Volume 4, Issue 2, February 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Object Tracking Using Partial Least Squares Analysis

M.Rutharesh
ME Communication Systems
Kumaraguru college of technology
Coimbatore, Tamilnadu, India

R.Naveenkumar
ME VLSI Design
Sri Shakthi Institute of Engg and Tech
Coimbatore, Tamilnadu, India

S.krishnakumar
ME Communication Systems
Kumaraguru college of technology
Coimbatore, Tamilnadu, India

Abstract*— The main objective of object tracking is to detect and track target objects in the current frame in the video sequence under various environmental conditions. The tracking of objects is difficult due to the impact of various factors like appearance change, occlusion, change in illumination, fast movement of object, etc. These problems can be overcome by effective appearance model for the target object which defines the appearance of object over time. The detection of object in the current frame is done by effective filtering stages. This is done by using two methods: particle least square analysis for the appearance model and filtering of object using particle filter to get the target object in the current frame. As the target object is continuously repeated over time, background subtraction method is used, which tests the system whether the target object is tracked properly. Subjective and objective measures are observed and calculated using this algorithm.

Keywords*— Appearance model, Particle filter, Partial least squares analysis, Object tracking, Background subtraction.

I. INTRODUCTION

Object tracking is one of the important tasks in the field of computer vision. As the use of high end computers, high quality video cameras, and the need of automatic video analysis which uses different object tracking algorithms.

There are three major steps in object tracking: detection of moving objects, tracking it on each frame, and analysis of object behaviour. The uses of object tracking are, automatic video surveillance, vehicle navigation, video indexing, traffic cam monitoring. Objects tracking are complex due to, Noise in images, Complex object motion, Non-rigid shape of the objects, Partial and full target object occlusions, Complex shape of the object, illumination changes in environment, and Real-time processing requirements.

There are several algorithms for object tracking have been proposed. These differ from each other based on object representation, appearance model used, target object to be tracked and motion of the object. This paper uses a tracking method that learns the object representation by partial least squares analysis [4] [12] and adapts to appearance change of the target and background while reducing drift. Feature selection is importance for generating an effective low-dimensional discriminative subspace. In this paper, we did this by learning a feature subspace with positive and negative samples in the high-dimensional feature space by particle least square analysis PLSA. The learned feature subspace is then utilized to construct an appearance model. As appearance of an object in consecutive frames is temporally correlated and likely to repeat over time, we learn and use multiple appearance models with PLSA for robust tracking.

The main contributions to this paper are as follows, the PLSA is used to learn low-dimensional discriminative feature subspace for object representation. Since object tracking is a task to separate the target object from the background, object representation with PLSA is effective than the widely used generative models such as principle component analysis (PCA) [6]. As no special search method is carried out to select or combine features, our representation scheme which is more efficient than existing discriminative methods [7]. Second, we represent, object with multiple appearance model for better tracking. To account for large and complex appearance change of a target object, the use more than one appearance model is more effective than existing methods with one single linear representation [9]. Third, we propose a two-stage particle filtering method. This tracking method use the appearance model which is initialized in the first frame and image observations obtained online, so tracking drift problem during model update is reduced. The proposed tracking algorithm achieves favourable performance with higher success rates and lower tracking errors.

II. OBJECT REPRESENTATION

Partial least squares analysis is a statistical method for modeling relations between the sets of variables via some latent quantities. In PLS analysis, the observed data was assumed to be generated by a process driven by a small number of latent variables. In this paper, we design object tracking as a classification problem with PLS analysis to learn a low-dimensional and discriminative feature subspace.

A. Partial Least Squares Analysis

Consider $X \in R_m$ be an m-dimensional space of variables and $Y \in R_n$ be an n-dimensional space of other variables. With N observed samples from each space $x \in X$ and $y \in Y$ forms blocks of variables, $X \in R_{N \times m}$, PLS methods find new spaces where most variations of the observed samples can be preserved, and the learned latent variables from two blocks are correlated than in the original spaces.

$$X = TPP^T + E$$
$$Y = UQ^T + F \quad (1)$$

Where $T \in R_{N \times p}$ and $U \in R_{N \times p}$ are factor matrices, $P \in R_{m \times p}$ and $Q \in R_{n \times p}$ are loading matrices, and $E \in R_{N \times m}$ and $F \in R_{N \times n}$ are error terms. With PLS analysis, each variable is represented by a p-dimensional vector.

To decompose X and Y by Equation (1), PLS algorithms computes the weight vectors $\mathbf{w}_1$ and $\mathbf{c}_1$ such that most variations in X and Y can be retained by $t_1 = X_{w1}$ and $u_1 = Y_{c1}$

$$\max var(t_1)$$
$$\max var(u_1) \quad (2)$$

where $\mathbf{t}_1$ and $\mathbf{u}_1$ are the first columns of T and U, respectively, and Var(.) denotes the variance.PLS analysis requires $\mathbf{t}_1$ to best explain $\mathbf{u}_1$

$$\max \rho(t_1, u_1) \quad (3)$$

Combining Equation (2) and Equation (3), PLSA analysis maximizes the covariance between $\mathbf{t}_1$ and $\mathbf{u}_1$ in the first step

$$\max \mathrm{cov}(t_1, u_1) = \max \sqrt{(\mathrm{Var}(t_1)\, \mathrm{var}(u_1)\, \rho(t_1, u_1))} \quad (4)$$

w_1 and c_1 is derived by solving the given optimization problem

$$max\{Xw_1, Yc_1\}$$
$$s.t\ w_1^T w_1 = 1,\ c_1^T c_1 = 1 \quad (5)$$

where (X_{w1}, Y_{c1}) defines the inter product of X_{w1} and Y_{c1}. The optimal weight vector $\mathbf{w}_1$ for the optimization problem is the first eigenvector of the following Eigen value problem.

$$X^T Y Y^T X w_1 = \lambda w_1 \quad (6)$$

Similarly, $\mathbf{c}_1$ can be obtained by solving another eigen value problem

$$Y^T X X^T c_1 = \lambda c_1 \quad (7)$$

After the first step, PLS method iteratively computes other weight vectors. The data matrices X and Y are deflated by subtracting their rank-one approximations

$$X \leftarrow X - t_1 p_1^T$$
$$Y \leftarrow Y - u_1 q_1^T \quad (8)$$

The new X and Y are used to compute w_2, c_2 based on Equation (6) and Equation (7). This process is repeated until the residuals are small or a predefined number of weight vectors $(w_1, \ldots, w_p)$ and $(c_1, \ldots, c_p)$ are obtained

B. Learning Appearance Models With PLS Analysis

In this paper, the object tracking is a classification problem which labels the target and background feature variables with different values. The PLS analysis denotes a low-dimensional space can be learned where the latent quantities from different sets of observed variables are more correlated than the original spaces [4]. Therefore, we can use PLS analysis to model the correlation of object appearance and class label due to its capacity for both dimensionality reduction and classification.

(a) (b)

Fig. 1. (a) Target object is marked by a red polygon from which a positive sample can be collected. (b) Negative samples are collected from the black box

In PLS formulation, the variables in our tracking task consists of two classes including feature vectors and class label. In the following sections, we can use $X \in R_m$ to denote the feature space for object description, and $Y \in R_n$ to denote the class label space of an object. After the target object is manually or automatically located in the first frame, we need a positive sample $\mathbf{x}_1$ by extracting a feature vector from the warped image specified by the state parameter [10]. If more positive samples are needed for training, we have to generate virtual data by small perturbations and extract corresponding feature vectors. To collect negative samples, we draw samples from an annular region defined by $\gamma < \| l_{neg} - l \| < \beta$ (γ and β are inner and outer radiuses, respectively), in which **l** is the target location, and I_{neg} is the location of a negative sample. Fig. 1 illustrates the positive and negative samples are obtained in the frame. With the obtained data set, we have to use PLS analysis to determine an appearance model of the target object.

Once the weight matrix $W = [w_1, w_2, \ldots, w_p]$ is computed, then the initial appearance model should be denoted as $A_1 = \{x_{1p}, x_1, W\}$, where X_{1P} is the mean of the positive samples. A test sample, $X \in R_m$, can be projected onto the learned latent feature space to get a latent feature vector $Z = W X_C \in R_p$, where $X_C = X - X_1$. Using the latent feature space $Z \in R_p$ with lower dimensionality, a target object should be more easily discriminated from the background than in the original feature space $X \in R_m$.

The weight vector $w_i \in R_m\,(i = 1 \ldots p)$ of W displays the importance of each real feature variable for object description and classification. If each feature variable in the selected feature space X is a function of pixel location in an object region, then the importance of feature variable is compared to the discriminability between the target and the background classes at a given location. Therefore, we have to use w_i to generate a saliency map, which shows the discriminative strength of different locations in an object region. If each variable denotes the intensity of one pixel and a feature vector represents the ensemble of pixel intensities in an object region, a subspace can be learned by PLS analysis with some positive and negative samples, and the saliency maps specified by w_i (e.g., i = 1, . . . , 10) that are shown in Fig. 1(b). In this figure red pixels indicates higher importance of a feature variable (i.e., with more discriminative strength). It is worth noticing the red pixels of the saliency map with w_i concentrates on the target object and blue pixels (with less importance) appear in the background region. With the learned subspace, a feature vector must be decomposed, where the coefficients are values of the learned latent variables. The discriminative strength of the latent variables is shown in decreasing order.

C. Particle Filtering

The main issue for any adaptive appearance model is that it is using noisy or misaligned observations for update and thereby causing tracking drift gradually. For online tracking, the only ground truth is the labelled target object in the first frame. Remaining other samples are obtained online is mostly different from the ground truth data. To reduce tracking drift, we need to present a two-stage particle filtering method for state prediction [1]. In this method, the appearance model $A_1 = \{x_{1p}, x_1, W_1\}$ initialized in the first frame is used to construct a static likelihood function using $p_s(x \mid s) \propto \exp(-d_s)$. The adaptive appearance model set A is used to construct another likelihood function using $p_s(x \mid s) \propto \exp(-d_s)$ where $d_a = \min\{d_{aI} \mid i = 1, \ldots, k\}$ and d_{ai} is the distance from **x** to the i -th appearance model. The adaptive likelihood function $p(x_t \mid s_t)$ is computed based on new adaptive appearance model. With distance metric is the likelihood function computed by

$$P(x_t / s_t)\ \alpha\ \exp(-d_t) \tag{9}$$

Where dt is the distance between the test sample **x**t and the learned appearance model set A at time t. The likelihood function adapts over time as a result of the proposed appearance model with online update.

At each frame, we need to estimate an initial tracking result using a particle filter with the adaptive likelihood function $p_s(x \mid s)$. With the initial estimate, we need to use another particle filter with the static likelihood function $p(x \mid s)$ to determine the final predicted state in the second stage. The filter with adaptive likelihood function can avoid the local minimum problem since the appearance change between two consecutive frames is not expected to be too large. The filter with static likelihood function can alleviate the drift problem since it requires the final tracking result to be as similar as the only ground truth value obtained in the first frame. Similar strategy has been successfully demonstrated to reduce drift.

III. THREE-STAGE TRACKING METHOD

Based on the adaptive appearance model, object tracking is done by using this algorithm. To reduce tracking drift, we need to present a two-stage particle filtering method to estimate the tracking result using both the initial and adaptive appearance models.

Algorithm 1 Adaptive Appearance Model

I. Initialize model with PLS analysis when t = 1(first frame).
II. for t = 2 to T (rest of frames)

III. Find new value to old value in appearance model which have the smallest (d_s) and largest d_l distances to the target sample x_s, respectively.
IV. if d_s < Threshold
V. Keep the same target value
VI. Update x_s using PLS analysis else

Learn a new value to appearance model by d_l, and replace it

Algorithm 2 Two-stage particle filtering

I. Input: Image frames $F_1, \ldots, F_T$.
II. Output: Tracking results s_t at time t.
III. for t = 1, . . . , T do
IV. if t = 1 then
V. Label the target manually or using a detector. Collect positive as well as negative samples, and compute the static appearance model $A_1 = \{x_{1p}, x_1, W\}$.
VI. else
VII. Stage 1. Perform particle filtering to estimate an initial result s_t using the previous tracking result s_{t-1} and the adaptive likelihood function $p(x_t \mid s_t)$.
VIII. Stage 2. Perform particle filtering to determine the final tracking result s_t with the initial tracking result st and the static likelihood function $p(x_t \mid s_t)$.
IX. Output the tracking result s_t .
X. Update the adaptive appearance model set A with s_t using Algorithm 1.
XI. end if
XII. end for

Algorithm 3 Background subtraction

I. Find median for current frame using same formula

$$(D) = \text{median}\,(I) \qquad (C) = \text{median}\,(B) \tag{10}$$

II. Compare the median value of both background and current frame value

$$N = \begin{cases} 1 \text{ if } [I(I, j) - B(i, j)] < 0 \\ 0 \text{ if } [I(I, j) - B(i, j)] > 0 \end{cases} \tag{11}$$

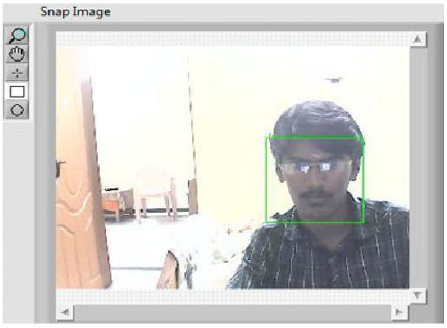
a. Where N is the value which is replaced to the value of pixel in the current frame value which is converted into Binary image.
b. If background median value is greater
c. do
d. Compare each pixel value of image with the threshold value "true" and get the output
e. Else
f. Compare each pixel value of image with the threshold value "false" and get the output

III. Get the location of target image and draw a rectangle on the target location in output.

IV. RESULTS

The project was done in LABVIEW 2009 version by using a i-ball webcam C 8.0.the captured image has resolution 0f 640×480/33fps.

(a)



(b)

(c) (d)

Fig . 2. (a) captured frame 2(b) target object is manually initialized (c) template of the target region 2(d) tracked region is highlighted

During program execution the various appearance of the object to be tracked is loaded into the program and then 'algorithm 1' will proceed to monitor the various appearance of the targeted object. At the starting period of program execution webcam capture the video of targeted object. This video has been displayed on the first panel location in LABVIEW as like as in fig 2a. Then the first frame of the video has been snapped and displayed in the front panel part in LABVIEW.

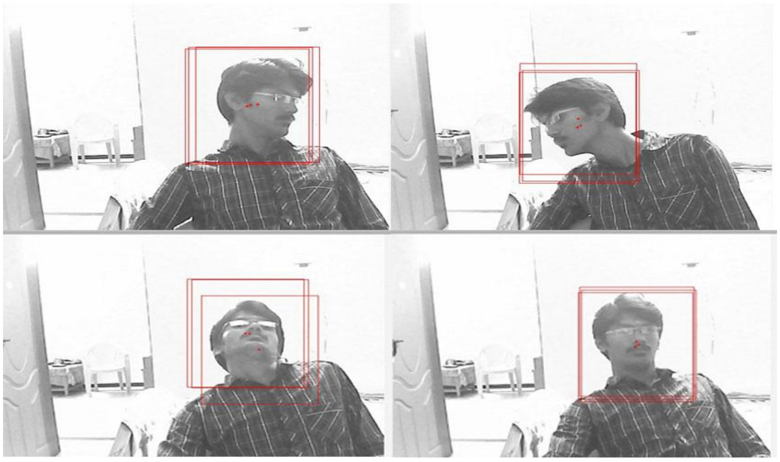


Fig. 3. Tracked results for object when it is far from camera

At the first step we have to manually initialize the target object in the first frame which is illustrated in fig 2c. Initially each frame has been divided into various regions to make accurate result. Then the marked targets object has taken and displayed in fig 2d. After target object initialization 'algorithm 2' has been proceed to filter the unwanted objects appeared in the video. Here the filter action has made with respect to the parameter contrast value and score value in region. Here I have taken contrast value as 10 and score value as 400. The region will discard or filter out if any one of these two parameter value become less than fixed level. If both values are greater than fixed level then we proceeds 'algorithm 3' and checks whether the object is present or not. If there is target object present in the region then it will be highlighted by red box as like as in fig 2d.

Fig. 4. Result of object tracking when it is close to camera

With the above actions we have accurately tracked the targeted object. The object tracking will not affected by the different kind of object appearance. Here fig 3 and 4 shows the successive object tracking with different kind of target object appearances.

V. CONCLUSION

Conventional algorithms uses linearly classified object instead of non-linear classified object. Three stage tracking is performed to extract the target object in the current frame. The third stage is by using Background subtraction technique by using Median value. By using multiple stage of tracking, tracking drift will be overcome. For face tracking we have used the score value as 400 and contrast 10. This algorithm can be used for different application but the appearance model and the parameter value differs. For the multiple object tracking implementation of the parallel processing concept to track the object effectively. But the parallel processing will significantly increase the computational cost.
In the real time application such as traffic monitoring, control applications using object tracking can be enhanced by this method. Though complexity slightly increased it provides better result than previous method

REFERENCES

[1] Pushe Zhao, Hongbo Zhu, He Li, and Tadashi Shibata, "A Directional-Edge-Based Real-Time Object Tracking System Employing Multiple Candidate-Location Generation", *IEEE Trans.* Circuits And Systems For Video Technology, *Intell.,* vol. 23, no. 3, pp. 503-517, March 2013.

[2] Sayed Hossein Khatoonabadi and Ivan V. Baji´c, "What Are We Tracking: A Unified Approach of racking and Recognition," *IEEE Trans.* Image Processing, *Intell.,* vol. 22, no. 2, pp. 549-560, February 2013.

[3] Sayed Hossein Khatoonabadi and Ivan V. "Video Object Tracking in the Compressed Domain Using Spatio-Temporal Markov Random Fields." *IEEE Trans.* Image Processing, *Intell.,* vol. 22, no. 1, pp. 300-313, January 2013.

[4] Qing Wang and Feng Chen, "Object Tracking via Partial Least Squares Analysis," *IEEE Trans*. Image Processing, *Intell.,* vol. 21, no. 10, pp. 4454-4465 October 2012.

[5] Alper YilmazOhio, "Object Tracking: A Survey," in Proc .ACM Computing Surveys, *Conf.*, 2010, pp.310-354

[6] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust visual tracking based on incremental tensor subspace learning," in *Proc. IEEE Comput Vis. Pattern Recognit. Conf.*, Rio de Janeiro, Brazil, , pp. 1–8 October 2007.

[7] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc.IEEE Comput. Vis. Pattern Recognit. Conf.*, June. 2006, pp. 260–267.

[8] R. Rosipal and N. Kramer, "Overview and recent advances in partial least squares," in *Latent Structures Feature Selection*. New York: Springer-Verlag, 2006, pp. 34–51

[9] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005

[10] Xi Li, "A Survey of Appearance Models in Visual Object Tracking," in Proc. ACM Intelligent Systems and Technology, *Conf.*, 2004, pp. 801–808.

[11] Nguyen and D. Rocke, "Tumor classification by partial least squares using microarray gene expression data," *Bioinformatics*, vol. 18, no. 1, pp. 39–50, 2002

[12] H. Wold, "Partial least squares," in *Encyclopedia of Statistical Science*, vol. 6, S. Kotz and N. L. Johnson, Eds. New York: Wiley, 1985, pp 581–591.

[13] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

