Dynamic Coupled Component Analysis

Fernando De la Torre* **Michael J. Black†**

*Departament de Comunicacions i Teoria del Senyal, Escola d'Enginyeria la Salle,
Universitat Ramon LLull, Barcelona 08022, Spain. ftorre@salleURL.edu

†Department of Computer Science, Brown University, Box 1910,
Providence, RI 02912, USA. black@cs.brown.edu

Abstract

We present a method for simultaneously *learning linear models of multiple high dimensional data sets and the dependencies between them. For example, we learn asymmetrically coupled linear models for the faces of two different people and show how these models can be used to animate one face given a video sequence of the other. We pose the problem as a form of Asymmetric Coupled Component Analysis (ACCA) in which we simultaneously learn the subspaces for reducing the dimensionality of each dataset while coupling the parameters of the low dimensional representations. Additionally, a dynamic form of ACCA is proposed, that extends this work to model temporal dependencies in the data sets. To account for outliers and missing data, we formulate the problem in a statistically robust estimation framework. We review connections with previous work and illustrate the method with examples of synthesized dancing and the animation of facial avatars.*

1 Introduction

Learning low-dimensional linear models from high dimensional training data has proven useful in computer vision for solving problems such as recognition and tracking. In particular, Principal Component Analysis (PCA) is one of the primary techniques used to construct these linear models. PCA finds the linear subspace of maximum variation within a data set. However there exist problems in computer vision where it is important to exploit correlations, linear relationships, or non-linear dependencies between multiple data sets. Figure (1), for example, shows two people who have different facial appearance, however they share some hidden variable which captures their expression. Many problems of this form appear in computer vision and

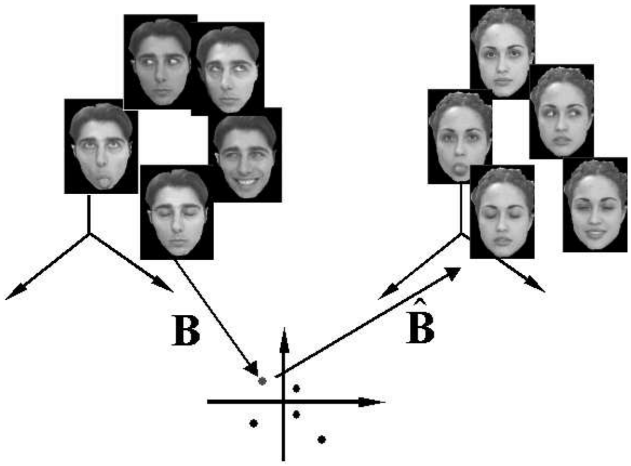


Figure 1: Two people (two data sets) showing different facial expressions in a high dimensional space coupled in a hidden space (see text).

modeling the dependencies in the high-dimensional space of images is impractical. In such a situation, we would like an algorithm that both reduces the dimensionality of the data sets while modeling and preserving the coupling between them. This coupling can take a number of forms as described below.

Given the high dimensional nature of images, modeling non-linear dependencies in the image space is often infeasible due either to limited training data or computational complexity. For high dimensional data, dimensionality reduction is often necessary. One approach is to independently learn low-dimensional linear models for each data set using PCA. The coupling between the coefficients of the linear approximations to the training data can be modeled using a neural network or other supervised learning technique. Applying PCA separately to each set preserves the directions of maximum variance within the sets but these do not necessarily correspond to the directions of maximum covariation between sets (or higher order generalizations). That is, independently learning the low-dimensional models may result in a loss of important detail relevant to the coupling between data sets.

Another common approach is to jointly model the data with a single linear subspace [4, 5]. This is done by performing PCA on augmented data vectors containing corresponding data from multiple data sets. Typically, the data in each training set is first normalized so that the variance of each set is similar (although many variations are possible). Once this joint model has been learned, we can use it to make linear predictions of one set given the other. Although this approach is the optimal linear solution for the joint representation, it is not necessarily optimal when we need asymmetric prediction; that is, predicting one data set from the other.

The purpose of this paper, is to describe an Asymmetric Coupled Component Analysis (ACCA) method for learning dependencies between high dimensional data sets in the hidden parameter space rather than the observation space. There are three main contributions. First, we formulate ACCA in such a way that the hidden coefficients are made explicit. This differs from and generalizes previous work in that it allows us to impose constraints on the coupling. Second, we formulate ACCA in a robust statistical estimation framework that improves resistance to outliers. This approach exploits an energy minimization framework that allows further generalizations of the method as discussed in the conclusions. Finally, by making the coupling coefficients explicit, we can extend the energy minimization approach to account for temporal dependencies in dynamic data sets. The approach complements recent work on robust PCA and robust Singular Value Decomposition (SVD) [7].

We illustrate the method by learning a coupled model of the faces in Fig. 1. This model can be used to animate one face using an image sequence of the other. Also, we illustrate the results by learning a coupled, dynamic model of two people swing dancing. Then given a new sequence of one of the dancers' motions, we can approximate the appropriate motion of their partner.

2 Previous Work

This paper is related to previous work on symmetric and asymmetric PCA. It is beyond the scope of the paper review all possible applications of PCA, therefore we just briefly describe the theory and point to related work for further information.

Let $\mathbf{D} = [\mathbf{d}_1 \ \mathbf{d}_2 \ ... \ \mathbf{d}_n] = [\mathbf{d}^1 \ \mathbf{d}^2 \ ... \ \mathbf{d}^d]^T$ be a matrix $\mathbf{D} \in \Re^{d \times n}$ [1], where each column $\mathbf{d}_i$ is a data sample (or image), n is the number of training images, and d is the number of pixels (variables) in each image. We assume that $\mathbf{D}$ is zero mean and then generalize to non-zero mean data later in the paper. Let the first k principal components of $\mathbf{D}$ be $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_k] \in \Re^{d \times k}$. The columns of $\mathbf{B}$ span the subspace of maximum variation of the data $\mathbf{D}$. Although a closed form solution for computing the principal components ($\mathbf{B}$) can be achieved by computing the k largest eigenvectors of the covariance matrix $\mathbf{D}\mathbf{D}^T$ [8], here it is useful to exploit work that formulates PCA as the minimization of an energy function [7, 8, 9, 11, 21, 25, 27]. Related formulations have been studied in various communities: machine learning [21, 25], statistics [9, 11], neural networks [8] and computer vision [7, 27]. In spirit, all these approaches essentially minimize the following energy function (although with different noise models, deterministic or Bayesian frameworks, or different metrics for the error):

$$E_{pca}(\mathbf{B}, \mathbf{C}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{c}_i||_2^2 \qquad (1)$$

where $||.||_2$ denotes the L_2 norm and $\mathbf{C} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \cdots \mathbf{c}_n]$ where each $\mathbf{c}_i$ is a vector of coefficients used to reconstruct the data vector $\mathbf{d}_i$.

If the effective rank of $\mathbf{D}$ is much less than d, we can approximate the column space of $\mathbf{D}$ with $k << d$ principal components. The data $\mathbf{d}_i$ can be approximated as a linear combination of the principal components as $\mathbf{d}_i^{rec} = \mathbf{B}\mathbf{B}^T\mathbf{d}_i$ where $\mathbf{c}_i = \mathbf{B}^T\mathbf{d}_i$ are the linear coefficients obtained by projecting the training data onto the principal subspace ($\mathbf{B}$ is an orthogonal matrix); substituting them in (1) gives as

$$E_{pca}(\mathbf{B}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{B}^T\mathbf{d}_i||_2^2. \qquad (2)$$

This formulation is common in the neural network community [8, 27].

Many methods exist for minimizing (1) (Alternated Least Squares (ALS), criss-cross regression [11] or Expectation-Maximization (EM) [21, 25]), but in the case of PCA, share the same basic philosophy. These algorithms alternate between solving for the coefficients $\mathbf{C}$ with the bases $\mathbf{B}$ fixed and then solving for

[1] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the j-th column of the matrix $\mathbf{D}$ and $\mathbf{d}^j$ is a column vector representing the j-th *row* of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in row i and column j of the matrix $\mathbf{D}$ and the scalar i-th element of a column vector $\mathbf{d}_j$. d_{ji} is the i-th scalar element of the vector $\mathbf{d}^j$. All non-bold letters represent scalar variables. $\circ$ represents the Hadamard (point wise) product. $||\mathbf{d}||_{\mathbf{W}}$ denotes the weighted L_2 norm of the vector $\mathbf{d}$, that is $\mathbf{d}^T\mathbf{W}\mathbf{d}$.

the bases $\mathbf{B}$ with $\mathbf{C}$ fixed. Typically, both updates are computed by solving a linear system of equations.

On the other hand, when two (or more) related data sets $\hat{\mathbf{D}} \in \Re^{d_1 \times n}$ and $\mathbf{D} \in \Re^{d_2 \times n}$ with an equal number of observations are available, a natural question which arises is how to exploit the correlations between them (e.g. to estimate one set from the other). An example of this is the modeling of both the shape and appearance of human faces [4, 5]. In many cases, the dimensionality, complexity, and energy of the data sets is different. These issues in conjunction with high dimensional data sets present a number of challenges.

A common approach is to construct a joint model by concatenating the two data sets into a new matrix $\ddot{\mathbf{D}} = \left[\begin{smallmatrix}\mathbf{D}\\ \hat{\mathbf{D}}\end{smallmatrix}\right]$ and perform PCA in the augmented matrix $\ddot{\mathbf{D}}$ (several variations of this technique are possible [4, 5]). With appropriate normalization of the data sets, this approach could be sufficient for jointly representing the data for reconstruction or recognition. For prediction of one data set from the other (the asymmetric case), this approach is not optimal. At this point, observe that the subspace found by joint PCA (or joint Singular Value Decomposition, SVD) will be obtained by minimizing the following energy function, $E_{svd}(\mathbf{B}, \hat{\mathbf{B}}, \mathbf{C})$:

$$E_{svd} = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{c}_i||_2^2 + \sum_{i=1}^{n} ||\hat{\mathbf{d}}_i - \hat{\mathbf{B}}\mathbf{c}_i||_2^2 \qquad (3)$$

where the coefficients are shared but the bases are specialized to the different data sets. Note that this approach constrains the model such that the maximum possible rank of the individual subspaces is the same.

An alternative approach is to reduce the dimensionality of each set independently and then learn the relationship between the coefficients of each data set using some non-linear fitting methods [20]. The problem with this approach is that information discarded in the residual subspaces (the $n - k$ smallest eigenvectors) of each data set may provide significant information about the relationship between the two sets.

In contrast, the purpose of this paper is to explore the use of linear models for learning relations between two (or more) given data sets while coupling the coefficients in various ways. This problem has been studied in the signal processing community [14, 22] and neural network community [8] and it is known as reduced rank Wiener filtering [22] or Asymmetric PCA (APCA) [8]. These can be formulated as the minimization of

$$E(\mathbf{B}, \hat{\mathbf{B}}) = \sum_{i=1}^{n} ||\hat{\mathbf{d}}_i - \hat{\mathbf{B}}\mathbf{B}^T\mathbf{d}_i||_2^2. \qquad (4)$$

Observe that if $\hat{\mathbf{d}}_i = \mathbf{d}_i$ and $\mathbf{B} = \hat{\mathbf{B}}^T$ then minimizing (4) leads to the standard (symmetric) PCA [8]. Observe also, that APCA imposes a rank restriction on the mapping $\mathbf{B}\hat{\mathbf{B}}^T$ which is advantageous when working with high dimensional data such as images.

Closed form solutions to this problem typically assume that $\mathbf{D}$ has full rank and $\mathbf{D}\mathbf{D}^T$ is invertible [8, 22], which in the case of images is often not true due to the lack of training samples. It can also be solved with the generalized singular value decomposition of the matrices $\hat{\mathbf{D}}\mathbf{D}^T$ and $\mathbf{D}^T$. Again this can be impractical for high dimensional data as images.

In the following section we address many of these problems by extending this approach further and formulating it robustly. We refer to this method as ACCA and show how it can be extended to learn linear dynamics in the hidden parameter space of coefficients. The approach essentially regularizes temporal data and we refer to it as Dynamic ACCA (DACCA).

3 Beyond PCA

In this section, following previous work on asymmetric PCA [8] and reduced rank-Wiener filtering [22], we extend APCA to model linear dependencies between two data sets, $\mathbf{D}$ and $\hat{\mathbf{D}}$, in the hidden parameter space of coefficients. In order to take into account possible missing data and outliers which occur at a pixel level in $\hat{\mathbf{D}}$ (we refer to these as *intra-sample outliers* [7]), we formulate the problem in the context of robust statistics.

3.1 Robust Asymmetric CCA

Let $\hat{\mathbf{D}} \in \Re^{d_1 \times n}$ and $\mathbf{D} \in \Re^{d_2 \times n}$ be two multidimensional simultaneous observations of a particular event (e.g. two people dancing, multi-view images, shape and appearance, etc). ACCA should find the linear transformations $\hat{\mathbf{B}} \in \Re^{d_1 \times k_1}$ which reduce the dimensionality of $\hat{\mathbf{D}}$ and simultaneously find the linear transformation $\mathbf{B} \in \Re^{d_2 \times k_2}$ which makes $\mathbf{D}$ correlated with $\hat{\mathbf{D}}$ in the hidden parameter space. In the simplest case we will assume $k_1 = k_2$. To achieve this we first replace $\mathbf{B}^T\mathbf{d}_i$ in (2) with the explicit coefficient $\mathbf{c}_i$ and then impose a new constraint on these coefficients

$$E_{acca}(\mathbf{B}, \hat{\mathbf{B}}, \mathbf{C}) = \sum_{i=1}^{n} ||\hat{\mathbf{d}}_i - \hat{\mathbf{B}}\mathbf{c}_i||_2^2 + \lambda \sum_{i=1}^{n} ||\mathbf{c}_i - \mathbf{B}^T\mathbf{d}_i||_2^2. \quad (5)$$

Minimizing this gives the bases $\hat{\mathbf{B}}$ for reconstructing the column space of $\hat{\mathbf{D}}$ and the bases $\mathbf{B}$ on which we project the data set $\mathbf{D}$. The constant, λ, weights the importance of each term in the energy function and is related to the noise in both datasets (although not in a straightforward manner because the noise of $\mathbf{d}$ is filtered by the matrix $\mathbf{B}$).

After the model has been learned, estimating, or predicting, $\hat{\mathbf{d}}_i$ from $\mathbf{d}_i$ is simply done by projecting $\mathbf{d}_i$ on the the bases $(\mathbf{B}^T\mathbf{d}_i)$ and using the resulting coefficients to reconstruct the data using the bases $\hat{\mathbf{B}}$.

In contrast to the previous Wiener filter [14, 22] and asymmetric PCA [8], we have explicitly introduced the coefficients $\mathbf{C}$. Explicitly solving for the coefficients $\mathbf{C}$, allows an easy generalization to the dynamic case and can also permit the addition of prior information over the coefficients; for example, one could impose sparseness constraints [18]. Experimentally, we have observed that minimizing (5) gives faster convergence than minimizing (2) for the same initial conditions.

We further generalize (5) by removing the assumption that the data is zero mean and by replacing the L2 norm with a robust function. This robust formulation can account for possible pixel-level outlying data in $\hat{\mathbf{D}}$ (see [7] for the benefits of the robust formulation). This necessitates robust estimates of the means $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$. Consequently we minimize $E_{racca}(\mathbf{B}, \hat{\mathbf{B}}, \mathbf{C}, \boldsymbol{\mu}, \hat{\boldsymbol{\mu}}, \boldsymbol{\sigma}, \hat{\boldsymbol{\sigma}})$ where this is defined as

$$E_{racca} = \sum_{i=1}^{n}\sum_{p=1}^{d_1} \rho(\hat{d}_{pi} - \hat{\mu}_p - \sum_{j_1=1}^{k_1} \hat{b}_{pj_1} c_{j_1 i}, \hat{\sigma}_p)$$
$$+\lambda \sum_{i=1}^{n}\sum_{j_1=1}^{k_1} \rho(c_{j_1 i} - \sum_{j_2=1}^{d_2} b_{j_2 j_1}(d_{j_2 i} - \mu_{j_2}), \sigma_p). \tag{6}$$

The robust function, $\rho(x, \sigma_p)$, downweights outlier data at the pixel level and here is taken to be the Geman-McClure error function [12] given by $\rho(x, \sigma_p) = \frac{x^2}{x^2+\sigma_p^2}$, where σ_p is a parameter that controls the convexity of the robust function and is used for deterministic annealing in the optimization process [7].

To derive the learning algorithm, we formulate the robust M-estimation problem in (6) as an iteratively reweighted least-squares [16] optimization problem and minimize:

$$E_{wacca} = \sum_{i=1}^{n} \left(||\hat{\mathbf{e}}_i||_{\hat{\mathbf{W}}_i} + \lambda ||\mathbf{e}_i||_{\mathbf{W}_i}\right) \tag{7}$$
$$= \sum_{p=1}^{k_1} ||\hat{\mathbf{e}}^p||_{\hat{\mathbf{W}}^p} + \lambda \sum_{p=1}^{d_1} ||\mathbf{e}^p||_{\mathbf{W}^p} \tag{8}$$

where $\hat{\mathbf{e}}_i$ represents the columns of the data error matrix $\hat{\mathbf{E}} = \hat{\mathbf{D}} - \hat{\mathbf{B}}\mathbf{C} \in \Re^{d_1 \times n}$ and $\hat{\mathbf{e}}^p$ the rows of the same matrix. Similarly $\mathbf{e}_i$ and the matrix $\mathbf{E} \in \Re^{k_1 \times n}$ represent the error $\mathbf{C} - \mathbf{B}^T\mathbf{D}$. $\hat{\mathbf{W}} \in \Re^{d_1 \times n}$ is a real positive matrix containing the weights of each pixel of $\hat{\mathbf{D}}$. $\hat{\mathbf{W}}_i \in \Re^{d_1 \times d_1} = diag(\hat{\mathbf{w}}_i)$ is a diagonal matrix containing the weighting coefficients for the data sample $\mathbf{d}_i$, and $\hat{\mathbf{W}}^p \in \Re^{n \times n} = diag(\hat{\mathbf{w}}^p)$ are diagonal matrices containing the weighting factors for the p-th pixel over $\hat{\mathbf{D}}$. The matrix $\hat{\mathbf{W}}$ is calculated for each iteration as a function of the previous residuals $\hat{\mathbf{E}}$ and is related to the "influence" [13] of pixels on the solution. See [7] for more detailed information. $\mathbf{W}$ is constructed similarly for the second term in (7) and (8).

In the more general case (arbitrary weight matrices $\mathbf{W}$ and $\hat{\mathbf{W}}$), equations (7) or (8) do not have a closed form solution in terms of eigenvectors of a weighted covariance matrix. However, observe that the function E_{wacca} is convex in each of the parameters, but it is no longer convex as a joint function of these variables. Therefore, for solving the previous equations we use the Alternated Least-Squares technique which updates one parameter while the others are fixed. This technique will monotonically reduce the cost function E_{wacca}, although is not guaranteed to converge to the global minimum.

Differentiating E_{wacca} w.r.t. its parameters we obtain the necessary conditions for the minimum. The derivatives of E_{wacca} (7) w.r.t. $\boldsymbol{\mu}, \mathbf{c}_i$ and $\mathbf{b}^p$, give the following closed form updates:

$$\hat{\boldsymbol{\mu}} = (\textstyle\sum_{i=1}^{n} \hat{\mathbf{W}}_i)^{-1} \sum_{i=1}^{n} \hat{\mathbf{W}}_i(\hat{\mathbf{d}}_i - \hat{\mathbf{B}}\mathbf{c}_i) \tag{9}$$
$$(\mathbf{C}\hat{\mathbf{W}}^{j_1}\mathbf{C}^T)\hat{\mathbf{b}}^{j_1} = \mathbf{C}\hat{\mathbf{W}}^{j_1}(\hat{\mathbf{d}}^{j_1} - \hat{\mu}_d \mathbf{1}_n) \ \ \forall\ j_1 \tag{10}$$
$$(\hat{\mathbf{B}}^T\hat{\mathbf{W}}_i\hat{\mathbf{B}} + \lambda\mathbf{W}_i)\mathbf{c}_i = (\hat{\mathbf{B}}^T\hat{\mathbf{W}}_i\hat{\mathbf{d}}_i + \lambda\mathbf{W}_i\mathbf{B}^T\mathbf{d}_i)\forall i \tag{11}$$

where $i = 1 \ldots n$ and $j_1 = 1 \ldots d_1$. Equation (9) and (10) are uncoupled equations between data sets. However the optimal coefficients $\mathbf{c}_i$ (11) are a weighted combination of the information of the two sets.

Similarly, differentiating (7) w.r.t the columns of $\mathbf{B}$ gives:

$$\mathbf{D}\mathbf{W}^p\mathbf{D}^T\mathbf{b}_p = \mathbf{D}\mathbf{W}^p\mathbf{c}^p \ \ \forall\ p = 1 \ldots k_1 \tag{12}$$

At this point observe that computing a closed form solution for $\mathbf{B}$ will involve solving k_1 linear systems of d_2 equations and d_2 unknowns which may be prohibitive in space and time. Therefore, we incrementally update $\mathbf{B}$ with a gradient descent method with the following learning rules:

$$\mathbf{B}^{n+1} = \mathbf{B}^n + \eta\mathbf{D}\big((\mathbf{D}^T\mathbf{B}^n - \mathbf{C}^T) \circ \mathbf{W}\big) \tag{13}$$
$$\boldsymbol{\mu}^{n+1} = \boldsymbol{\mu}^n + \eta\mathbf{B}\textstyle\sum_{i=1}^{n} \mathbf{W}_i(\mathbf{c}_i - \mathbf{B}^T(\mathbf{d}_i - \boldsymbol{\mu}^n)). \tag{14}$$

After each update of $\mathbf{B}$ or $\boldsymbol{\mu}$, we update the error $\mathbf{E}$ and weights $\mathbf{W}$. η is set up by hand to control the rate of convergence. Typically several iterations for each update of $\mathbf{B}$ or μ are needed. Observe that the computational cost of one iteration is $\mathcal{O}(nk_2d_2)$.

4 Robust Dynamic ACCA

Modeling the dynamics of the hidden states (coefficients $\mathbf{C}$) has proven to be useful in many applications such as recognition or tracking. In this section, we temporally couple the coefficients over time; this coupling will act as a regularization term for smoothing the hidden values. Assuming linear temporal dynamics, the DACCA formulation becomes

$$E_{dwacca} = E_{wacca} + \lambda_2 \sum_{i=2}^{n-1} ||\bar{\mathbf{e}}_i||_{\bar{\mathbf{W}}_i} \quad (15)$$

where $\bar{\mathbf{e}}_i = \mathbf{c}_i - \mathbf{A}\mathbf{c}_{i-1}$, $\mathbf{A}$ is the dynamic transition matrix that must be estimated, and analogous to the static case, $\bar{\mathbf{W}}$ are the weights. This effectively couples the hidden parameters $\mathbf{c}_i$ over time imposing a linear transformation between two time instants.

Taking derivatives of E_{dwacca} w.r.t. the parameters, the resulting equations for $\mathbf{B}$ and $\hat{\mathbf{B}}$ are the same as E_{wacca}. However for $\mathbf{C}$, now we have additional terms due to the coupling of coefficients $\mathbf{c}$ over time which have to be taken into account. The resulting system of equations to update $\mathbf{C}$ and $\mathbf{A}$ are:

$$\begin{bmatrix} \Pi_1 & 0 & 0 & \cdots & 0 \\ \Phi_2 & \Pi_2 & \Psi_2 & \cdots & 0 \\ 0 & \Phi_3 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \Pi_{n-1} & \Psi_{n-1} \\ 0 & 0 & \cdots & 0 & \Pi_n \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \cdots \\ \mathbf{c}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \\ \cdots \\ \boldsymbol{\tau}_n \end{bmatrix} \quad (16)$$

$$(\mathbf{C}^{t-1}\bar{\mathbf{W}}_i(\mathbf{C}^{t-1})^T)\mathbf{a}^{j_1} = (\mathbf{C}^{t-1}\bar{\mathbf{W}}_i(\mathbf{C}^t)^{j_1}) \quad (17)$$

where $j_1 = 1 \ldots k_1$ and

$$\boldsymbol{\tau}_i = \left[\ \hat{\mathbf{B}}^T\hat{\mathbf{W}}_i\hat{\mathbf{d}}_i + \lambda\mathbf{W}_i\mathbf{B}^T\mathbf{d}_i\ \right] \quad (18)$$

$$\Phi_i = \Psi_i^T = \left[\ -\lambda_2\bar{\mathbf{W}}_i\mathbf{A}\ \right]$$

$$\Pi_i = \left[\ \hat{\mathbf{B}}^T\hat{\mathbf{W}}_i\hat{\mathbf{B}} + \lambda\mathbf{W}_i + \lambda_2(\bar{\mathbf{W}}_i + \mathbf{A}^T\bar{\mathbf{W}}_i\mathbf{A})\ \right]$$

$$\mathbf{C}^{t-1} = [\mathbf{c}_1\ \mathbf{c}_2\ \ldots\ \mathbf{c}_{n-2}] \quad \mathbf{C}^t = [\mathbf{c}_2\ \mathbf{c}_3\ \ldots\ \mathbf{c}_{n-1}].$$

Observe that Π_i is the same expression as left hand side of (11) with the additional regularization term $\lambda_2(\bar{\mathbf{W}}_i + \mathbf{A}^T\bar{\mathbf{W}}_i\mathbf{A})$, but now due to the temporal coupling of the coefficients equation (16) results in a large sparse system of equations of size $(nk_1 \times nk_1)$. If $\lambda_2 = 0$ we obtain a uncoupled (diagonal) system of equations in (11). We use a "blocked" version of the iterative Gauss-Seidel method [26] for solving it; the solution can be solved iteratively as:

$$\mathbf{c}_i^{n+1} = \Pi_i^{-1}(\boldsymbol{\tau}_i - \Psi_i\mathbf{c}_{i-1}^{n+1} - \Phi_i\mathbf{c}_{i+1}^{n}) \ \ \forall\ i = 2 \ldots n-1$$

The initial solution is chosen to be the uncoupled one, that is, when $\Psi_i = \Phi_i = \mathbf{0}$. In our experimental results the Gauss-Seidel method has always converged, however in the more general case it is not guarantee to do so. A necessary and sufficient condition for the algorithm to converge requires that the largest (in absolute value) eigenvalue of the iteration matrix be smaller than 1 [26]. Observe that in this case, in order to compute the coefficients $\mathbf{c}_i$ in new test data, we have to jointly estimate all the coefficients (eq. 16).

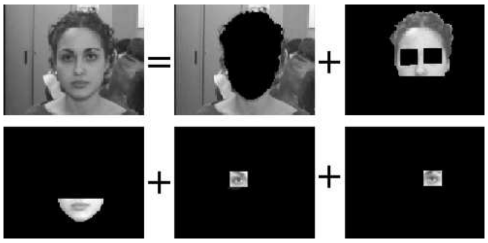


Figure 2: Faces are modeled using separate regions for the eyes and mouth.

5 Experimental Results

The methods described here are useful for learning low-dimensional models in situations where (i) there are multiple, high-dimensional, data sets that provide information about the same event, (ii) the inherent dimensionality of these data sets may differ, (iii) the noise in each set may differ, (iv) the data may have temporal dependencies, and (v) the data may be corrupted by noise. After training, the method is specifically useful for predicting one data set from another. The approach assumes that the data and the dependencies can be well approximated by linear models. It is interesting to note that linear models have been successful for modeling complex data that is, in fact, non-linear (e.g. [24]).

5.1 Learning Coupled Facial Appearance Models

In this experiment we test the ability of ACCA to learn the relationship between two sets of faces of two different people performing similar facial expressions. In general it is hard to model and animate faces, even when they are cartoons characters. Usually complex models encoding the physical underlying musculature of the face are used (e.g. Candide model [1, 10]). Recently De la Torre [6] has used eigenfeatures [15, 17] to automatically learn person-specific appearance face models and dynamically animate them. The face is modeled using separate eigenfeatures since facial features such as the eyes and mouth undergo almost independent changes in appearance [6, 17]. The facial feature appearance models are automatically learned from an input image sequence given the starting regions in the first frame (Figure 2) [6].

Given two sets of faces of dimensions ($\hat{\mathbf{d}} \in \Re^{21128\times 1}$ and $\mathbf{d} \in \Re^{27858\times 1}$), we manually select all pairs of corresponding images and store the image pixels for a given region from each image in $\mathbf{d}$ and $\hat{\mathbf{d}}$. Once we have $\mathbf{D}$ and $\hat{\mathbf{D}}$, we compute ACCA by simultaneously minimizing (9), (11), (10), (14) and (13). In this case we have empirically chosen $\lambda = 0.1$.

Given an ACCA model, and a new test sequence, we can compute the coefficients of the test sequence and derive the coupled coefficients for the other face. Fig. 4 shows frames of a virtual female face animated by the appearance of the input male face. The first column shows the original input stream ($\hat{\mathbf{d}}$); the second one, ($\mathbf{d}$), is the result of animating the face with ACCA plus the affine motion of the head. As we can observe this approach allows us to model the rich texture present on the face providing fairly realistic animations.

As discussed in Section 2, another possible solution would involve computing PCA separately on $\mathbf{D}$ and $\hat{\mathbf{D}}$, projecting the data into each independent PCA basis and calculating the linear regression between the coefficients. Alternatively, we could compute the joint SVD. However in these cases the resulting reconstruction of $\mathbf{d}$ given just one sample of $\hat{\mathbf{d}}$ will result in a larger error. For instance, the normalized reconstruction error in the test sequence for the mouth area (9 basis) is 0.045 for ACCA, 0.12 for the joint SVD and 0.11 for the independent PCA plus linear regression between coefficients. Moreover, ACCA provides a formalism which allows us to incorporate regularization terms (as in DACCA) or to extend it to several training sets.

5.2 Can you dance without me?

Learning models of how humans move, has been an active area of research during the last decade. Several approaches have been proposed in the literature for learning human motion based on linear dynamics [3], switching linear models [19] and more complex non-linear methods [2]. In this section we explore learning linear dependencies between two dancers and illustrate ACCA by animating one dancer knowing the motion of the other. In this case we make use of DACCA since temporal smoothness is a reasonable assumption given the continuity of the motion.

The training and test data consist of joint angles of professional dancers gathered with a motion capture system. The human body configuration is determined by 25 parameters. Six parameters represent to the 3D translation and 3D rotation of the body and the remaining 19 parameters correspond to the relative joint angles between connected limbs (see [23]).

In this experiment we assume that there is no correlation between the 3D translation, 3D rotation and the rest of the parameters describing the joint angles. Figure 3a shows the correlation between the x and y translation, as we can observe there is a correlation between these two variables. However the same phenomenon does not hold for the x translation and any of the joint angles, where the joint distribution over the angles is roughly spherical, see Figure 3b.

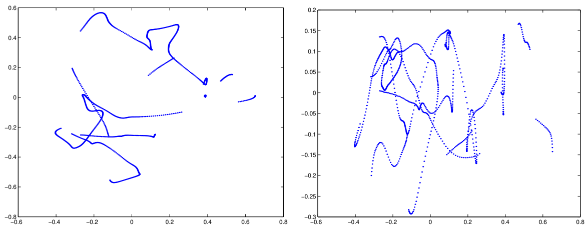


Figure 3: *(a)* Correlation between the x-translation and y-translation. *(b)* Correlation between x-translation and joint angle of the left hand

We train DCCA, minimizing (15), on the first 2700 samples and use the rest (around 700 samples) for testing purposes. Figure 5a shows some frames of the test sequence where two people are dancing. In Figure 5b just one of the dancers is given to the algorithm and we can seen DACCA infers a reasonable estimate of the pose of the other dancer. In this case the predicted dancer is the low intensity one, the closer to the camera in the first frame.

6 Conclusions and Future work

In this paper, following previous work on asymmetric PCA, we have proposed ACCA for learning dependencies between two data sets by coupling them in the hidden parameter space. ACCA can be especially useful when working with high dimensional spaces, since a first common step in many algorithms is to reduce the dimensionality (usually with the arbitrary PCA coordinate frame), and afterwards to perform the processing in this low dimensional space. Experiments with facial avatars suggest that the linear model may suffice for these arguably complex data sets (provided that a good training set is given).

Our current work is exploring a Bayesian formulation of the problem which can give a probabilistic interpretation and can exploit statistical methods and provide formal ways to automatically determine, all the parameters of interest (e.g. λ).

Also we are working on two extensions to the case of symmetric coupled component analysis. That is, in order to construct a joint model where we can do predictions *bidirectionally*, we could minimize:

$$E(\mathbf{B},\mathbf{C},\hat{\mathbf{B}},\hat{\mathbf{C}}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{c}_i||_2^2 + \lambda||\hat{\mathbf{d}}_i - \hat{\mathbf{B}}\hat{\mathbf{c}}_i||_2^2 + \lambda_2||\mathbf{c}_i - \hat{\mathbf{B}}^T\hat{\mathbf{d}}_i||_2^2 + \lambda_3||\hat{\mathbf{c}}_i - \mathbf{B}^T\mathbf{d}_i||_2^2$$

This is just one of many possible extensions. We are exploring the use of another symmetric formulation, that can be posed as the minimization of

$$E(\mathbf{B},\mathbf{C},\hat{\mathbf{B}},\hat{\mathbf{C}},\mathbf{\Gamma},\hat{\mathbf{\Gamma}},\mathbf{H}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{c}_i||_2^2 + \lambda||\hat{\mathbf{d}}_i - \hat{\mathbf{B}}\hat{\mathbf{c}}_i||_2^2 + \lambda_1||\mathbf{c}_i - \mathbf{\Gamma}\mathbf{h}_i||_2^2 + \lambda_2||\hat{\mathbf{c}}_i - \hat{\mathbf{\Gamma}}\mathbf{h}_i||_2^2$$

This method can be useful as a continuous model for recognition or joint representation when both data sets are presented. Observe that we have introduced a hierarchical structure in the coefficients, in which local coefficients $\mathbf{c}_i$ and $\hat{\mathbf{c}}_i$ reconstruct each of the sets (with different dimensionality, rank) and coefficients $\mathbf{h}_i$ reconstruct both sets. This could be useful in the case of modeling the shape and texture [4], where it is likely that the dimensionality of the shape space is lower than then dimensionality of the texture.

Another obvious extension of this work would employ a more complex, non-linear model of the coupling between coefficients (e.g. mixture of Gaussians, multilayer perceptron, radial basis functions, etc.). Finally, the method can be extended to model the dependences in more than two data sets.

Acknowledgments. We are grateful to the reviewers for their remarkably detailed and thorough reviews which have improved this manuscript. We thank Michael Gleicher for providing the 3D motion capture data used in the dance experiments.

References

[1] M. Brand. Voice puppetry. *SIGGRAPH*, pp. 21–28, 1999.

[2] M. Brand and A. Hertzmann. Style machines. *SIGGRAPH*, pp. 183–192, 2000.

[3] C. Bregler. Learning and recognizing human dynamics in video sequences. *CVPR*, pp. 568–574, 1997.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *ECCV*, pp. 484–498, 1998.

[5] M. Covell. Eigen-points: Control-point location using principal component analysis. *Int. Conf. on Automatic Face and Gesture Recognition*, pp. 122– 127, 1996.

[6] F. De la Torre. Automatic learning of appearance face models. *Second Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, 2001.

[7] F. De la Torre and M. Black. Robust principal component analysis for computer vision. *ICCV*, pp. 362–369, 2001.

[8] K. Diamantaras. *Principal Component Neural Networks (Theory and Applications).* John Wiley & Sons, 1996.

[9] C. Eckardt and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[10] P. Eisert and B. Girod. Model-based estimation of facial expression parameters from image sequences. *ICIP*, pp. 418–421, 1997.

[11] K. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics, Vol. 21, pp.*, 21:489–498, 1979.

[12] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the Int. Statistical Institute*, LII:4:5, 1987.

[13] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions.* Wiley, New York., 1986.

[14] S. Haykin. *Adaptive filter theory.* Prentice-Hall, 1996.

[15] T. Jebara, K. Russell, and A. Pentland. Mixtures of eigenfeatures for real-time structure from texture. *ICCV*, pp. 128–135, 1998.

[16] G. Li. Robust regression. D. Hoaglin, F. Mosteller, and J. Tukey, eds., *Exploring Data, Tables, Trends and Shapes.* John Wiley & Sons, 1985.

[17] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *PAMI*, 19(7):696–710, 1997.

[18] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, (37):3311–3325, 1997.

[19] V. Pavlovic, J. Rehg, and J. MacCormick. Learning switching linear models of human motion. *NIPS*, pp. 626–632, 2000.

[20] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. Hand pose estimation using specialized mappings. *ICCV*, Vol. I, pp. 378–385, 2001.

[21] S. Roweis. EM algorithms for PCA and SPCA. *NIPS*, pp. 626–632, 1997.

[22] L. Scharf. *The SVD and reduced rank signal processing. SVD and Signal Processing, II.* (R. Vaccaro, ed.), Elsevier, 1991.

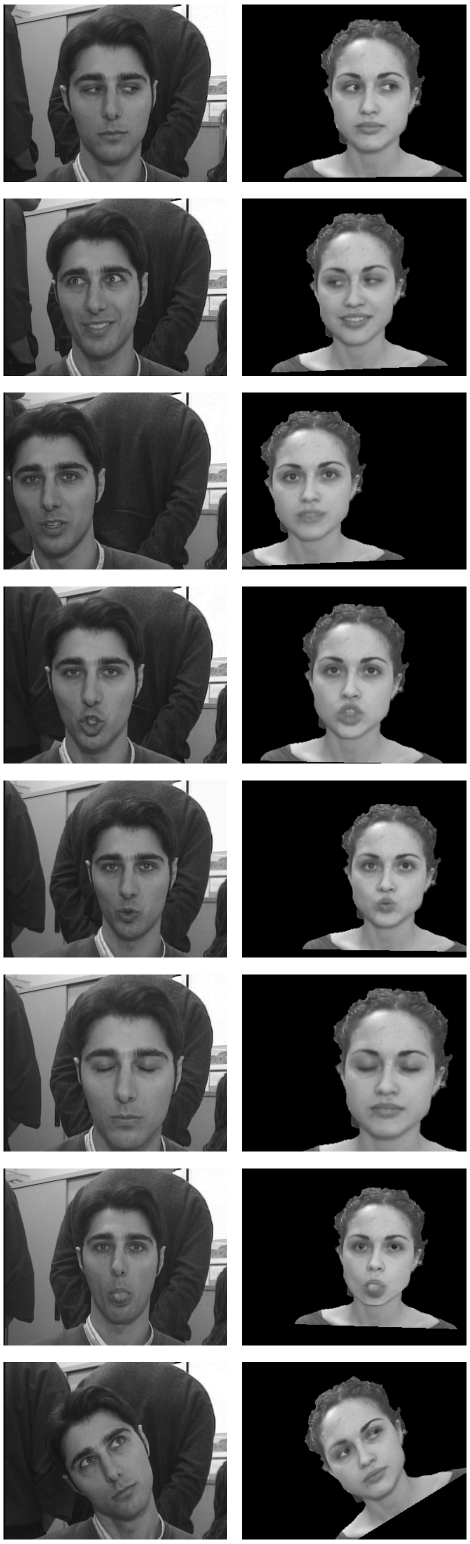
[23] H. Sidenbladh, F. De la Torre, and M. Black. A framework for modeling the appearance of 3D articulated figures. *Int. Conf. on Auto. Face and Gesture Recognition*, pp. 368–375, 2000.

[24] S. Soatto, G. Doretto, and Y.Wu. Dynamic textures. *ICCV*, pp. 439–446, 2001.

[25] M. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. Royal Stat. Soc. B*, 61:611–622, 1999.

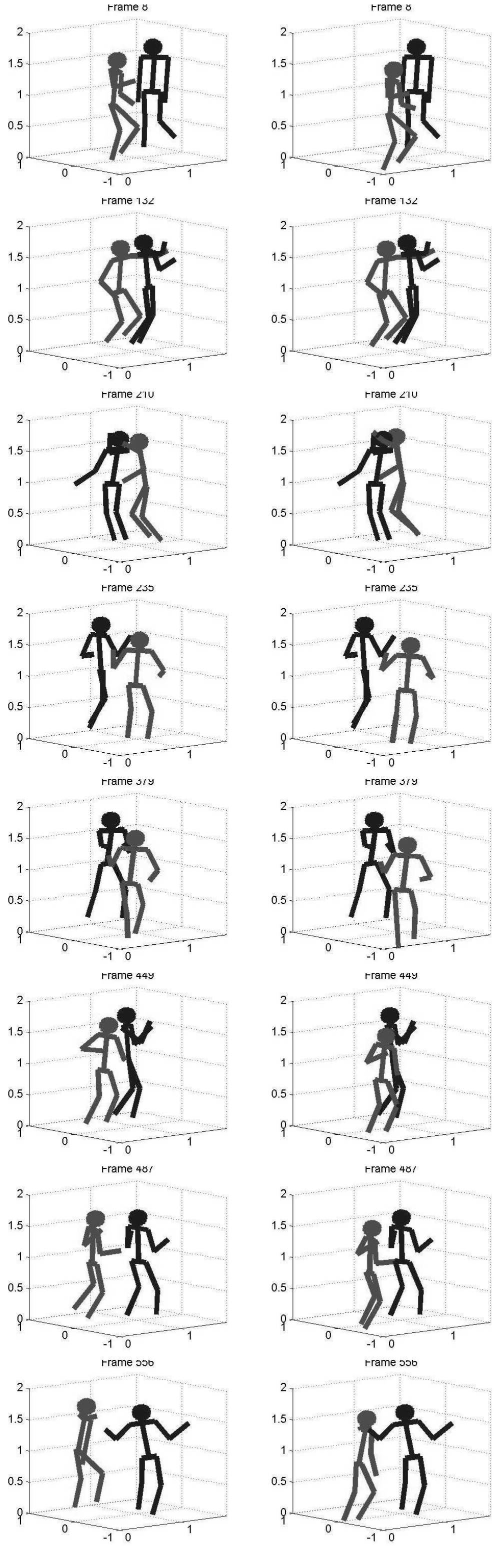
[26] R. Varga. *Matrix Iterative Analysis.* Springer-Verlag, 2000.

[27] L. Xu and A. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. Neural Networks*, 6(1):131–143, 1995.



a *b*

Figure 4: a) Original face. b) Animated virtual face.



a *b*

Figure 5: a) Original data of the test set. b) Predicted data(see text).