Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(7):659-669

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

A novel kernel-PLS method for object tracking

Yi Ouyang*, Yun Ling and Biyan Wu

School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China

ABSTRACT

In this paper, we propose a On-line kernel-PLS approach to improving both the robustness and accuracy of object tracking which is appropriate for real-time video surveillance. Typical tracking with color histogram matching provides robustness but has insufficient accuracy, because it does not involve spatial information. On the other hand, tracking with pixel-wise matching achieves accurate performance but is not robust against deformation of a target object. To tackle these problems, this paper presents a tracking method that combine histogram-wise matching and pixel-wise template matching via leans a robust object representation by Kernel-PLS analysis and adapts to appearance change of the target. In this paper, we propose a novel On-line Kernel-PLS analysis, for generating a low-dimensional discriminative feature subspace. As object appearance is temporally correlated and likely to repeat over time, we learn and adapt multiple appearance models with On-line Kernel-PLS analysis for robust tracking.

Key words: Image processing; PLS; Object Tracking; Classification; Feature spaces

INTRODUCTION

Target tracking has been an important topic in computer vision for several years. Recent years have been significant progress in tracking. To distinguish targets with backgrounds and with each other, most visual tracking methods focus on tracking target appearance separately; they usually try to find proper appearance models that distinguish object with all other targets or backgrounds, and adopt meanshift or particle filtering like approach to online adjust target appearance models, and use updated models to continuously track targets.

The most tracking methods reported to handle this difficulty, thus far, is to adaptively update the target appearance model at each frame: learn a new appearance model with time invariant characteristics extracted from historic observed target samples, and adopt the model to the current frame. Such as, IVT [1] algorithm uses a subspace model by adding adaptively modify the appearance of the model. Tracking-Learning-Detection (TLD) [2] method to track task is decomposed into three sub-processes: tracking, learning and learning, each sub- task as a separate task , each sub-task can be performed s tracking will fail . (2) the target disappeared, detection operator continues . (3) less real-time tracking and other issues . In order to solve the above problems , Kalal [3] analyze a variety of information in video images and proposed a new learning framework called P-N Learning for training a binary classifier from labeled and unlabeled examples. The learning process is guided by positive experts and negative experts constraints which restrict the labeling of the unlabeled set. P-N learning evaluates the classifier on the unlabeled data, identifies examples that have been classified in contradiction with structural constraints and augments the training set with the corrected samples in an iterative process. Learning processes for any errors exist , the expert operator P-N due to mutual compensation of the error probability is limited within a certain range in order to achieve stability. Based on P-N learning method, Saigo [4] proposed PLS method to obtain a better target tracking. The Fragment-based tracker [11] aims to solve partial occlusion with a representation based on histograms of local patches. The tracking task is carried out by combing votes of matching local patches using a template. There are some sparse matrixes combined with particle filter for target tracking applications L1 [5].
Kernel-based pattern recognition methods [14] such as Support Vector Machines (SVMs) [15], Kernel-PCA

(KPCA)[16] and Kernel-Partial Least Squares analysis (KPLS) [17,18] have previously been applied in a multitude of contexts for exploratory analysis and classification, including biological applications [19].

Target tracking problem is challenging as tracking processing needs to deal with appearance variations caused by many factors such as illumination, pose deformation, occlusion, background clutter, and camera motion. This paper focuses on the issue of nonlinear K PLS analysis for target tracking. To tackle these challenges, we present a tracking method that learns a robust object representation by Kernel partial least squares analysis and adapts to appearance change of the target. Firstly, we extend these earlier works[4] by embedding nonlinear kernel analysis for PLS tracking. To improve the existing work, we perform the color histogram probability density function for the object color constraint is modeled as a smooth function that indicates how well the candidate set images and target is met. To improve stability for standard belief propagation, the color constraint function node is connected to four gradient variable nodes.

Although detection results can be improved by utilizing overlapping blocks for low-level feature extraction within the detection window, the dimensionality of the feature vector becomes extremely high. As a result, the speed of the human detector decreases significantly due to the time needed to extract features and project them. To overcome this problem, we employ a On-line learning approach. In a fast first stage, based on a small number of features, the majority of detection windows (those with low probability of containing humans) are discarded. The remaining windows are evaluated during a second stage where the complete set of features allows challenging samples to be correctly classified.

ON-LINE KERNEL-PLS ANALYSIS(KPLS)

The PLS method employs the descriptor matrix $X \in R^{N \times K}$, where N denotes the number of samples and K the number of variables in X, to predict the response matrix $Y \in R^{N \times M}$, where M denotes the number of variables in Y. The unique property of the PLS method compared to other linear regression methods is its ability to separate the modeling of covariation from structured noise, defined as systematic Y-orthogonal variation, while simultaneously maxim the covariance between X and Y.

The K-PLS algorithm follows the principles of the linear PLS algorithm while it is written in dual form. This implies that all expressions have been rewritten so that the input matrix X is expressed as the outer product XX^T in all instances. The outer product XX^T is subsequently replaced by the kernel matrix K in the K-PLS algorithm. K is deflated for the Y-orthogonal components. In simple terms, in the algorithm for model estimation, K consists of two instances of the transformed data matrix. One of these instances represents the predictive weights W and should thus be retained throughout the calculations, whereas the other should be deflated accordingly by the Y-orthogonal variation. Subsequently, XX^T

is substituted for the kernel Gram matrix K with entries $K_{i,j} = k(x_i, x_j)$, where x_i and x_j corresponds to the i-th and j-th row vector in the descriptor matrix X, respectively, and $k(x_i, x_j)$ represents the kernel function. Hence, one can avoid explicitly mapping X to higher-dimensional spaces as well as computing dot products in the feature space, which is computationally beneficial. The transformation to higher dimensional spaces is performed implicitly by the kernel

function $k(x_i, x_j)$; where common kernel functions use Gaussian functions as follow:

$$k(x, y) = exp(-\| x - y \|^2 / 2\sigma^2) \qquad (0.1)$$

The kernel functions in Equations (1.1) depend on the parameters σ, which influences the predictive ability of the kernel-based method. The traditional approach to kernel parameter selection is to perform an exhaustive grid search over the entire parameter space based on predefine parameter. The generalization properties of the model are evaluated using e.g. cross-validation [19] to identify the parameter setting yielding the lowest possible generalization error, which can result in a large number of calculation and run times. The projection direction in the feature space at each stage is given by the vector u_j, which is in the primal space while we must work in dual space,and express a multiple of u_j as

$$a_j u_j = X_j^T \beta_j \qquad (0.2)$$

which is clearly consistent with the derivation of u_j in the primal PLS algorithm. For the dual PLS algorithm we must implement the deflation of Y. This redundant step for the primal will be needed to get the required dual representations. We use Y_j to denote the j-th deflation.

$$\beta = Y_j Y_j^T X_j X_j^T \beta = Y_j Y_j^T K_j \beta \qquad (0.3)$$

with the normalization $\beta = \dfrac{\beta}{\|\beta\|}$,then compute a rescaled t_j and c_j.

$$t_j = a_j X_j u_j = X_j X_j^T \beta_j = K_j \beta_j$$
$$c_j = \frac{Y_j^T t_j}{t_j^T t_j} = \frac{Y_j^T X_j u_j}{a_j u_j^T X_j^T X_j u_j} = \frac{c_{j'}}{a_j} \qquad (0.4)$$

where $c_{j'}$ denotes the same weight vectors c in PLS algorithm. t_j can be consider as a rescaled dual representation of the output vector $c_{j'}$.

Let u_j, v_j, σ_j be the first singular vector of $X_j^T Y$,the deflation of X_j as

$$X_{j+1} = (I - \frac{t_j t_j^T}{t_j^T t_j}) X_j \qquad (0.5)$$

with an equivalent deflation of the kernel matrix

$$K_{j+1} = X_{j+1} X_{j+1}^T = (I - \frac{t_j t_j^T}{t_j^T t_j}) K_j (I - \frac{t_j t_j^T}{t_j^T t_j}) \qquad (0.6)$$

In this paper, we formulate object tracking as a classification problem with On-line Kernel PLS analysis to learn a low-dimensional and discriminative feature subspace. S.Wold et al. [12,13]were the first to extend the linear PLS model to its nonlinear form. They have done this by replacing the linear inner relation between the score vectors t and u by a nonlinear model.

$$u = k(t) + e = k(X, w) + e \qquad (0.7)$$

where k represents a continuous nonlinear function, e denotes a vector of residuals. The relation between each pair of latent variables is modeled separately. We use radial basis function to model k. It can be observed that the vector of weights of w, computed in the first step of the NIPALS algorithm, represents the sample covariance between the output space score vector u and the input space data matrix X. However, the use of a nonlinear model to relate the score vectors in the inner relation affects the computation of w. Although w represents the association among variables of X and u also in nonlinear PLS, this association will be closely related to covariance values only if the nonlinear mapping between latent variables is monotonic and slightly nonlinear.

In our model, a kernel function modeling the inner relation is used to update an initial KPLS estimate of the weight vector w. S.Wold [12,13] proposed to update w by means of a Newton-Raphson linearization of g. The procedure thus consists of a first-order Taylor series expansion of g, followed by the calculation of the correction term $\square w$ which is used to update w. So, consider the nonlinear inner relation, where $k(t) = k(X, w)$ is continuous and differentiable with respect to w. The second-order Taylor expansion of has the form

$$\hat{u} = u_0 + \frac{\partial k}{\partial w} \Delta w \qquad (0.8)$$

where $u_0 = k(t)$ is the value of g at the known value of t. Similarly, $\frac{\partial k}{\partial w}$ stands for the partial derivatives of g numerically evaluated at the same known value of t. The second term of 12 can be written element-wise as

$$\frac{\partial k}{\partial w} = \sum_{i=1}^{N} \frac{\partial k}{\partial w_i} \Delta w \qquad (0.9)$$

At this point several different methods to compute the correction Δw were proposed. To simplify further notation consider the matrix form of the linear approximation $\hat{u}$

$$\hat{u} = Zv \qquad (0.10)$$

where $Z = [u_0 \frac{\partial k}{\partial w}]$ and $v = [1 \Delta w]^T$. The following variants to compute Δw were suggested:

$$k(x, y) = \sum_{i=1}^{D} \alpha_i \phi(x) \phi(y) = \Phi(x)^T \Phi(y)$$
$$= < \Phi(x), \Phi(y) > \qquad (0.11)$$

where $\{\phi_i\}$ is a sequence of linearly independent functions, $\{\alpha_i\}$ are positive numbers and $D \leq N$ is the dimension of the space H. Following this relation the feature map Φ can be written as

$$\Phi : X \to F \qquad (0.12)$$
$$x \to \Phi(x) = (\sqrt{a_1}\phi_1(x), \sqrt{a_2}\phi_2(x), ..., \sqrt{a_N}\phi_N(x)) \qquad (0.13)$$

Thus, if we are only interested in the computation of dot products in F, it does not matter how F was constructed and simply all dot products can be replaced by a unique kernel function associated with F . This is important to note because different feature spaces associated with the same kernel function can be constructed. In literature, this replacement of a dot product with the kernel function value is known as the kernel trick method.

A modified version of the NIPALS algorithm where steps 1 and 3 are merged and the score vectors t and u are scaled to unit norm instead of scaling the weight vectors w and c. The obtained kernel form of the NIPALS algorithm is as follows

Step1: $t = \Phi\Phi^T u = Ku$

Step2: $\| t \| \to 1$

Step3: $c = Y^T t$

Step4: $u = Yc$

Step5: $\| u \| \to 1$

Although step 2 guarantee orthogonality of the score vectors can be rescaled to follow the standard linear NIPALS algorithm with the unit norm weight vectors vector w. Step 3 and 4 can be further merged which may become useful in applications where an analogous kernel mapping Φ of the Y-space data is considered; that is, the Gram matrix $K_y = \Phi\Phi^T$ of the cross dot products between all mapped output data is constructed. Then, the kernel NIPALS algorithm consists of the following four steps. Step 3 and 4 can be further merged which may become useful in applications where an analogous kernel mapping Φ of the Y-space data is considered; that is the Gram matrix $K_y = \Phi\Phi^T$ of the cross dot products between all mapped output data is constructed. Then, the kernel NIPALS algorithm consists of the following four steps

Step1: $t = Ku$
Step2: $\| t \| \to 1$
Step3: $u = K_y t$
Step4: $\| u \| \to 1$

The On-line KPLS algorithm learns a threshold in linear function in a kernel-defined feature space.

$$h(x) = sgn < w, \phi(x) > \qquad (0.14)$$

If the weight vector after t updates is denoted by w_t then the update rule for the (t+1) update when an sample (x_i, y_i) is misclassified is given by

$$w_{t+1} = w_t + y\phi(x_i) \qquad (0.15)$$

Hence, the corresponding dual update rule is simple

$$a_i = a_i + 1 \qquad (0.16)$$

if we assume that the weight vector is expressed as

$$w_t = \sum_{i=1}^{N} a_i y_i \phi(x_i) \qquad (0.17)$$

Once the weight matrix $W = [w_1, ..., w_k]$ is computed, the initial appearance model can be denoted by $A_1 = \{x_p, \hat{x}, W\}$, where x_p is the mean of the positive samples. A test sample, $x \in R^m$, can be projected onto the learned latent feature space specified by A_1 to get a latent feature vector $z = W^T(x - \hat{x})$. Using Z with lower dimensionality, a target object can be more easily discriminated from the background than in the original feature space X.

Algorithm 1 Pseudo-code for the On-line K-PLS algorithm

Estimate the predictive Y-weights (c_1) by eigen-vector decomposition of $Y^T KY$
Project Y onto c_1 to achieve the predictive score matrix of $Y : u_1 \leftarrow Yc_1$
Calculate the kernel matrix of $K_{ij} = k(x_i, x_j)$
$K_1 = K$
for t=2:k **do**
  $\beta_t = Y_1$
  normalization $||\beta_t|| \rightarrow 1$
  repeat
    $\beta_t = YY^T K_j \beta_j$
    normalization $||\beta_t|| \rightarrow 1$
  until convergence
  $t_j = K_j \beta_j$
  $c_j = Y^T t_j / ||t_j||^2$
  $Y = Y - t_j c_j$
  $K_{t+1} = (I - \frac{t_j t_j^T}{t_j^T t_j}) K_t (I - \frac{t_j t_j^T}{t_j^T t_j})$
end for
$\Lambda = [\beta_1, ..., \beta_k]$,$T = [t_1, ..., t_k]$
$w = \Lambda(T^T K \Lambda)^{-1} T^T Y$

Figure 1.Pseudo-code for the On-line KPLS algorithm

TARGET APPEARANCE MODEL

Since the appearance change of an object during a long period of time may be quite nonlinear and complex, one linear appearance model is not likely to suffice. However, appearance of a target object may be temporally correlated and may repeat over time. We therefore learn multiple appearance models for more effective object representation.

Video sequence of a target over a long period can be divided into multiple sets. Within the i-th set, the object appearance dosed not change much and we use KPLS analysis to learn a discriminative appearance model $A_i = \{x_i^p, x_i, W_i\}$. Therefore, the appearance of a target object can be represented by multiple appearance models $A = \{A_1, ..., A_k\}$, where k is the number of appearance models. The proposed representation scheme is more effective than existing methods base on single linear appearance model.

In this paper, the distance between a test sample $x \in R^m$ and the learned appearance model set A is defined as

$$Dist = \sum_{i=1}^{k} ||W_i^T (x - \overline{x}) - W_i^T (\tilde{x}^p - \overline{x})||_2^2 \quad (0.18)$$

where $\tilde{x}^p$ is the mean of the positive samples used in training A_i, $\overline{x_i}$ is the mean of all the samples in training A_i, and $||\cdot||_2$ is Euclidean norm.

The target and background appearances may change due to factors such as illumination, pose, occlusion, camera motion, and so on. To deal with this problem, we propose an adaptive object representation method. Let the current set of appearance models be $A = \{A_i \mid i = 1, ..., k, k \leq K\}$.

When the tracking result at time t is obtained, we use the corresponding target observation x_t to update A. Since we have computed the distances from the target observation x_t to all the appearance models in A for determining the tracking result, we select the appearance model A_s with the smallest distance d_s and the appearance model A_l with the largest distance d'. If d_s is less than a predefined threshold, x_t is utilized to update A_s.

The update process includes three components: the mean of the positive sample $\overline{x_i^p}$,the mean of all the training samples $\overline{x}$, and the weight matrix W . $\overline{x^p}$ can be updated by using a random update probability.

Both $\hat{x}$ and W can be updated by KPLS method with the positive and negative samples. If $\| A_s - A_l \|$ is larger than the predefined threshold and $k < K$, a new appearance model A_{k+1} is added to A. If $\| A_s - A_l \|$ is larger than the predefined threshold and $k = K$, a new appearance model is initialized to replace A_l in A. The proposed adaptive appearance model is summarized as follow:

Algorithm 2 Adaptive Appearance Model Based on K-PLS analysis

Initalize A with KPLS analysis when t=1.
for t=2:T **do**
  Find $A_s = arg\min_{x=min(Dist)}\{x_i^p, x_i, W_i\}$
  Find $A_l = arg\max_{x=max(Dist)}\{x_i^p, x_i, W_i\}$
  if $||A_s - A_l|| < Threshold$ **then**
    Get a random update probability $q \in [0, 1]$
    $x_s^p \leftarrow qx_s^p + (1-q)x_t;$
    Update x_s and W_s using KPLS analysis
  else
    if $k < K$ **then**
      Learn a new appearance model A_{k+1}, and add it to A
    else
      Learn a new appearance model and use it to update A_l in A
    end if
  end if
end for

Figure 2.Adaptive Appearance Model Based on KPLS analysis

ON-LINE KPLS TRACKING METHOD VIA MAP

Given the observation set of the target $x_{1:t} = [x_1, \ldots x_t]$ up to time t, the tracking result s_t can be determined by the Maximum A Posteriori (MAP) estimation,

$$\hat{s}_t = argmax\ p(s_t \mid x_{1:t}) \qquad (0.19)$$

where $p(s_t \mid x_{1:t})$ is inferred by the Bayes theorem recursively with follow equation.

$$p(s_t \mid x_{1:t}) \propto p(x_t \mid s_t) p(s_t \mid x_{1:t}) \qquad (0.20)$$

$$p(s_t \mid x_{1:t-1}) = \int p(s-t \mid s_{t-1}) p(s_{t-1} \mid x_{1:t-1}) ds_{t-1} \qquad (0.21)$$

This inference is governed by the dynamic model $p(s_t \mid s_{1:t-1})$ which describes the temporal correlation of the tracking results in consecutive frames, and the likelihood function (i.e., observation model) $p(x_t \mid s_t)$ which denotes the likelihood of s_t observing x_t .
The main issue for any adaptive appearance model is that it is likely to use noisy or mis-aligned observations for update, thereby causing tracking drift gradually. For online tracking, the only ground truth at our disposal is the labeled target object in the first frame.

Let X as random variable of a video sequence, defining the state vector $X_t = \{L_{(x,y)}, s_t\}$,where $L_{(x,y)}$ is the target center location, s_t is the scale factor, $p(y_t \mid x)$ denotes the probability of cliques belongs to target. Motion model $p(x_t \mid x_{t-1})$ generate a predict x, which denotes the correlation of target's temporal structure in the video. We assume that the motion model obeys Gaussian distribution:

$$p(x_t \mid x_{t-1}) = N(m(x_t, x_{t-1}), \Lambda) \qquad (0.22)$$

where Λ is the diagonal covariance matrix, $m(x_t, x_{t-1}$ denotes the means of two random variable. We use Bayes filtering method to track target .

Prediction stage:

$$Bel(x) = \int p(x_t \mid x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Update stage:

$$Bel(x_t) \propto a_t p(y_t \mid x_t) Bel(x_t)$$

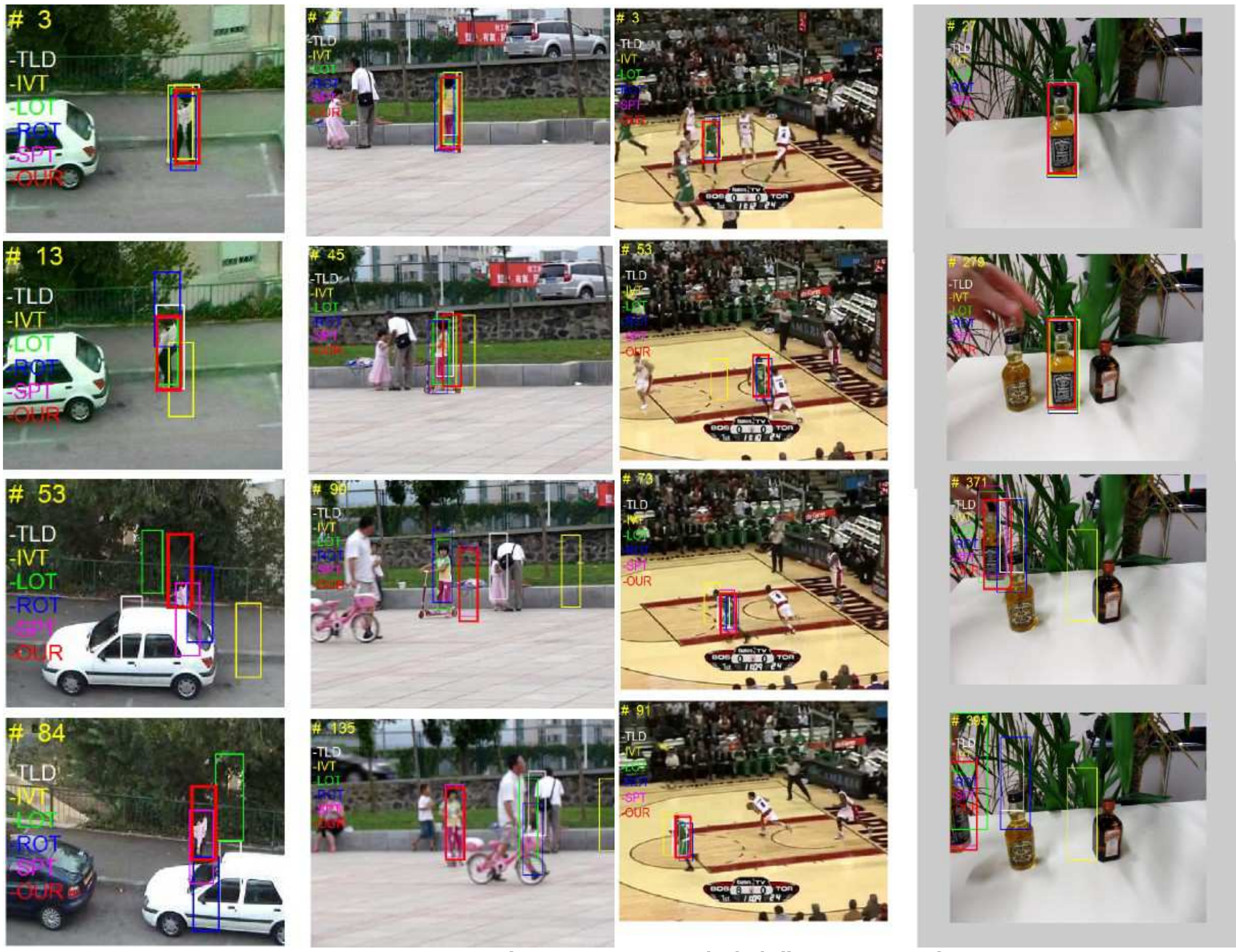
Observation Markov assumptions: Observation depends only on the state of the current observations, i.e. $p(y_{1:t} \mid x_t) = p(y_t \mid x_t) p(z_{1:t-1} \mid x_t)$, the filtering process is mainly determined by the dynamic model $p(x_t \mid x_{t-1})$, which describes the state temporal correlation of the target between frames .

EXPERIMENTAL ANALYSIS

To evaluate On-line KPLS, we compile a set of 15 challenging tracking sequences (e.g. car4, woman) that are publicly available online. Due to space constraints, we will only show results on 4 of these sequences. These videos are recorded in indoor and outdoor environments and include challenging appearance variations due to changes in pose, illumination, scale, and the presence of occlusion. The center distance error results of proposed tracking method are compared with that of current state-of-the-art methods IVT [1], TLD [2], LOT [7] and ROT [8]. Experimental video sequences singer1 and basketball from VTD[9], lemming and liquor from PROST[10], woman from FragTrack[11] and girl_mov from SPT [6]. Implemented in MATLAB on an Intel Core 2.26 GHz computer with 4GB RAM and no code optimization.

Table 1. The mean of Center D denote average errors of center position

Seq\Method	IVT	TLD	SPT	LOT	ROT	OUR
Basketball	94	7	5	6	6	7
Girl_mov	216	128	26	36	105	31
Bolt	83	-	7	12	150	25
Liquor	54	30	8	9	34	12
Woman	161	-	11	119	113	20



woman girl_mov basketball liquor

Figure 3. Tracking results comparison during tracking process

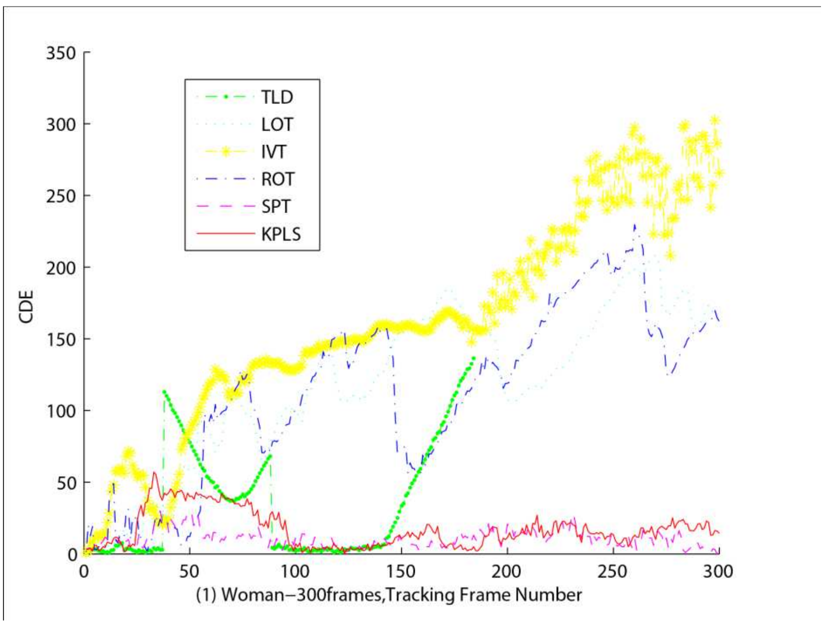


Figure 4. CDE results comparison during tracking process

Fig 3 woman and liquor subsequence occurred with the camera movement, IVT tracking will fail. The key reason is no full use of the target and background appearance model, and making the tracking accuracy is limited. On-line KPLS

tracking method considers local and environmental features surrounding, and adaptively adjusting the target appearance model, which can more accurately distinguish target and improve tracking accuracy.
In Figure3 basketball sequence, because the target of the block is not serious, the proposed method and SPT almost unanimously, in girl_mov, when traced to 115. In liquor object appears in a similar situation, the proposed method with SPT similar. The presence of the target under occlusion, the proposed method is superior to other methods.

In Table-1and Figure 4, the paper uses Center Distance Error (CDE)to analysis the tracking performance, where '-' indicates partial frame detection failure tracking process.CDE represents the center position coordinates errors between the tracking results and the reference standard (ground truth) , where$||.||$ is the Euler distance, '-' indicates some frames tracking failure in the tracking process.

The target objects are partially occluded in the woman sequences, which has a partially occlusion phenomenon. Using On-line KPLS method can eliminate the target because of changes in background appearance model for the impact, and the adaptive appearance learning approach through dynamic update appearance models, target tracking accuracy of better than TLD, IVT, LOT and ROT methods to detect and deal with serious obscured targets.

The results (Figure 4) show that it is normal condition in the center and the ground truth detection frame error is basically the same, when there is heavy occlusion, our method is significantly better than other tracking algorithms.

CONCLUSION

This paper presents a novel On-line Kernel PLS analysis method for object tracking, and utilizes the positive samples and negative samples for adaptively updated information on the target - background appearance model. To solving the these problems, such as occlusion, illumination changing, and the shape changing during the process of the target tracking. We employ the image KPLS weight scores as confidence response map to determine the target location information. Although traditional tracking method tracking detection results can be improved by utilizing overlapping blocks for low-level feature extraction within the detection window, the dimensionality of the feature vector becomes extremely high. As a result, the speed of the human detector decreases significantly due to the time needed to extract features and project them.

To overcome this problem, we employ a On-line learning approach. In a fast first stage, based on a small number of features, the majority of detection windows (those with low probability of containing humans) are discarded. The remaining windows are evaluated during a second stage where the complete set of features allows challenging samples to be correctly classified.

Acknowledgment
The authors would like to thanks the anonymous reviewers for their constructive and useful comments, which helped in improving the presentation of our work.

REFERENCES

[1] Ross D A ,Lim J , Lin R S, et al. *International Journal of Computer Vision*, **2008**, 77(1-3): 125-141.
[2] Kalal Z, Mikolajczyk K and Matas J. *IEEE Transactions* on. **2012**, 34(7): 1409-1422.
[3] Kalal Z ,Matas J and Mikolajczyk K. Pn learning: Bootstrapping binary classifiers by structural constraints, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, **2010**: 49-56.
[4] Saigo H, Kr mer N and Tsuda K. Partial least squares regression for graph mining. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, **2008**: 578-586.
[5] Wang Yu-Xiang, Xu Han and Ma Ting-Huai. *Journal of Electronics & Information Technology*,**2012**,V34(1): 223-226.
[6] Wang S ,Lu H and Yang F, et al. Superpixel tracking[C].Computer Vision (ICCV), 2011 IEEE International Conference on, **2011**: 1323-1330.
[7] Oron S ,Bar-Hillel A and Levi D, et al. Locally orderless tracking. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, **2012**: 1940-1947.
[8] Ning J, Zhang L, Zhang D and Wu C. Robust object tracking using joint color-texture histogram. International Journal of Pattern Recognition and Artificial Intelligence. **2009**, 23(07): 1245-1263.
[9] Kwon J and Lee K M. Visual tracking decomposition. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, **2010**: 1269-1276.
[10] Santner J ,Leistner C and Saffari A, et al. Prost: Parallel robust online simple tracking. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, **2010**: 723-730.

[11] Adam A ,Rivlin E and Shimshoni I. Robust Fragments-based Tracking using the Integral Histogram.Computer Vision and Pattern Recognition **2006** IEEE Computer Society Conference on, **2006**:798-805.
[12] Trygg J, Wold S. *Journal Chemometrics*,**2002**, 16:119-128.
[13] Trygg J, Wold S. *Journal Chemometrics*,**2003**, 17:53-64
[14] Shawe-Taylor J, Cristianini N: Kernel methods for pattern analysis, Cambridge University Press; **2004**:462.
[15] Schölkopf B, Smola A: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.Cambridge , MIT Press; **2001**.
[16] Rosipal R, Girolami M,Trejo LJ, Cichocki A: *Neural Comput Appl* **2001**, 10(3):231-243.
[17] Lindgren F, Geladi P, Wold S: *Journal Chemometrics*,**1993**, 7(1):45-59.
[18] Rosipal R, Trejo LJ: *J Mach Learn Res*,**2002**,2(2):97-123.
[19] Anderson DC, Li W, Payan DG, Noble WS*: J Proteome Res* **2003**,2(2):137-146