

Subspace Analysis Of Arbitrarily Many Linear Filter Responses with an application to Face Tracking

Stefanos Zafeiriou[†]

[†]Dept. of Computing,
Imperial College London
180 Queen's Gate
London SW7 2AZ, U.K.

{s.zafeiriou, gt204, m.pantic}@imperial.ac.uk

Georgios Tzimiropoulos[†]

Maja Pantic^{†,*}

*EEMCS
University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

PanticM@cs.utwente.nl

Abstract

Multi-scale/orientation local image analysis methods are valuable tools for obtaining highly distinctive image-based representations. Very often, these features are generated from the responses of a bank of linear filters corresponding to different scales and orientations. Naturally, as the number of filters increases, so does the feature dimensionality. Further processing is often feasible only when dimensionality reduction is performed by subspace learning techniques, such as Principal Component analysis (PCA) or Linear Discriminant Analysis (LDA). The major problem stems from the fact that as the number of features increases, so does the computational complexity of these methods which, in turn, limits the number of scales and orientations examined. In this paper, we show how linear subspace analysis on features generated by the response of linear filter banks can be efficiently re-formulated such that complexity does not depend on the number of filters used. We describe computationally efficient and exact versions of PCA while the extension to other subspace learning algorithms is straightforward. Finally, we show how the proposed methods can boost the performance of algorithms for appearance based tracking.

1. Introduction

A significant amount of research in computer vision has revolved around providing efficient solutions to the following problem: given samples of a high-dimensional space estimate a low-dimensional space which preserves the intrinsic structure of the input data. The assumption that input data points are actually samples from a low-dimensional manifold embedded in a high-dimensional space is not unreasonable since large amounts of collected visual data of-

ten result from changes in very few degrees of freedom. This, in turn, attributes input data with a well-defined and probably predictable structure. Subspace learning methods perform dimensionality reduction with the goal of finding this underlying structure.

The necessity for dimensionality reduction in computer vision becomes more evident if we consider standard feature extraction techniques based on multi-scale/orientation local image analysis. Typically, the application of such methods generates highly redundant representations which, in turn, results in a further increase in dimensionality. As an example, due to their excellent localization properties in space and frequency as well as orientation selectivity properties, Gabor filters have become one of the defacto choices for feature extraction in many applications such as face recognition, action and gait recognition as well as facial expression analysis [13, 3, 20, 10, 19, 23, 24, 5, 7, 22, 1]. Applying a set of 40 Gabor filters to 64×64 images, as for example suggested in [9, 11, 12], results in a feature space of dimensionality equal to 163840. Therefore, further processing is feasible only after dimensionality reduction has been performed.

Nevertheless, the application of dimensionality reduction techniques poses serious computational concerns. Popular subspace learning methods, such as PCA or LDA and their kernel-based counterparts (Kernel PCA and Kernel FDA [9, 10]), require operations such as matrix multiplications and eigen-analysis of covariance matrices which not only require a huge amount of memory but also become computationally intractable for very high dimensional spaces. Thus, practical approaches either perform feature extraction in region of interests only or decrease the number of features using approximations such as downsampling [9, 11, 12].

A recent work [1] has shown how a bank of linear fil-

ters can be used to manipulate the margin in linear support vector machines. Inspired by [1], we show how linear subspace analysis on features generated by the response of linear filter banks can be efficiently re-formulated such that complexity does not depend on the number of filters used. Our approach touches upon classical results from linear algebra previously used for the efficient implementation of PCA and the efficient calculation of inner products in the Fourier domain [15]. We show how these results can be used in order to formulate computationally efficient and exact versions of PCA. Finally, we show how the proposed methods can boost the performance of algorithms for appearance based face tracking.

Summarizing, the contributions of this paper are

- An exact complex PCA of arbitrarily many linear filter responses. Our complex formulation applies PCA to both the magnitude and the phase of the filter responses. On the contrary, previously proposed algorithms do not scale well with the number of filters, are inexact and make use of magnitude information only [3, 20, 9, 11, 12, 10]. Note that recent papers have shown that the phase of linear filter responses conveys useful information for recognition [18, 25].
- A general framework for incremental PCA of arbitrarily many linear filter responses. We show how to apply this framework in order to formulate an appearance based tracking algorithm using arbitrarily many linear filter responses [17].

2. Principal Component Analysis

We assume that we are given a population of N images $\{f_1, \dots, f_N\}$ with $f_i(\mathbf{n}) : \mathcal{D} \subseteq \mathcal{Z}^2 \rightarrow \mathfrak{R}$, where \mathcal{Z} and \mathfrak{R} are the sets of integers and real numbers respectively. We denote by \mathbf{f}_i be the F -dimensional vector obtained by writing image f_i in lexicographic ordering and by $\mathbf{F} = [\mathbf{f}_1 | \dots | \mathbf{f}_N] \in \mathfrak{R}^{F \times N}$ the data matrix. Let us also denote by $\bar{\mathbf{f}} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}_i$ and $\bar{\mathbf{F}}$ the sample mean and the centralized data matrix, respectively. PCA finds a set of $p < F$ (usually, $p \ll F$) bases $\mathbf{U} = [\mathbf{u}_1 | \dots | \mathbf{u}_p] \in \mathfrak{R}^{F \times p}$ by solving the following optimization problem

$$\begin{aligned} \mathbf{U}_o = \arg \max_{\mathbf{U}} \text{tr} [\mathbf{U}^T \mathbf{S}^f \mathbf{U}] \\ \text{subject to } \mathbf{U}^T \mathbf{U} = \mathbf{I} \end{aligned} \quad (1)$$

where $\text{tr}[\mathbf{A}]$ is the trace of matrix \mathbf{A} and \mathbf{S}^f is the covariance matrix $\mathbf{S}^f = \bar{\mathbf{F}} \bar{\mathbf{F}}^T$. The solution is given by the eigenvectors corresponding to the p largest eigenvalues obtained from the eigen-decomposition of \mathbf{S}^f . The low dimensional representation of $\bar{\mathbf{F}}$ is given by $\mathbf{C} = \mathbf{U}^T \bar{\mathbf{F}}$.

3. PCA Of Arbitrarily Many Linear Filter Responses (AMFR-PCA)

We denote by $\{h_1(\mathbf{n}), \dots, h_m(\mathbf{n})\}$ a set of m filters with $h_l(\mathbf{n}) : \mathcal{D} \rightarrow \mathbb{C}$ where \mathbb{C} is the set of complex numbers. Using this filter bank, we build a set of responses $\{g_1(\mathbf{n}), \dots, g_m(\mathbf{n})\}$

$$g_l(\mathbf{n}) = f(\mathbf{n}) \star h_l(\mathbf{n}), \quad l = 1, \dots, m \quad (2)$$

where \star is the convolution operation. Let \mathbf{g}_l be the lexicographic ordering of $g_l(\mathbf{n})$ and $\mathbf{x} = [\mathbf{g}_1^T \dots \mathbf{g}_m^T]^T \in \mathbb{C}^{mF}$ the vector that contains the responses of the m filters. The total scatter matrix is $\mathbf{S}^x = \bar{\mathbf{X}} \bar{\mathbf{X}}^H$ where $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1 | \dots | \bar{\mathbf{x}}_N]$ and $\bar{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$. Finally, the PCA bases are obtained from the eigen-analysis of \mathbf{S}^x . This has a cost $O(m^3 F^3)$ which, as the number of filters increases, becomes prohibitive.

To obtain an efficient algorithm, we start by considering the efficient implementation of PCA ($O(N^3)$ instead of $O(m^3 F^3)$) proposed in [21]. Rather than computing the eigen-analysis of $\bar{\mathbf{X}} \bar{\mathbf{X}}^H$, we compute the eigen-analysis of $\mathbf{X}^H \mathbf{X}$ and make use of the following theorem [4]²

Theorem I. Define matrices \mathbf{A} and \mathbf{B} such that $\mathbf{A} = \mathbf{\Gamma} \mathbf{\Gamma}^H$ and $\mathbf{B} = \mathbf{\Gamma}^H \mathbf{\Gamma}$ with $\mathbf{\Gamma} \in \mathbb{C}^{m \times r}$. Let \mathbf{U}_A and \mathbf{U}_B be the eigenvectors corresponding to the non-zero eigenvalues Λ_A and Λ_B of \mathbf{A} and \mathbf{B} , respectively. Then, $\Lambda_A = \Lambda_B$ and $\mathbf{U}_A = \mathbf{\Gamma} \mathbf{U}_B \Lambda_A^{-\frac{1}{2}}$.

The application of Theorem I requires the computation of $\bar{\mathbf{X}}^H \bar{\mathbf{X}}$ which has a cost $O(mFN)$. To reduce this to $O(NF)$, we make use of the properties of the Fourier transform [15, 18, 1] which enables the computation of inner products in the Fourier domain. Let us first denote by $\mathbf{D} \in \mathbb{C}^{F \times F}$ the matrix of the Discrete Fourier Transform (DFT) [18]. Since \mathbf{D} is a unitary, we have $\mathbf{D}^{-1} = \mathbf{D}^H$ and we can write (2) as

$$\mathbf{g}_l = \mathbf{D}^{-1} (\mathbf{D} \mathbf{f} \odot \mathbf{D} \mathbf{h}_l) = \mathbf{D}^H (\mathbf{H}_l \mathbf{D} \mathbf{f}) \quad (3)$$

where \mathbf{h}_l is the lexicographic ordering of filter h_{σ_l} , \mathbf{H}_l is an $F \times F$ diagonal matrix with diagonal elements $\mathbf{D} \mathbf{h}_l$ and \odot is the Hadamard product. Using the above we have (based on the Parseval's theorem, the authors in [1] derived a very similar result)

$$\begin{aligned} \bar{\mathbf{x}}_i^H \bar{\mathbf{x}}_j &= (\mathbf{x}_i - \bar{\mathbf{x}})^H (\mathbf{x}_j - \bar{\mathbf{x}}) \\ &= \sum_{l=1}^n (\mathbf{g}_{il} - \bar{\mathbf{g}}_l)^H (\mathbf{g}_{jl} - \bar{\mathbf{g}}_l) \\ &= \sum_{l=1}^n (\mathbf{D}^H (\mathbf{H}_l \mathbf{D} \mathbf{f}_i - \mathbf{H}_l \mathbf{D} \bar{\mathbf{f}}))^H \\ &\quad \times \mathbf{D}^H (\mathbf{H}_l \mathbf{D} \mathbf{f}_j - \mathbf{H}_l \mathbf{D} \bar{\mathbf{f}}) \\ &= (\mathbf{D} \mathbf{f}_i - \mathbf{D} \bar{\mathbf{f}})^H \sum_{l=1}^n \mathbf{H}_l^2 (\mathbf{D} \mathbf{f}_j - \mathbf{D} \bar{\mathbf{f}}) \\ &= (\mathbf{H}^{1/2} \mathbf{D} \bar{\mathbf{f}}_i)^H \mathbf{H}^{1/2} \mathbf{D} \bar{\mathbf{f}}_j \\ &= \bar{\mathbf{z}}_i^H \bar{\mathbf{z}}_j \end{aligned} \quad (4)$$

¹ T denotes matrix transposition and H denotes conjugate matrix transposition.

²The version of this theorem for real PCA has been used in [21].

where $\mathbf{H} = \sum_{l=1}^n \mathbf{H}_l^2$ and $\bar{\mathbf{z}}_i$ is given by

$$\bar{\mathbf{z}}_i = \mathbf{H}^{1/2} \mathbf{D} \bar{\mathbf{f}}_i. \quad (5)$$

The vectors $\bar{\mathbf{z}}_i$ can be computed very efficiently from the Hadamard product of the diagonal of $\mathbf{H}^{1/2}$ and the 2D DFT of f_i which can be computed fast using the FFT (for example for square images the FFT has a cost $O(F \log(\sqrt{F}))$). This product does not depend on the number of the filters while \mathbf{H} can be pre-computed. Finally, if we write $\bar{\mathbf{Z}} = [\bar{\mathbf{z}}_1 | \dots | \bar{\mathbf{z}}_N]$, then, from (4), we have $\bar{\mathbf{X}}^H \bar{\mathbf{X}} = \bar{\mathbf{Z}}^H \bar{\mathbf{Z}}$ whose computation has a cost $O(FN)$.

Let us denote by \mathbf{V}_p the eigenvectors corresponding to the p largest eigenvalues Λ_p obtained from the eigen-decomposition of $\bar{\mathbf{Z}}^H \bar{\mathbf{Z}}$. Then, according to Theorem I, the PCA bases are given by $\mathbf{U}_p = \bar{\mathbf{X}} \mathbf{V}_p \Lambda_p^{-\frac{1}{2}}$. As we may easily observe, this calculation is still very computationally expensive. Note however that, in order to find the low dimensional embedding, the explicit computation of \mathbf{U}_p is not necessary. Let us denote by f_y a test image and by $\bar{\mathbf{y}}$ the concatenation of the filter responses after the subtraction of the mean $\bar{\mathbf{x}}$. To compute its low-dimensional representation $\mathbf{c}_p \in \mathbb{C}^p$ efficiently, we expand

$$\begin{aligned} \mathbf{c}_p &= \mathbf{U}_p^H \mathbf{y} \\ &= \Lambda_p^{-1/2} \mathbf{V}_p^H \bar{\mathbf{X}}^H \bar{\mathbf{y}} \\ &= \Lambda_p^{-1/2} \mathbf{V}_p^H [\bar{\mathbf{x}}_1^H \bar{\mathbf{y}}, \dots, \bar{\mathbf{x}}_N^H \bar{\mathbf{y}}]^T \\ &= \Lambda_p^{-1/2} \mathbf{V}_p^H [\bar{\mathbf{z}}_1^H \bar{\mathbf{z}}_y, \dots, \bar{\mathbf{z}}_N^H \bar{\mathbf{z}}_y]^T \\ &= \Lambda_p^{-1/2} \mathbf{V}_p^H \bar{\mathbf{Z}}^H \bar{\mathbf{z}}_y \end{aligned} \quad (6)$$

where $\bar{\mathbf{z}}_y = \mathbf{H}^{1/2} \mathbf{D} \bar{\mathbf{y}}$. Finally, the reconstruction error $\epsilon^x(\mathbf{U}_p)$ is given by

$$\begin{aligned} \epsilon^x(\mathbf{U}_p) &= \|\bar{\mathbf{X}} - \mathbf{U}_p \mathbf{U}_p^H \bar{\mathbf{X}}\|_F^2 \\ &= \text{tr}[(\bar{\mathbf{X}} - \mathbf{U}_p \mathbf{U}_p^H \bar{\mathbf{X}})^H (\bar{\mathbf{X}} - \mathbf{U}_p \mathbf{U}_p^H \bar{\mathbf{X}})] \\ &= \text{tr}[\bar{\mathbf{X}}^H \bar{\mathbf{X}} - \bar{\mathbf{X}}^H \mathbf{U}_p \mathbf{U}_p^H \bar{\mathbf{X}}] \\ &= \text{tr}[\bar{\mathbf{Z}}^H \bar{\mathbf{Z}} - \bar{\mathbf{Z}}^H \mathbf{Q}_p \mathbf{Q}_p^H \bar{\mathbf{Z}}] \\ &= \|\bar{\mathbf{Z}} - \mathbf{Q}_p \mathbf{Q}_p^H \bar{\mathbf{Z}}\|_F^2 \\ &= \epsilon^z(\mathbf{Q}_p) \end{aligned} \quad (7)$$

where $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{Q}_p = \bar{\mathbf{Z}} \mathbf{V}_p \Lambda_p^{-1/2}$.

Algorithm 1 summarizes the steps of our efficient algorithm. Note that steps 1-3 are computed only once.

4. Experimental Results

We evaluated the performance of our subspace learning algorithms for the applications of appearance based face tracking. We used multi-scale/orientation Gabor filters and multi-scale derivatives of the Gaussian. In the spatial domain (x, y) , a Gabor filter is a complex exponential modu-

Algorithm 1 AMLFR-PCA

Inputs: A set of N images $f_i, i = 1, \dots, n$, of F pixels, the matrix $\mathbf{H} = \sum_{l=1}^n \mathbf{H}_l^2$, the number p of principal components and a test image f_y of F pixels.

Step 1. Compute $\bar{\mathbf{z}}_i = \mathbf{H}^{1/2} \mathbf{D} \bar{\mathbf{f}}_i$, form the matrix of the transformed data $\bar{\mathbf{Z}} = [\bar{\mathbf{z}}_1 | \dots | \bar{\mathbf{z}}_N] \in \mathbb{C}^{F \times n}$ and compute the matrix $\mathbf{K} = \bar{\mathbf{Z}}^H \bar{\mathbf{Z}} \in \mathbb{C}^{n \times n}$.

Step 3. Compute the eigen-decomposition $\mathbf{K} = \mathbf{V} \Lambda \mathbf{V}^T$ and denote by $\mathbf{V}_p \in \mathbb{R}^{n \times p}$ and $\Lambda_p \in \mathbb{R}^{p \times p}$ the p -reduced set.

Step 4. Compute $\mathbf{z}_y = \mathbf{H}^{1/2} \mathbf{D} \bar{\mathbf{y}}$.

Step 5. Compute the embedding $\mathbf{c}_p = \Lambda_p^{-1/2} \mathbf{V}_p^H \bar{\mathbf{Z}}^H \bar{\mathbf{z}}_y$.

lated by a Gaussian function defined as

$$\begin{aligned} \psi_{\omega, \theta}(x, y) &= \frac{1}{2\pi\sigma^2} e^{-\frac{(x \cos \theta + y \sin \theta)^2 + (-x \sin \theta + y \cos \theta)^2}{2\sigma^2}} \\ &\cdot \left[e^{i\omega(x \cos \theta + y \sin \theta)} - e^{-\frac{\omega^2 \sigma^2}{2}} \right] \end{aligned} \quad (8)$$

where ω is the radial center frequency of the complex exponential, θ is the orientation of the Gabor wavelet, and σ is the standard deviation of the Gaussian function. The first derivative Gaussian filter bank is defined as [16, 6]

$$\psi_{\sigma}(x, y) = -\sigma(x - y\mathbf{i})e^{-\sigma(x^2 + y^2)}. \quad (9)$$

where σ is the scale of the filter. In all experiments, we used Gabor filters with 16 scales and 16 orientations and Gaussian filters of 16 scales.

The appearance-based approach to tracking has been one of the de facto choices for tracking objects in image sequences. Prominent examples of such an approach include subspace-based techniques [2, 14, 17]. Here, we propose a subspace-based tracking algorithm closely related to the incremental visual tracker in [17]. As such, our tracker can deal with drastic appearance changes, does not require off-line training, continually updates a compact object representation and uses the Condensation algorithm to robustly estimate the object's location [17].

Similarly to [17], the proposed tracker is essentially an eigen-tracker [2], where the eigen-space is adaptively learned and updated online. The key element which makes our approach equally fast but significantly more robust, is how the eigen-space is generated. The method in [17] uses the incremental version of standard intensity-based PCA. On the contrary, the proposed tracker is based on the eigen-space generated by the responses of a bank of Gabor and Gaussian filters. In the following, we show how this subspace can be learned and updated using an approach which is as computationally efficient as the incremental version of intensity-based PCA.

Let us assume that given N images $\{f_1, \dots, f_N\}$, we have already computed and stored the data matrix $\bar{\mathbf{Z}}_N$,

Algorithm 2 INCREMENTAL AMLFR-PCA

Inputs: The \mathbf{Q}_p and Σ_p from SVD of $\bar{\mathbf{Z}}_N$ (\mathbf{Q}_p and Σ_p can be calculated by the application of Algorithm 1 since \mathbf{Q}_p is the principal subspace and Σ_p is $\Lambda_p^{1/2}$) and mean vector $\bar{\mathbf{z}}_N$, a set of new images $\{f_{N+1}, \dots, f_{N+M}\}$ and the number p of principal components

Step 1. From set $\{f_{N+1}, \dots, f_{N+M}\}$ compute the matrix of the transformed data $\mathbf{Z}_M = [\mathbf{z}_{N+1} | \dots | \mathbf{z}_{N+M}]$ and the mean $\bar{\mathbf{z}}_M$.

Step 3. Compute the new mean vector $\bar{\mathbf{z}}_{N+M} = \frac{N}{N+M}\bar{\mathbf{z}}_N + \frac{M}{N+M}\bar{\mathbf{z}}_M$ and form the matrix

$$\mathbf{G} = \left[(\mathbf{z}_{N+1} - \bar{\mathbf{z}}_M) | \dots | (\mathbf{z}_{M+N} - \bar{\mathbf{z}}_M) \right] \sqrt{\frac{NM}{N+M}} (\bar{\mathbf{z}}_M - \bar{\mathbf{z}}_N)$$

Step 4. Compute $\tilde{\mathbf{G}} = \text{orth}(\mathbf{G} - \mathbf{Q}_p \mathbf{Q}_p^H \mathbf{G})$ and

$$\mathbf{R} = \begin{bmatrix} \Sigma_p & \mathbf{Q}_p^H \mathbf{G} \\ \mathbf{0} & \tilde{\mathbf{G}}^H (\mathbf{G} - \mathbf{Q}_p \mathbf{Q}_p^H \mathbf{G}) \end{bmatrix} \quad (\text{where orth performs orthogonalization})$$

Step 5. Compute $\mathbf{R} \stackrel{svd}{=} \tilde{\mathbf{U}} \Sigma^{N+M} \tilde{\mathbf{V}}^H$ (where Σ^{N+M} are new singular values).

Step 6. Compute the new principal subspace $\mathbf{U}^{N+M} = [\mathbf{U}_p \tilde{\mathbf{G}}] \tilde{\mathbf{U}}$.

the mean vector $\bar{\mathbf{z}}_N$, the eigenvalues Λ_p and the matrix $\mathbf{Q}_p = \bar{\mathbf{Z}}_N \mathbf{V}_p \Lambda_p^{-1/2}$, where we have made the dependency of the quantities on N explicit. Then, given a new image set $\{f_{N+1}, \dots, f_{N+M}\}$, our target is to obtain \mathbf{U}^{N+M} , Λ^{M+N} and $\bar{\mathbf{z}}_{N+M}$ corresponding to $\{f_1, \dots, f_{N+M}\}$ without re-applying Algorithm 1. Based on our analysis in Section 3 and the derivations in [8, 17], Algorithm 2 summarizes the steps of our efficient incremental PCA of arbitrarily many linear filter responses (Incremental AMLFR-PCA).

Finally, the proposed tracker combines our incremental AMLFR-PCA with a variant of the Condensation algorithm for dynamical estimation of the object's location [17].

To evaluate the performance of our tracker quantitatively, we used the ground truth provided for the position of seven facial fiducial points for the popular Dudek video sequence³. We used the proposed tracker as well as the original implementation in [17] to track a face region manually defined by a rectangle in the first frame. We then used the tracked region to estimate an affine transformation for all subsequent frames and used this transformation in order to obtain an estimate of the new position of the fiducial points. Finally, we evaluated the tracking accuracy from the mean value of the root mean square (RMS) error between the locations of the points as given by the ground truth and the estimated transformation. Table 1 summarizes the obtained results, while Fig. 1 shows representative examples illus-

³The Dudek video sequence with annotations is available from: <http://www.cs.toronto.edu/~dross/ivt/>

trating the tracking process for the case of the Gaussian filter bank. As we may observe, the proposed tracker performs significantly better at no further computational cost.

	Intensity	Gabor	Gaussian
RMS ERROR	7.44	7.12	6.32

Table 1. Mean RMS Error on Dudek Video Sequence.

5. Conclusion

We proposed an exact and very efficient framework for subspace analysis of features generated by arbitrarily many linear filter responses such as Gabor and Gaussian filter banks. Previously proposed methods [20, 9, 11, 12, 10] are based on approximations and do not scale well with the number of filters. Thus, compared to previous work, our formulation is not only exact but also significantly faster. We showed how the proposed framework can boost the performance of appearance based algorithms for tracking.

6. Acknowledgements

This work has been funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB).

References

- [1] A. Ashraf, S. Lucey, and T. Chen. Re-interpreting the application of gabor filters as a manipulation of the margin in linear support vector machines. *IEEE transactions on pattern analysis and machine intelligence*, pages 1335–1341, 2010. 1, 2
- [2] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998. 3
- [3] B. Duc, S. Fischer, and J. Bigun. Face authentication with Gabor information on deformable graphs. *IEEE Transactions on Image Processing*, 8(4):504–516, 1999. 1, 2
- [4] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins Univ Pr, 1996. 2
- [5] G. Goudelis, S. Zafeiriou, A. Tefas, and I. Pitas. Class-Specific Kernel-Discriminant Analysis for Face Verification. *IEEE Transactions on Information Forensics and Security*, 2(3 Part 2):570–587, 2007. 1
- [6] J. Koenderink and A. Van Doorn. Receptive field families. *Biological cybernetics*, 63(4):291–297, 1990. 3
- [7] I. Kotsia, I. Buciu, and I. Pitas. An analysis of facial expression recognition under partial facial image occlusion. *Image and Vision Computing*, 26(7):1052–1067, 2008. 1
- [8] A. Levey and M. Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9:1371–1374, 2000. 4
- [9] C. Liu. Gabor-based kernel PCA with fractional power polynomial models for face recognition. *IEEE Transactions on*

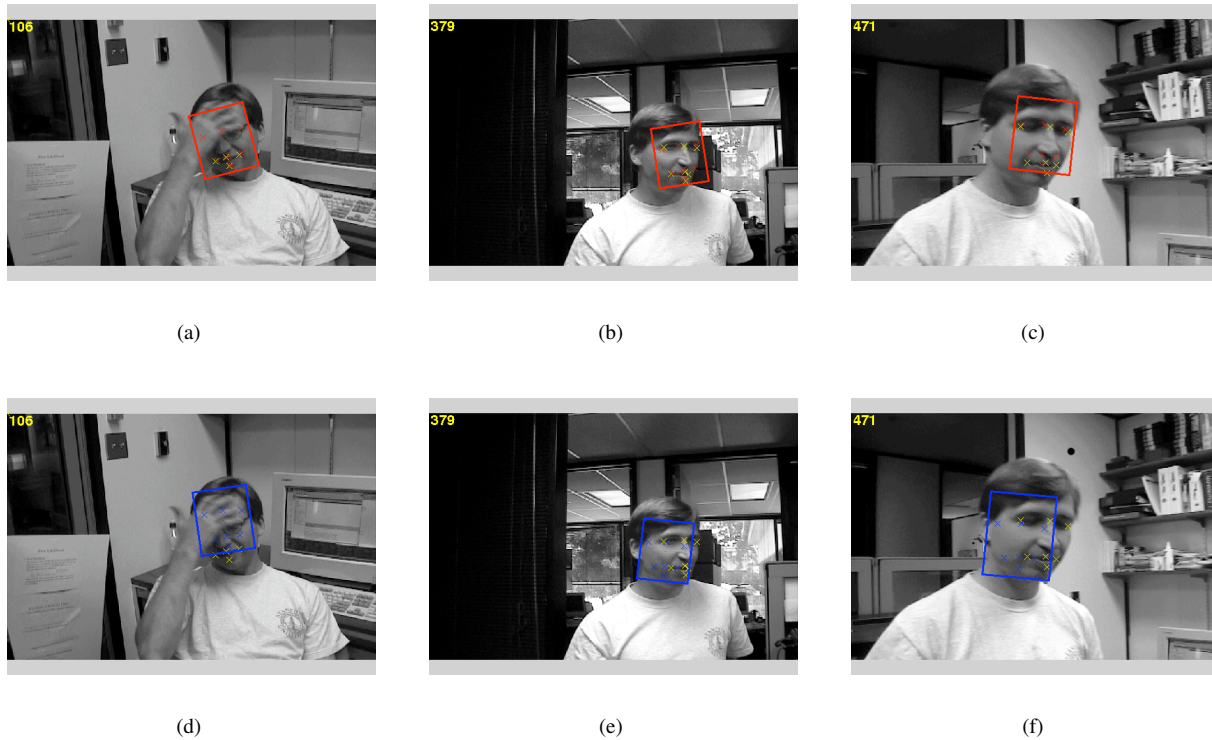


Figure 1. Appearance-based tracking using incremental PCA. (a)-(c) Tracking using a bank of multi-scale Gaussian filters. (d)-(f) Tracking using pixel intensities.

- Pattern Analysis and Machine Intelligence*, 26(5):572–581, May 2004. 1, 2, 4
- [10] C. Liu. Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. *IEEE transactions on pattern analysis and machine intelligence*, pages 725–737, 2006. 1, 2, 4
- [11] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476, Apr. 2002. 1, 2, 4
- [12] C. Liu and H. Wechsler. Independent component analysis of gabor features for face recognition. *IEEE Transactions on Neural Networks*, 14(4):919–928, July 2003. 1, 2, 4
- [13] M. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1357–1362, 2002. 1
- [14] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. 3
- [15] A. Oppenheim and R. Schaffer. Discrete-time signal processing. *Prentice Hall Signal Processing*, page 1120, 2009. 2
- [16] M. Reisert and H. Burkhardt. Complex derivative filters. *Image Processing, IEEE Transactions on*, 17(12):2265–2274, 2008. 3
- [17] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1):125–141, 2008. 2, 3, 4
- [18] M. Savvides, B. Kumar, and P. Khosla. Eigenphases vs eigenfaces. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004 (ICPR 2004)*, volume 3, 2004. 2
- [19] D. Tao, X. Li, X. Wu, and S. Maybank. General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1700–1715, 2007. 1
- [20] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2002. 1, 2, 4
- [21] M. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 2
- [22] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan. Toward practical smile detection. *IEEE transactions on pattern analysis and machine intelligence*, pages 2106–2111, 2009. 1
- [23] S. Zafeiriou and I. Pitas. Discriminant graph structures for facial expression recognition. *IEEE Transactions on Multimedia*, 10(8):1528–1540, 2008. 1
- [24] S. Zafeiriou, A. Tefas, and I. Pitas. Minimum class variance support vector machines. *Image Processing, IEEE Transactions on*, 16(10):2551–2564, 2007. 1
- [25] B. Zhang, S. Shan, X. Chen, and W. Gao. Histogram of Gabor phase patterns (HGPP): a novel object representation

approach for face recognition. *IEEE Transactions on Image Processing*, 16(1):57–68, 2006. [2](#)