Pattern Recognition 45 (2012) 1050–1060

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Correlation-based incremental visual tracking

Minyoung Kim

Department of Electronic & Information Engineering, Seoul National University of Science & Technology, Seoul 139-743, South Korea

ARTICLE INFO

Article history:
Received 20 February 2011
Received in revised form
22 July 2011
Accepted 24 August 2011
Available online 1 September 2011

Keywords:
Canonical correlation analysis
Incremental subspace learning
Visual tracking
Particle filtering

ABSTRACT

Generative subspace models like probabilistic principal component analysis (PCA) have been shown to be quite effective for visual tracking problems due to their representational power that can capture the generation process for high-dimensional image data. The recent advance of incremental learning has further enabled them to be practical for real-time scenarios. Despite these benefits, the PCA-based approaches in visual tracking can be potentially susceptible to noise such as partial occlusion due to their compatibility judgement based on the goodness of fitting for the entire image patch. In this paper we introduce a novel appearance model that measures the goodness of target matching as the correlation score between partial sub-patches within a target. We incorporate the canonical correlation analysis (CCA) into the probabilistic filtering framework in a principled manner, and derive how the correlation score can be evaluated efficiently in the proposed model. We then provide an efficient incremental learning algorithm that updates the CCA subspaces to adapt to new data available from the previous tracking results. We demonstrate the significant improvement in tracking accuracy achieved by the proposed approach on extensive datasets including the large-scale real-world YouTube celebrity video database as well as the novel video lecture dataset acquired from British Machine Vision Conference held in 2009, where both datasets are challenging due to the abrupt changes in pose, size, and illumination conditions.

1. Introduction

The visual tracking is a very important problem in computer vision whose goal is to localize the regions of the object of interest (e.g., face) in a stream of video frames. Central to the success of higher-level tasks such as object recognition and activity classification, it has received significant attention in the vision community for decades. However, the visual tracking still remains a challenging problem due to the intrinsic difficulty in modeling potentially varying appearance of a target object along video frames, typically originated from dynamic changes in pose and illumination as well as shape deformation for non-rigid objects.

In tackling the visual tracking problem, two essential issues that need to be considered are: (1) how to model the target appearance (e.g., subspace representation, kernel-weighted histogram), and (2) how to choose (i.e., search for) the best target location among a set of candidates in each video frame (e.g., temporal filtering, gradient search). For instance, [1] used a contour model for target representation while the density of the tracking state (or target location) is estimated and propagated within a probabilistic state-space model, often referred to as *particle filtering* or *Condensation*. In [2], a gradient-based search strategy called *mean-shift* was adopted to find the best-matching target patches represented as color histograms.

The former issue of the appearance modeling is known to be a more critical factor than the search strategy, and often significantly affects the tracking performance [3–6]. Developing a robust observation model has been the matter of primary interest in the recent visual tracking research. Existing appearance models include, among others, the view-based models [7], 3D models [8], mixture models [5], the kernel representation [9], and the Gaussian density based model [10].

In designing an appearance model, the crucial properties that a tracker needs to meet are robustness and adaptability to changes in target appearance (e.g., pose, illumination). Recent tracking methods such as the Incremental Visual Tracker (IVT) [6] aimed to achieve these goals by incorporating an adaptive appearance model. Similar attempts to incremental modeling of appearance changes have been suggested in [11–13]. In particular, the IVT represented a target as a low-dimensional subspace that captures the principal components of possible appearance variations, where the subspace is updated adaptively using the image patches tracked in the previous frames. Unlike many non-adaptive approaches that employ fixed appearance template models such as the eigentracking of [7], the IVT alleviates the burden of constructing a target model prior to tracking with a large number of expensive offline data, and tends to yield higher tracking

E-mail address: mikim21@gmail.com

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.026

accuracies. More recently, [11] extended the IVT by introducing offline target models (so-called *visual constraints*) to the adaptive one, addressing the IVT's known problem of susceptibility to drift due to the gradual adaptation to non-targets.

The appearance model of the IVT is essentially the probabilistic principal component analysis (PPCA) [14], a generative probabilistic model that aims to represent the image patch of the target from a low-dimensional latent subspace. Despite its representational power that can capture low-dimensional intrinsic variability for high-dimensional image data, the PPCA in visual tracking can be potentially susceptible to noise (e.g., partial occlusion). This is because the PPCA judges the goodness of target matching based on the compatibility score of the entire image patch with respect to its generation process. In the presence of partial occlusion, for instance, although a considerable portion of the target image patch remains intact, the occluded portion may severely degrade the compatibility score with respect to the PPCA generative model.

Although there have been some previous attempts to address the partial occlusion problem (e.g., [12]), in this paper we suggest a novel appearance model that judges the goodness of the target matching based on the *correlation* score between partial sub-patches within a target image patch. The sub-patches can be chosen a priori, say two half-patches obtained by vertical or horizontal split of the target (see Fig. 2). The intuition is that the statistical relationships between pairs of partial sub-patches tend to be less affected by environment changes (e.g., illumination variation or occlusion) since those sub-patches would undergo identical random noise processes that govern the changes in environment. Our approach can thus be more robust to noise than the PPCA which only judges how well the entire image patch fits to the underlying model.

For a reliable estimation of the correlation score for the high-dimensional image data, we basically consider the canonical correlation analysis (CCA) [15,16], a low-dimensional dyadic subspace model that captures the maximal correlation between two variates (sub-patches). In order to incorporate the correlation model into a probabilistic filtering framework in a principled manner, we utilize the probabilistic CCA (PCCA), the probabilistic extension of the CCA recently introduced by [17].

Our major contribution is two-fold: we first derive how the correlation score can be evaluated efficiently for a given image patch (that is, a particle in the filtering framework), which serves as the essential particle-reweighting part in our PCCA-based filtering model that gauges the relevance of each particle. We then provide an efficient incremental algorithm for updating the PCCA subspaces, which avoid the computationally demanding procedure of building the subspaces naively from the scratch. We call our approach the *correlation-based incremental visual tracking*.

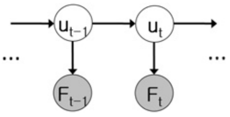

Fig. 1. First-order state-space model for tracking.

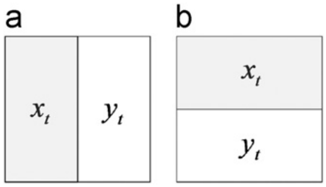

Fig. 2. Possible split of a given image patch $\mathbf{h}_t$ into a pair of correlation sub-patches: (a) vertical split, and (b) horizontal partition.

In an extensive set of experiments including the large-scale real-world YouTube celebrity video database [11] as well as the novel challenging video lecture dataset obtained from the British Machine Vision Conference held in 2009, we demonstrate that our approach significantly outperforms the IVT in terms of the tracking accuracy.

The paper is organized as follows: in the next section, the formal description of the probabilistic framework for visual tracking is provided. We then briefly review the IVT [6] in Section 3 focusing on the main ingredients, the PPCA subspace model and the incremental learning algorithm. In Section 4, we introduce our approach, the correlation-based incremental visual tracker, with the emphasis on the derivation for the observation likelihood score and the incremental PCCA update algorithm. The experimental evaluation results are provided in Section 5.

2. Probabilistic framework for visual tracking

In a probabilistic framework [1], the object tracking can be posed as an on-line temporal filtering problem where we aim to estimate

$$P(u_t|F_{0\ldots t}), \quad \text{for } t=1,2,\ldots. \tag{1}$$

Here F_t is the input image frame and u_t is the tracking state at time t with the initial state u_0 assumed known a priori either by manual mark-up or a object detector. The state u_t specifies an image patch in F_t that tightly surrounds the target object to be tracked. A typical choice is to form a square bounding box parameterized by the similarity transformation vector $u_t=[c_x,c_y,s,\phi]^T$, where the first two elements are the center position of the tracking bounding box, s is the scale w.r.t. the reference patch size (e.g., (48 pixels × 48 pixels)), and ϕ is the in-plane rotation angle from the horizontal axis. Hence the tracker is required to localize the target object in space (c_x,c_y) as well as in size and orientation. The cropped image patch specified by u_t in F_t is denoted by $\mathbf{h}_t$ which can be obtained from the affine warping function $\mathbf{h}_t=\omega(u_t,F_t)$.

For tractable estimation of Eq. (1), one typically assumes the first-order Markov dependency for the state dynamics whose graphical model representation is depicted in Fig. 1. In this model, two potential functions need to be specified: (1) the temporal dynamic (transition) model $P(u_t|u_{t-1})$, and (2) the appearance (emission) model $P(F_t|u_t)$. Among various dynamic models, smooth changes in motion can be well justified by the use of a random-walk (Gaussian) model, namely,

$$P(u_t|u_{t-1})=\mathcal{N}(u_t;u_{t-1},\Sigma_0), \tag{2}$$

with a proper choice of the covariance matrix Σ_0.

The emission model essentially judges the compatibility score of the tracking state u_t w.r.t. the underlying target appearance model, through the representative cropped image patch $\mathbf{h}_t=\omega(u_t,F_t)$ obtained by applying the affine warping function to u_t. To form an emission probability, one typically adopts the Gibbs energy form with a proper scaling parameter σ_0:

$$P(F_t|u_t)\propto e^{-E(\omega(u_t,F_t);\theta)/\sigma_0^2}, \tag{3}$$

where $E(\mathbf{h}_t;\theta)$ is the energy function that assigns a lower (higher) value when $\mathbf{h}_t$ is more (less) compatible with the target observation model θ.

Equipped with transition and emission models, one can then estimate Eq. (1) using the standard Bayesian recursion:

$$P(u_t|F_{0\ldots t})\propto P(F_t|u_t)\cdot\int P(u_t|u_{t-1})\cdot P(u_{t-1}|F_{0\ldots t-1})\,du_{t-1}. \tag{4}$$

The exact computation of Eq. (4) is in general intractable due to the non-Gaussianity of $P(F_t|u_t)$ (in terms of u_t). Among several approximated solutions, the sampling-based approaches such as the particle filtering are the most popular. In particle filtering [1], the conditional density $P(u_{t-1}|F_{0\ldots t-1})$ at the previous stage is approximately represented as a set of n weighted particles (samples) $\{(w^i,u^i)\}_{i=1}^n$. To estimate the belief $P(u_t|F_{0\ldots t})$ at the current time t, we perturb these particles using the Gaussian dynamics $P(u_t|u_{t-1})$, which are then reweighted by the observation likelihood $P(F_t|u_t)$. This procedure is recursively repeated for forthcoming frames in an online fashion.

In this framework, the energy function $E(\mathbf{h}_t)$ (that is, the appearance model) is the most critical part that directly affects the tracking performance since it judges the quality or goodness of each candidate particle. There have been numerous approaches suggested to build a robust appearance model. Traditionally, trackers based on template matching employed a fixed template target model, typically the cropped image of the initial frame $\mathbf{h}_0$, and defined the energy as the distance from the template (i.e., $E(\mathbf{h}_t) = d(\mathbf{h}_t,\mathbf{h}_0)$), where $d(\cdot,\cdot)$ is a properly chosen distance measure in the image space. The crucial drawback of such approach is that the trackers are not flexible enough to adjust the object's appearance changes originated from 3D motion or lighting condition variation. An immediate remedy may be to replace the template with the most recent one, $\mathbf{h}_{t-1}$, however, this can make the tracker too susceptible to appearance variations.

To satisfy the requirements of both robustness and adaptability to changes in target appearance, the adaptive appearance models have been proposed and studied intensively. The IVT [6] is one of the most successful ones, where it represents a target as a low-dimensional subspace that captures the principal components of possible appearance variations. Utilizing the principal component analysis (PCA) in visual tracking has been studied considerably in the computer vision community [7], however, the IVT introduced an efficient incremental singular value decomposition (SVD) algorithm that updates the PCA subspace adaptively using the image patches tracked in the previous frames. Some brief technical details of the IVT follow in the next section.

3. Incremental visual tracking (IVT)

The goal of the IVT [6] is to build and maintain a low-dimensional subspace at each time that captures the principal variations of the object appearance thus far. This can be achieved by learning a PCA subspace at each time t, denoted as $\mathcal{S}_t = (\mathbf{m}_t,\mathbf{B}_t)$ where we often hide the subscript t in notation for brevity, using the previously tracked images $\mathcal{D} = \{\mathbf{h}_0,\ldots,\mathbf{h}_{t-1}\}$. That is, $\mathbf{m}$ is the mean vector of $\mathcal{D}$, and $\mathbf{B}$ takes in its columns a few major eigenvectors of the covariance matrix estimated from $\mathcal{D}$. In the course of particle filtering, the goodness score of a particle u (corresponding to $\mathbf{h} = \omega(u,F)$) can then be measured by the so-called reconstruction error, namely,

$$E(\mathbf{h};\mathcal{S}) = \|\mathbf{h}-(\mathbf{m}+\mathbf{B}\mathbf{B}^\top(\mathbf{h}-\mathbf{m}))\|^2. \tag{5}$$

The origin of this PCA subspace model is the factor analysis [18] that aims at representing high-dimensional data through relatively low-dimensional latent variables. The dimension of the image patch $\mathbf{h}$, denoted by d, is typically several hundreds or thousands (e.g., the patch size of (48×48) pixels yields a 2304-dim vector). A more relevant formulation directly compatible with the probabilistic tracking framework can be derived by the use of the probabilistic extension of PCA, known as the PPCA [14].

In the PPCA, the observed data $\mathbf{h}$ is assumed to be generated by the low-dimensional latent variables $\mathbf{z} \in \mathbb{R}^q$ $(q \ll d)$ by the following linear equation:

$$\mathbf{h} = \mathbf{W}\mathbf{z}+\mathbf{m}+\boldsymbol{\varepsilon}, \tag{6}$$

where $\mathbf{W}$ is a $(d\times q)$ matrix that relates $\mathbf{z}$ to $\mathbf{h}$, and $\boldsymbol{\varepsilon}$ is the white noise. The conventional assumption of spherical Gaussianity, specifically $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I}_q)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}_d)$ (here, $\mathbf{I}_m$ indicates an m-dimensional identity matrix, and σ^2 is the noise variance), leads to the marginal observation model:

$$P(\mathbf{h}|\mathbf{W}) = \mathcal{N}(\mathbf{h};\mathbf{m},\mathbf{W}\mathbf{W}^\top+\sigma^2\mathbf{I}_d). \tag{7}$$

A link to the PCA can be established by the maximum likelihood estimate (MLE) for the PPCA. Given the data $\mathcal{D}$, the MLE can be derived as (see [14]):

$$\mathbf{W} = \mathbf{B}(\mathbf{L}-\sigma^2\mathbf{I}_q)^{1/2}\mathbf{R}, \tag{8}$$

where $\mathbf{B}$ is the $(d\times q)$ matrix comprised of q major PCA eigenbasis, $\mathbf{L}$ is the $(q\times q)$ diagonal matrix of the corresponding eigenvalues $(\lambda_1,\ldots,\lambda_q)$, and $\mathbf{R}$ is a $(q\times q)$ arbitrary rotation matrix.

Under this model, the appearance (log-)likelihood can be written as follows (up to some constant):

$$\log P(F_t|u_t) = \log P(\mathbf{h}_t|\mathbf{W}) = -\tfrac{1}{2}(\mathbf{h}_t-\mathbf{m})^\top\mathbf{C}^{-1}(\mathbf{h}_t-\mathbf{m}), \tag{9}$$

where $\mathbf{h}_t = \omega(u_t,F_t)$ and $\mathbf{C} = \mathbf{B}(\mathbf{L}-\sigma^2\mathbf{I}_q)\mathbf{B}^\top+\sigma^2\mathbf{I}_d$. Furthermore, it can be shown that the quadratic function in Eq. (9) is equivalent to

$$(\mathbf{h}_t-\mathbf{m})^\top\mathbf{C}^{-1}(\mathbf{h}_t-\mathbf{m}) = \mathbf{z}_t^\top\mathbf{L}^{-1}\mathbf{z}_t+\frac{1}{\sigma^2}\|(\mathbf{h}_t-\mathbf{m})-\mathbf{B}\mathbf{z}_t\|^2, \tag{10}$$

where $\mathbf{z}_t = \mathbf{B}^\top(\mathbf{h}_t-\mathbf{m})$. Notice that the second term in the right hand side of Eq. (10) is equal to the PCA's reconstruction error (Eq. (5)), while the first term accounts for the within-subspace Mahalanobis distance from the origin $\mathbf{m}$.

Another important observation from Eq. (10) is that the PPCA can be seen as a low-dimensional approximation of the full Gaussian modeling (i.e., $(\mathbf{h}-\mathbf{m})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{h}-\mathbf{m})$) that relaxes the full covariance $\boldsymbol{\Sigma}$ estimated from $\mathcal{D}$ by the PPCA matrix $\mathbf{C}$ with far fewer model parameters ($O(dq)$ vs. $O(d^2)$).

The remaining and the most critical issue required for real-time tracking is how to update efficiently the PPCA model every time a new tracked image is gleaned. Instead of performing the PCA computation — essentially conducting the singular value decomposition (SVD) of the data matrix — from the scratch, the incremental SVD learning algorithm, also known as the sequential Karhunen–Loeve (SKL) algorithm [19], was suggested in the IVT. The main idea of the SKL is to decompose the new data into orthogonal and dependent components with respect to the current PCA basis, perform the SVD on a low-dimensional matrix of the projected components, and use the factors to adjust basis to the new data. The time complexity of the update algorithm is independent on the number of training data accumulated thus far, allowing real-time tracking feasible. Readers are encouraged to refer to [6] for technical details.

Despite its representational power that can capture low-dimensional intrinsic variability for high-dimensional image data, a potential drawback of the PPCA in visual tracking is its susceptibility to noise such as partial occlusion. As PPCA's judgement on the goodness of target matching is based on how compatible the generation process of the entire image patch is, a small perturbation on the target image may affect severely the compatibility score with respect to the underlying generation process. To address this problem, we suggest a novel appearance model that judges the goodness of the target matching based on the *correlation* score between partial sub-patches within a target image patch. This will be discussed in the following section.

4. Correlation-based incremental visual tracking

Our main idea is to partition the image patch $\mathbf{h}_t$ into two sub-patches $\mathbf{x}_t$ and $\mathbf{y}_t$, and build a subspace that captures the correlation between two. The sub-patches can be chosen a priori, say two half-patches obtained by a vertical or horizontal split (see Fig. 2). Measuring the intrinsic statistical relationship between these high-dimensional variates can be effectively done by the subspace model called the canonical correlation analysis (CCA) [15,16], where its probabilistic interpretation [17], the probabilistic CCA abbreviated as PCCA, can be better suited for our belief propagation framework. Before we discuss our two major contributions, namely the derivation of how to evaluate the correlation score efficiently for a given image patch, and the novel incremental update algorithm for the PCCA subspaces, we present brief reviews on CCA and PCCA first.

4.1. Reviews on CCA and PCCA

The canonical correlation analysis (CCA) aims at finding projections for two random vectors that can yield the maximal correlation. More formally, for random vectors $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y} \in \mathbb{R}^{d_y}$, CCA finds embedding vectors $\mathbf{u}_x \in \mathbb{R}^{d_x}$ and $\mathbf{u}_y \in \mathbb{R}^{d_y}$ such that $\mathrm{Corr}(\mathbf{u}_x^\top \mathbf{x}, \mathbf{u}_y^\top \mathbf{y})$ is maximized, which can be expressed as the following optimization problem:

$$\max\ \mathbf{u}_x^\top \mathbf{\Sigma}_{xy} \mathbf{u}_y \quad \text{s.t.}\ \mathbf{u}_x^\top \mathbf{\Sigma}_{xx} \mathbf{u}_x = \mathbf{u}_y^\top \mathbf{\Sigma}_{yy} \mathbf{u}_y = 1, \tag{11}$$

where $\mathbf{\Sigma}_{ab} = \mathrm{Cov}(\mathbf{a},\mathbf{b})$ indicates the population (or sometimes, empirically estimated) covariance matrix for $\mathbf{a}$ and $\mathbf{b}$.

Using the Lagrangian multipliers, it can be shown that the optimal $\mathbf{u}_x$ and $\mathbf{u}_y$ are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigensystems (refer to [16]):

$$\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{yx}\mathbf{u}_x = \lambda_x \mathbf{\Sigma}_{xx}\mathbf{u}_x \quad \text{and} \quad \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xy}\mathbf{u}_y = \lambda_y \mathbf{\Sigma}_{yy}\mathbf{u}_y. \tag{12}$$

The eigenvalues of the above two equations are always equal, and the square root of the largest eigenvalue is the maximum correlation score which we denote by $\rho (= \lambda_x^{1/2} = \lambda_y^{1/2})$.

Having found the directions that yield the largest correlation, one can then recursively compute the subsequent projections subject to zero correlation with the previous ones. Considering to find q directions, the optimal CCA solutions are the square roots of the q largest eigenvalues ($\rho_1 \geq \cdots \geq \rho_q$) and the corresponding eigenvectors ($\{(\mathbf{u}_{xi},\mathbf{u}_{yi})\}_{i=1}^{q}$) of Eq. (12) [17].

We will often use the matrix notation: $\mathbf{U}_x = [\mathbf{u}_{x1}, \ldots, \mathbf{u}_{xq}]$ and $\mathbf{U}_y = [\mathbf{u}_{y_1}, \ldots, \mathbf{u}_{y_q}]$ are the CCA projection matrices of size $(d_x \times q)$ and $(d_y \times q)$, respectively, while $\mathbf{P}$ is the $(q \times q)$ diagonal matrix containing $\rho_1, \ldots, \rho_q$ in the diagonal entries. The CCA essentially reduces the dimensionality of data $\mathbf{x}$ and $\mathbf{y}$ to q ($\ll d_x, d_y$) by the projections $\mathbf{U}_x^\top \mathbf{x}$ and $\mathbf{U}_y^\top \mathbf{y}$.

Similar to giving birth to the probabilistic PCA (PPCA) by interpreting PCA within a probabilistic framework, the probabilistic CCA (PCCA) can be derived from the latent probabilistic modeling. In [17], the low-dimensional latent variable $\mathbf{z} \in \mathbb{R}^q$ is introduced, where the observed data $\mathbf{x}$ and $\mathbf{y}$ are generated from $\mathbf{z}$. More specifically, the PCCA can be specified as

$$\mathbf{x},\mathbf{y}\,|\,\mathbf{z} \sim \mathcal{N}(\mathbf{x};\mathbf{W}_x\mathbf{z}+\mathbf{m}_x,\mathbf{\Psi}_x)\cdot\mathcal{N}(\mathbf{y};\mathbf{W}_y\mathbf{z}+\mathbf{m}_y,\mathbf{\Psi}_y),$$

$$\mathbf{z} \sim \mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I}_q), \tag{13}$$

where $\mathbf{W}_x$ is the $(d_x \times q)$ coefficient matrix, $\mathbf{m}_x$ is the mean vector for $\mathbf{x}$, and $\mathbf{\Psi}_x$ is the $(d_x \times d_x)$ Gaussian covariance matrix (similarly for $\mathbf{y}$). By marginalizing out $\mathbf{z}$ in this model, one can get the following joint model:

$$\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_x\\ \mathbf{m}_y\end{pmatrix}, \begin{pmatrix}\mathbf{W}_x\mathbf{W}_x^\top+\mathbf{\Psi}_x & \mathbf{W}_x\mathbf{W}_y^\top\\ \mathbf{W}_y\mathbf{W}_x^\top & \mathbf{W}_y\mathbf{W}_y^\top+\mathbf{\Psi}_y\end{pmatrix}\right). \tag{14}$$

The following MLE for the model parameters essentially has a direct link to the CCA solution (i.e., $\mathbf{U}_x$, $\mathbf{U}_y$, and $\mathbf{P}$):

$$\mathbf{W}_x = \mathbf{\Sigma}_{xx}\mathbf{U}_x\mathbf{M}_x, \quad \mathbf{\Psi}_x = \mathbf{\Sigma}_{xx} - \mathbf{W}_x\mathbf{W}_x^\top, \tag{15}$$

$$\mathbf{W}_y = \mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{M}_y, \quad \mathbf{\Psi}_y = \mathbf{\Sigma}_{yy} - \mathbf{W}_y\mathbf{W}_y^\top, \tag{16}$$

where $\mathbf{M}_x$ and $\mathbf{M}_y$ are arbitrary $(q \times q)$ matrices that satisfy: $\mathrm{spr}(\mathbf{M}_x\mathbf{M}_x^\top) < 1$, $\mathrm{spr}(\mathbf{M}_y\mathbf{M}_y^\top) < 1$, and $\mathbf{M}_x\mathbf{M}_y^\top = \mathbf{P}$. Here $\mathrm{spr}(\mathbf{A})$ indicates the spectral radius of the matrix $\mathbf{A}$.

4.2. Appearance model based on PCCA

Our appearance model at time t is the CCA subspaces $(\mathbf{U}_x,\mathbf{U}_y,\mathbf{P})$ learned from the pairs of sub-patches, $\{(\mathbf{x}_\tau,\mathbf{y}_\tau)\}_{\tau=0}^{t-1}$, extracted from the previously tracked images $\{\mathbf{h}_t\}_{\tau=0}^{t-1}$. The sub-patches $(\mathbf{x}_t,\mathbf{y}_t)$ corresponding to the particle u_t (and hence $\mathbf{h}_t$) is then assigned the log-likelihood score that indicates the degree of consistency with respect to the current correlation model. Based on the PCCA model in Eq. (14), the compatibility score can be written as (denoting $\overline{\mathbf{x}}_t = \mathbf{x}_t - \mathbf{m}_x$ and $\overline{\mathbf{y}}_t = \mathbf{y}_t - \mathbf{m}_y$):

$$\log P(F_t\,|\,u_t) = -\frac{1}{2}\,(\overline{\mathbf{x}}_t\,\overline{\mathbf{y}}_t)^\top\,\mathbf{\Gamma}^{-1}\,(\overline{\mathbf{x}}_t\,\overline{\mathbf{y}}_t), \tag{17}$$

$$\text{where} \quad \mathbf{\Gamma} = \begin{pmatrix}\mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xx}\mathbf{U}_x\mathbf{P}\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}\\ \mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{P}\mathbf{U}_x^\top\mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{yy}\end{pmatrix}. \tag{18}$$

There are two computational issues in evaluating Eq. (17). First, a direct inversion of the matrix $\mathbf{\Gamma}$ of size $((d_x+d_y)\times(d_x+d_y))$, is prohibitive as it requires cubic time in the image dimension (d_x+d_y). Secondly, even though $\mathbf{\Gamma}^{-1}$ is available, naive approaches to computing the products in Eq. (17) require quadratic time $O(d_x^2+d_y^2)$ for each and every particle $(\mathbf{x}_t,\mathbf{y}_t)$. Considering that we usually sample hundreds or thousands of particles for each frame, it is imperative to reduce the time to linear (or sub-linear).

To address these issues, we first show that $\mathbf{\Gamma}^{-1}$ can be derived as follows:

$$\mathbf{\Gamma}^{-1} = \begin{pmatrix}\mathbf{\Sigma}_{xx}^{-1}+\mathbf{U}_x\mathbf{Q}\mathbf{U}_x^\top & -\mathbf{U}_x\mathbf{R}\mathbf{U}_y^\top\\ -\mathbf{U}_y\mathbf{R}\mathbf{U}_x^\top & \mathbf{\Sigma}_{yy}^{-1}+\mathbf{U}_y\mathbf{Q}\mathbf{U}_y^\top\end{pmatrix}, \tag{19}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are $(q \times q)$ diagonal matrices with $\mathbf{Q}_{ii} = \rho_i^2/(1-\rho_i^2)$ and $\mathbf{R}_{ii} = \rho_i/(1-\rho_i^2)$ for $i=1,\ldots,q$. The proof is presented in Appendix A.

The quadratic form in Eq. (17) is then expressed as

$$\begin{pmatrix}\overline{\mathbf{x}}\\ \overline{\mathbf{y}}\end{pmatrix}^\top \mathbf{\Gamma}^{-1}\begin{pmatrix}\overline{\mathbf{x}}\\ \overline{\mathbf{y}}\end{pmatrix} = \overline{\mathbf{x}}^\top\mathbf{\Sigma}_{xx}^{-1}\overline{\mathbf{x}}+\overline{\mathbf{y}}^\top\mathbf{\Sigma}_{yy}^{-1}\overline{\mathbf{y}}+\mathbf{z}_x^\top\mathbf{Q}\mathbf{z}_x+\mathbf{z}_y^\top\mathbf{Q}\mathbf{z}_y-2\mathbf{z}_x^\top\mathbf{R}\mathbf{z}_y, \tag{20}$$

where $\mathbf{z}_x = \mathbf{U}_x^\top\overline{\mathbf{x}}$ and $\mathbf{z}_y = \mathbf{U}_y^\top\overline{\mathbf{y}}$ are $(q \times 1)$ vectors obtained by projecting $\mathbf{x}$ and $\mathbf{y}$ on to the corresponding CCA subspaces. In the right-hand side of Eq. (20), while the last three terms can be computed in linear time $O(q(d_x+d_y))$, the first two terms still require quadratic time $O(d_x^2+d_y^2)$. To remedy this, we utilize the low-dimensional approximation motivated from the PPCA subspace modeling. That is, we approximate $\overline{\mathbf{x}}^\top\mathbf{\Sigma}_{xx}^{-1}\overline{\mathbf{x}}$ (similarly for $\mathbf{y}$), the score originated from the full Gaussian model $\mathbf{x} \sim \mathcal{N}(\mathbf{m}_x,\mathbf{\Sigma}_{xx})$, by the PPCA's factored representation (c.f. Eq. (10)):

$$\overline{\mathbf{x}}^\top\mathbf{\Sigma}_{xx}^{-1}\overline{\mathbf{x}} \approx (\mathbf{B}_x^\top\overline{\mathbf{x}})^\top\mathbf{L}_x^{-1}(\mathbf{B}_x^\top\overline{\mathbf{x}})+\frac{1}{\sigma^2}\|\overline{\mathbf{x}}-\mathbf{B}_x(\mathbf{B}_x^\top\overline{\mathbf{x}})\|^2, \tag{21}$$

where $(\mathbf{B}_x,\mathbf{L}_x)$ are the PCA estimates (of dimension referred to as $q_x(\ll d_x)$) learned from the $\mathbf{x}$ data. As this factored form can now be calculated in $O(q_x d_x)$, and similarly for $\mathbf{y}$, the overall particle scoring (Eq. (20)) can be performed in linear time in d_x and d_y.

The last remaining issue is to incrementally update the correlation model, specifically the CCA models $(\mathbf{U}_x,\mathbf{U}_y,\mathbf{P})$ as well as the PPCA models for $\mathbf{x}$ and $\mathbf{y}$, that is, $(\mathbf{B}_x,\mathbf{L}_x)$ and $(\mathbf{B}_y,\mathbf{L}_y)$, needed for approximation in Eq. (21). The latter can be done by directly applying the SKL algorithm of the IVT. In what follows, we derive an efficient incremental update algorithm for the CCA subspaces.

4.3. Incremental learning of CCA subspaces

In this section we discuss how to update the CCA subspace for $\mathbf{x}$, while the derivation for $\mathbf{y}$ can be done straightforwardly by simply interchanging $\mathbf{x}$ and $\mathbf{y}$. The CCA subspace is the solution to Eq. (12), which can be equivalently written in a matrix form as

$$\mathbf{A}\mathbf{U}_x = \mathbf{U}_x\mathbf{P} \quad \text{where } \mathbf{A} = \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}. \tag{22}$$

The task is to update $\mathbf{A}$ estimated with the existing data $\mathbf{D} = \{(\mathbf{x}_0,\mathbf{y}_0),\ldots,(\mathbf{x}_{t-1},\mathbf{y}_{t-1})\}$ to a new one, denoted by $\mathbf{A}^{new}$, learned from the augmented data, $\mathbf{D}^{new} = \{(\mathbf{x}_0,\mathbf{y}_0),\ldots,(\mathbf{x}_{t-1},\mathbf{y}_{t-1}),(\mathbf{x}_t,\mathbf{y}_t)\}$. Once $\mathbf{A}^{new}$ is available, the updated CCA subspace can be obtained by solving the generalized eigensystem in Eq. (22).

The main computational requirement here is that updating $\mathbf{A}$ as well as solving the eigensystem should be done quickly. Fortunately, unlike the correlation score evaluation in the previous section which is computed for each and every particle, the subspace update occurs only once for each frame, allowing quadratic running time acceptable. In accordance with this, finding the q largest eigenvalues/vectors in the generalized eigensystem can be done in $O(qd_x^2)$ using the Implicitly Restarted Lanczos Method (IRLM) [20,21]. However, naive methods for updating $\mathbf{A}$, for instance, through direct evaluation/inversion/multiplication of the covariance matrices ($\boldsymbol{\Sigma}_{xx}$, $\boldsymbol{\Sigma}_{xy}$, and $\boldsymbol{\Sigma}_{yy}$), would take cubic time impractical for tracking.

We suggest an efficient algorithm for updating $\mathbf{A}$ that can be performed in quadratic time. Our algorithm maintains the following matrices at the previous stage: (1) $\mathbf{m}_x$ and $\mathbf{m}_y$, (2) $\boldsymbol{\Sigma}_{xy}$ and $\boldsymbol{\Sigma}_{yx}(=\boldsymbol{\Sigma}_{xy}^\top)$, (3) $\boldsymbol{\Sigma}_{xx}^{-1}$ and $\boldsymbol{\Sigma}_{yy}^{-1}$, (4) $\mathbf{A}_0 = \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$, and (5) $\mathbf{A} = \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$. When the new data point $(\mathbf{x}_t,\mathbf{y}_t)$ arrives, the estimates of the means and the covariances become

$$\mathbf{m}_x^{new} = \frac{t}{t+1}\mathbf{m}_x + \frac{1}{t+1}\mathbf{x}_t, \quad (\text{similarly for } \mathbf{m}_y), \tag{23}$$

$$\boldsymbol{\Sigma}_{xy}^{new} = \frac{t}{t+1}\boldsymbol{\Sigma}_{xy} + \frac{t}{(t+1)^2}(\mathbf{x}_t-\mathbf{m}_x)(\mathbf{y}_t-\mathbf{m}_y)^\top. \tag{24}$$

For the inverse covariances, we use the Sherman–Morrison rank-one update formula, resulting in

$$(\boldsymbol{\Sigma}_{xx}^{new})^{-1} = \frac{t+1}{t}\boldsymbol{\Sigma}_{xx}^{-1} - \frac{\tilde{a}\tilde{a}^\top}{\alpha}, \quad (\boldsymbol{\Sigma}_{yy}^{new})^{-1} = \frac{t+1}{t}\boldsymbol{\Sigma}_{yy}^{-1} - \frac{\tilde{b}\tilde{b}^\top}{\beta}, \tag{25}$$

where

$$\alpha = 1 + a^\top\tilde{a}, \quad a = \frac{\sqrt{t}(\mathbf{x}_t-\mathbf{m}_x)}{t+1}, \quad \tilde{a} = \frac{t+1}{t}\boldsymbol{\Sigma}_{xx}^{-1}a,$$

$$\beta = 1 + b^\top\tilde{b}, \quad b = \frac{\sqrt{t}(\mathbf{y}_t-\mathbf{m}_y)}{t+1}, \quad \tilde{b} = \frac{t+1}{t}\boldsymbol{\Sigma}_{yy}^{-1}b.$$

The update equation for $\mathbf{A}_0$ can be derived as follows:

$$\mathbf{A}_0^{new} = \frac{t}{t+1}\mathbf{A}_0 + \frac{t}{(t+1)\beta}(a\hat{b}^\top + \hat{b}a^\top) + \frac{\beta-1}{\beta}aa^\top - \frac{t^2}{(t+1)^2\beta}\hat{b}\hat{b}^\top, \tag{26}$$

where $\hat{b} = \boldsymbol{\Sigma}_{xy}\tilde{b}$. Finally $\mathbf{A}^{new}$ can be obtained from the recursion:

$$\mathbf{A}^{new} = \mathbf{A} - \frac{t\tilde{a}(\tilde{a}^\top\mathbf{A}_0)}{(t+1)\alpha} + \tilde{a}(\eta\hat{b}^\top + \nu a^\top) + (\boldsymbol{\Sigma}_{xx}^{-1}\hat{b})\left(\frac{a^\top}{\beta} - \frac{t\hat{b}^\top}{(t+1)\beta}\right), \tag{27}$$

where

$$\eta = \frac{t}{(t+1)\alpha\beta} + \frac{t^2\tilde{a}^\top\hat{b}}{(t+1)^2\alpha\beta}, \quad \nu = \frac{\beta-1}{\alpha\beta} - \frac{t\tilde{a}^\top\hat{b}}{(t+1)\alpha\beta}.$$

We leave the proofs of Eqs. (25)–(27) in Appendix B. It is not difficult to see that all the update procedures from Eq. (23) to Eq. (27) can be completed in quadratic time.

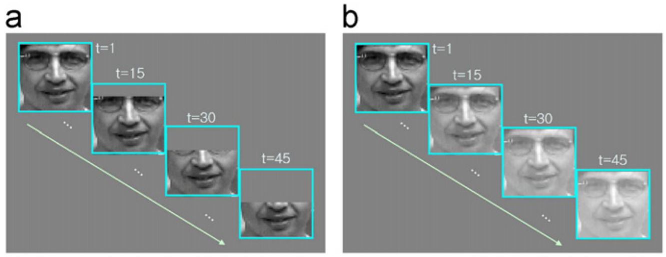

Fig. 3. Two synthetic data settings: (a) partial occlusion, (b) illumination change. See text for details.

5. Evaluation

In this section we empirically demonstrate the performance of the proposed correlation-based tracking method on both synthetic and real-world videos. We particularly deal with the face tracking problems, which are of greater importance and often even more popular than tracking non-face objects due to the intrinsic changeability in 3D pose as well as non-rigid facial expression.

We first consider two synthetic settings that simulate the conditions of partial occlusion and gradual illumination changes. Next, we test our algorithm on the extensive real-world video datasets including the large-scale real-world YouTube celebrity database [11] as well as the novel video lecture dataset acquired from British Machine Vision Conference held in 2009. The latter dataset is especially challenging due to abrupt changes in pose, size, and illumination conditions originated from dynamic motions of lecturers.

In these experiments, we focus on contrasting the tracking accuracy of our correlation-based incremental visual tracker which we denote by *CorrIVT*, with the *IVT* of [6]. Both algorithms are implemented in MATLAB with MEX. The tracking parameters common to both models (e.g., initial tracking state, image patch size, the number of particles, Condensation scale σ_0, etc.) are identically set for both trackers. In particular, the image patch size for all datasets is fixed as (48×48).[1] We use the vertical split (Fig. 2(a)) throughout the experiments, except for Section 5.1 where we illustrate the impact of splitting in an extreme case. The dimensions of the PCA and CCA subspaces are chosen empirically so as to yield the best performance for each model.

5.1. Synthetic dataset

To illustrate the robustness of *CorrIVT* to noise and appearance change, we devise two synthetic setups. In the first experiment, we emulate partial occlusion. As depicted in Fig. 3(a), given the image frame of uniform gray intensity, a face template, initially located at the upper-left corner ($t=1$), moves toward the lower-right corner ($t=45$). The upper portion of the template undergoes gradual occlusion by a grey patch with intensity equal to that of background. In the second setup, we simulate illumination change by gradually increasing the intensities of the pixels uniformly in the target template as illustrated in Fig. 3(b).

Our tracker and the IVT were applied to these synthetic videos, where the tracking state u is comprised of only two parameters, the center position of the target, as there is no variation in rotation or size. For each frame one hundred particles are randomly sampled centered at the previous position with the standard deviation set to 10 pixels along X, Y directions. We empirically choose the PPCA subspace dimension as 8, while the CCA subspace dimension is set to 4. Starting with the ground-truth initial position, the tracking results of both models are shown in Fig. 4.

[1] In [6], they used (32×32) template and 16-dim PPCA subspace.

Unlike *IVT* that eventually drifts away, our *CorrIVT* exhibits nearly perfect target tracking for both scenarios, demonstrating that the correlation modeling can be effective and less sensitive to noise in the appearance. We also measured the quantitative tracking error, the Euclidean distance (in pixel) between the ground-truth position and the predicted one averaged over entire frames, which is: *IVT* (9.57) vs. *CorrIVT* (1.76) for the partial occlusion setup, and *IVT* (9.35) vs. *CorrIVT* (1.37) for the changing illumination setup.

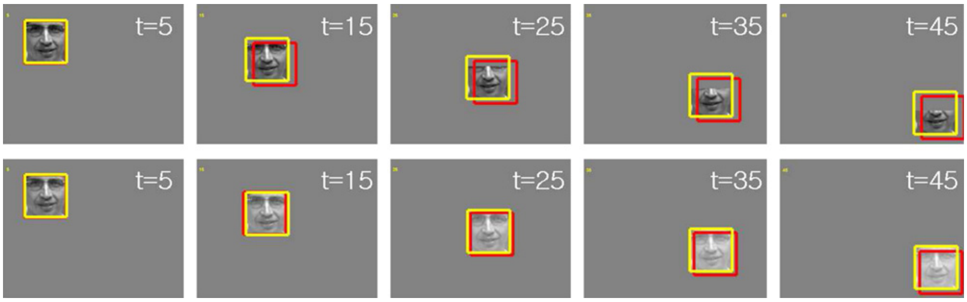

Fig. 4. Tracking results on the partial occlusion setup (top row) and the illumination change (bottom) for some selected frames. The yellow (brighter) box indicates *CorrIVT*, while the red (darker) is *IVT*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

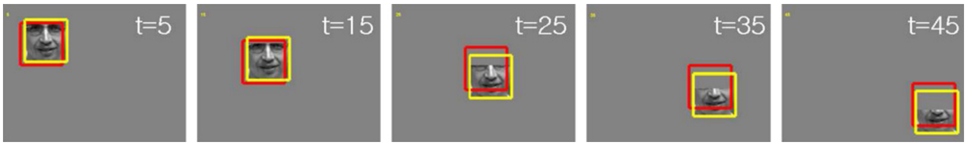

Fig. 5. Tracking results illustrating impact of patch split in the partial occlusion setup for some selected frames. The yellow (brighter) box indicates vertically split *CorrIVT*, while the red (darker) is horizontally split *CorrIVT*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

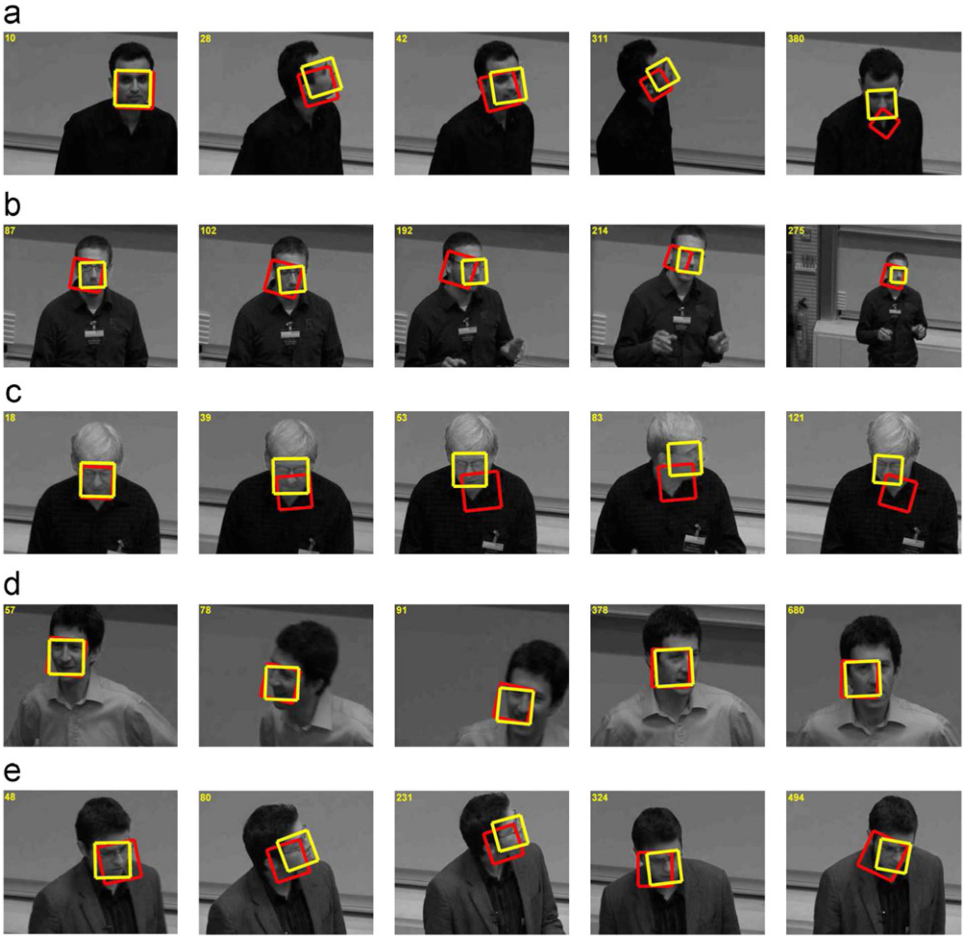

Fig. 6. Tracking results on the BMVC'09 lecture videos. Selected frames are highlighted where the yellow (brighter) box indicates *CorrIVT*, while the red (darker) is *IVT*. (a) Bashir. (b) Dickscheid. (c) Efros. (d) Fitzgibbon. (e) Gaidon. (f) Hidayat. (g) Mei. (h) Mortazavian. (i) Taylor. (j) Trinh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

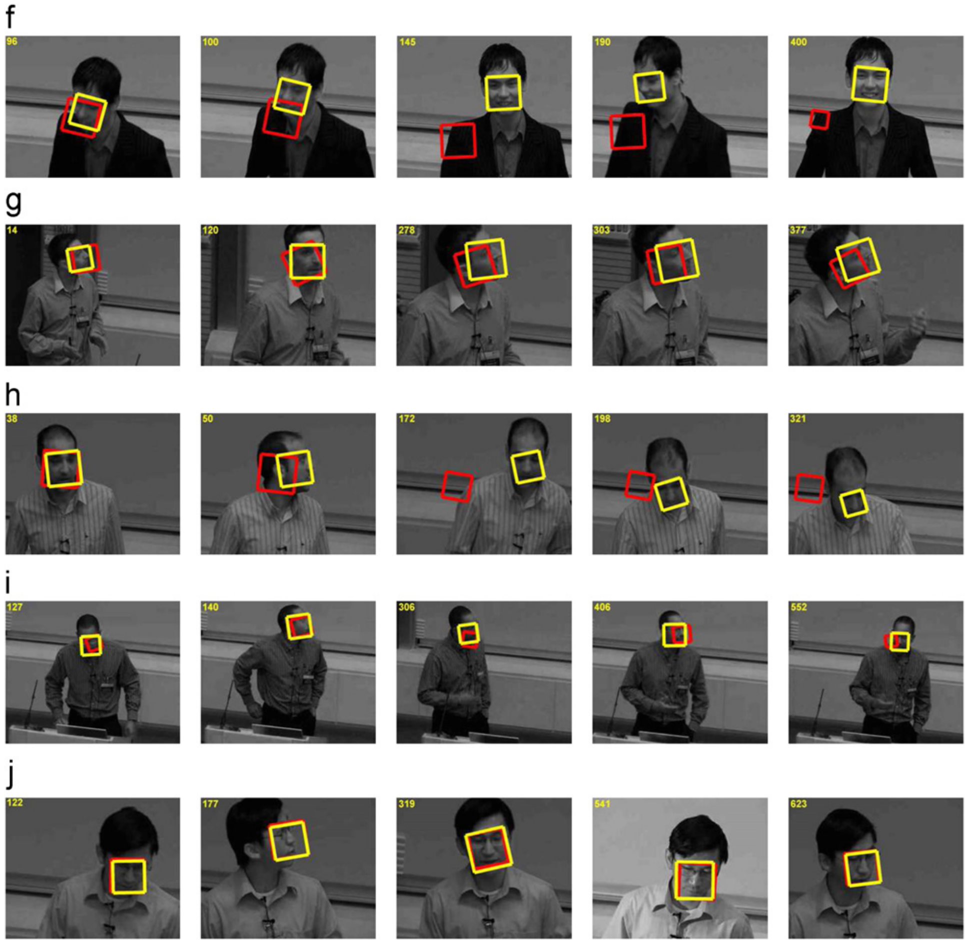

Fig. 6. (*continued*)

Table 1
Average tracking errors in pixel for the BMVC'09 lecture dataset. The numbers in parentheses indicate the numbers of tracking failures/reinitializations.

Lecturer	IVT	CorrIVT	Lecturer	IVT	CorrIVT
Bashir	6.38 (1)	3.93 (0)	Hidayat	6.22 (2)	2.89 (0)
Dickscheid	10.54 (0)	3.71 (0)	Mei	12.55 (0)	5.76 (0)
Efros	11.26 (1)	5.36 (0)	Mortazavian	6.36 (1)	5.58 (0)
Fitzgibbon	13.98 (2)	6.29 (1)	Taylor	7.13 (0)	5.61 (0)
Gaidon	9.73 (0)	4.24 (0)	Trinh	4.00 (0)	3.25 (0)

For the setting of gradual partial occlusion along vertical direction, we also tested *CorrIVT* with horizontal split to see the impact of patch splitting on tracking performance in an extreme case. Unlike vertical split where the occlusion equally affects both sub-patches, horizontal split in this setting undergoes changes in only one sub-patch. The average tracking error of the horizontally split *CorrIVT* is 8.38, which is slightly better than *IVT* (9.57) but not as good as the vertical-split *CorrIVT* (1.76). The tracking results that contrast horizontal and vertical splits are depicted in Fig. 5.

Although this is a rather extreme situation, the horizontal split degraded the tracking performance, in which the sub-patch correlation is not captured well. From this observation, the performance of our model can be affected by the choice of patch splitting. However, the correlation can be generally learned well under the circumstance that one sub-patch does not undergo completely different process of appearance changes from the other, which can be a mild assumption in practice (see the next two sections for real video tracking).

5.2. BMVC'09 video lecture dataset

We now test our tracking algorithm on real videos. We build a novel video lecture dataset[2] from British Machine Vision Conference (BMVC) held in 2009. After collecting videos of 10 presenters, each of length 1 ~ 2 min, the first step is the segmentation of the video frames as the raw videos usually contain frames with no lecturers (e.g., presentation slides only). We manually segment them into 500 ~ 1000 frames long video clips where the lecturers always appear. The frame size is (352×288).

These videos introduce significant challenges for the task of tracking: First, the lecturers constantly speak and move around, which results in appearance change in size, pose, and deformation. There are lots of extreme poses (e.g., profile and up/down views) as the lecturers look at the screen and audience frequently. Also, the occasional camera flash produces abrupt illumination change. The camera view point changes that focus on the lecturers lead to significant appearance variation as well.

The tracking state is composed of four parameters (two for center position and the others for scale and rotation), where the initial-frame state of each video is manually marked and provided with the trackers. We run the trackers with the empirically chosen PPCA subspace dimension 8 and the CCA dimension 8. The number of particles is set to 500 for both models. Running on a standard 2.4 GHz machine, the per-frame processing time is 0.1628 s (IVT)

[2] We will make the dataset together with the ground-truth face positions available for public access soon.

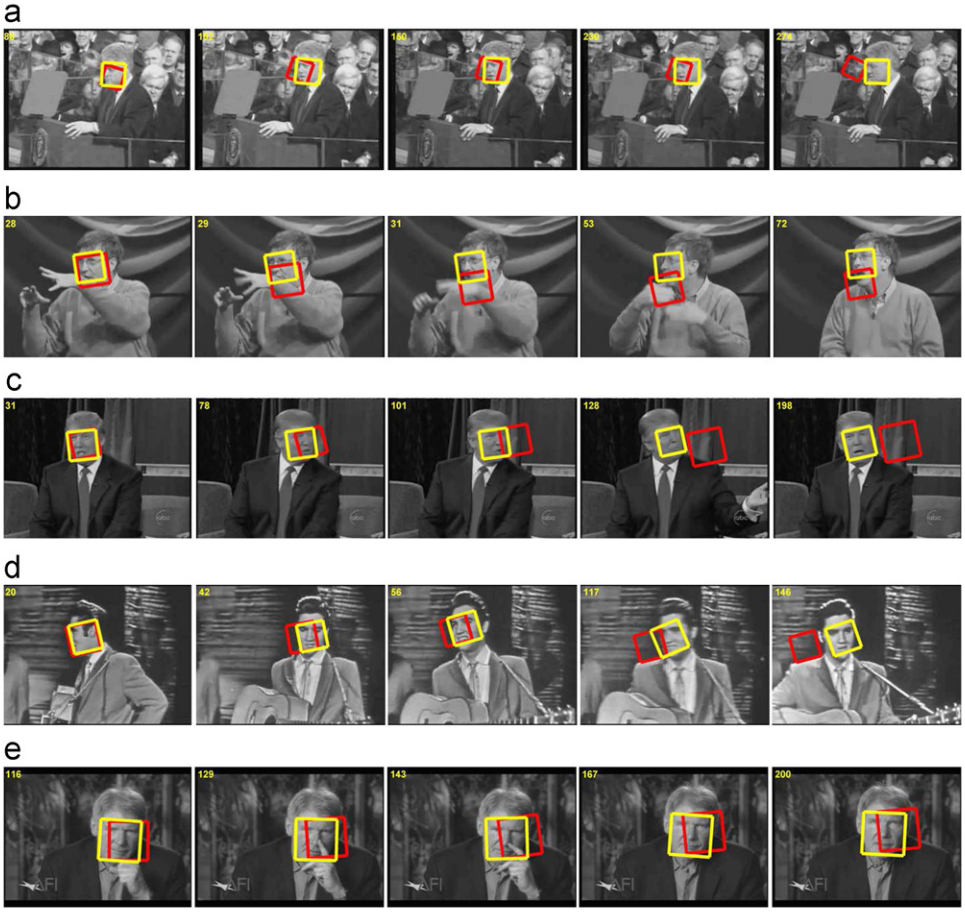

Fig. 7. Tracking results on the YouTube celebrity videos. Selected frames are highlighted where the yellow (brighter) box indicates *CorrIVT*, while the red (darker) is *VC-IVT*. (a) 09_bill_clinton_03_004. (b) 10_bill_gates_01_005. (c) 13_donald_trump_03_003. (d) 14_elvis_presley_03_001. (e) 19_harrison_ford_01_004. (f) 26_jet_li_02_001. (g) 39_paul_mccartney_01_004. (h) 43_steven_spielberg_01_013. (i) 44_sylvester_stallone_02_017. (j) 46_victoria_beckham_02_004. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and 0.5764 s (CorrIVT) on average. When visually inspected, our *CorrIVT* achieved successful tracking for nine videos among the 10 videos, while *IVT* succeeded for only five videos. The tracking results for some selected frames are depicted in Fig. 6 where the tracking states of both algorithms are superimposed.

For quantitative error measure, we manually recorded the center position of the face for every fifth frame. We then measure the Euclidean distance between the ground-truth position and the predicted center position. For rigorous performance comparison, we reinitialize the trackers when they completely drift away visually, and we keep record of the number of failures as well as the average Euclidean distance errors. These are reported in Table 1. As shown, our approach yielded significantly better performance in both failure rates and error rates on average than the IVT for most of the videos. The videos where the IVT fails often exhibit severer changes in appearance. This experiment demonstrates that not merely adaptation to appearance change, but also the role of correlation modeling is crucial for successful tracking being robust to changes in size, pose, and illumination.

5.3. YouTube celebrity database

Next we evaluate our approach on the YouTube celebrity dataset[3] [11], a large set of real-world videos, collected from the YouTube website. The dataset is comprised of 1910 videos from 47 celebrities, mostly actors/actresses and politicians, where the initial tracking states are also provided in the dataset. Each video clip consists of hundreds of frames, all containing the celebrity of interest, where the frame size ranges from (180×240) to (240×320).

This dataset is known to be challenging mainly due to the low quality, low resolution recorded at high compression rates and significant changes in facial pose and expression. In addition, there is a variety of noise effects ranging from poor illumination conditions, occlusion, and cluttered background. The visually constrained IVT tracker (referred to as *VC-IVT*) of [11], the extension of the IVT by incorporating the non-adaptive models of facial pose subspaces and the alignment SVM, is the best performing tracker for this dataset thus far, yielding successful tracking for about 80% of the videos.

The PPCA subspace dimension is empirically set as 16 and the CCA as 8. All the other peripheral parameters are chosen identically for both *VC-IVT* and our *CorrIVT* similarly to the previous section. Without video-wise parameter tuning (the same as the setup of [11]), the proportions of the successfully tracked videos are: *VC-IVT* (82.88%) vs. *CorrIVT* (93.14%). Fig. 7 highlights some examples: our approach exhibits strong robustness especially to occlusion ((b), (e), (h)) and abrupt pose changes ((d), (f), (i)). Considering the high variability in appearance and dynamics in this dataset, the performance improvement achieved by our tracker is outstanding, again emphasizing the impact of correlation modeling that can capture the statistical relationship between subsets of the target in tracking.

[3] Available on http://seqam.rutgers.edu/projects/motion/face/face.html

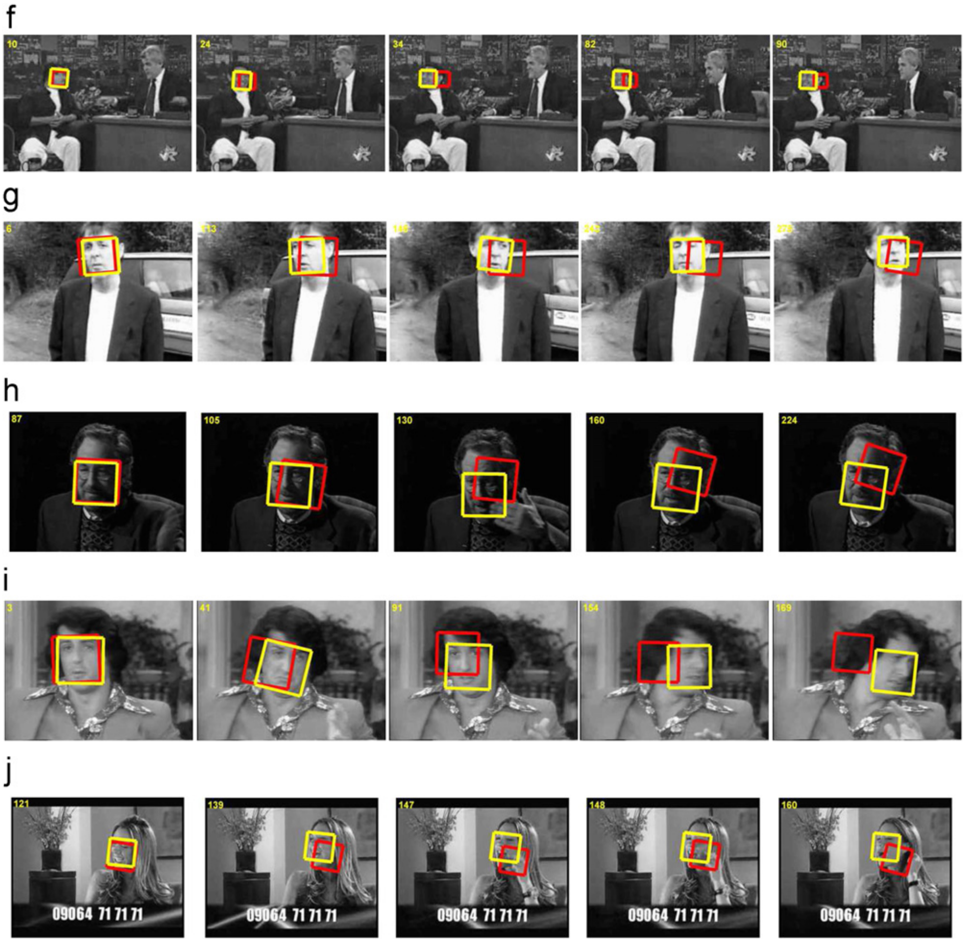

Fig. 7. (*continued*)

6. Concluding remarks

In this paper we have proposed a novel correlation-based appearance model in visual tracking problem that can be robust to noise such as partial occlusion. Compared to the existing generative models such as PCA subspaces which only judge how well the entire image fits to the underlying generation process, our model captures the statistical relationship between partial patches within a target, which is shown to be crucial for accurate and robust tracking. In addition, we have introduced an efficient incremental learning algorithm for CCA subspaces that can adjust the correlation to the circumstance change. Throughout the extensive experiments on both synthetic and real-world datasets, we have shown that our approach can improve the performance of the IVT significantly.

As our model captures the correlation of the sub-patches, it performs the best under the circumstances where one sub-patch does not undergo completely different process of appearance changes from the other. To be resilient to this condition, the intelligent patch splitting strategies can be adopted: for instance, one can split a patch adaptively or randomly select pixels for a sub-patch. These are all interesting research topics that needs to be pursued further in the future. Another potential drawback of our approach is that the correlation update algorithm is relatively much slower than the incremental SVD algorithm of the IVT due to the quadratic-time model update equations. Further development of new update algorithms may be desirable for real-time applications, which we leave as our future work.

Appendix A. Proof of Eq. (19)

We use the Schur's lemma for a block matrix inversion: for square matrices A,B,C (symmetric A and C with $\det(A) \neq 0$), it holds that

$$\begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1}+A^{-1}BS^{-1}B^\top A^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}B^\top A^{-1} & S^{-1} \end{pmatrix}, \tag{28}$$

where $S = C - B^\top A^{-1}B$ is the Schur complement. By letting $A = \mathbf{\Sigma}_{xx}$, $B = \mathbf{\Sigma}_{xx}\mathbf{U}_x\mathbf{P}\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}$, and $C = \mathbf{\Sigma}_{yy}$, the left-hand side of Eq. (28) becomes $\mathbf{\Gamma}^{-1}$ while S can be written as

$$S = \mathbf{\Sigma}_{yy} - \mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{P}\underbrace{\mathbf{U}_x^\top\mathbf{\Sigma}_{xx}\mathbf{U}_x}_{=\mathbf{I}_q}\mathbf{P}\mathbf{U}_y^\top\mathbf{\Sigma}_{yy} = \mathbf{\Sigma}_{yy} - \mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{P}^2\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}. \tag{29}$$

To compute S^{-1}, the Woodbury identity[4] is applied as follows:

$$\begin{aligned} S^{-1} &= \mathbf{\Sigma}_{yy}^{-1}(\mathbf{I}_{d_y} - \mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{P}^2\mathbf{U}_y^\top)^{-1} \\ &= \mathbf{\Sigma}_{yy}^{-1}\left(\mathbf{I}_{d_y} + \mathbf{\Sigma}_{yy}\mathbf{U}_y\left(\mathbf{P}^{-2} - \underbrace{\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}\mathbf{U}_y}_{=\mathbf{I}_q}\right)^{-1}\mathbf{U}_y^\top\right) \\ &= \mathbf{\Sigma}_{yy}^{-1}(\mathbf{I}_{d_y} + \mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{Q}\mathbf{U}_y^\top s) \end{aligned} \tag{30}$$

$$= \mathbf{\Sigma}_{yy}^{-1} + \mathbf{U}_y\mathbf{Q}\mathbf{U}_y^\top, \tag{31}$$

[4] $(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1}$.

where in Eq. (30), we denote $(\mathbf{P}^{-2}-\mathbf{I}_q)^{-1}$ by $\mathbf{Q}$ which is the $(q \times q)$ diagonal matrix whose ith entry is $\rho_i^2/(1-\rho_i^2)$. Since the magnitude of the correlation ρ is no greater than 1 (if the non-practical case of perfect correlation (i.e., $\rho_i^2=1$) is discarded), we have non-zero denominators.

The other blocks of $\mathbf{\Gamma}^{-1}$ in Eq. (28) can then be derived as

$$\begin{aligned}-A^{-1}BS^{-1} &= -\mathbf{\Sigma}_{xx}^{-1}\mathbf{\Sigma}_{xx}\mathbf{U}_x\mathbf{P}\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}(\mathbf{\Sigma}_{yy}^{-1}+\mathbf{U}_y\mathbf{Q}\mathbf{U}_y^\top)\\ &= -\mathbf{U}_x\mathbf{P}\mathbf{U}_y^\top-\mathbf{U}_x\mathbf{P}\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{Q}\mathbf{U}_y^\top\\ &= -\mathbf{U}_x(\mathbf{P}+\mathbf{P}\mathbf{Q})\mathbf{U}_y^\top = -\mathbf{U}_x\mathbf{R}\mathbf{U}_y^\top, \end{aligned} \tag{32}$$

where $\mathbf{R}=\mathbf{P}+\mathbf{P}\mathbf{Q}$ is the diagonal matrix whose ith entry is $\rho_i/(1-\rho_i^2)$. And,

$$\begin{aligned}A^{-1}+A^{-1}BS^{-1}B^\top A^{-1} &= \mathbf{\Sigma}_{xx}^{-1}+\mathbf{U}_x\mathbf{R}\mathbf{U}_y^\top\mathbf{\Sigma}_{yy}\mathbf{U}_y\mathbf{P}\mathbf{U}_x^\top\mathbf{\Sigma}_{xx}\mathbf{\Sigma}_{xx}^{-1}\\ &= \mathbf{\Sigma}_{xx}^{-1}+\mathbf{U}_x\mathbf{R}\mathbf{P}\mathbf{U}_x^\top = \mathbf{\Sigma}_{xx}^{-1}+\mathbf{U}_x\mathbf{Q}\mathbf{U}_x^\top. \end{aligned} \tag{33}$$

This completes the proof. □

Appendix B. Proofs of Eqs. (25)–(27)

Since $\mathbf{\Sigma}_{xx}^{new}=t/(t+1)\mathbf{\Sigma}_{xx}+t/(t+1)^2(\mathbf{x}_t-\mathbf{m}_x)(\mathbf{x}_t-\mathbf{m}_x)^\top$, Eq. (25) immediately follows from the Sherman–Morrison formula,[5] by letting $A=t/(t+1)\mathbf{\Sigma}_{xx}$ and $u=v=\sqrt{t}/(t+1)(\mathbf{x}_t-\mathbf{m}_x)$.

To prove Eq. (26), we first let $\mathbf{C}_0=\mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{yx}$, where its update equation can be derived as

$$\begin{aligned}\mathbf{C}_0^{new} &= (\mathbf{\Sigma}_{yy}^{new})^{-1}\mathbf{\Sigma}_{yx}^{new}\\ &= \left(\frac{t+1}{t}\mathbf{\Sigma}_{yy}^{-1}-\frac{\tilde{b}\tilde{b}^\top}{\beta}\right)\left(\frac{t}{t+1}\mathbf{\Sigma}_{yx}+ba^\top\right)\\ &= \mathbf{\Sigma}_{yy}^{-1}\mathbf{\Sigma}_{yx}+\frac{t+1}{t}\mathbf{\Sigma}_{yy}^{-1}ba^\top-\frac{t}{(t+1)\beta}\tilde{b}(\tilde{b}^\top\mathbf{\Sigma}_{yx})-\frac{\tilde{b}(\tilde{b}^\top b)a^\top}{\beta}\\ &= \mathbf{C}_0+\tilde{b}a^\top-\frac{t}{(t+1)\beta}\tilde{b}(\tilde{b}^\top\mathbf{\Sigma}_{yx})-\frac{\tilde{b}(\beta-1)a^\top}{\beta}\\ &= \mathbf{C}_0+\frac{\tilde{b}a^\top}{\beta}-\frac{t}{(t+1)\beta}\tilde{b}(\tilde{b}^\top\mathbf{\Sigma}_{yx}). \end{aligned} \tag{34}$$

Using this result, it follows that

$$\begin{aligned}\mathbf{A}_0^{new} &= \mathbf{\Sigma}_{xy}^{new}\mathbf{C}_0^{new}\\ &= \left(\frac{t}{t+1}\mathbf{\Sigma}_{xy}+ab^\top\right)\left(\mathbf{C}_0+\frac{\tilde{b}a^\top}{\beta}-\frac{t\tilde{b}(\tilde{b}^\top\mathbf{\Sigma}_{yx})}{(t+1)\beta}\right)\\ &= \frac{t}{t+1}\mathbf{A}_0+\frac{ta(\tilde{b}^\top\mathbf{\Sigma}_{yx})}{t+1}+\frac{t(\mathbf{\Sigma}_{xy}\tilde{b})a^\top}{(t+1)\beta}\\ &\quad+\frac{(\beta-1)aa^\top}{\beta}-\frac{t^2(\mathbf{\Sigma}_{xy}\tilde{b})(\tilde{b}^\top\mathbf{\Sigma}_{yx})}{(t+1)^2\beta}-\frac{t(\beta-1)a(\tilde{b}^\top\mathbf{\Sigma}_{yx})}{(t+1)\beta}\\ &= \frac{t}{t+1}\mathbf{A}_0+\frac{ta\hat{b}^\top}{t+1}+\frac{t\hat{b}a^\top}{(t+1)\beta}+\frac{(\beta-1)aa^\top}{\beta}-\frac{t^2\hat{b}\hat{b}^\top}{(t+1)^2\beta}\\ &\quad-\frac{t(\beta-1)a\hat{b}^\top}{(t+1)\beta}\\ &= \frac{t}{t+1}\mathbf{A}_0+\frac{t(a\hat{b}^\top+\hat{b}a^\top)}{(t+1)\beta}+\frac{(\beta-1)aa^\top}{\beta}-\frac{t^2\hat{b}\hat{b}^\top}{(t+1)^2\beta}. \end{aligned}$$

Finally, the recursion for $\mathbf{A}$ can be obtained from the following derivation:

$$\begin{aligned}\mathbf{A}^{new} = (\mathbf{\Sigma}_{xx}^{new})^{-1}\mathbf{A}_0^{new} &= \left(\frac{t+1}{t}\mathbf{\Sigma}_{xx}^{-1}-\frac{\tilde{a}\tilde{a}^\top}{\alpha}\right)\left(\frac{t}{t+1}\mathbf{A}_0+\frac{t(a\hat{b}^\top+\hat{b}a^\top)}{(t+1)\beta}\right.\\ &\quad\left.+\frac{(\beta-1)aa^\top}{\beta}-\frac{t^2\hat{b}\hat{b}^\top}{(t+1)^2\beta}\right)\\ &= \mathbf{A}-\frac{t\tilde{a}(\tilde{a}^\top\mathbf{A}_0)}{(t+1)\alpha}+\frac{(\mathbf{\Sigma}_{xx}^{-1}a)\hat{b}^\top}{\beta}-\frac{t(\tilde{a}^\top a)\tilde{a}\hat{b}^\top}{(t+1)\alpha\beta}\\ &\quad+\frac{(\mathbf{\Sigma}_{xx}^{-1}\hat{b})a^\top}{\beta}-\frac{t(\tilde{a}^\top\hat{b})\tilde{a}a^\top}{(t+1)\alpha\beta}+\frac{(t+1)(\beta-1)(\mathbf{\Sigma}_{xx}^{-1}a)a^\top}{t\beta}\\ &\quad-\frac{(\beta-1)(\tilde{a}^\top a)\tilde{a}a^\top}{\alpha\beta}-\frac{t(\mathbf{\Sigma}_{xx}^{-1}\hat{b})\hat{b}^\top}{(t+1)\beta}+\frac{t^2(\tilde{a}^\top\hat{b})\tilde{a}\hat{b}^\top}{(t+1)^2\alpha\beta}\\ &= \mathbf{A}-\frac{t\tilde{a}(\tilde{a}^\top\mathbf{A}_0)}{(t+1)\alpha}+\left(\frac{t}{(t+1)\alpha\beta}+\frac{t^2(\tilde{a}^\top\hat{b})}{(t+1)^2\alpha\beta}\right)\tilde{a}\hat{b}^\top\\ &\quad+\left(\frac{\beta-1}{\alpha\beta}-\frac{t(\tilde{a}^\top\hat{b})}{(t+1)\alpha\beta}\right)\tilde{a}a^\top+\frac{(\mathbf{\Sigma}_{xx}^{-1}\hat{b})a^\top}{\beta}-\frac{t(\mathbf{\Sigma}_{xx}^{-1}\hat{b})\hat{b}^\top}{(t+1)\beta}. \end{aligned}$$

This proves Eq. (27). □

[5] For invertible A and two vectors u,v, $(A+uv^\top)^{-1}= A^{-1}(-A^{-1}uv^\top A^{-1}/1+v^\top A^{-1}u)$.

References

[1] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, in: European Conference on Computer Vision, Cambridge, UK, 1996.

[2] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–575.

[3] M.J. Black, D.J. Fleet, Y. Yacoob, A framework for modeling appearance change in image sequence, in: Proceedings of IEEE International Conference on Computer Vision, 1998, pp. 660–667.

[4] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 681–685.

[5] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appearance models for visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1296–1311.

[6] D. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, International Journal of Computer Vision 77 (1) (2008) 125–141.

[7] M.J. Black, A.D. Jepson, EigenTracking: robust matching and tracking of articulated objects using a view-based representation, in: European Conference on Computer Vision, Cambridge, UK, 1996.

[8] M. La Cascia, S. Sclaroff, V. Athitsos, Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (4) (2000) 322–336.

[9] G.D. Hager, M. Dewan, C. V. Stewart, Multiple kernel tracking with SSD, in: IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, 2004.

[10] B. Han, D. Comaniciu, Y. Zhu, L.S. Davis, Sequential kernel density approximation and its application to real-time visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7) (2008) 1186–1197.

[11] M. Kim, S. Kumar, V. Pavlovic, H.A. Rowley, Face tracking and recognition with visual constraints in real-world videos, in: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 2008.

[12] K. Hotta, Adaptive weighting of local classifiers by particle filters for robust tracking, Pattern Recognition 42 (5) (2009) 619–628.

[13] A. Yao, G. Wang, X. Lin, X. Chai, An incremental Bhattacharyya dissimilarity measure for particle filtering, Pattern Recognition 43 (4) (2010) 1244–1256.

[14] M.E. Tipping, C.M. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society, Series B 61 (3) (1999) 611–622.

[15] H. Hotelling, Relations between two sets of variates, Biometrika 28 (1936) 321–377.

[16] D. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: an overview with application to learning methods, Neural Computation 16 (12) (2004) 2639–2664.

[17] F. Bach, M.I. Jordan, A Probabilistic Interpretation of Canonical Correlation Analysis, Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

[18] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, Wiley, New York, 1984.

[19] A. Levy, M. Lindenbaum, Sequential Karhunen–Loeve basis extraction and its application to images, IEEE Transactions on Image Processing 9 (8) (2000) 1371–1374.

[20] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. Vorst (Eds.), Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.

[21] S. White, P. Smyth, A spectral clustering approach to finding communities in graph, in: SIAM Data Mining, Newport Beach, CA, 2005.

Minyoung Kim received the BS and MS degrees both in Computer Science and Engineering in Seoul National University, South Korea. He earned the PhD degree in Computer Science from Rutgers University in 2008. From 2009 to 2010 he was a postdoctoral researcher at the Robotics Institute of Carnegie Mellon University. He is currently an Assistant Professor in the Department of Electronic and Information Engineering at Seoul National University of Science and Technology in Korea. His primary research interest is machine learning and computer vision. His research focus includes graphical models, motion estimation/tracking, discriminative models/learning, kernel methods, and dimensionality reduction.