To appear in Pattern Recognition.

Weighted and robust learning of subspace representations

Danijel Skočaj *,[a] Aleš Leonardis [a] Horst Bischof [b]

[a] *University of Ljubljana, Faculty of Computer and Information Science, Tržaška 25, SI-1001 Ljubljana, Slovenia*

[b] *Inst. for Computer Graphics and Vision, Graz University of Technology, Austria*

Abstract

A reliable system for visual learning and recognition should enable a selective treatment of individual parts of input data and should successfully deal with noise and occlusions. These requirements are not satisfactorily met when visual learning is approached by appearance-based modeling of objects and scenes using the traditional PCA approach. In this paper we extend standard PCA approach to overcome these shortcomings. We first present a weighted version of PCA, which, unlike the standard approach, considers individual pixels and images selectively, depending on the corresponding weights. Then we propose a robust PCA method for obtaining a consistent subspace representation in the presence of outlying pixels in the training images. The method is based on the EM algorithm for estimation of principal subspaces in the presence of missing data. We demonstrate the efficiency of the proposed methods in a number of experiments.

Key words: appearance-based modeling, robust learning, principal component analysis, weighted PCA, missing pixels, robust PCA

1 Introduction

The construction of suitable object and scene representations plays a crucial role in the process of visual learning and recognition. Previous experience, prior knowledge, and the information resulting from other cognitive processes affect the level to which the newly acquired information is incorporated in the representation. One may also expect more recent (or more reliable, or more informative, or more noticeable) experiences to have a stronger influence on the model than others. The psychophysical studies of object recognition suggest that human perception is more tuned to some (e.g., more frequently experienced) views than to the others [1,2]. Therefore, a learning algorithm should enable a selective influence of individual training images in the process of learning. It should also enable a selective treatment of individual pixels as well.

In real world applications, it is often the case that parts of the data are unavailable. The learning algorithm should compensate for the missing data and build a consistent representation. In addition, images of objects and scenes

* Corresponding author. Tel. :+386-1-4776631; Fax: +386-1-4264647.
Email address: danijel.skocaj@fri.uni-lj.si.

may contain noise, occlusions, specular reflections or other undesirable effects. The ultimate system for visual learning and recognition should be able to determine the objects of interest in the learning stage and include in the representation only relevant information.

Visual learning is often approached by the appearance-based modeling of objects and scenes. One popular approach to model building is principal component analysis (PCA). However, PCA in its original form has several shortcomings with respect to the premises mentioned above.

In this paper we present approaches that overcome these problems. In Section 2 we first present a generalized *weighted* version of PCA, which, unlike the standard approach, considers individual pixels and images selectively, depending on the corresponding weights. In Section 3 we present a special case of weighted PCA, which is adopted for *learning from partial data*. Since standard PCA is intrinsically non-robust to non-Gaussian noise, we present in Section 4 a method for *robust learning* that is able to detect inconsistencies in the training images and build the representations from consistent data only. The proposed methods are evaluated in Section 5. In the last section we summarize the paper and outline some work in progress.

1.1 Related work

In PCA, the basis vectors of the principal subspace, i.e., the principal directions in the input space, can be estimated by minimizing the reconstruction error of all reconstructed input vectors. A similar, but probabilistically oriented approach, was taken also by Roweis, who derived an algorithm [3] for calculating principal subspace, which is based on EM (expectation-maximization) algorithm [4]. A very similar Probabilistic PCA algorithm was also independently proposed by Tipping and Bishop [5].

The two-step structure of the EM algorithm allows us to introduce weights in order to perform weighted learning or learning from incomplete data. In this respect, several methods with different derivations but very similar realizations have already been proposed [6–10]. Our basic algorithm for *weighted learning*, which we will derive by modifying the EM algorithm [3], is closely related to these approaches, since in principle they all minimize the same error function – the weighted squared reconstruction error. Like in [10], we also present different algorithms, which are specialized for different types of weights (temporal, spatial). In addition, we adapt the EM algorithm for *learning from incomplete data* and further extend this algorithm with the regularization term, which adequately constrains the reconstructed values in missing pixels. Furthermore, we present a new approach to learning from incomplete data by iterative reconstruction of missing pixels.

The weights or outlying pixels are, however, very often not known in advance and they have to be determined in the learning process. And since in the *learning* stage the model of the object or the scene is just being built, there is no reliable previous knowledge, which could be used to estimate outliers[1]. Nevertheless, some authors have tackled also the problem of the *robust learning* [19,8]. More recently, De la Torre and Black proposed a method for robust principal component analysis based on M-estimation [20,10], which performs

[1] Different approaches have been proposed to improve the robustness of the *recognition* [11–18]. All of them, however, assume that the images in the learning stage were ideal and that the visual model is correct and available.

well on images with sufficient temporal correlation, but it is very time consuming. Lately, several incremental methods for robust subspace learning have been presented [21,22] that tackle the problem of robust learning in an incremental way. Finally, several methods that are used for robust factorization have also been proposed (e.g., [23,24]). Though tailored to a different type of problem, some of their principles are relevant for robust eigenspace learning as well. The method for *robust learning*, which we will present in this work, is most related to [20], however it is simpler and faster while still producing similar results. Our proposed method iteratively detects outliers in all images, estimates the representation from inliers only, and reconstructs outlying values.

1.2 Notation and background

We first briefly outline the standard PCA approach and introduce the notation.

Each image is represented as a vector $\mathbf{x} \in \mathbb{R}^M$. Let all N training images be aligned in the data matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{M \times N}$. The low-dimensional principal subspace of this high-dimensional input space is usually calculated by the eigenvalue decomposition of the covariance matrix[2] $\mathbf{C} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^\top$ of the input data $\mathbf{X}$, where $\hat{\mathbf{x}}_j$, the columns of $\hat{\mathbf{X}}$, are the input images $\mathbf{x}_j$ with subtracted mean image $\boldsymbol{\mu}$. We denote the eigenvectors of $\mathbf{C}$ by $\mathbf{u}_i$ (eigenimages), and the corresponding eigenvalues by λ_i. As the variance is mainly contained in the first eigenimages (those with the largest eigenvalues), only

[2] If the image size M is larger than the number of training images N a similar approach based on the inner product matrix is used [25].

$k \ll N$ eigenimages are retained, thus $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{M \times k}$. All the training images are then projected into the eigenspace and the coefficient vectors $\mathbf{a}_j = [a_{1j}, \ldots, a_{kj}]^\top \in \mathbb{R}^k$, $j = 1 \ldots N$ (aligned in $\mathbf{A} \in \mathbb{R}^{k \times N}$) are obtained:

$$\mathbf{a}_j = \mathbf{U}^\top \hat{\mathbf{x}}_j = \mathbf{U}^\top (\mathbf{x}_j - \boldsymbol{\mu}) \ . \tag{1}$$

Therefore, each image $\mathbf{x}_j$ is approximated with the linear combination of the eigenimages

$$\mathbf{x}_j \approx \sum_{i=1}^{k} a_{ij} \mathbf{u}_i + \boldsymbol{\mu} \ . \tag{2}$$

For a given dimension of the subspace k, PCA finds such principal axes $\mathbf{u}_l$, $l = 1 \ldots k$ and coefficient vectors $\mathbf{a}_j \in \mathbb{R}^k, j = 1 \ldots N$ that maximize the variance of the projections of the training images and minimize the total squared reconstruction error

$$\mathcal{E} = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\hat{x}_{ij} - \sum_{p=1}^{k} u_{ip} a_{pj} \right)^2 \ . \tag{3}$$

Thus, the principal axes and the principal components can alternatively be estimated by minimizing (3). This is a nonlinear minimization problem and can be solved by iterating the two-step procedure where first the coefficients are estimated and then the principal axes are computed. Such an EM algorithm as proposed by Roweis [3] looks as follows:

- **E-step**: $\mathbf{A} = (\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \hat{\mathbf{X}}$
- **M-step**: $\mathbf{U} = \hat{\mathbf{X}} \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}$.

2 Weighted PCA

Standard PCA can be extended to a weighted version by introducing weights into Eq. (3). The weights are assembled in the matrix $\mathbf{W} \in \mathbb{R}^{M\times N}$, where w_{ij} is the weight of the i-th pixel in the j-th image. The goal is to minimize the *weighted* squared reconstruction error

$$\mathcal{E} = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\left(\hat{x}_{ij} - \sum_{p=1}^{k} u_{ip}a_{pj}\right)^2 \quad . \tag{4}$$

Here, the values of the matrix $\hat{\mathbf{X}}$ are obtained by subtracting the *weighted* mean vector $\boldsymbol{\mu}$ from the training images $\mathbf{x}_j$.

In practice, it is useful to deal with two types of weights: *temporal* weights ${}^{\mathbf{t}}\mathbf{w} \in \mathbb{R}^{1\times N}$, which put different weights on individual images, and *spatial* weights ${}^{\mathbf{s}}\mathbf{w} \in \mathbb{R}^{M}$, which put different weights on individual pixels within an image[3]. Since different types of weights yield different algorithms for estimating the weighted principal subspace, we will discuss both types of weights separately.

2.1 Temporal weights

Temporal weights determine the importance of individual images for estimation of the principal subspace. If the temporal weight of an image is larger than the weights of other images, the reconstruction error of that image should be proportionally smaller than the reconstruction errors of the other images. Similarly, the contribution of its principal components to the estimation of the

[3] The left superscript is used to distinguish between temporal (${}^{\mathbf{t}}\mathbf{w}$) and spatial (${}^{\mathbf{s}}\mathbf{w}$) weights. ${}^{\mathbf{t}}\mathbf{w}$ is a row vector, while ${}^{\mathbf{s}}\mathbf{w}$ is a column vector.

variances should be larger in comparison with other principal components. For instance, if a training image has the weight 2, while all other images have the weight 1, the result of the weighted algorithm should be the same as the result of the standard algorithm having two copies of the particular image in the training set.

Since all the pixels in an image are treated equally, the weight matrix $\mathbf{W}$ contains the row vector ${}^{\mathbf{t}}\mathbf{w}$ in each row thus[4] $\mathbf{W} = \mathbf{1}_{M\times 1}{}^{\mathbf{t}}\mathbf{w}$. The principal subspace can be obtained by maximizing the *weighted variance*, which can be achieved by eigendecomposition of the *weighted covariance matrix* as presented in Algorithm 1.

Algorithm 1 : TWPCA – temporally weighted PCA

```
Input: data matrix $\mathbf{X}$, temporal weights ${}^{\mathbf{t}}\mathbf{w}$
Output: weighted mean vector $\boldsymbol{\mu}$, eigenvectors $\mathbf{U}$, eigenvalues $\boldsymbol{\lambda}$.
Estimate the weighted mean vector: $\boldsymbol{\mu} = \frac{1}{\sum_{j=1}^{N} {}^{t}w_j} \sum_{j=1}^{N} {}^{t}w_j \mathbf{x}_j$ .
Scale the input data centered around the weighted mean:
${}^{\mathbf{t}}\hat{\mathbf{x}}_j = \sqrt{{}^{t}w_j}(\mathbf{x}_j - \boldsymbol{\mu}), \;\; j = 1 \ldots N$ .
if $M \leq N$ then
    Estimate the weighted covariance matrix: $\mathbf{C} = \frac{1}{\sum_{j=1}^{N} {}^{t}w_j} {}^{\mathbf{t}}\hat{\mathbf{X}}\,{}^{\mathbf{t}}\hat{\mathbf{X}}^{\top}$ .
    Perform SVD on $\mathbf{C}$. Obtain the eigenvectors $\mathbf{U}$ and the eigenvalues $\boldsymbol{\lambda}$.
else
    Estimate the weighted inner product matrix: $\mathbf{C}' = \frac{1}{\sum_{j=1}^{N} {}^{t}w_j} {}^{\mathbf{t}}\hat{\mathbf{X}}^{\top}\,{}^{\mathbf{t}}\hat{\mathbf{X}}$ .
    Perform SVD on $\mathbf{C}'$. Obtain the eigenvectors $\mathbf{U}'$ and the eigenvalues $\boldsymbol{\lambda}'$.
    Determine the principal axes $\mathbf{U}$: $\mathbf{u}_i = \frac{{}^{\mathbf{t}}\hat{\mathbf{X}}\mathbf{u}'_i}{\sqrt{\sum_{j=1}^{N} {}^{t}w_j}\sqrt{\lambda'_i}}$ , $i = 1 \ldots N$ .
    Determine the eigenvalues $\boldsymbol{\lambda} = \boldsymbol{\lambda}'$ .
end if
```

[4] $\mathbf{1}_{M\times N}$ denotes a matrix of the dimension $M \times N$, where every element equals 1.

Before we derive the algorithm for estimating *principal subspace* using arbitrary weights, we will discuss the weighed estimation of *subspace coefficients* considering spatial weights. Spatial weights control the influence of individual pixels within an image. Therefore, if a part of an image is unreliable or not important for the estimation of principal components, its influence can be diminished by decreasing the weights of the corresponding pixels.

The coefficients are traditionally estimated by the standard projection (1) of an image **x** onto the principal axes. The coefficient estimation can, however, also be formulated as a problem of finding a coefficient vector **a** that minimizes the squared reconstruction error between the original image **x** and its reconstruction [18]. Now, it is straightforward to introduce weights in this minimisation process. The goal is to find a coefficient vector **a** that minimizes the *weighted* squared reconstruction error:

$$e = \sum_{i=1}^{M} {}^{s}w_i \left(\hat{x}_i - \sum_{p=1}^{k} u_{ip} a_p \right)^2 \quad . \tag{5}$$

The solution to this minimization problem (the vector **a**) can be found by solving an over-constrained system of linear equations:

$$\sqrt{{}^{s}w_i} \hat{x}_i = \sqrt{{}^{s}w_i} \sum_{p=1}^{k} u_{ip} a_p \quad , \quad i = 1 \dots M \quad . \tag{6}$$

In a similar way we can also approach to the weighted estimation of principal subspace considering arbitrary weights, thus to minimizing (4). The weighted squared reconstruction error (4) can be minimized using a modified EM algorithm. First, let us note that solving an over-constrained system of linear equations is equivalent to calculating the pseudoinverse of the corresponding matrix. Now, by noting that $\mathbf{a}_j = (\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top\hat{\mathbf{x}}_j = \mathbf{U}^\dagger\hat{\mathbf{x}}_j$ we can replace E-step of the EM algorithm by solving the corresponding system of linear equations for each $\mathbf{a}_j$ $, j = 1 \ldots N$. A very similar observation holds also for M-step. Therefore, if we consider weights as well, we can estimate the principal subspace by iteratively solving the following systems of linear equations:

- **E-step**: Estimate $\mathbf{A}$ in the following way: For each image j, $j = 1 \ldots N$, solve the following system of linear equations in the least squares sense:

$$\sqrt{w_{ij}}\hat{x}_{ij} = \sqrt{w_{ij}}\sum_{p=1}^{k} u_{ip}a_{pj} \ \ , \ \ i = 1 \ldots M \ \ . \tag{7}$$

- **M-step**: Estimate $\mathbf{U}$ in the following way: For each pixel i, $i = 1 \ldots M$, solve the following system of linear equations in the least squares sense:

$$\sqrt{w_{ij}}\hat{x}_{ij} = \sqrt{w_{ij}}\sum_{p=1}^{k} u_{ip}a_{pj} \ \ , \ \ j = 1 \ldots N \ \ . \tag{8}$$

At convergence, the columns of $\mathbf{U}$ span the space of the first k principal axes. By expressing (7) and (8) in a more concise way (using pseudoinverse), the modified EM algorithm for weighted PCA is shown[5] in Algorithm 2.

3 PCA on incomplete data

In real world applications, it is often the case that not all data are available. The values of some pixels are missing or are totally unreliable. Such pixels

[5] Subscript $\mathbf{x}_i$ denotes the i-th *column* vector in the matrix $\mathbf{X}$, while $\mathbf{x}_{i:}$ denotes the i-th *row* vector in the matrix $\mathbf{X}$. $.\sqrt{\mathbf{A}}$ is an operator that calculates the square root of each element of the matrix $\mathbf{A}$. $\mathbf{A} \circ \mathbf{B}$ denotes the Hadamard (entrywise) product between two matrices of equal dimension.

```
Algorithm 2 : WPCA – weighted PCA
Input: data matrix $\mathbf{X}$, weight matrix $\mathbf{W}$, number of principal axes to be estimated $k$.
Output: weighted mean vector $\boldsymbol{\mu}$, $\mathbf{U}$ spanning principal subspace.
Estimate the weighted mean vector: $\mu_i = \frac{\sum_{j=1}^{N} w_{ij} x_{ij}}{\sum_{j=1}^{N} w_{ij}} \ , \ i = 1 \ldots M \ .$
Center the input data around the mean: $\hat{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu} \mathbf{1}_{1\times N} \ .$
Set the elements of $\mathbf{U} \in \mathbb{R}^{M\times k}$ to random values.
repeat
    E-step: $\mathbf{a}_j = ((.\sqrt{\mathbf{w}_j} \mathbf{1}_{1\times k}) \circ \mathbf{U})^{\dagger} (.\sqrt{\mathbf{w}_j} \circ \hat{\mathbf{x}}_j), \ j = 1 \ldots N \ .$
    M-step: $\mathbf{u}_{i:} = (.\sqrt{\mathbf{w}_{i:}} \circ \hat{\mathbf{x}}_{i:})((\mathbf{1}_{k\times 1} .\sqrt{\mathbf{w}_{i:}}) \circ \mathbf{A})^{\dagger}, \ i = 1 \ldots M \ .$
until convergence.
```

are referred to as *missing pixels*[6] . The estimation of the principal subspace in the case of incomplete data can be regarded as a special case of weighted PCA where the weights of missing pixels are set to zero. In this section we will consider this special case of weighted learning more thoroughly.

3.1 Modified EM algorithm

First let us denote the sets of indices of non-missing (known) and missing pixels in the j-th image with $\mathcal{I}_j^{\bullet}$ and $\mathcal{I}_j^{\circ}$, respectively, and the sets of indices of non-missing and missing pixels in the i-th row of the data matrix $\mathbf{X}$ with

[6] E.g., in a range image, the depth is not defined in some pixels; in a panoramic image obtained by a camera-mirror setup, the image is occluded by a mirror handle – the corresponding pixels are not a part of the environment; when a robot holds an object with a griper, the object is occluded by the gripper's fingers – these pixels can be detected and considered as *missing pixels* (non-valid, undefined data), as opposed to other valid, defined – *known pixels.*

$\mathcal{I}_{i:}^{\bullet}$ and $\mathcal{I}_{i:}^{\circ}$, respectively.

Now, the goal is to minimize the reconstruction error of known pixels, thus in E and M steps of the EM algorithm we have to set up the equations arising from the non-missing pixels only:

- **E-step**: Estimate **A** in the following way: For each image j, $j = 1 \ldots N$, solve the following system of linear equations in the least squares sense:

$$\hat{x}_{ij} = \sum_{p=1}^{k} u_{ip} a_{pj} \ , \ i \in \mathcal{I}_{j}^{\bullet} \ . \tag{9}$$

- **M-step**: Estimate **U** in the following way: For each pixel i, $i = 1 \ldots M$, solve the following system of linear equations in the least squares sense:

$$\hat{x}_{ij} = \sum_{p=1}^{k} u_{ip} a_{pj} \ , \ j \in \mathcal{I}_{i:}^{\bullet} \ . \tag{10}$$

Here, the mean image is obtained by estimating the mean over the known pixels:

$$\mu_i = \frac{1}{|\mathcal{I}_{i:}^{\bullet}|} \sum_{j \in \mathcal{I}_{i:}^{\bullet}} x_{ij} \ , \ i = 1 \ldots M \ . \tag{11}$$

When dealing with images containing a considerable number of missing pixels, such a formulation results in an ill-posed problem. The principal axes are optimized to ensure the optimal reconstruction error in known pixels. Since the reconstruction error in missing pixels is not considered in the minimization process, the reconstructed missing pixels can have arbitrary values. Therefore, the generalization ability of this algorithm is relatively weak. Thus, although such an algorithm can produce very small reconstruction errors in known pixels, it may at the same time produce very strange reconstructed values in missing pixels.

To alleviate this problem we impose additional application dependent constraints to the minimization process. When the images are ordered, as in the case of image sequences, we can extend the algorithm to also include a smoothness prior so as to ensure that the values of reconstructed missing pixels are changing smoothly over time. Thus, in the M-step we minimize the second derivative of the reconstructed missing pixels. The new M-step looks as follows:

- **M-step**: Estimate $\mathbf{U}$ in the following way: For each pixel i, $i = 1 \ldots M$, solve the following system of linear equations in the least squares sense:

$$\begin{aligned} \hat{x}_{ij} &= \sum_{p=1}^{k} u_{ip} a_{pj} \ , \ \ j \in \mathcal{I}^{\bullet}_{i:} \\ 0 &= \alpha \sum_{p=1}^{k} u_{ip}(a_{p,j-1} - 2a_{pj} + a_{p,j+1}) \ , \ \ j \in \mathcal{I}^{\circ}_{i:} \ , \end{aligned} \tag{12}$$

  where α is the parameter which weights the influence of the smoothness constraint.

Thus, the overall algorithm minimizes the following error function:

$$\mathcal{E} = \sum_{j=1}^{N} \sum_{i \in \mathcal{I}^{\bullet}_{j}} \left(\hat{x}_{ij} - \sum_{p=1}^{k} u_{ip} a_{pj} \right)^2 + \alpha \sum_{j=1}^{N} \sum_{i \in \mathcal{I}^{\circ}_{j}} \left(\sum_{p=1}^{k} u_{ip} a''_{pj} \right)^2 \ . \tag{13}$$

To summarize the algorithm in a more concise way we will introduce some new notation. Let us partition a training image (a *column* in the data matrix) $\hat{\mathbf{x}}_j$ into $\hat{\mathbf{x}}^{\bullet}_j$ and $\hat{\mathbf{x}}^{\circ}_j$ vectors of $M^{\bullet}$ known and M° unknown values in $\hat{\mathbf{x}}_j$, respectively, and assign the corresponding *rows* of $\mathbf{U}$ to $\mathbf{U}^{\bullet}$ and $\mathbf{U}^{\circ}$. Similarly, we can partition a *row* in the data matrix $\hat{\mathbf{x}}_{i:}$ into $\hat{\mathbf{x}}^{\bullet}_{i:}$ and $\hat{\mathbf{x}}^{\circ}_{i:}$ row vectors of the $N^{\bullet}$ known and N° unknown values in $\hat{\mathbf{x}}_{i:}$, respectively, and assign the corresponding *columns* of $\mathbf{A}$ to $\mathbf{A}^{\bullet}$ and $\mathbf{A}^{\circ}$. Using this notation, the modified EM algorithm for the estimation of the principal subspace from incomplete data is given in Algorithm 3.

Algorithm 3 : MPPCAtmpSm – PCA on incomplete data by temporal smoothing

Input: data matrix $\mathbf{X}$, weight matrix $\mathbf{W}$ with binary values, number of principal axes to be estimated k.

Output: mean vector $\boldsymbol{\mu}$, $\mathbf{U}$ spanning principal subspace.

```
Estimate the weighted mean vector: $\mu_i = \frac{1}{N^\bullet} \sum_{j=1}^{N^\bullet} \mathbf{x}^\bullet_{ij}$ , $i = 1 \ldots M$ .
Center the input data around the mean: $\hat{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}_{1\times N}$ .
Set elements of $\mathbf{U} \in \mathbb{R}^{M\times k}$ to random values.
repeat
    E-step: $\mathbf{a}_j = \mathbf{U}^{\bullet\dagger}\hat{\mathbf{x}}^\bullet_j$, $j = 1 \ldots N$ .
    M-step: $\mathbf{A}' = [\mathbf{a}_2 \ldots \mathbf{a}_N, \mathbf{a}_N] - 2\mathbf{A} + [\mathbf{a}_1, \mathbf{a}_1 \ldots \mathbf{a}_{N-1}]$
                 $\mathbf{u}_{i:} = [\hat{\mathbf{x}}^\bullet_{i:}\ \mathbf{0}_{1\times N^\circ}][\mathbf{A}^\bullet\ \alpha\mathbf{A}'^\circ]^\dagger$ , $i = 1 \ldots M$ .
until convergence.
Orthogonalise $\mathbf{U}$.
Project input data on $\mathbf{U}$: $\mathbf{a}_j = \mathbf{U}^{\bullet\dagger}\hat{\mathbf{x}}^\bullet_j$, $j = 1 \ldots N$ .
Perform PCA on $\mathbf{A}$. Obtain $\mathbf{U}'$ .
Rotate $\mathbf{U}$ for $\mathbf{U}'$: $\mathbf{U} = \mathbf{U}\mathbf{U}'$.
```

In the recognition stage, an input image $\mathbf{x}$ is projected into the principal subspace $\mathbf{U}$ in an analogical way. The coefficients a_j are obtained by solving an over-constrained system of linear equations, which arise from the non-missing pixels only:

$$\hat{x}_i = \sum_{j=1}^{k} a_j u_{ij} \ , \ \ i \in \mathcal{I}^\bullet \ , \tag{14}$$

or, expressing it with the pseudoinverse:

$$\mathbf{a} = \mathbf{U}^{\bullet\dagger}\hat{\mathbf{x}}^\bullet \ . \tag{15}$$

Alternatively, the principal axes in the M-step can be obtained by applying standard PCA to all pixels providing that missing pixels are filled-in. The question is how to optimally fill-in the values of the missing pixels. Since not all the pixels of an image are known, some coordinates of the corresponding point in the image space are undefined. Thus, the position of the point is constrained to the subspace defined by the missing pixels. Given the principal subspace $\mathbf{U}$, which models the input data, the optimal location is a point in the missing pixels subspace which is closest to the principal subspace. This point is obtained by replacing the missing pixels with the reconstructed values, which are calculated by (2) using the coefficients $\mathbf{a}_j$ estimated in E-step of the current iteration and the principal axes $\mathbf{u}_j$ obtained in the previous iteration. Therefore, the new M-step looks as follows:

- **M-step**: Estimate $\mathbf{U}$ by applying standard PCA to $\mathbf{X}$ with the reconstructed missing pixels:

$$x_{ij} = y_{ij} \ , \ j = 1 \dots N \ , \ i \in \mathcal{I}_j^{\circ} \ \text{where} \ \mathbf{Y} = \mathbf{U}\mathbf{A} + \boldsymbol{\mu}\mathbf{1}_{1 \times N} \ . \tag{16}$$

The convergence of the algorithm can be sped up by a more efficient initialization of the principal axes $\mathbf{U}$. Instead of simply setting its elements to random values, one can estimate the initial values of $\mathbf{U}$ from an estimate of $\mathbf{A}$ obtained by performing SVD on the inner product matrix $\mathbf{C}'$, which is estimated from the non-missing pixels only. The entire procedure is outlined in Algorithm 4.

An advantage of this approach is that it does not assume a smoothness prior, therefore it is appropriate for visual learning from unordered image sequences as well.

Algorithm 4 : MPPCAitRec – PCA on incomplete data by iterative reconstruction

```
Input: data matrix $\mathbf{X}$, weight matrix $\mathbf{W}$ with binary values, number of principal axes to be estimated $k$.
Output: mean vector $\boldsymbol{\mu}$, principal axes $\mathbf{U}$.
Estimate the weighted mean vector: $\mu_i = \frac{1}{N^\bullet} \sum_{j=1}^{N^\bullet} \mathbf{x}^\bullet_{ij}$ , $i = 1 \ldots M$ .
Center the input data around the mean: $\hat{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}_{1\times N}$ .
Estimate the inner product matrix $\mathbf{C}'$ from the known data:
    $c'_{ij} = \frac{M}{N|\mathcal{P}|} \sum_{p \in \mathcal{P}} \hat{x}_{pi}\hat{x}_{pj}$ , $\mathcal{P} = \{p \mid p \in \mathcal{I}^\bullet_i \wedge p \in \mathcal{I}^\bullet_j\}$, $i = 1 \ldots N$, $j = 1 \ldots N$.
Perform SVD on $\mathbf{C}'$ yielding eigenvectors as an estimate for $\mathbf{A}$.
Estimate an initial $\mathbf{U}$: $\mathbf{u}_{i:} = \hat{\mathbf{x}}^\bullet_{i:}\mathbf{A}^{\bullet\dagger}$ , $i = 1 \ldots M$ .
repeat
    E-step: Estimate $\mathbf{A}$: $\mathbf{a}_j = \mathbf{U}^{\bullet\dagger}\hat{\mathbf{x}}^\bullet$, $j = 1 \ldots N$ .
    M-step: Estimate $\mathbf{U}$ by applying standard PCA on $\mathbf{X}$ with reconstructed missing pixels: $x_{ij} = y_{ij}$ , $j = 1 \ldots N$ , $i \in \mathcal{I}^\circ_j$ where $\mathbf{Y} = \mathbf{U}\mathbf{A} + \boldsymbol{\mu}\mathbf{1}_{1\times N}$ .
    M-step: Estimate $\mathbf{U}$ by applying PCA on $\mathbf{X}$ with reconstructed missing pixels: $x_{ij} = y_{ij}$ , $j = 1 \ldots N$ , $i \in \mathcal{I}^\circ_j$ where $\mathbf{Y} = \mathbf{U}\mathbf{A} + \boldsymbol{\mu}\mathbf{1}_{1\times N}$ .
until convergence.
```

4 Robust PCA

So far we have presented algorithms for weighted learning and learning from incomplete data. These algorithms assume that the weights are known in advance. However, in a real world environment, this is often not the case. Images may contain various outliers (occlusions, reflections, etc.) whose exact positions in an image are not known. Since standard PCA is intrinsically sensitive to non-Gaussian noise, such disturbances may considerably degrade the results of the visual learning and recognition.

In this section we will present a robust algorithm that has the capability of detecting outliers in training images during eigenspace learning. These outliers are then treated as missing pixels and the principal subspace is estimated from inliers only, using the algorithms for learning from incomplete data presented in the previous section.

If we have some additional knowledge about the object or scene, which is being modeled, or an algorithm for outlier detection tailored for that particular problem, we can take advantage of it and use such an algorithm for detection of outliers. However, if we have no additional knowledge we have to rely solely on the information contained in the training images. We can detect outliers by checking the consistency over the entire image sequence. The pixels, whose reconstruction error significantly deviates from the distribution of the reconstruction errors of all pixels, are treated as outliers. In other words, the outliers are pixels with large reconstruction errors.

Based on this, our robust PCA approach is outlined in Algorithm 5. Usually, only a few iterations (even only a single one) of this algorithm are sufficient for convergence. At each iteration the data matrix $\mathbf{X}$ gets improved; many outliers are detected and replaced with reconstructed values, which are better approximations of the correct values. Therefore, in the subsequent iteration additional outliers can be detected and reconstructed even better.

In the initial step of this algorithm standard PCA is performed on the entire set of training images. The obtained initial subspace $\mathbf{U}'$ is then used for the detection of outliers. Since the outlying pixels are not consistent with other pixels they usually appear as a high-frequency noise in a sequence of train-

Algorithm 5 : RPCA – robust PCA

```
Input: data matrix $\mathbf{X}$, number of principal axes to be estimated $k$.
Output: mean vector $\boldsymbol{\mu}$, eigenvectors $\mathbf{U}$, eigenvalues $\boldsymbol{\lambda}$, coefficients $\mathbf{A}$.
repeat
    Perform standard PCA on $\mathbf{X}$ and obtain $\boldsymbol{\mu}'$, $\mathbf{U}' \in \mathbb{R}^{M \times k'}$ and $\mathbf{A}' \in \mathbb{R}^{k' \times N}$.
    Reconstruct the training images using $\boldsymbol{\mu}'$, $\mathbf{U}'$ and $\mathbf{A}'$ and calculate the reconstruction error.
    Detect outliers considering reconstruction errors.
    Treat outliers as missing pixels and perform PCA using an algorithm for learning from incomplete data (MPPCA) to obtain $\boldsymbol{\mu}$, $\mathbf{U}$, $\boldsymbol{\lambda}$, and $\mathbf{A}$ from inliers only.
    Reconstruct the training images using $\boldsymbol{\mu}$, $\mathbf{U}$ and $\mathbf{A}$ and replace missing pixels in $\mathbf{X}$ with reconstructed values.
until the change in the outlier set is small.
```

ing pixels. Thus, they are modeled predominately with the eigenvectors, which correspond to small eigenvalues, while the consistent pixels (signal) are mostly modeled with the eigenvectors containing most of the variance. For that reason, only the first k' eigenvectors ($k' < k$) are used for the detection of outliers. The value of k' and the threshold for outlier detection depend on the content and the type of the training images and on the amount and the degree of the deviation of outliers, as well as on the application's goal. Therefore, these values are application dependent and cannot be automatically determined in general.

A potential drawback of this algorithm is the initial step, which still relies on standard non-robust PCA. If the training set consists of a certain number of images that contain a proportionally large number of outliers, they can distort the principal subspace in such a way that the detection of outliers becomes very unreliable. A solution to this problem is to divide each iteration of the algorithm into two stages. In the first stage, the *outlying images* are detected (images, which contain a large portion of outliers, and are not consistent with other images) and then the initial eigenspace, which is used for detecting outlying pixels, is detected from other images only. In this way, the obtained initial eigenspace is less distorted by outliers. Consequently, the detection of *outlying pixels* in all images in the second stage is more reliable.

5 Experimental results

In this work we described algorithms for weighted subspace learning, learning from incomplete data and robust learning. We performed a number of experiments in order to evaluate the proposed methods. In this section we will present some of the results.

5.1 Weighted PCA

We applied the weighted PCA approach in a categorisation task. The goal was to categorise images in one of 25 categories shown in Fig. 1. The category members were obtained by morphing between 3-D models of two members from the same basic level category.

The images were created by Dahl, Graf et al. [26,27] for conducting psychophysical experiments. They investigated whether categorisation performance depends systematically on topological transformation, i.e., on the po-

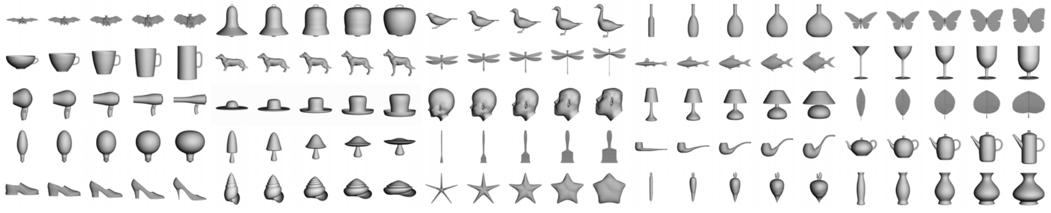
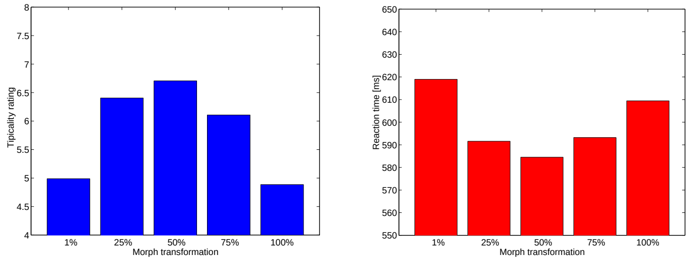


Fig. 1. 25 categories used in psychophysical and computer vision categorisation experiments [26,27].

sition within the morphing sequence [26]. Their conclusion was that more typical objects were categorised faster than objects that were rated as less typical category members. This is also obvious from the Fig. 2, which depicts the typicality ratings (TRs) obtained in the typicality task and reaction times (RTs) obtained in the word-matching task.



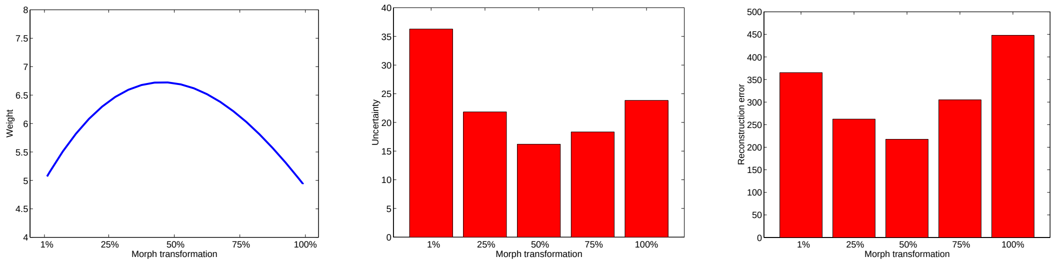
(a) (b)

Fig. 2. (a) Typicality rates, (b) reaction times in psychophysical experiments [26,27].

We applied the proposed temporally-weighted PCA learning algorithm and set the weights according to the typicality ratings of the training images. Since typicality ratings tend to be higher for objects that were perceived more frequently [1], it seems reasonable to weight the training images to simulate different numbers of observations of individual training images.

In the training stage we used 22 morphed training images per category. We weighted these images and build for each category a two-dimensional subspace (the average weights are plotted in Fig. 3(a) and correspond to the empirical typicality ratings shown in Fig. 2(a)). In the categorisation stage we used the same five images per category, which were used in the psychophysical experiment. We projected every test image into each two-dimensional eigenspace. For every image the smallest reconstruction error was obtained when the image was projected into the eigenspace representing the correct category. We were thus able to correctly categorise all test images only by considering the reconstruction errors (i.e., distances to the subspaces). To also evaluate the reliability of categorisation we calculated the uncertainty defined as the ratio of the smallest to the second smallest reconstruction error, thus the ratio of the distances to the closest and to the second closest subspace (plane).

The overall results (average uncertainty and mean reconstruction error over all test images with the same morph distance) are plotted in Figs. 3(b,c). They show that the categorisation of images which are similar to the training images with higher weights (more typical images) is more reliable. These results nicely resemble the RTs in the picture-name matching psychophysical experiment with human subjects (Fig. 2(b)). They demonstrate that the weighted learning approach tuned the subspace representations in such a way that more typical category members can be categorised more reliably.



(a) (b) (c)

Fig. 3. (a) Weights, (b) uncertainty, (c) reconstruction errors in computer vision experiment.

5.2 Learning from incomplete data

In this subsection we will present the results of the proposed methods for *learning from incomplete data*. The goal was to learn a 9-D subspace representation of the 'duck' object from 72 images from the COIL20 database (see Fig. 4(a)). From each image we "erased" a randomly placed square, which covered 20% of the image (Fig. 4(b)) and considered erased pixels as missing pixels.

First we applied standard PCA algorithm on the training images with missing pixels filled-in using the simple mean substitution (referred to as *MS* in the figures). The missing pixels were replaced with the mean values calculated from non-missing pixels only. Then we applied the proposed algorithms: Algorithm 4, which iteratively reconstructs missing pixels (*itRec*), the modified EM algorithm which calculates the principal subspace from non-missing pixels only (*EM*), and Algorithm 3, which assumes the smoothness prior. The latter algorithm was tested in the circumstances when the smoothness prior held (the training images were aligned in the sequential order – *EMts*) and when this assumption was disadvantageous (the training images were aligned in a random order – *rndEMts*).

Figs. 4(c–g) depict some training images with the missing pixels filled-in with the reconstructed values, which were obtained using the methods described above. One can observe that both proposed algorithms (*itRec* and *EMts*) reconstructed missing squares significantly better than standard PCA on the training images with mean-substituted missing pixels (*MP*). The modified EM algorithm without smoothness prior (*EM*) performed well too for the most part, however in some pixels it failed to correctly reconstruct the pixel values to a large extent. As expected, when the training images were randomly ordered, the incorrectly assumed smoothness prior significantly degraded the results (*rndEMts*).

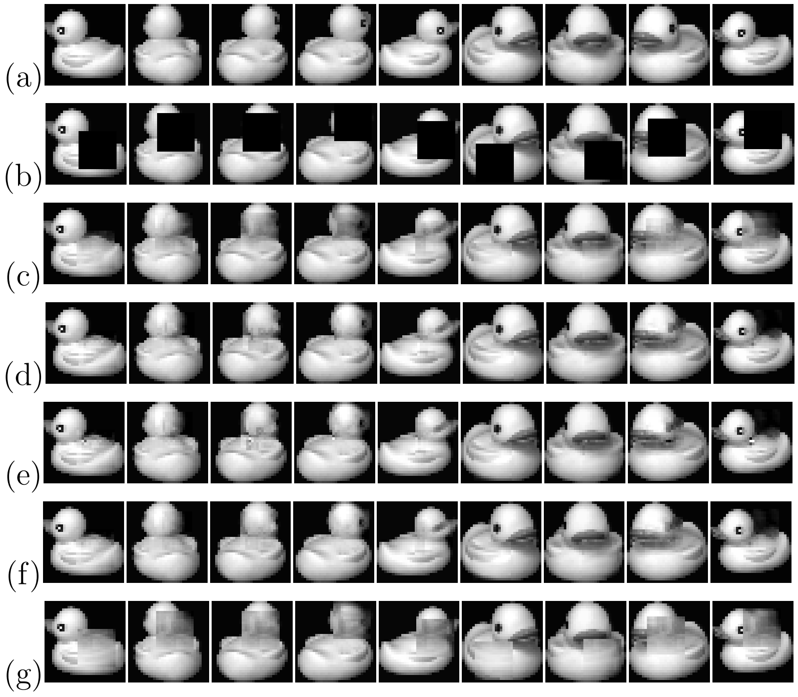
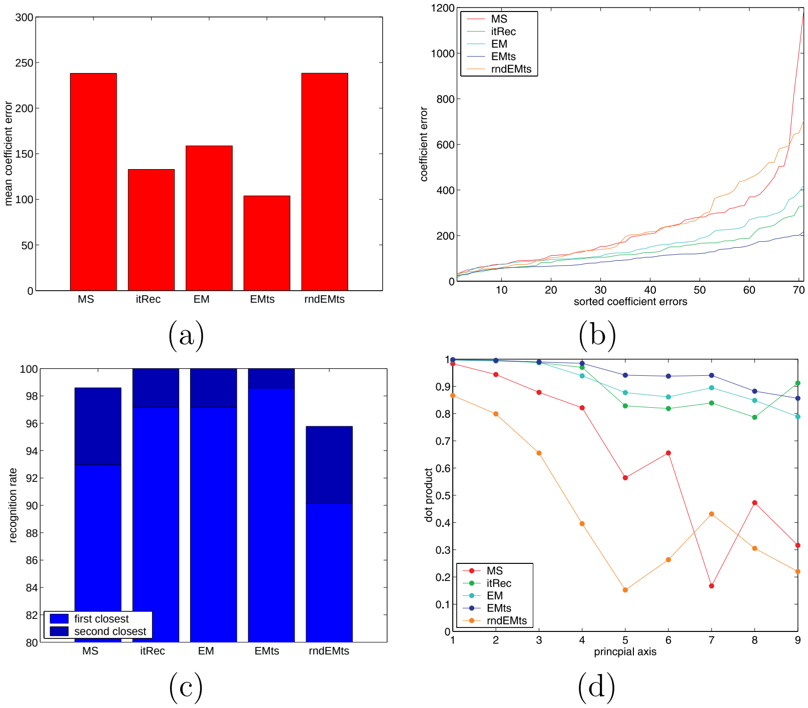


Fig. 4. (a) Nine complete images, (b) training images with missing squares, reconstructed missing pixels using (c) *MS*, (d) *itRec*, (e) *EM*, (f) *EMts* and (g) *rndEMts* approaches.

The same conclusions can be drawn also from Fig. 5, which presents quantitative results. For comparison, we projected the original images without missing pixels into the obtained eigenspaces. Ideally, we would obtain exactly the same coefficients as we would by projecting the training images considering the non-missing pixels only. Fig. 5(a) compares the obtained mean coefficient error, i.e., the mean distance between the corresponding coefficient vectors, while Fig. 5(b) plots the values of all 72 distances in increasing order. Fig. 5(c) plots the recognition rate, i.e. how many times the projected complete images were the closest to the corresponding projected non-complete training images. Fig. 5(d) indicates the similarity between the estimated principal axes and optimal ones, which were obtained by performing PCA on the complete training images without missing pixels. For each of the first nine principal axes, the dot product between the ideal and the estimated principal axis is depicted. All these figures clearly indicate that the proposed methods for learning from incomplete data perform well and considerably improve the results of the simple mean-substitution approach.



(a)

(b)

(c)

(d)

Fig. 5. Performance of learning algorithms: (a) mean coefficient error, (b) sorted coefficient errors, (c) recognition rate, (d) dot products between corresponding estimated and optimal principal axes.

We also tested the performance of the algorithms with regard to the amount of missing pixels present in the training images. We erased from 10% to 90% of randomly selected spatially incoherent pixels in each image. All the algorithms mentioned above were applied at each level of missing pixels.

One such trial at 50% missing pixels is shown in Fig. 6. One complete and one training image are shown as well as the same training image with reconstructed missing pixels. Again, it is evident that *itRec*, *EM*, and *EMts* approaches reconstruct missing pixels significantly better than *MS* and *rndEMts*.

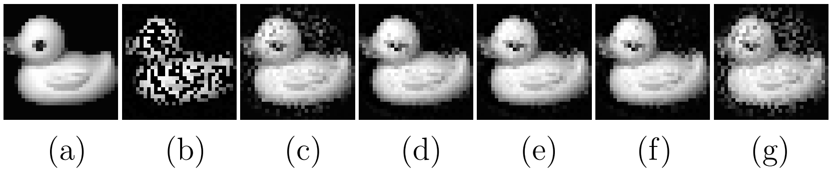


Fig. 6. (a) One complete image, (b) training image containing 50% of missing pixels, reconstructed missing pixels using (c) *MS*, (d) *itRec*, (e) *EM*, (f) *EMts* and (g) *rndEMts* approaches.

Fig. 7 summarizes the results obtained at all levels of missing pixels. For each level of missing pixels it plots the mean coefficient error, the mean squared reconstruction error in missing pixels, the recognition rate and the mean dot product between the corresponding estimated and optimal principal axes. All the plots show that *itRec*, *EM*, and *EMts* perform very well when the amount of missing pixels is 50% or less. Note that in this case the missing pixels are spatially incoherent, which alleviates the problem to some extent. It can be observed that the unconstrained EM-based approach *EM* breaks down when we have 70% missing pixels. At this level, some of the missing pixels are reconstructed very erratically, since no constraints are given. By contrast, if the reconstructed values are constrained with the smoothness prior (*EMts*), the best results are achieved. Of course, if this smoothness prior does not hold true, this algorithm does not perform well (*rndEMts*). In such cases, the best choice is iterative reconstruction of missing pixels (*itRec*).

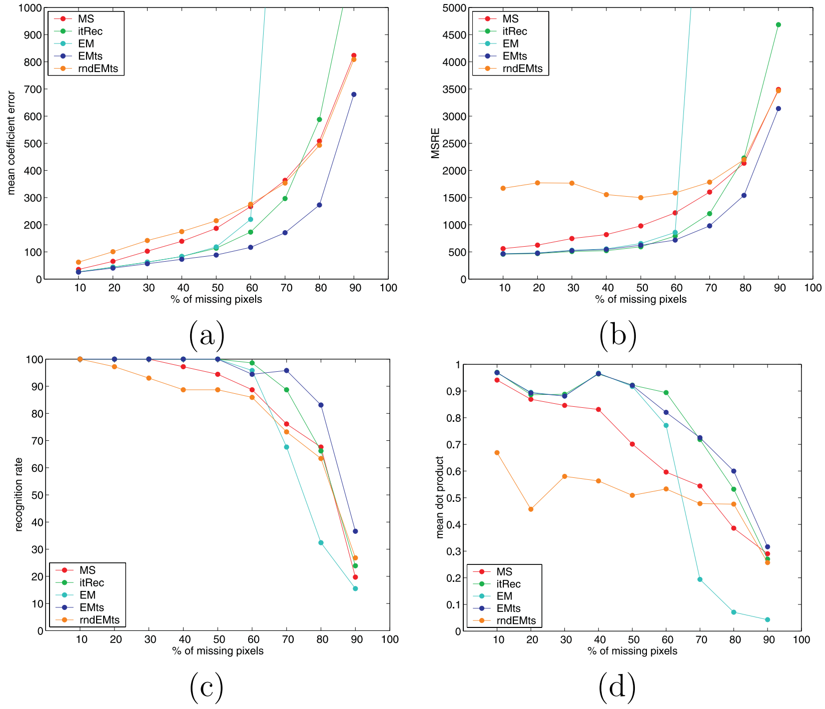


Fig. 7. Performance of learning algorithms with regard to amount of missing pixels present in training images: (a) mean coefficient error, (b) mean squared reconstruction error in missing pixels, (c) recognition rate, (d) mean dot product between corresponding estimated and optimal principal axes.

5.3 Robust PCA

The performance of the robust method we first tested on the images with known ground truth is shown in Fig. 8. We captured 30 panoramic images of size 100 × 100 in the faculty hall. Then, we synthetically applied gradual illumination changes and nonlinear illumination changes (a shadow—the horizontal “cloud”) to this set of images. In addition, we added as an outlier area, a square on a randomly chosen position in every image. The goal was to learn the background representation capturing the illumination variations (linear and nonlinear) but discarding the outliers. Since these images are temporally well correlated, we included the smoothness prior in the M-step of the EM algorithm for the calculation of the principal axes from incomplete data.

The results are depicted in Fig. 8. In the images reconstructed from the first principal component obtained with the standard PCA, one can clearly observe that the linear illumination changes are modeled, but not the nonlinear ones. If the model consists of the first 8 principal axes produced by standard PCA, then all illumination changes are captured in the reconstructions, however, the model also contains the outliers (squares). On the other hand, using our robust algorithm, one can observe from the reconstructions based on the first 8 principal axes, that all illumination changes are captured in the model, while the outliers are not, which is exactly what we want to achieve.

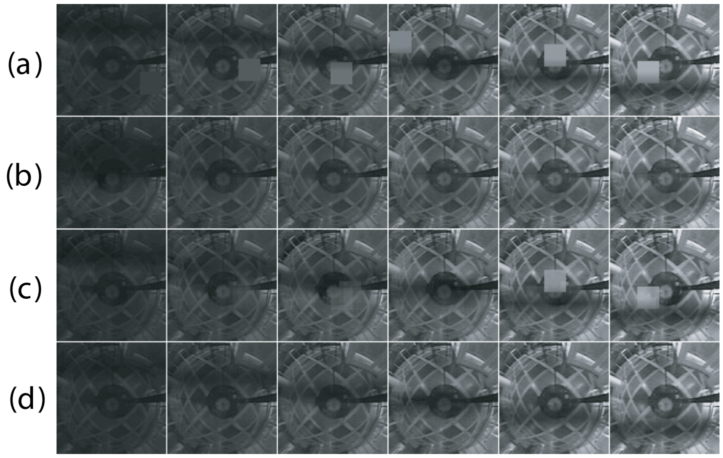


Fig. 8. Comparison of our method with standard PCA. (a) Input images (every fifth image from training set). (b) Reconstruction based on first principle axis (PA) using standard PCA. (c) Reconstruction based on first 8 PA using standard PCA. (d) Reconstruction based on first 8 PA using *robust PCA*.

We also calculated the mean absolute reconstruction error of the pixels belonging to the background and of the pixels that were occluded by the squares. Ideally, the reconstruction error of the background pixels should be as small as possible (meaning that the model fits the data well), while the reconstruction error of the occluded pixels should be large (meaning that the outlying pixels do not fit to the model), which would lead to the efficient detection of outliers. Table 1 compares the reconstruction errors with the errors obtained using the optimal principal axes, which were estimated from the data without outliers (ground truth). It is evident that the robust PCA outperforms the standard one since the errors obtained with the proposed algorithm are much closer to the optimal ones.

Table 1

Comparison of the reconstruction errors obtained using standard and robust PCA.

		num.	rec. error in	
data	method	of PA	background	squares
ground truth	standard PCA	8	1	2805
with outliers	standard PCA	1	146	2601
with outliers	standard PCA	8	21	540
with outliers	robust PCA	8	6	2608

We also performed an experiment where we applied our method on the same image sequence[7] of 506 images of size 120 × 160 pixels as De la Torre and Black [20], who also proposed a method for robust learning of appearances based on PCA. Our goal was to model the background, capturing the gradual illumination changes, while excluding the people that appear in the images.

Some of the results are presented in Fig. 9. A subset of input images is depicted in Figs. 9(a,f). One can observe in Figs. 9(b,g) that using standard non-robust

[7] Images were obtained from http://www.salleurl.edu/∼ftorre/papers/rpca2.html.

PCA approach the people are not excluded from the background and still appear as ‘ghosts’. By contrast, the robust approach is able to detect the outliers by most part, thus the robust model encompasses predominantly the background and does not contain the people. As a consequence, the outliers depicted as white pixels in Figs. 9(d,i) mainly correspond to the people and extreme illumination changes.

For comparison, we also present the results of De la Torre and Black [20] in Figs. 9(e,j). Their method produces similar results to ours. Also, the Equal Error Rate of the two methods is similar. We manually labeled (segmented) people that appear in the images presented in Fig. 9 and calculated EER. Our algorithm produced 39,94% EER, while the algorithm from [20] produced 40,52%. This similarity is not surprising, since both algorithms rely on PCA and detection of outliers based on reconstruction error. However, the approach to the robust estimation of the PCA subspace is different. Their approach is ‘softer’ since it is based on robust M-estimation that iteratively and gradually differentiates more reliable pixels from less reliable ones. Our algorithm is ’harder’ in this respect, since it detects and completely discards the detected outliers and imputes their values with the values predicted by the model in an EM fashion. Consequently, our algorithm is faster; in this experiment it took less than 2 minutes to finish the task, while the algorithm of De la Torre and Black [20] calculated the corresponding model in 21 minutes on a 3.2 GH Pentium IV.

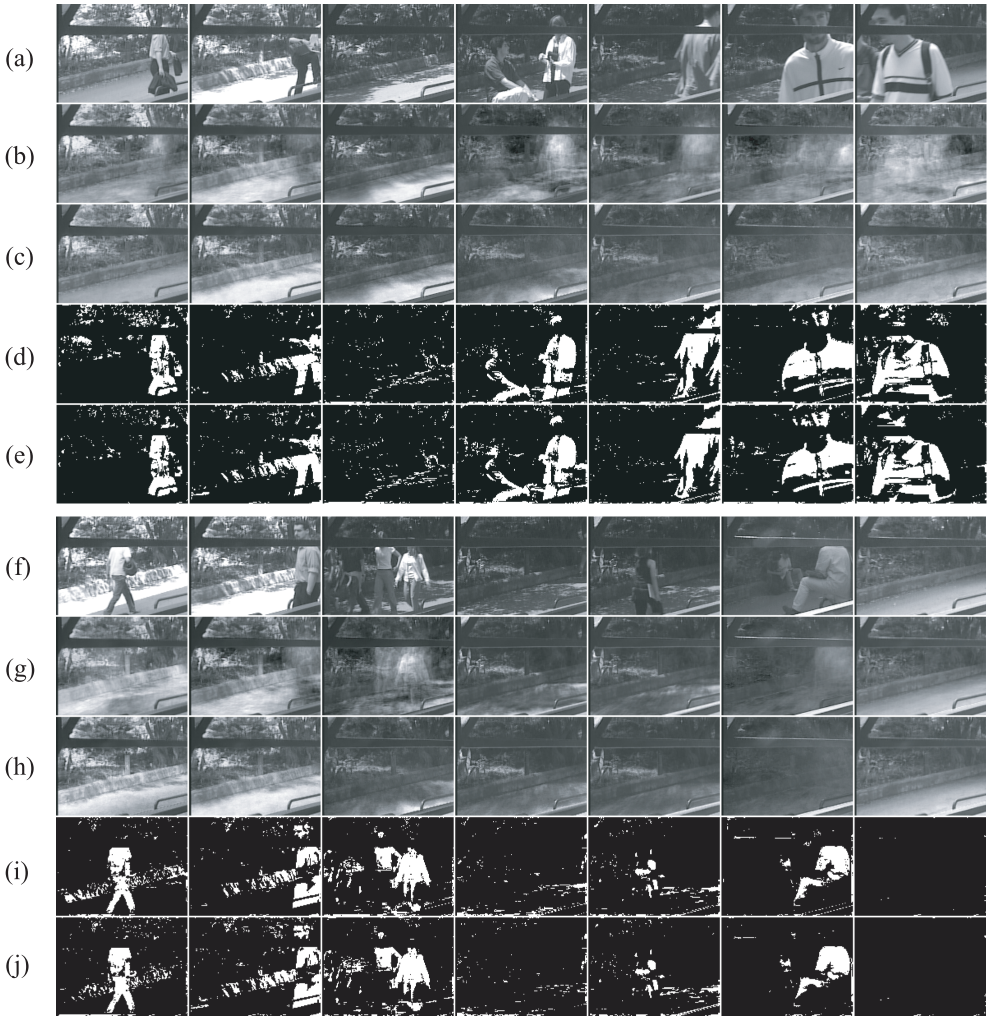


Fig. 9. (a),(f) Original data from [20], (b),(g) standard PCA reconstruction, (c),(h) our robust PCA reconstruction, (d),(i) outliers obtained by our method, (e),(j) outliers obtained by [20].

6 Conclusion

Appearance-based modeling using PCA has been a popular approach to visual learning and recognition in recent years. A shortcoming of standard PCA approach is that it treats all pixels of an image equally. Also, all training images have equal influence on the estimation of principal axes. This is not well suited for applications where some images or some parts of images should be considered more or less important than others. In this work, we presented a generalized PCA approach, which estimates principal axes and principal components considering weighted pixels and images, enabling selective influence of training images as well as pixels in individual images during the process of learning. We also considered the special case of weighted learning, where some pixels are totally unreliable and proposed methods for learning from incomplete data. Finally, we put the learning algorithms into a robust framework. Since the images of objects and scenes are not always ideal and as such, they may contain noise or occlusions, we proposed a robust method for eigenspace learning, which detects outliers and estimates the principal subspace from the consistent data only.

In this work the PCA approach was always used for processing grey level images. It can be used, however, for processing images of other modalities as well. Moreover, it can be applied to any set of vectors that are temporally correlated. Instead of images, feature vectors obtained using various methods for feature extraction, can be used. All the algorithms presented in this paper can be applied to such data domains as well. In this way, a pure holistic view-based approach can be extended by exploiting local geometric features and structural information. Our current research has been taking this approach, which has the potential to overcome many drawbacks of view-based methods enabling efficient visual learning and recognition.

Acknowldegment

This research has been supported in part by the following funds: Research program Computer Vision P2-0214 (RS), the EU FP6-004250-IP project CoSy, CONEX project, and SI-A project. The experiment on weighted PCA was performed in cooperation with M. Graf, C. Dahl and H. H. Bülthoff from Max-Planck Institute for Biological Cybernetics, Tübingen, Germany.

References

[1] L. Barsalou, Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories, Journal of Experimental Psychology: Learning, Memory, & Cognition 11 (1985) 629–649.

[2] D. I. Perret, M. W. Oram, E. Ashbridge, Evidence accumulation in cell popoulations responsive to faces: an account of generalisation of recognition withouth mental transformations, in: Object recognition in man, monkey, and machine, MIT/Elsevier, 1998, pp. 111–145.

[3] S. Roweis, EM algorithms for PCA and SPCA, in: Neural Information Processing Systems 10 (NIPS'97), 1997, pp. 626–632.

[4] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society series B 39 (1977) 1–38.

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

[6] T. Wiberg, Computation of principal components when data are missing, in: Proc. Second Symp. Computational Statistics, 1976, pp. 229–236.

[7] H. Shum, K. Ikeuchi, R. Reddy, Principal component analysis with missing data and its application to polyhedral object modeling, PAMI 17 (9) (1995) 854–867.

[8] K. Gabriel, S. Zamir, Lower rank approximation of matrices by least squares with any choice of weights, Technometrics 21 (21) (1979) 489–498.

[9] H. Sidenbladh, F. de la Torre, M. J. Black, A framework for modeling the appearance of 3D articulated figures, in: AFGR00, 2000, pp. 368–375.

[10] F. De la Torre, M. J. Black, A framework for robust subspace learning, IJCV 54 (1) (2003) 117–142.

[11] A. P. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: Proceedings of CVPR 1994, pp. 84–91, 1994.

[12] K. Ohba, K. Ikeuchi, Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects, PAMI 19 (9) (1997) 1043–1048.

[13] H. Murase, S. K. Nayar, Image spotting of 3D objects using parametric eigenspace representation, in: SCIA95, 1995, pp. 325–332.

[14] J. L. Edwards, H. Murase, Coarse-to-fine adaptive masks for appearance matching of occluded scenes, MVA 10 (5-6) (1998) 232–242.

[15] M. J. Black, A. D. Jepson, Eigentracking: Robust matching and tracking of articulated objects using a view-based representation, IJCV 26 (1) (1998) 63–84.

[16] R. Dayhot, P. Charbonnier, F. Heitz, Robust visual recognition of color images, in: Proceedings of CVPR 2000, 2000, pp. 685–690.

[17] R. P. N. Rao, Dynamic appearance-based recognition, in: CVPR 1997, 1997, pp. 540–546.

[18] A. Leonardis, H. Bischof, Robust recognition using eigenimages, CVIU 78 (2000) 99–118.

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

[20] F. De la Torre, M. J. Black, Robust principal component analysis for computer vision, in: ICCV'01, 2001, pp. I: 362–369.

[21] D. Skočaj, A. Leonardis, Weighted and robust incremental method for subspace learning, Ninth IEEE International Conference on Computer Vision ICCV 2003 II (2003) 1494–1501.

[22] Y. Li, On incremental and robust subspace learning, Pattern Recognition 37 (7) (2004) 1509–1518.

[23] H. Anæs, R. Fisker, K. Åström, J. M. Carstensen, Robust factorization, PAMI 24 (9) (2002) 1215–1225.

[24] P. Anandan, M. Irani, Factorization with uncertainty, IJCV 49 (2) (2002) 101–116.

[25] H. Murakami, V. Kumar, Efficient calculation of primary images from a set of images, PAMI 4 (5) (1982) 511–515.

[26] C. Dahl, M. Graf, H. Bülthoff, Categorisation performance depends systematically on shape transformation, Perception 32 Supplement 117 (ECVP).

[27] M. Graf, H. Bülthoff, Object shape in basic level categorisation, in: Proceedings of the European Cognitive Science Conference, 2003, p. 390.