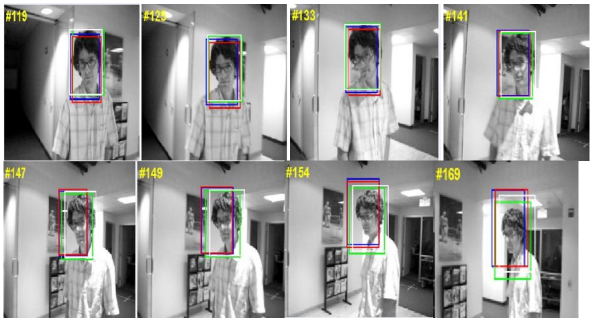
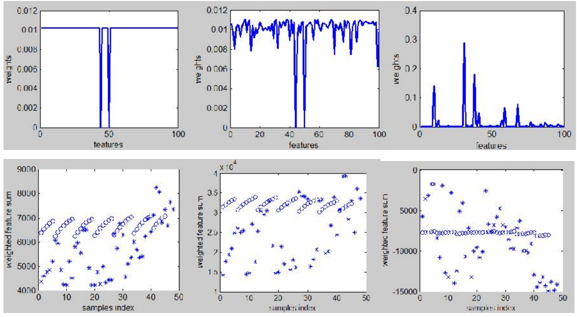


Figure 4. Comparison of the ability to deal with cluttered background and view changes. The white boxes are the ground truths, and the red/blue/green ones are by SigKK, CT, and FP, respectively.

In Fig. 4, we captured 8 frames from the tracking results that involve cluttered background and view changes. We can also see that FP outperforms the other two algorithms.

4.2 Ability of classification

We also obtained some experimental results that show our proposed feature weighting algorithm of FP really works better than the other two previous works, CT and SigKK. The following figures are obtained by using the tracking data, with 45 positive samples (within the distance of 4 pixels from the ground truth) and 50 negative samples (with distance between 8-30 pixels from the ground truth) around the target.



(a) (b) (c)

Figure 5. Feature weighting for classification. There are three results in each column from left to right: (a) by using uniform weights; (b) by SigKK and (c) by FP. The top row shows feature weights. The bottom row shows the linear combination of weighted feature components. Tiny circles are positive samples, and tiny stars are negative samples.

The top figure of 5(a) is the result when the uniform weight is given to every feature component, without discriminative weighting, i.e., we used $W_{\text{fda}}=I$, where I is the vector of all one's. However, since features are generated automatically, there are two features with approximately 0 values, which should be removed by using weight zero. As the bottom figure of 5(a) shows, it is difficult to distinguish positive samples from the negative ones, because they are somewhat mixed. The figures in the column 5(b) are by SigKK, which show better separation. Finally those in the column 5(c) are by FP using the proposed value of w_{fda} in (3-1), and they show great improvement in classification.

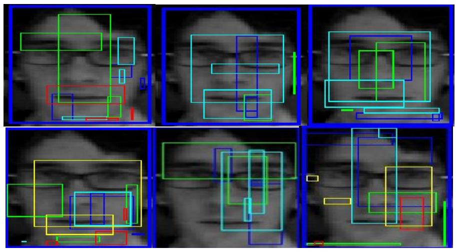


Figure 6. The most significant 3-5 features with the hightst rank from each groups of 100 features. One feature comprises the rectangles in the same color.

Our feature weighting formula has an important meaning in that it allows us to consider some significant features only. The algorithm randomly generates 20 groups of features, each having 100 features. By the proposed feature weighting formula, the algorithm ranks every group and picks up the most significant 3-5 features from every group. Figure 6 shows a few rectangles that represent those significant features. For example, since a face, which is one of significant features, is symmetric, we can find some symmetric rectangles in the figure. For other significant features like a nose, a mouth or an eye, we can see corresponding rectangles that represent them.

5. Conclusion and Future Works

In this paper, we proposed a tracking algorithm based on image classification involving online feature weighting. The proposed method can be characterized by the following three aspects.

Firstly, we used an extension of the Fisher discriminative analysis, i.e., we formulated features' discriminative power with the training data, and use it to weight features. This feature weighting is independent of any specific application, and it can be combined with any representation of feature space to train classifiers. Secondly, it uses a two-stage detection method, in which FDA and PCA are used one after the other to get better classification and matching results. This two-stage detection perfectly combines both the generative and the discriminative methods, each of which supplements each other's shortcomings. Lastly, it implements online model update smoothly to deal with a drift problem.

Experimental results show that the proposed method FP gives better detection and tracking results when compared with some competitive algorithms. To realize longer term tracking, some of the directions might be to combine the AdaBoost algorithm for better features selection. In addition, we may adopt a particle filter method as a motion model for better prediction of the object's position. Nonetheless, there always exists a tradeoff between accuracy and efficiency, which we need to balance on our way.

Acknowledgements:

The authors would like to thank Dr. K. Zhang for making their videos publicly available with no restriction. This work was supported by the GRRC program of Gyeonggi province [GRRC SUWON 2013-B1, Cooperative CCTV Image Based Context-Aware Process Technology].

Corresponding Author:
Dr. Pil Seong Park
Department of Computer Science
University of Suwon, Gyunggi-do 445-743, Korea
E-mail: pspark@suwon.ac.kr

References

1. Abdi H, Williams LJ. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2010; 2(4): 433–59.
2. Bao C, Wu Y, Ling H, Ji H. Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach. CVPR 2012;1830 –7.
3. Chen Y, Park PS. A Semi-Supervised Classification Boosted by Discriminative Features for Long Term Object Tracking. Proceedings of the IW-STEIC 2013; 34-42.
4. Chen Y, Park PS, Li A. A Novel Approach of On-line Discriminative Tracking Feature Selection. IJCIT 2013;2(2):392-7.
5. Dollar P, Tu Z, Tao H, Belongie S. Feature mining for image classification. Proceedings of IEEE Conference on CVPR 2007;1-8.
6. Fisher RA. The use of multiple measurements in taxonomic problems. Annals of Eugenics 1936; 7: 179-88.
7. Huang K, Aviyente S. Sparse representation for signal classification. NIPS 2006.
8. Li G, Liang D, Huang Q, Jiang S, Gao W. Object tracking using incremental 2D-LDA learning and Bayes inference. ICIP 2008; 1568-71.
9. Lin RS, Yang MH, Levinson SE. Object tracking using incremental Fisher discriminant analysis. ICPR 2004;2:757-60.
10. Qiang H, Zhao H. A Survey of Visual Tracking. ACTA AUTOMATICA SINICA 2006;32(4): 603-17.
11. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. IEEE Computer Society Conference 2001;1:511-8
12. Wang J, Chen X, Gao W. Online Selecting Discriminative Tracking Features Using Particle Filter. IEEE CVPR 2005; 2:1037–42
13. Welling M. Fisher linear discriminant analysis. Department of Computer Science, University of Toronto, 2005.
14. Yang H, Shao L, Zheng F, Wang L, Song Z. Recent advances and trends in visual tracking: A review. Neuro computing 2011;74(18):3823-31.
15. Yilmaz A, Javed O, Shah M. Object tracking: A survey. ACM Computing Surveys CSUR 2006; 38(4):No. 13.
16. Zhang K, Zhang L, Yang M. Real-Time Compressive Tracking. ECCV 2012.
17. Zhao W, Chellappa R, Krishnaswamy A. Discriminant analysis of principal components for face recognition. 3rd IEEE Automatic Face and Gesture Recognition 1998;336-41.

5/16/2014