Online Improved Eigen Tracking

Subarna Tripathi subarna.tripathi@gmail.com
Santanu Chaudhury schaudhury@gmail.com
Sumantra Dutta Roy sumantra@cse.iitd.ac.in
Electrical Engineering Department, IIT Delhi

Abstract

We present a novel predictive statistical framework to improve the performance of an Eigen Tracker which uses fast and efficient eigen space updates to learn new views of the object being tracked on the fly using candid co-variance free incremental PCA. The proposed system detects and tracks an object in the scene by learning the appearance model of the object online motivated by non-traditional uniform norm. It speeds up the tracker many fold by avoiding non-linear optimization generally used in the literature.

1. Introduction

There are numerous tracking algorithms proposed in the literature like mean-shift or camshift algorithms, appearance based tracker etc. An appearance-based tracker (*EigenTracker* [1]) can track moving objects undergoing appearance changes powered by dimensionality reduction techniques. The Isard and Blake CONDENSATION algorithm [2] can represent simultaneous multiple hypothesis. There can be several ways by virtue of which the power of EigenTracker and particle filter can be combined like [7] and [8]. But these have the overhead of non-linear optimization. [6] proposes a fast appearance tracker which eliminates non-linear optimizations completely but it lacks the benefit of predictive framework. We enhance the capabilities of the EigenTracker by augmenting it with a CONDENSATION-based predictive framework to increase its efficiency and also make it fast by avoiding non-linear optimization like [6]. The main features of our approach are the tracker initialization, presence of prediction framework, effective subspace update algorithm [4] and avoidance of non-linear optimizations.

2. On-Line Prediction in the Tracker

2.1. The Prediction Mechanism

The tracking area is described by a rectangular window parameterized by $[x_t, y_t, w_t, h_t, \theta_t]$, and modeled by the 7 dimensional state vector $X_t = [x_t, x'_t, y_t, y'_t, w_t, h_t, \theta_t]$, where (x_t, y_t) represents the position of the tracking window, (w_t, h_t) represents the width and height of the tracking window, (x'_t, y'_t) represents the horizontal and vertical component of the velocity and θ_t represents the 2*D* rotation angle of the tracking window. These 5 motion parameters can track the object with its bounding box being an oriented rectangle. This seed point is needed for sampling windows around it. The predictive framework helps generating better seed values for diverse object dynamics. We use a simple first-order AR process to represent the state dynamics (t represents time):

$X_t = A_t X_{t-1} + w_t$, where w_t is a zero-mean, white, Gaussian random vector. The measurement is the set of five motion parameters obtained from the image, Z_t. The observation model has Gaussian peaks around each observation, and constant density otherwise.

We estimate the values of the five motion parameters based on their predicted values and the measurements done. These estimated values serve as seeds to the next frame. For every frame, we get sampled version of conditional state density (S_t), and corresponding weights ($\prod_t$) for conditional probability propagation or CONDENSATION. The state estimate is used to generate the predictions for the next frame. The prediction framework we used is motivated by predictive Eigen tracker [7].

2.2. Initialization of the tracker

Accurate tracker initialization is a difficult problem. Our coding solution currently can detect the most moving object automatically by analyzing the first three frames, i.e. with the overhead of additional two frames buffering at the beginning of the tracking process which is quite acceptable. We have used a moving object segmentation method based on the improved PCA which is a simplified version of the methodology used in [3] for moving object detection and segmentation. For this technique to work the background should be still or changing slowly such as grassplot or cloud for the analyzing frames. The principle component analysis is improved to adapt to the motion detection. The definition of traditional covariance matrix is modified to:

$$C = (X1 - X2)^T(X1 - X2) + (X2 - X3)^T(X2 - X3) + (X1 - X3)^T(X1 - X3) \quad (1)$$

Where, Xi is a one dimensional vector obtained by vectorizing the original image sequence. Secondly, the calculation result is improved in the following way. Say, E1 and E2 as the first two eigenvectors calculated. The element wise product of these two eigenvectors is:

E = E1 × E2. E effectively eliminates the blur of the eigen images of the moving object. And after formation of E, a simple thresholding usually gives a good initialization of the object's rectangular bounding box.

2.3. On-the-fly Eigen space Updates

In most tracking problems, the object of interest undergoes changes in appearance over time. It is not feasible to learn all possible poses and shapes even for a particular domain of application, off-line. Therefore, one needs to learn and update the relevant Eigen spaces on the fly. Since a naive $O(mN^3)$ algorithm (for N images having m pixels each) is time-consuming, we use an efficient-estimation motivated by optimal incremental principal component analysis of O(mNk) algorithm (for k most significant singular values) proposed by Juyang Weng *et al.* [4].

At each time frame F_{i+1}, the IPCA method iteratively computes the new principal components $vj(i+1)$ (for $j = 1, 2, ...d$), as follows:

1. $u1(i + 1) = Oi+1$.
2. For $j = 1, 2, ..., min(d, i + 1)$ do,

(a) If $j = i + 1$,
initialize the jth eigenvector as $vj(i + 1) = uj(i + 1)$;
(b) Otherwise,

$$vj(i+1) = \frac{i-1}{t} vj(i) + \frac{1+1}{i+1} uj(i+1)\, uj'(i+1) \frac{vj(i)}{\| vj(i) \|} \quad (2)$$

$$uj+1(i+1) = uj(i+1) - u'j(i+1) \frac{vj(i+1)}{\| vj(i+1) \|} \frac{vj(i+1)}{\| vj(i+1) \|} \quad (3)$$

where l is the amnesic parameter giving larger weights to newer samples, and $||v||$ is the eigenvalue of v. Intuitively, eigenvectors $v_j(i)$ are pulled towards the data $u_j(i+1)$, for the current eigenvector estimate $v_j(i + 1)$ in eq (3). Since the eigenvectors have to be orthogonal, therefore eq (4) shifts the data $u_{j+1}(i+1)$ normal to the estimated eigenvector $vj(i+1)$. This data $u_{j+1}(i + 1)$ is used for the estimating the $(j+1)$ th eigenvector $v_{j+1}(i + 1)$. The IPCA method converges to the true eigenvectors in fewer computations than PCA (proof in [5]).

Since the real mean of the image data is unknown, we incrementally estimate the sample mean m'(n) by

$$m'(n) = \frac{n-1}{n} m'(n-1) + \frac{1}{n} x(n) \quad (4)$$

Where x(n) is the nth sample image. The data entering the IPCA algorithms are the scatter vectors,
u(n) = x(n) – m'(n) for n=1,2,…

2.4. The Overall Tracking Scheme

The following section outlines our overall tracking scheme. In the first frame, we initialize the tracker (Section 2.2). For all subsequent frames, the next step is to obtain the measurements – taking the minimum distant prediction from the learnt sub-space (in RGB plane) as the description of the tracked object. We then update the eigen-spaces incrementally. Finally, we predict the motion parameters values for the next frame. The idea behind the subspace construction for the appearance based tracking is the uniform $L2$ reconstruction error norm

$$Error^{\infty}(L, \{x1, \cdots xN\}) = \max_i d^2(L, xi) \quad (5)$$

To define the quality of approximation, we use the uniform reconstruction error norm $Error^{\infty}$ introduced in Equation 5 in our approach. If N denotes the number of previous frames whose tracking results are retained and $\delta > 0$ is a threshold parameter, we can specify a pair of input parameters (N, δ). We can define the subspace L to be *any* subspace such that the uniform reconstruction error norm between L and $\{x1, \cdots, xN\}$ is less than the threshold δ. *i.e.*

$$Error^{\infty}(L, \{x1, \cdots xN\}) < \delta. \quad (6)$$

This definition of L is general and the solution is generally not unique. As along as δ is greater than zero, there exists at least one L that satisfies the inequality in Equation 6, the subspace L spanned by the entire collection of samples $\{x1, \cdots, xN\}$. One of the great advantages of this non-uniqueness of the solution is that we only need to find one such L, and it allows us to design a simple and computationally inexpensive algorithm to find just one such L. Having a computationally inexpensive update algorithm is necessary if the tracking algorithm is expected to run in real-time.

4. Remark and Discussions

The computational complexity of the algorithm is dominated by the number of windows generated from the sampling. Like all appearance-based tracker it cannot handle situation like sudden pose or illumination changes or fully occlusion, but it can handle partial occlusion and gradual pose or illumination changes (Figures 1, 2, 3). There are three important free parameters in our algorithm, *N*, the number of samples to pick and l, amnesic parameter for the subspace update and k, the number of principal components. In the experiments we reported below, we let *l* range from 2 to 6 and *N* range from 150 to 200 and k range from 3 to 10.

6. Experiments and Results

We implemented the proposed method in MATLAB 7. Our current implementation runs at about 0.25 to 0.5 frames/sec with 320x240 and 176x144 video input respectively on a standard Intel centrino P4 1.8 MHz machine and thus it is quite expected that C implementation easily can run on real time. Our test cases contain scenarios which a real-world tracker encounters, including changes in appearance, large pose variations, significant lighting variation and shadowing, partial occlusion, object partly leaving field of view, large scale changes, cluttered backgrounds, and quick motion resulting in motion blur.

video	Frames tracked		Avg Time/frame	
	No prediction	With prediction	No prediction	With prediction
Coast guard	80	100	4.2 sec	4.2 sec
hall	82	112	4.5 sec	4.6 sec

Table 1: comparison of predictive and non-predictive framework (N = 150 windows sampled for each case)

It is evident from the above table that incorporation of predictive framework makes the tracker more robust. Coastguard sequence has presence of the boat up to frames 100 out of total 300 frames and then it disappears (figure 1). Hall is the sequence where a person (tracking object) appears in frame 25 and disappears after 140th frame, and in that interval it changes poses heavily. If we increase the number of windows to be sampled by 250, no prediction framework (with almost double time complexity) shows almost similar robustness that of predictive framework with 150 samples.

7. Summary and conclusions

In this paper, we have introduced a technique for predictively learning the statistical distribution on-line with an Eigen subspace representation of an object that is being tracked with a fast EigenSpace update technique. The resulting tracker is both simple and fast. The method can robustly track an object in the presence of large viewpoint changes, partial occlusion, lighting variation, changes to the shape of the object shaky cameras, and motion blur. Moreover avoidance of non-linear optimization makes our tracking task faster than that of [7].

8. References

[1] M. J. Black and A. D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation", *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63 - 84, 1998.

[2] M. Isard and A. Blake, "CONDENSATION – Conditional Density Propagation For Visual Tracking", *International Journal of Computer Vision*, vol. 28, no. 1, pp. 5 - 28, 1998.

[3] Chun-Ming Li, Yu-Shan Li, Qing-De Zhuang, Qiu-Ming Li, Rui-Hong Wu, Yang Li . "Moving Object Segmentation and Tracking In Video", *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou,* 18-21 August 2005, pp. 4957-4960

[4] J. Weng, Y. Zhang, and W. Hwang,. "Candid covariance-free incremental principal component analysis" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.25(8), pp.1034-1040, 2003.

[5] Y. Zhang and J. Weng, "Convergence Analysis of Complementary Candid Incremental Principal Component Analysis," *Technical Report MSU-CSE- 01-23, Dept. of Computer Science and Eng., Michigan State Univ., East Lansing,* Aug. 2001.

[6] Jeffrey Ho, Kuang-Chih Lee, Ming-Hsuan Yang, David Kriegman, "Visual Tracking Using Learned Linear Subspaces", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04),* Vol 1 pp. 782-789

[7] Namita Gupta, Pooja Mittal, Kaustubh S. Patwardhan, Sumantra Dutta Roy, Santanu Chaudhury and Subhashis Banerjee, "On Line Predictive Appearance-Based Tracking". *Proc.IEEE Int'l Conf. on Image Processing (ICIP 2004),* pp 1041 - 1044

[8] Kaustubh Srikrishna Patwardhan, Sumantra Dutta Roy, "Hand gesture modelling and recognition involving changing shapes and trajectories, using a Predictive EigenTracker", *Pattern Recognition Letters,* vol. 28, no. 3, pp. 329 - 334, February 2007

Frame 1 Frame 21 Frame 35

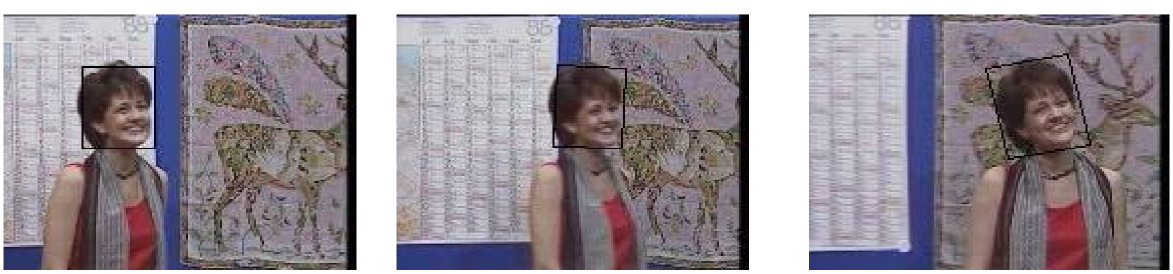
Frame 67 Frame 86 Frame 108

Figure 1: Sequence of tracking a boat (sequence coastguard) which shows high background motion, background clutter as well as object partly going out of the field of view

Frame 1 Frame 210 Frame 237

Frame 261 Frame 264 Frame 271

Figure 2 Sequence of tracking a helicopter in a changing background and which goes under partial occlusion



Frame 1 Frame 25 Frame 84

Figure 3: Sequence of tracking a woman's face (sequence Renata) which shows apparent pose changes