Accepted in 5th Int. Conference on Imaging for Crime Detection and Prevention (ICDP), 2013

A Comparison of Adaptive Appearance Methods for Tracking Faces in Video Surveillance

M. Ali Akber Dewan*, E. Granger*, F. Roli†, R. Sabourin*, and G. L. Marcialis†

*Laboratoire d'imagerie, de vision et d'intelligence artificielle, École de technologie supérieure, Université du Québec, Montréal, Canada
dewan@livia.etsmtl.ca, eric.granger@etsmtl.ca, robert.sabourin@etsmtl.ca

†Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, Cagliari, Italy
roli@diee.unica.it, marcialis@diee.unica.it

Keywords: Biometrics, Face Tracking, Spatiotemporal Face Recognition, Video Surveillance, On-Line and Incremental Learning, Adaptive Appearance Methods.

Abstract

Face recognition is increasingly employed by public safety organizations in decision support systems for video surveillance, to detect the presence of individuals of interest. In the context of spatiotemporal face recognition, tracking is an important function used to locate, follow and regroup faces of different individuals in a scene. Techniques for face tracking in video surveillance should be robust to changes in pose, expression and illumination, as well as occlusion in cluttered scenes. Given these challenges, trackers based on adaptive appearance modelling (AAM) typically improve target's state estimation because they initiate and update an internal face model per individual according to changes in facial appearance. In this paper, the performance of three AAM trackers – Incremental Visual Tracking (IVT), Tracking Learning Detection (TLD) and Discriminative Sparse Coding based Tracking (DSCT) – are compared for face tracking with video surveillance applications in mind. These methods are evaluated according to area overlap error, tracking error and time complexity using Chokepoint videos collected in uncontrolled video-surveillance environments, where individuals walk through portals. Results indicate that IVT outperforms the others in its ability to accurately track faces in the presence of occlusion, and under variations in pose, scale and lighting. Further characterization of IVT indicates that using a small batch size and forgetting factor during update provide better tracking accuracy when face tracks changes in their capture conditions. When conditions change more gradually, IVT benefits from assessing facial quality before updating face models.

1 Introduction

Given the current demand for security and surveillance technologies, decision support systems for video surveillance are being considered by many public safety organizations for enhanced situation analysis. In many applications, automated face recognition is increasingly employed to alert a human operator as to the presence of individuals of interest appearing in either live (real-time monitoring) or archived (post-event analysis) videos. Face recognition is relevant in a range of video surveillance applications still-to-video face recognition (e.g., watch-list screening) and video-to-video face recognition (e.g., person re-identification). In practice, face recognition in video surveillance (FRiVS) is challenging because accurate responses are required for faces captured under semi-constrained (e.g., inspection lane, portal and checkpoint entry) and unconstrained (e.g., cluttered free-flow scene at an airport or casino) conditions.

In the recent years, face tracking has become an important tool for recognition. It is used to locate and follow faces of different individuals in motion, and regroup the facial information for spatiotemporal face recognition. This information is useful to generate reliable facial models; mitigate effects of non-cooperative capture conditions; model facial behaviour; generate better facial models from multiple views; and accumulate decision on multiple frames to achieve improved recognition [1]. However, variations in pose, scale, expression, and illumination, and the occlusions in cluttered scenes can degrade tracking performance. To address these challenges, many robust face tracking methods have been proposed, although only partial solutions to these key issues exist. Beyond the need for reliable face tracking, efficient techniques are required for various real-time applications because video surveillance networks are comprised of a growing number of cameras, and potentially cluttered scenes.

Face tracking involves (1) facial model (FM) representation, (2) prediction filtering (PF), and (3) data association (DA). In FM, facial captures are represented with distinctive features in order to be located and tracked from frame to frame, whereas PF allows predicting the state (size and location) of a face in the current frame based on information from previous frames and on some underlying state transition model. Finally, DA allows to link facial captures to predicted face locations in order to actually locate different faces of a frame.

Among state-of-the-art trackers, adaptive appearance modelling (AAM) methods have been shown to efficiently address current challenges of face tracking since they adapt internal face models for enhanced PF as well as DA [2]. These methods employ some off-line and on-line learning algorithms to adapt changes in facial appearance in a scene. However, AAM methods are computationally expensive and seek to fulfil the contradicting goals of rapid learning and stable memory for the AAMs, which is often referred to as the stability-plasticity dilemma. Finally, AAMs carry the risk of adapting to inputs from other targets or the background,

leading tracks to drift from the target. This is referred to as knowledge corruption. These key issues must be considered when selecting an AAM method that is suitable for FRiVS.

This paper presents an empirical comparison of performance for trackers based on AAM. State-of-the-art methods are reviewed, and an algorithmic description is presented for three representative AMM trackers – Incremental Visual Tracking (IVT) [3], Tracking-Learning-Detection (TLD) [4], and Discriminative Sparse Coding–Based Tracking (DSCT) [5] highlighting their pros and cons for FRiVS. Among all AAM–based trackers, IVT, TLD and DSCT are considered as the baseline trackers representing three subgroups of the AAM taxonomy. These trackers are compared according to tracking quality and computation time using the Chokepoint video dataset for unconstrained video surveillance applications. Since IVT outperforms the other trackers, this paper also characterizes the impact of its key parameters (i.e., batch size and forgetting factor) on IVT tracking quality. In addition, this paper explores the benefits of updating IVT's internal FM using quality assessment provided by a core module in all systems for FRiVS – the face detection module.

2 Adaptive Appearance Model-Based Trackers

Trackers based on AAM use an internal FM of different individuals, and can adapt these models to changes in the scene. The main benefit of using AAM for FRiVS applications is to improve accuracy for DA in changing face-capture condition. According to DA mechanisms, AAM trackers in literature are categorized into two main groups: generative and discriminative. AAM trackers with generative DA typically use 1-class classification (or density estimation) to model only target face samples, whereas the discriminative typically use 2-class classification to model faces in a scene with target and non-target samples. Typically, generative modelling requires more (target) samples to attain a high level of performance.

Generative methods learn the appearance of faces and track them by searching for the region most similar to the target face appearances in each frame. They do not exploit any background information for DA. In this category, Eigen tracker [6] learns a low dimensional subspace offline for FM of each target at some fixed views, whereas the sum of squared difference is used for DA. Wandering-Lost-Stable Model–based Tracker [7] uses a Gaussian Mixture Model (estimated with an online Expectation Maximization algorithm) for FM in order to handle appearance variations during tracking. Euclidean distance along with a robust estimator is used for DA. Methods in [6] and [7] do not involve PF. IVT [3] presents grey-scale intensities of target faces in a low dimensional subspace for FM which is updated adaptively using the faces tracked in the previous frames. Particle filter is used for PF, and Euclidean and Mahalanobis distances are used for DA. Riemannian Manifold tracker [8] presents an online subspace learning algorithm for face representation, which is based on a covariance matrix descriptor. In this method, particle filtering is used for PF and Log Euclidean-Riemannian metric is used for DA.

With discriminative methods, trackers localize the target using a classifier that learns a decision boundary between the appearance of target and that of background and non-target samples. In this category, Support Vector Machine–based Tracker [9] trains SVMs off-line for DA, and uses histogram of oriented gradient and the RGB colour components for FM representation. In addition, Ensemble tracker [10] presents an online boosting algorithm to classify pixels belonging to foreground and background. Multiple-Instance-Learning tracker [11] performs multiple instance learning to handle ambiguously labelled target and non-target samples from the scene to reduce visual drift caused by classifier update. Haar-like features are used for facial representation and AdaBoost algorithm is used for DA. DSCT [5] uses target and non-target samples to generate sparse codes for face representation and compares them using adaptive and a static observation model to achieve robust DA. All the above methods use particle filtering for PF. Finally, TLD [4] performs tracking by refining the results obtained from a tracker and a detector. Target and non-target image patches are used as FM. A learning component exploits p-expert and n-expert to select target and non-target patches during to update FM and detector online. A classifier based on random forest is used for DA.

Considering the challenges of real world video surveillance applications, both generative and discriminative trackers face several issues. For generative trackers, numerous scaled and aligned facial samples from consecutive frames are required in order to learn a FM online; otherwise, the drift problem is likely to occur if the appearance of an individual's face changes significantly in the scene. Since these trackers do not use background and non-target information, the FMs that they generate are less discriminant and may cause interchange of the tracks in multi-face tracking. Discriminative trackers outperform in such conditions if sufficient reference samples are available for design. However, having enough data of the target, and the background and non-target information in advance is also not realistic. Both generative and discriminative trackers suffer from tracking drift problem when FMs are updated with noisy and potentially misaligned samples through online learning.

In this paper, IVT [3], TLD [4], and DSCT [5] are selected for further analysis. IVT and TLD are selected as baseline AAM trackers in the generative and discriminative categories. Also from the discriminative category, DSCT is selected as a representative of a broad subgroup of tracking techniques based on sparse coding.

2.1 Incremental Visual Tracking:

The IVT method (see Algorithm 1) incrementally updates FMs in a low dimensional sub-space, and adapts changes in capture condition during tracking. Incremental update is performed using Sequential Karhunen–Loeve (SKL) [12]. Once a new face is initially detected in the scene, IVT tracks the face in the first n frames using template matching, and then defines a data block $\mathbf{A} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, where components are the vectors representing the tracked face regions with states $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$. The initial FM of the target is represented

as a eigenspace **U**, computed from the singular value decomposition (SVD) of the centered data matrix of **A**. To find face correspondence, particle filter–based affine motion parameters are used in a Bayesian framework, where Euclidean and Mahalanobis distances are used for DA. When a new data block $\mathbf{B} = \{\mathbf{p}_{n+1}, \ldots, \mathbf{p}_{n+m}\}$ becomes available after tracking for m more frames, the updated appearance $\mathbf{U}'$ of the FM is obtained by using the augmented data matrix $[\mathbf{A}\ \mathbf{B}]$, where the update exploits computationally efficient SKL algorithm. Two key parameters – forgetting factor, f, and batch size, m, determine the plasticity of FM. The parameter m defines the number of observations whereas f the amount of contribution from older observations to be considered in updating the face model for tracking.

Algorithm 1: IVT for a single face in a scene.

Input: Frames $\{\mathbf{I}_1,...,\mathbf{I}_\infty\}$. The target face is labelled in the first frame $\mathbf{I}_1$ using a face detection.

Output: States $\{\mathbf{X}_1,...,\mathbf{X}_\infty\}$ of the target face in the frames.

for $t = 2, \ldots, \infty$ **do**

  if $t \le n$ **then**

    –Perform template based tracking and define the state $\mathbf{X}_t$.

    –Construct data block $\mathbf{A}$ with the tracked facial region $\mathbf{p}_t$

$$\mathbf{A} \leftarrow \mathbf{A} + \mathbf{p}_t$$

  end if

  Compute $\mathbf{M}_C = (\mathbf{A} - \bar{\mathbf{p}}_A) = 1/(n-1)\sum_{i=1}^{n}(\mathbf{p}_i - \bar{\mathbf{p}}_A)(\mathbf{p}_i - \bar{\mathbf{p}}_A)^T$

  where $\bar{\mathbf{p}}_A = (1/n)\sum_{i=1}^{n}\mathbf{p}_i$.

  Compute $\mathbf{M}_C \xrightarrow{SVD} \mathbf{U}\Sigma\mathbf{V}^T$, and use $\mathbf{U}$ as the initial FM.

  if $t > n$ **then**

    –Predict states $\hat{\mathbf{X}}_t$ at $\mathbf{I}_t$ by modelling the affine parameters $(x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t)$ with Gaussian distribution:

$$p(\hat{\mathbf{X}}_t \mid \mathbf{X}_{t-1}) = N(\hat{\mathbf{X}}_t, \mathbf{X}_{t-1}, \psi)\ ;\ \psi = \{\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2\}$$

    –Given a facial region $\mathbf{p}_t$ (defined by $\hat{\mathbf{X}}_t$), compute the probability of $\mathbf{p}_t$ being generated from $\mathbf{U}$ and centered at $\boldsymbol{\mu}$:

$$P(\mathbf{p}_t \mid \mathbf{X}_t) = N(\mathbf{p}_t; \boldsymbol{\mu}, \mathbf{U}\mathbf{U}^T + \varepsilon\mathbf{I})\, N(\mathbf{p}_t; \boldsymbol{\mu}, \mathbf{U}\Sigma^{-2}\mathbf{U}^T)$$

    –Estimated $\mathbf{X}_t$ as a hidden state using Bayes' theorem, given that a set of observed image patches $\Gamma_t = \{\mathbf{p}_1,...,\mathbf{p}_t\}$:

$$P(\mathbf{X}_t \mid \Gamma_t) \propto P(\mathbf{p}_t \mid \mathbf{X}_t) \int P(\mathbf{X}_t \mid \mathbf{X}_{t-1}) P(\mathbf{X}_{t-1} \mid \Gamma_{t-1})\, d\mathbf{X}_{t-1}$$

    if $P(\mathbf{X}_t \mid \Gamma_t) < \delta$ **then** // δ is a predefined threshold

      –Drop the track and exit.

    else

      –Update data block $\mathbf{B}$ with $\mathbf{B} \rightarrow \mathbf{B} + \mathbf{p}_t$

      if size of $\mathbf{B}$ becomes equal to m **then** update $\mathbf{U}$:

$$\bar{\mathbf{p}}_B = 1/m\sum_{i=n+1}^{n+m}\mathbf{p}_i\ ,\ \bar{\mathbf{p}}_c = \frac{fn}{fn+m}\bar{\mathbf{p}}_A + \frac{m}{fn+m}\bar{\mathbf{p}}_B$$

$$\hat{\mathbf{B}} = \left[(\mathbf{p}_{n+1} - \bar{\mathbf{p}}_B) \ldots (\mathbf{p}_{n+m} - \bar{\mathbf{p}}_B)\ \sqrt{\{nm/(n+m)\}}(\bar{\mathbf{p}}_B - \bar{\mathbf{p}}_A)\right]$$

$$\tilde{\mathbf{B}} = \text{orth}\left(\hat{\mathbf{B}} - \mathbf{U}\mathbf{U}^T\hat{\mathbf{B}}\right) \text{ and } \mathbf{R} = \begin{bmatrix} f\Sigma & \mathbf{U}^T\hat{\mathbf{B}} \\ \mathbf{0} & \tilde{\mathbf{B}}\left(\hat{\mathbf{B}} - \mathbf{U}\mathbf{U}^T\hat{\mathbf{B}}\right) \end{bmatrix}$$

      SVD of $\mathbf{R}: \mathbf{R} \xrightarrow{SVD} \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^T$

      $\mathbf{U}' = \left[\mathbf{U}\ \ \tilde{\mathbf{B}}\right]\tilde{\mathbf{U}}$ and $\Sigma' = \tilde{\Sigma}$

    end if

  end if

  end if

end for

In IVT, the face representation based on eigenspaces is robust to illumination and pose changes and clutter. Moreover, the on-line learning incrementally updates FMs according to the changes in the scene. However, IVT is susceptible to drift as it can gradually adapt to non-targets regions since this FMs representation is built solely on target samples from previous tracked frames. Moreover, IVT lacks mechanisms for detecting and correcting drift as it does not incorporate global constraints on the overall appearance of the target FM. Such constraints can be learned from a set of generic (non specific) well-cropped and well aligned face images that span possible variations in pose, illumination and expressions [13]. However, it requires off-line training, and performance would rely heavily on the training datasets.

2.2 Tracking-Learning-Detection:

With the TLD method (see Algorithm 2), FM is represented with a collection of target and non-target patches observed, $\mathbf{M} = \{\mathbf{p}_1^+, \mathbf{p}_2^+, ..., \mathbf{p}_m^+; \mathbf{p}_1^-, \mathbf{p}_2^-, ..., \mathbf{p}_n^-\}$. Two similarity measures – the relative similarity $S^r = S^+ / (S^+ + S^-)$ and conservative similarity $S^c = S_{50\%}^+ / (S_{50\%}^+ + S^+)$ – are employed throughout TLD to measure the similarity of an arbitrary patch $\mathbf{p}$ to the appearance in the FM, **M,** where S, S^+, S^-, and $S_{50\%}^+$ are defined as follows:

$$S(\mathbf{p}_i, \mathbf{p}_j) = 0.5\left(NCC(\mathbf{p}_i, \mathbf{p}_j) + 1\right) \quad (1)$$

$$S^+(\mathbf{p}_i, \mathbf{M}) = \max\left\{S(\mathbf{p}_i, \mathbf{p}_j^+) : \mathbf{p}_j^+ \in \mathbf{M}\right\} \quad (2)$$

$$S^-(\mathbf{p}_i, \mathbf{M}) = \max\left\{S(\mathbf{p}_i, \mathbf{p}_j^-) : \mathbf{p}_j^- \in \mathbf{M}\right\} \quad (3)$$

$$S_{50\%}^+(p, \mathbf{M}) = \max\left\{S(p, p_i^+) : p_i^+ \in M \wedge i \le m/2\right\} \quad (4)$$

Here, *NCC* refers to the normalized cross-correlation of the image patches.

The TLD framework consists of three main components: *tracking*, *detection*, and *learning*. The *tracking* component uses median-flow tracker [14] to find the face correspondence in frames. The *detection* component employs a three layer cascaded classifier, where at each layer a number of candidate patches are refined. Finally, *detection* selects the patch most similar to the target FM, **M,** using a nearest neighbour classification, where the similarity is measured using S^r. Among the two regions obtained from *detection* and *tracking*, the maximally confident region is selected as the tracking target, where the confidence is determined using conservative similarity S^c. The *learning* component employs p-*expert* and n-*expert* to select reliable target and non-target patches to update **M**, where the reliability is determined using S^c.

Since the TLD framework uses a threshold to update FM, this method tracks faces as long as the appearance does not differ considerably from the appearance observed so far in previous frames. Moreover, by using a 2-class classifier, this method learns the appearance of the target with respect to non-target samples, and thus can automatically determine the presence or disappearance of the faces in the scene. In low-clutter scenes, the vulnerability of TLD to drift is lower, at the expense of being less adaptive. Moreover, TLD performs an exhaustive search of faces through the whole image which also increases its processing time. Tracking failure may occur if an object with similar appearance to the target face is present in the scenario.

Algorithm2: TLD for a single face in a scene.

Input: Frames $\{\mathbf{I}_1,...,\mathbf{I}_\infty\}$. The target face is labelled in the first frame $\mathbf{I}_1$ using a face detection.

Output: States $\{\mathbf{X}_1,...,\mathbf{X}_\infty\}$ of the target face in the frames.

for t = 1, …, ∞ do
  if $t = 1$ **then**
    –Define $\mathbf{X}_1$ with the labeled face. A set of target $\mathbf{p}_1^+,...,\mathbf{p}_m^+$ and non-target $\mathbf{p}_1^-,...,\mathbf{p}_n^-$ patches are collected using $\mathbf{X}_1$
    –Initial FM is represented as $\mathbf{M} = \{\mathbf{p}_1^+,...,\mathbf{p}_m^+,\mathbf{p}_1^-,\mathbf{p}_2^-,...,\mathbf{p}_n^-\}$
  else
  A. Detection
    –Detect $\{\mathbf{p}_1, …, \mathbf{p}_z\}$ by varying the location and size of a search window in $\mathbf{I}_t$
    –Reject the patches with $V(\mathbf{p}_i) < V(\mathbf{p}_{\mathbf{X}_{t-1}})/2$, where $\mathbf{p}_{\mathbf{X}_{t-1}}$ is the patch defined by $\mathbf{X}_{t-1}$ and $V(\mathbf{p}_i) = E(\mathbf{p}_i^2) - E^2(\mathbf{p}_i)$
    –Reject the patches whose posterior probability generated by an ensemble classifier $\mathbf{C}$ is less than 0.5
    –Select $\mathbf{p}_r$ as the new location of FM in $\mathbf{I}_t$ which generates minimum S^r
  B. Tracking
    –Estimate FM location in $\mathbf{I}_t$ using *median flow* tracker [14]
    –Drop a track if $median|d_i - d_m| > 10$, where d_i and d_m are a single displacement and median displacement of the FM
  C. Integrator
    –Integrate *detection* and *tracking* results by selecting the window with maximum S^c.
    –Define $\mathbf{X}_t$ of the FM at $\mathbf{I}_t$ with the selected window
  D. Learning
    –P-*Expert* selects target patches from reliable tracks, where a track is considered reliable if $S^c > \delta$
    –N-*Expert* selects non-target patches, where the overlap between the target and non-target patches is less than 0.2
    – Update $\mathbf{M}$ and $\mathbf{C}$ with the selected patches
  end if
end for

2.3 Discriminative Sparse Coding–Based Tracking:

DSCT (see Algorithm 3) uses sparse codes to represent the target FM for tracking. When a face is initially detected, a set of positive samples are drawn from the locations $\mathbf{I}_{pos}$ specified by a Gaussian perturbation of the target location in the first frame $\mathbf{I}_1$ that satisfy $\|\mathbf{l}_{pos}-\mathbf{l}_1\|<\gamma$, and a set of negative samples are drawn from the locations that satisfy $\gamma<\|\mathbf{l}_{neg}-\mathbf{l}_1\|<\eta$, where the γ and η are the thresholds that define the area for collecting the positive and negative samples, respectively. The image patches specified by the locations $\mathbf{l}_{pos}$ and $\mathbf{l}_{neg}$ are then cropped and computed the sparse codes $\{\mathbf{z}_i, y_i\}_{i=1}^m$, where $\mathbf{z}_i \in \Re^{n+2d}$, $y_i \in \{+1,-1\}$, m is the number of training samples, d is the dimensionality of the image vectors, and n is the number of image vectors. With this data, a linear classifier is trained to identify the target in future frames. This is known as a static observation model. The DSCT also exploits an adaptive observation model which is constructed by accumulating several of the most recent frames. The sample in the following frame is first processed by adaptive observation model and then further examined with the static observation model to achieve tracking in the DSCT.

Both the observation models in DSCT depend on the strong assumption that all future states are similar to some extent to the ground truth in the first frame, which is not necessarily true in video surveillance. Furthermore, the adaptive model often fails, which leads to malfunctioning of the static model. The dimensionality of feature vectors in DSCT is high which translates to high computational complexity, especially when these vectors are further used to train a classifier.

Algorithm 3: Discriminative Sparse Coding–Based Tracking (DSCT) for a single face in a scene.

Input: Frames $\{\mathbf{I}_1,...,\mathbf{I}_\infty\}$. The target face is labeled in the first frame $\mathbf{I}_1$ using a face detection

Output: States $\{\mathbf{X}_1,...,\mathbf{X}_\infty\}$ of the target face in the frames

for t = 1, …, ∞ **do**
  if t = 1
    –Construct an initial over-complete dictionary $\mathbf{D}_t$
    –Learn a linear classifier with the sparse-codes extracted from $\mathbf{D}_t$ and parameter $\mathbf{w}_t$
  else
    –Perform particle filtering to estimate target state $\mathbf{X}_t^s$ by using the previous tracking result $\mathbf{X}_{t-1}$, and the adaptive observation model parameterized by $\mathbf{D}_{t-1}$ and $\mathbf{w}_{t-1}$
    –Set $\hat{\mathbf{X}}_{t-1} \leftarrow \mathbf{X}_t^s$.
    –Perform particle filtering again with $\hat{\mathbf{X}}_{t-1}$ and the static observation model parameterized by $\mathbf{D}_t$ and $\mathbf{w}_t$ to determine the final state $\mathbf{X}_t$
    if *tracking quality* < δ **then** // δ is a predefined threshold
      –Drop the track and exit
    else
      –Update the adaptive tracker to get $\mathbf{D}_t$ and $\mathbf{w}_t$ with the tracking result $\mathbf{X}_t$
    end if
  end if
end for

3 Performance Analysis

3.1 Experimental Setup

To compare the performance of the 3 trackers, 18 video sequences – 9 from entering and 9 from leaving (see Table 1) – are used from the Chokepoint dataset [15]. Each sequence views 5 individuals walking through the portal, one at a time. The Viola-Jones face detection algorithm [16] is used to initiate new tracks. These video sequences are recorded for face recognition applications under real world surveillance conditions, where an array of 3 cameras was placed above several portals (natural choke points of pedestrian traffic) to capture subjects walking through portals. The sequences are named according to the recording conditions (e.g., P2E_S1_C3), where P, S, and C stand for portal, sequence and camera, respectively. E and L indicate subjects either entering or leaving the portal.

Entering Seq.	Frame Count	Leaving Seq.	Frame Count
P1E_S1_C1	1023	P1L_S1_C1	905
P1E_S1_C2		P1L_S1_C2	
P1E_S1_C3		P1L_S1_C3	
P1E_S2_C1	1035	P1L_S2_C1	860
P1E_S2_C2		P1L_S2_C2	
P1E_S2_C3		P1L_S2_C3	
P2E_S2_C1	1073	P2L_S2_C1	1000
P2E_S2_C2		P2L_S2_C2	
P2E_S2_C3		P2L_S2_C3	

Table 1: Video sequences selected for performance analysis.

Both qualitative and quantitative evaluations are presented for performance analysis. In each case, tracker parameters were optimized to achieve best accuracy on the security feeds. In qualitative evaluation, the robustness and failure modes are observed visually under different face tracking challenges. The 3 following performance metrics are used for quantitative evaluation.

(i) Area Overlap Error (AOE) for a frame is computed by the ratio of the overlapping in the facial regions of the target face defined by the tracker to that of the ground truth as

$$AOE = 1 - \left(2\left|B_T \cap B_{GT}\right|\right) / \left(\left|B_T\right| + \left|B_{GT}\right|\right) \quad (5)$$

where B_T and B_{GT} represent the set of pixels in the tracking window and the window obtained from ground truth data, respectively. Here, *AOE* equals 0 in case of perfect overlap and 1 is case of no overlap at all. In the experiments, average *AOE* is computed for all the frames of a video sequence.

(ii) Tracking Error (TE) is computed for a video by taking the ratio of the number of correct face-correspondences achieved by a tracker and the total number of face-correspondences in the sequence as follows:

$$TE = 1 - T_C / T_A \quad (6)$$

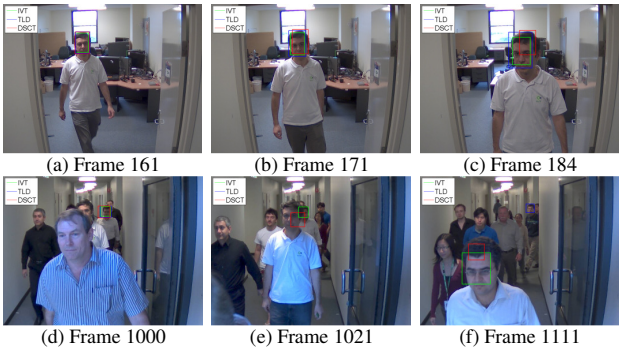
where T_C is the number of correct face-correspondences and T_A is the total number of face-correspondences that extend from the first to the last frame in the sequence. A face-correspondence is considered as a correct if $AOE > \partial$, where $\partial \in [0,1]$ is a predefined threshold.

(iii) Computation Time (CT) refers to the time needed by a tracker to process of one face in a frame. Trackers are implemented in Matlab code, and in order to compare the performance in terms of *CT*, the experiments are conducted on a 3.40 GHz Intel Core-i7 processor and 8 GB memory.

3.2 Experimental Results

To observe the behaviour of trackers under changes in scale, pose and illumination, the tracking results on the sequence P1L_S1_C1 are shown in Figures 1(a)-(c). In this sequence the illumination, pose, and the scale of the face region changes gradually. The IVT and TLD perform better on this sequence. Due to strong assumption that future states should be similar that of the ROI with a face of the first frame, the DSCT cannot adapt to changes of the capture condition and starts drifting as seen in Figure 1(c). To observe tracking performance under occlusion and clutter, the results on the sequence P2L_S5_C1 are shown in Figures 1(d)-(f). DSCT drifts and TLD stops tracking when a partial occlusion occurs as seen in Figure 1(e). In Figure 1 (f), TLD incorrectly retrieves a dropped track because features used as the face descriptor are less discriminant. IVT outperforms the other trackers over the whole sequence.

Table 2 shows the accuracy (*AOE* and *TE*) for the trackers on the 6 entering and 6 leaving sequences in Table 1, where the *TE* is computed by selecting the face-correspondences with $AOE > 0.7$. Since *AOE* is sensitive to small misalignment of the bounding boxes, $AOE \leq 0.7$ is considered as a satisfactory tracking result in this experiment. According to the average *AOE* and *TE*, it can be seen that IVT outperforms other methods. The robust tracking performance of IVT can be attributed to the eigenface representation of the face model which is robust to illumination change, pose change and clutter, and to the update mechanism [12] which incrementally updates the FM to adapt it with the changes of the scene. It is to be noted that, for all the tracking methods, the average *AOE* and *TE* are higher in the "entering" sequences than the "leaving" sequences; this is because the entering sequences are more challenging in terms of pose, scale and illumination changes.



(a) Frame 161 (b) Frame 171 (c) Frame 184

(d) Frame 1000 (e) Frame 1021 (f) Frame 1111

Figure 1: Face tracking provided by IVT, TLD, and DSCT on selected frames of P1L_S1_C1 and P2L_S5_C1 sequences.

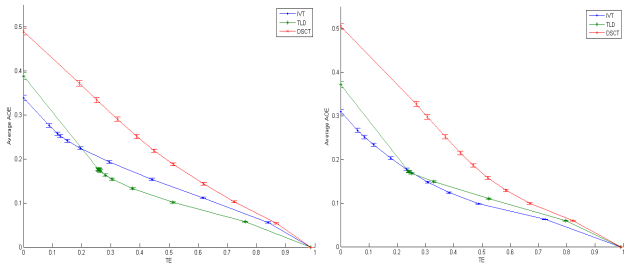
For a fair analysis, instead of fixing $\partial = 0.7$, this threshold is varied in the range [0, 1], and the corresponding *TE* and average *AOE* for the entering and leaving sequences are plotted in Figures 2(a) and (b), respectively. Note that an ideal tracker is represented by the point (0, 0) in these figures. In the sequences, when large variations in face appearance occurs due to any challenging condition, TLD stops tracking, whereas IVT and DSCT usually continue tracking with generating larger *AOE*. This results in higher average *AOE* for IVT and DSCT than TLD in the area of Figure 2(a) where *TE* is greater than 0.3. In real-world video surveillance applications, a face tracking method generating TE > 0.25 may not be suitable. In the areas of Figures 2(a) and (b), where $TE \leq 0.25$, IVT performs the best among methods.

	Entering Sequences			Leaving Sequences		
AAM	***Avg. AOE***	***TE***	***Avg. CT***	***Avg. AOE***	***TE***	***Avg. CT***
IVT	0.34 ± 0.005	0.14	0.08 ± 0.001	0.30 ± 0.005	0.17	0.09 ± 0.001
TLD	0.39 ± 0.008	0.26	1.03 ± 0.002	0.37 ± 0.007	0.24	0.97 ± 0.002
DSCT	0.49 ± 0.007	0.38	6.18 ± 0.001	0.50 ± 0.007	0.42	6.22 ± 0.002

Table 2: Avg. AOE, TE, and CT (in sec.) for the face tracking methods along with standard errors.

In terms of *CTs*, IVT also outperforms the other methods (Table 2). The computational effort required by the trackers is mainly found in processing steps for face model update and data association. For model update, IVT uses the SKL algorithm [12], which is computationally efficient. For DSCT, the large dimensionality of the final feature vector causes high computational cost when they are used to update a classifier [17]. TLD accumulates some positive and negative samples from a reliable tracking trajectory to update a random forest classifier to improve *detection*, which is computationally complex. For data association, IVT and DSCT generate 600 particles (candidate samples) each using particle filters, whereas TLD generates 50,000 candidate

patches for an image (240×320 pixels) by varying scale, size, and shifting parameters of an initial window, although these candidate patches are further refined by employing a cascaded classifier. The processing time for all the trackers depends on the update frequency. It is predefined for IVT and DSCT, but is adaptive for TLD according to changes observed in face tracks.



(a) Entering sequences (b) Leaving sequences

Figure 2: *TE* vs. average *AOE* of the trackers by varying $\partial \in [0,1]$

Since IVT outperformed others in terms of tracking accuracy and time complexity, further analysis is now presented. In particular, this section considers how this face tracking algorithms can be exploited within a system for FRiVS. One of the main limitations of IVT is track drift which occurs due to the integration of non-facial regions into the facial model during its update process. By controlling the plasticity of face models and assessing the quality of the facial regions that may be used to update a model, the problem of track drift can be mitigated. In the IVT, *batch size, m,* and *forgetting factor, f,* determine the plasticity of facial models. In addition, within a system for FRiVS, a face detection module may be exploited for facial quality assessment before a facial model is updated.

In IVT, *batch-size, m* refers to the number of frames used before updating a face model during tracking. In the original implementation of IVT (available in the authors' website, http://www.cs.toronto.edu/~dross/ivt/), this *m* is set to 5. However, this value affects a trade-off between computational efficiency and quality of facial modeling. Increase of *m* makes IVT computationally faster at the expense of being less adaptive, and vise-versa.

	Entering Sequences		Leaving Sequences	
Batch	***Avg. AOE***	***Avg. CT***	***Avg. AOE***	***Avg. CT***
5	0.338±0.005	0.081±0.001	0.304±0.005	0.091±0.001
10	0.357±0.006	0.077±0.005	0.312±0.006	0.073±0.003
15	0.368±0.005	0.071±0.003	0.337±0.008	0.066±0.004
20	0.379±0.007	0.063±0.008	0.341±0.003	0.062±0.007

Table 3: Average AOE and CT required by IVT for different batch sizes, *m.*

Table 3 presents the average *AOE* and average *CT* for six "leaving" and six "entering" sequences of Chokepoint dataset when batch sizes are *m* = 5, 10, 15, and 20. As *m* increases, the average *AOE* also increases while the *CT* decreases, because the internal facial models are updated less frequently. It is however not necessarily true that adapting face models more frequently can lead to higher tracking accuracy for all sequences. Rather, frequent updates may gradually incorporate non-facial regions and cause tracks to drift. For best tracking accuracy, when a video scene captures faces that move rapidly or abruptly, the face models should be adapted more frequently (with smaller batches) at the expense of *CT*. Moreover, the "entering" sequences result in higher *AOE* than the "leaving" sequences. This is due to the fact that the "entering" sequences include more variations in face appearances (pose, scale and illumination changes).

IVT exploits a parameter known as *forgetting factor*, $f \in [0, 1]$, where $f = 0$ is set to forget all previous observation, while $f = 1$ is set to remember them. This parameter determines the contributions of earlier observations while updating a face model. Figure 3 presents the *AOE* for the twelve videos of Chokepoint dataset when *f* is set to 0.00, 0.0.20, 0.40, 0.60, 0.80, and 1.00. This figure shows that, for most sequences, the *AOE* decreases, as the *f* increases. Results suggest that, though recent observations are more indicative of the current facial appearance, a contribution from both the recent and older observations in updating face model can be beneficial. Small *m* and *f* values are most suitable for abrupt changes, while larger *m* and *f* values are most suitable for tracks with gradual changes.

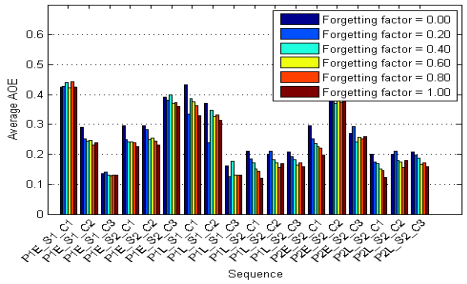


Figure 3: Average AOE of IVT for different forgetting factors.

Face detection is the front end of any system for FRiVS [18]. For quality assessment, the Viola-Jones face detection algorithm was considered. Figure 4 presents the *AOE* for the original *IVT*, as well as for *IVT with detection* on Chokepoint videos. In both of these cases, the face model is only updated for frames where the detection module finds a facial ROI that has significant overlap with the facial region predicted by the tracker. Figure 4(left) shows results when the FM is updated with the facial region obtained from the tracker, whereas in Figure 4(right) the update procedure is performed with the facial ROIs obtained by the face detection module. In both figures, it can be seen that the tracking accuracy for IVT with detection improves for sequences P1E_S1_C1, P1E_S1_C2, P1E_S2_C1, P1E_S2_C2, P1E_S2_C3, P2E_S2_C3, P1L_S1_C1, P1L_S2_C1, P1L_S2_C2, P1L_S2_C3 and P2L_S2_C1, where tracks are exposed to gradual changes. The detection module finds faces in most frames, and updates the facial model accordingly. Faces are modelled with a considerable number of high quality facial regions, and the tracker performs accurate DA, resulting in improved tracking accuracy.

In sequences P1L_S1_C2, P1L_S1_C3, P2E_S2_C1, P2E_S2_C2, P2L_S2_C2 and P2L_S2_C2, the pose, size, and illumination vary more abruptly and tracking accuracy is degraded. The detection module does not find faces in many frames, thus cannot efficiently update the facial models, leading to increased *AOE*. Both *IVT* and *IVT with detection* drift at the beginning of this sequence, resulting in large *AOE*. Comparing the results presented in Figure 4(left) with Figure

4(right), faces updated with ROIs (from the detection module) provide higher quality facial regions than the tracker. Although IVT with detection cannot perform well in all the sequences, the results in Figures 4(left) and (right) for the sequences P1E_S1_C1, P1E_S1_C2, P1L_S1_C1, P2E_S2_C3 and P2L_S2_C1 suggest that, by assessing the quality of facial regions before updating a facial model, it can improve tracking performance.

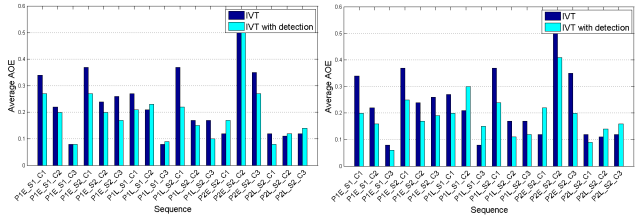


Figure 4: AOE of IVT and IVT with detection, where updating is driven by the face detection. (left) Face model is updated with the facial region obtained from tracker; (right) Face model is updated with the facial ROI obtained from face detection module.

4 Conclusion

This paper presents a comparison of three state-of-the-art AAM trackers – IVT, TLD, and DSCT – for FRiVS applications. The performance results for these trackers on Chokepoint video data indicates that IVT is the more suitable tracker since it outperforms the other methods in terms of tracking accuracy and computation time. The low discriminant power of the TLD face descriptor and the computational complexity of DSCT are the main limitations of these methods. An analysis of IVT has revealed the impact of its key parameters (i.e., batch size and forgetting factor) on its tracking quality. Results suggest that small batch size and forgetting factor values are most suitable for tracks with abrupt changes, while larger values are most suitable for tracks with larger ones. The quality of IVT's internal facial model is also shown to benefit from quality assessment of the face detection (i.e., the front-end face detection) module when face tracks are more gradual.

Although IVT performs better than TLD and DSCT, it still has some limitations. As with other generative AAM trackers, IVT relies on limited amount of data to learn and update the initial appearance FM. Many misaligned samples are likely to be learned incrementally which degrades the appearance model and causes track drift. Like all other AAM–based trackers, IVT is rather complex, yet lacks mechanism for detecting and correcting tracking error online as it sets no global constraints on the overall appearance of the target model. Their computational complexity grows with the number of facial tracks, size of faces, camera resolution and frame rate, and the number of cameras used by the surveillance system.

To improve the performance of IVT, face tracking can be triggered by the facial captures of a face detection module (e.g. Viola-Jones algorithm). ROIs from face detection can be used to construct and validate more reliable FM for IVT. Recently, contextual information has been exploited effectively in visual tracking [19] and can play an important role for the further improvement to IVT. Finally, this paper underscores the need for more empirical benchmarking studies to compare the numerous tracking methods presented in literature.

Acknowledgements

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada, and the Ministère du développement économique, de l'innovation et de l'exportation du Québec.

References

[1] R. Chellappa, M. Du, P. Turaga and S. K. Zhou, "Face tracking and recognition in video," in *Hand Book of Face Recognition*, Springer-Verlag, 2011, pp. 323-351.

[2] S. Salti and A. Cavallaro, "Adaptive appearance modeling for video tracking: survey and evaluation," *IEEE Trans. on Image Processing,* vol. 21(10), pp. 4334-4348, 2012.

[3] D. A. Ross, J. Lim, R.-S. Lin and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision,* vol. 77(1), pp. 125-141, 2008.

[4] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 34(7), pp. 1409-1422, 2012.

[5] Q. Wang, F. Chen, W. Xu and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *IEEE Workshop on Appl. of Computer Vision*, Beijing, China, January 2012.

[6] 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(1), pp. 63-84, 1998.

[7] A. D. Jepson, D. J. Fleet and T. F. El-Marahgi, "Robust online appearance models for visual tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 25(10), pp. 1296-1311, 2003.

[8] X. Li, W. Hu, Z. Zhang, X. Zhang and M. Zhu, "Visual tracking via incremental log-euclidean riemannian subspace learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Alaska, USA, June 2008.

[9] S. Avidan, "Support vector tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 26(8), pp. 1064-1072, 2004.

[10] S. Avidan, "Ensemble tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 29(2), pp. 261-271, 2007.

[11] B. Babenko, M.-H. Yang and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, California, USA, June 2009.

[12] A. Levy and M. Lindenbaum, "Sequential karhunen–loeve basis extraction and its application to images," *IEEE Trans. on Image Processing,* vol. 9(8), pp. 1371-1374, 2000.

[13] M. Kim, S. Kumar, V. Pavlovic and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Alaska, USA, June 2008.

[14] Z. Kalal, K. Mikolajczyk and J. Matas, "Forward-backward error: automatic detection of tracking failure," in *IEEE International Conf. on Pattern Recognition*, Istanbul, Turkey, August 2010.

[15] Y. Wong, S. Chen, S. Mau, C. Sanderson and B. C. Lovell, "Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition," in *IEEE Computer Vision and Pattern Recognition Workshops*, Colorado, USA, June 2011.

[16] P. Viola and M.J. Jones, "Robust real-time face detection," *International Journal of Computer Vision,* vol. 57(2), pp. 137-154, 2004.

[17] S. Zhang, H. Yao, X. Sun and X. Lu, "Sparse coding based visual tracking: review and experimental comparison," *Pattern Recognition,* vol. 46(7), p. 1772–1788, 2013.

[18] J. F. Connolly, E. Granger and R. Sabourin, "An adaptive classification system for video-based face recognition," *Information Sciences,* vol. 192, pp. 50-70, 2012.

[19] F. Li, X. Zhou, J. Ma and S. T. C. Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *IEEE Trans. on Medical Imaging,* vol. 29, no. 1, pp. 96-105, 2010.