Available online at www.sciencedirect.com
SCIENCE DIRECT®
Image and Vision Computing xx (2006) 1–12
image AND vision COMPUTING
www.elsevier.com/locate/imavis

Incremental and robust learning of subspace representations

Danijel Skočaj *, Aleš Leonardis

Visual Cognitive Systems Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, SI-1001 Ljubljana, Slovenia

Received 21 July 2004; received in revised form 7 June 2005; accepted 21 July 2005

Abstract

Learning is a fundamental capability of any cognitive system. To enable efficient operation of a cognitive agent in a real-world environment, visual learning has to be a continuous and robust process. In this article, we present a method for subspace learning, which takes these considerations into account. We present an incremental method, which sequentially updates the principal subspace considering weighted influence of individual images as well as individual pixels within an image. We further extend this approach to enable determination of consistencies in the input data and imputation of the inconsistent values using the previously acquired knowledge, resulting in a novel method for incremental, weighted, and robust subspace learning. We demonstrate the effectiveness of the proposed concept in several experiments on learning of object and background representations.

Keywords: Subspace learning; Incremental learning; Robust learning

1. Introduction

Cognitive vision has become an important emerging discipline over the last years caused by the growing need for visually enabled cognitive systems. The main scientific foundations of cognitive vision are, among others, the issues of visual architecture, representations, memory, learning, and recognition [1]. There is no doubt, however, that *learning* plays a major role in developing intelligent visual and cognitive systems. Cognitive systems need to acquire the information about the external world through learning or association, as the complex interrelationships between percepts and their contextual frames could never be specified explicitly through programming [2] due to their complexity and the need for adaptability. In this paper, we will focus on two important aspects of learning: incrementality and robustness.

In order to avoid evolutionary time-scales in the development of a cognitive system, it has to initially encompass a certain level of predefined functionality. It then has to be able to build new concepts upon the initial ones and keep developing throughout its lifetime. It is, therefore, important that the representations employed allow learning to be a continuous, open-ended, life-long process. In other words, the representations should not be learned once and for all in a training stage and then used in their fixed form for recognition, planning, and acting. They should be continuously updated over time, adapting to the changes in the environment, new tasks, user reactions, user preferences, etc. The learning process should facilitate such incremental way of building and updating of representations. So, there should be no strict distinction between the activities of learning and recognizing—these activities should be interleaved.

This is a non-trivial challenge. Most of the state-of-the-art algorithms for visual learning and recognition do not consider continuous learning. They follow the standard paradigm, dividing the off-line learning stage and the recognition stage [3–10]. Most of these approaches are not designed in a way that would enable efficient updating of the learned model, which is a basic prerequisite for incremental learning.

There are two issues, which are important for a reliable continuous learning. Firstly, the representation which is used for modeling the observed world has to allow for updating with newly acquired information. This update step should be efficient and should not require access to the previously observed data while still preserving the previously acquired knowledge. And secondly, a crucial issue is the quality of updating, which highly depends on the correctness of the interpretation of the current visual input. When the correct interpretation of the current visual input is given by a tutor, the update step is risk-free in the sense that the algorithm can

* Corresponding author. Tel.: +386 1 4776 631; fax: +386 1 4264 647.
E-mail addresses: danijel.skocaj@fri.uni-lj.si (D. Skočaj), alesl@fri.uni-lj.si (A. Leonardis).

0262-8856/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.imavis.2005.07.028

update the model with a high confidence. On the other hand, when the information, which is to be added to the representation, is autonomously extracted by the agent in an unsupervised manner (e.g. using a recognition procedure) there is a risk of propagating an erroneous extraction from the data through the learning process. Consequently, the representation could be corrupted with false data, resulting in a poorer performance and a less reliable representation. Robust mechanisms, which prevent such propagation of errors, play an important role in the process of continuous learning.

Robustness is thus another important capability of a cognitive system. A cognitive agent is equipped with imperfect sensors and effectors, and it is supposed to operate in a partially unpredictable real-world environment. Nevertheless, in the vision literature ideal training conditions are most commonly assumed. They enable the construction of a reliable model of the environment, which can then be used for determining outliers in new images and performing *robust recognition*. But yet, a fully operational cognitive system should be able also to build the representation under non-ideal conditions. *Robust learning* is, however, a greater challenge than robust recognition. During the learning process, a sufficient knowledge required for determining the relevance of visual input is still to be acquired. Therefore, robust learning should strongly be intertwined with the process of continuous learning, which could provide enough redundant information to determine statistically consistent data (this information can also be provided by a user acting as a tutor). Only the consistent data would then be used to build the representations of objects, enabling robust learning (and updating of the representations) under non-ideal real-world conditions.

Traditionally, learning is performed in a batch way. However, as discussed above, batch methods are in most cases not appropriate for cognitive vision systems; they are not biologically plausible and they are not feasible for very large sets of data. Nevertheless, they serve as a good basis for evaluation of incremental methods, since they have all original input data available and can thus extract the information, which is the most important for building a faithful representation. The incremental methods, on the other hand, have only one or a few original input images available, and only the representations of the previously seen (and learned) images. Therefore, one could expect that in general the incremental methods produce somehow inferior results than the batch methods. A very interesting and important question is, how severe these degradations of the results are? What factors influence the results; i.e. does the order of training images influence the results? These questions may be even more pronounced in the case of non-ideal training conditions: What happens if most of the training images are corrupted? How important is it that the images at the beginning of the learning sequence are of sufficient quality? When designing a representation to be used in a cognitive vision system, these questions need to be addressed.

A plethora of representations and approaches to visual learning and recognition has been proposed in the past. However, in the context of cognitive vision, it is very important that whatever the type of the representation is, it enables continuous and robust learning. In the following sections, we will present implementation of these principles in the case of subspace-based visual learning and recognition. Since, the PCA-based approach is originally designed as a batch method and is inherently non-robust to non-gaussian noise, we propose several extensions of the standard approach, which enable incremental and robust learning.

The paper is organized as follows. In Section 2, we present the subspace-based approach to visual learning and recognition, expose its shortcomings, review the previously proposed improvements, and outline our approach. In two following sections, we elaborate our approach in detail. In Section 3, we first present the basic algorithm for incremental learning. In Section 4, we extend this algorithm into a weighted algorithm, which considers temporal and spatial weights. Next, we present a special case of the algorithm, which can handle missing data. This algorithm is then advanced into a robust incremental algorithm, which can detect and discard inconsistencies in the input images. In Section 5, we present the experimental results. Finally, we summarize the paper, expose the contributions, and outline some work in progress.

2. Subspace-based modeling of objects and scenes

Appearance of an object combines effects of its shape, reflectance properties, pose in the scene, and illumination conditions [11]. It proves to be very difficult to separate all these factors from a set of images in order to obtain a view and illumination-invariant representation. In the appearance-based approach, the separation of these physical properties is circumvented. However, in order to obtain a complete appearance-based model of an object, one has to systematically observe the training object under different viewing and illumination conditions, which may result in a rather large set of images. Consequently, they have to be efficiently represented using a compact representation. A commonly used technique for compression of training images is based on principal component analysis (PCA) [12]. In PCA-based approach [11], an object is represented with the projections of the training images into the principal subspace, thus the object recognition is reduced to the searching of the closest point in this low-dimensional space.

2.1. Problem statement

The standard PCA approach in its original form has several shortcomings with respect to the premises mentioned in Section 1. PCA-based learning is traditionally performed in a batch mode, i.e. all training images are processed simultaneously, which means that all of them have to be given in advance; the obtained representation cannot be updated with new images without starting the process from the scratch. To make updating of the previously learned representation possible, one has to take an *incremental approach* to the principal component analysis.

Besides, in the standard PCA approach all pixels of an image receive an equal treatment. Also, all the training images

have equal influence on the estimation of principal axes. To enable a selective influence of individual images and pixels, PCA can be generalized into a *weighted approach*, which considers individual pixels and images diversely, depending on the corresponding weights.

PCA in its standard form is also intrinsically non-robust to non-gaussian noise. The recognition method can be extended such that non-gaussian noise in test images is detected, and the recognition is performed by considering relevant parts of the image only, providing that a consistent representation is given. However, if the training images are taken under non-ideal conditions, the non-desirable effects should be detected in the learning stage already and not included into the representation. Thus, we need a method for *robust learning*, which is able to detect inconsistencies in the training images and build the representations from consistent data only.

In Section 2.2, we will first review some existing extensions of the standard PCA approach, which can cope with the problems mentioned above. Then we will outline our approach to the solution of these problems, which will be described in detail in the following sections.

2.2. Previous work

Several authors faced the problem of decomposing large covariance matrices obtained from a huge number of training images. To overcome this problem several incremental algorithms for PCA have been proposed. The first algorithm for incremental PCA in the computer vision community was proposed by Murakami and Kumar [13]. Then, Chandrasekeran et al. proposed an algorithm, which is based on SVD updating [14]. Incremental singular value decomposition was often tackled also in the past (e.g. [16,17]) and recently [15]. All these methods keep the origin of the principal subspace in the origin of the image space, assuming that the mean of the input images is always zero. This is not true in general and this assumption may degrade the results of the classification [18]. By noting this problem, Hall et al. proposed a method for eigenspace updating, which sequentially shifts the origin of the eigenspace according to the new images, which are being added [18,19].

Several methods for weighted learning with different derivations but very similar realizations have also been proposed. Wiberg [20] has proposed a method for subspace learning when data are missing based on the weighted least squares technique. This method was later extended by Shum et al. [21]. Gabriel and Zamir [22] proposed a method for subspace learning with any choice of weights, where each data point can have a different weight determined on the basis of reliability. A similar approach was also used in the work of Sidenbladh et al. [23] and De la Torre and Black [24]. All these methods operate in a batch mode.

The only incremental methods that explicitly deal with spatial weights are the methods for incremental singular value decomposition of data with missing values introduced by Brand [15] and the method for incremental PCA very recently proposed by Li [25]. With respect to temporal weights, the latter method, as well as the incremental methods proposed by Liu and Chen [26] and Levy and Lindenbaum [27] are tailored for temporally weighted learning considering a decay parameter thus allowing newer images to have a larger influence on the estimation of the current subspace than the older ones. Therefore, their methods consider only a special case of temporal weights.

A severe limitation of the basic approach to the subspace visual modeling is its non-robustness to noise, occlusions, and cluttered background. Different approaches have been proposed to improve the robustness of the *recognition*: modular eigenspaces [28], eigenwindows [29], search-window [30], adaptive masks [31], *M*-estimation [32,33], hierarchial approach [34], and subsampling hypothesis-and-test-based approach [35]. However, all these methods introduce the robustness in the *recognition stage*. They assume that the images in the learning stage were ideal and that the correct visual model is available.

The *robust learning* is a much more difficult problem. Since, in the learning stage the model of the object or the scene is being built, there is no reliable model at the beginning of the learning process, which could be used to estimate outliers. The authors coped with this problem in different ways. Xu and Yuille proposed an algorithm, which introduced robustness on the image level [36]. During the learning stage, they discard images, which are inconsistent with the others. However, in many practical applications this is not satisfactory. The robustness on the pixel level should be assured meaning that only single pixels should be discarded and not the entire images. Gabriel and Zamir tried to solve this problem using a weighted singular value decomposition [22]. Recently, De la Torre and Black proposed a method for robust principal component analysis based on *M*-estimation [24,37]. This method, as well as the related method proposed by Skočaj et al. [38], perform well on images with sufficient temporal correlation, but are very time consuming. Very recently, Aanæs et al. [39] proposed a method for robust factorization, which is tailored for a different type of problems; however, some of its principles are relevant for robust eigenspace learning as well. These methods also operate in a batch mode, processing all training images simultaneously. Furthermore, they are executed in an iterative manner by repeating time consuming procedures on the entire set of training images. Therefore, the processing time is usually very long, and even becomes prohibitive when the number of training images is large. To overcome these problems, only very recently the incremental methods for robust estimation of the principal subspaces were proposed [25,40].

2.3. Our approach

The *incremental algorithm*, which we will present in this paper, produces the identical principal subspace as the method proposed by Hall et al. [18]. However, the subspace is obtained in a different way. A significant advantage of our method is that it is able to treat different images differently, which enables to extend it into a weighted incremental method. Furthermore, our

method maintains the low-dimensional representations of the previously learned images throughout the entire learning stage, therefore, enabling that each training image can be discarded immediately after the update.

Our *weighted incremental approach* considers arbitrary temporal and spatial weights, thus it is more general than the methods proposed in [25–27]. The incremental method is also adapted for learning from incomplete data, which in contrast to the method presented in [15], considers also the mean and updates its value at each step adequately. We also propose a method for incremental robust learning, which sequentially determines consistencies in the input images and reconstructs inconsistent pixels using the previously acquired knowledge. All the proposed methods will be described and evaluated in detail in the following sections.

3. Incremental PCA

In this section, we propose a method for incremental learning. It takes the training images sequentially and computes the new eigenspace from the subspace obtained in the previous step and the current input image.

Let us suppose that we have already built an eigenspace from the first n images. In the step $n+1$, we could calculate a new eigenspace from the *reconstructions*[1] of the first n input images and a new image using the standard batch method. The computational complexity of such an algorithm would be prohibitive, since at each step we would have to perform the batch PCA on a set of high-dimensional data. However, identical results can be obtained by using low-dimensional *coefficient vectors*[2] of the first n input images instead of their high-dimensional reconstructions, since coefficient vectors and reconstructed images encompass the same visual variability, i.e. they are just represented in different coordinate frames. Since, the dimension of the eigenspace is small, this algorithm is computationally very efficient.

The summarized procedure for one update of the current eigenspace is outlined in Algorithm 1.[3] This algorithm increases the dimension of the subspace by one. After the update, we can discard the least significant principal vector to preserve the dimension of the subspace [41].

The initial values of the mean image, the eigenvectors, and the coefficients can be obtained by applying the batch PCA on a small set of images. Alternatively, one can simply set the first training image as the initial eigenspace by assigning $\boldsymbol{\mu}^{(1)}=\mathbf{x}_1$, $\mathbf{U}^{(1)}=\mathbf{0}_{M\times 1}$, and $\mathbf{A}^{(1)}=0$. In this way, the algorithm is completely incremental, requiring only one image to be available at each time instant.

It is worth noting that this algorithm produces the identical principal subspace as the method proposed by Hall et al. [18]. However, the subspace is obtained in a different way. In contrast to our method, which between the learning steps passes coefficient vectors of all training images, the Hall's method passes only the eigenvalues. While one may consider this as an advantage, since less data is being passed from step to step and calculation of the covariance matrix is faster, this can also be disadvantageous, because the coefficients are not estimated and maintained during the learning process, thus less information is available. Our algorithm calculates the coefficients at that time instant, when the particular image is added to the model, and then maintains their values throughout the process of incremental learning. This is slightly slower, however, it has two advantages. The first advantage is, that each image can be discarded immediately after it has been used for updating the subspace. This is very appropriate (and possibly required) for on-line scenarios (e.g. navigation of mobile robots with limited memory resources). And finally, since more information is encompassed in the model, our method can be advanced into a method for weighted learning of eigenspaces, which can consider arbitrary temporal weights.

Algorithm 1. Incremental PCA

Input: Current mean vector $\boldsymbol{\mu}^{(n)}$, current eigenvectors $\mathbf{U}^{(n)}$, current coefficients $\mathbf{A}^{(n)}$, new input image $\mathbf{x}$.
Output: New mean vector $\boldsymbol{\mu}^{(n+1)}$, new eigenvectors $\mathbf{U}^{(n+1)}$, new coefficients $\mathbf{A}^{(n+1)}$, new eigenvalues $\boldsymbol{\lambda}^{(n+1)}$.

1: Project a new image $\mathbf{x}$ into the current eigenspace: $\mathbf{a}=\mathbf{U}^{(n)\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}^{(n)})$.
2: Renconstruct the new image: $\mathbf{y}=\mathbf{U}^{(n)}\mathbf{a}+\boldsymbol{\mu}^{(n)}$.
3: Compute the residual vector: $\mathbf{r}=\mathbf{x}-\mathbf{y}$. $\mathbf{r}$ is orthogonal to $\mathbf{U}^{(n)}$.
4: Append $\mathbf{r}$ as a new basis vector:

$$\mathbf{U}' = \begin{bmatrix} \mathbf{U}^{(n)} & \dfrac{\mathbf{r}}{\mathrm{r}} \end{bmatrix}$$

5: Determine the coefficients in the new basis:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}^{(n)} & \mathbf{a} \\ \mathbf{0} & \|\mathbf{r}\| \end{bmatrix}.$$

6: Perform PCA on $\mathbf{A}'$. Obtain the mean value $\boldsymbol{\mu}''$, the eigenvectors $\mathbf{U}''$, and the eigenvalues $\boldsymbol{\lambda}''$.
7: Project the coefficient vectors to the new basis: $\mathbf{A}^{(n+1)}=\mathbf{U}''^{\mathrm{T}}(\mathbf{A}'-\boldsymbol{\mu}''\mathbf{1}_{1\times n+1})$.
8: Rotate the subspace $\mathbf{U}'$ for $\mathbf{U}''$: $\mathbf{U}^{(n+1)}=\mathbf{U}'\mathbf{U}''$.
9: Update the mean: $\boldsymbol{\mu}^{(n+1)}=\boldsymbol{\mu}^{(n)}+\mathbf{U}'\boldsymbol{\mu}''$.
10: New eigenvalues: $\boldsymbol{\lambda}^{(n+1)}=\boldsymbol{\lambda}''$.

We will demonstrate the behavior of the proposed algorithm on a simple 2D example. The 2D input space contains 41 points

[1] An image can be reconstructed by transforming its subspace coefficient vector into the high-dimensional input image space.

[2] Coefficient vectors are composed of coefficients, which are obtained by projecting an image onto the principal axes spanning the eigenspace.

[3] $\boldsymbol{U}\in\mathbb{R}^{M\times k}$ denotes a matrix of k M-dimensional principal axes, $\mathbf{A}\in\mathbb{R}^{k\times n}$ is a matrix of n k-dimensional coefficient vectors, $\boldsymbol{\mu}\in\mathbb{R}^{M}$ is the mean image. Superscript denotes the step, which the data is related to ($\mathbf{U}^{(n)}$ denotes the values of $\mathbf{U}$ at the step n). $\mathbf{1}_{m\times n}$ denotes a $m\times n$ matrix of ones. $\|\mathbf{r}\|$ denotes the L_2 norm of the vector $\mathbf{r}$.

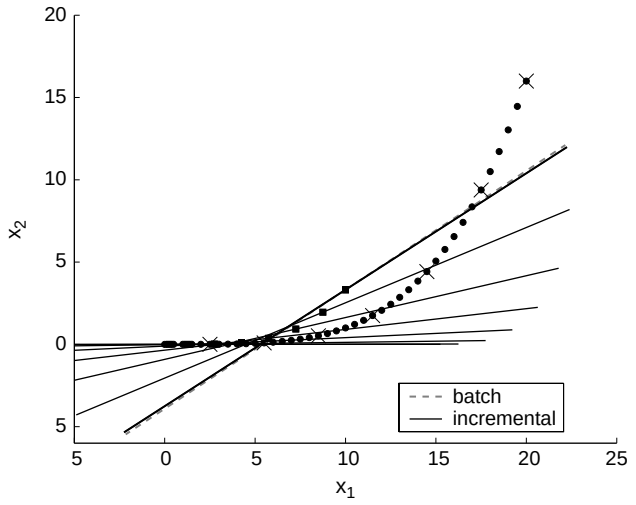

Fig. 1. Incremental learning.

shown as black dots in Fig. 1. The goal is to estimate 1D principal subspace, i.e. the first principal axis. The eigenspace is being built incrementally. At each step one point (from the left to the right) is added to the representation and the eigenspace is updated accordingly. Fig. 1 illustrates how the eigenspace evolves during this process. The principal axis, obtained at every sixth step, is depicted. The points, which were appended to the model at these steps, are marked with crosses. One can observe, how the origin of the eigenspace (depicted as a square) and the orientation of the principal axis change through time, adapting to the new points, which come into the process. At the end, the estimated eigenspace, which encompasses all training points, is almost identical to the eigenspace obtained using the batch method.

4. Weighted and robust approach

In order to achieve selective influence of pixels and images, the individual pixels as well as images can be weighted with different weights. In practice, it is useful to deal with two types of weights: *temporal* weights ${}^{t}\mathbf{w} \in \mathbb{R}^{1\times N}$, which put different weights on individual images and *spatial* weights ${}^{s}\mathbf{w} \in \mathbb{R}^{M}$, which put different weights on individual pixels within an image.[4]

4.1. Temporal weights

Temporal weights determine how important the individual images are for the estimation of principal subspace. If the temporal weight for one of the images is higher than the weights for the other images, the reconstruction error of this image should be smaller than the reconstruction errors of the other images. Similarly, the contribution of its principal components to the estimation of the variance should be larger in comparison with that of the other principal components.

From this observation, we can derive an algorithm for estimation of the principal subspace considering temporal weights. The principal axes, which maximize the *weighted variance* of the projections of the input images onto the principal axes, can be obtained by eigendecomposition (or, similarly, singular value decomposition) of the *weighted covariance matrix*. If the matrix $\hat{\mathbf{X}} \in \mathbb{R}^{M\times N}$ is composed from N re-scaled input vectors centered around the weighted mean

$$\hat{x}_j - = \sqrt{{}^{t}w_j}(x_j - \mu) \quad j = 1, \ldots, N \tag{1}$$

the weighted covariance matrix can be calculated as

$$C = \frac{1}{\sum_{j=1}^{N} {}^{t}w_j} \hat{\mathbf{X}}\hat{\mathbf{X}}^{\mathrm{T}}. \tag{2}$$

Using this algorithm, the estimated principal subspace does not depend on all training images equally. For instance, if a training image has the weight 2, while all the other images have the weight 1, the result of this algorithm equals the result of the standard PCA algorithm, which has two copies of the particular image in the training set.

It is quite straightforward to incorporate the temporal weights into the incremental algorithm. The core of this algorithm is still the standard batch PCA on low-dimensional data (step 6 of Algorithm 1). We can replace this standard batch PCA with the weighted algorithm, which considers temporal weights. This is feasible, because our incremental algorithm maintains low-dimensional coefficients of all input images throughout the process of the incremental learning (in contrast with the other incremental approaches). Therefore, the representation of each image can be arbitrarily weighted at each update.

To illustrate the behavior of the proposed algorithm, we put different weights on the training points from our simple 2D example. We set temporal weights to ${}^{t}w = j^2$, which gives a larger influence to the recent points. Fig. 2 depicts the evolution of the eigenspace. By comparing this figure with Fig. 1, it is evident how the weights affect the learning process. At the end

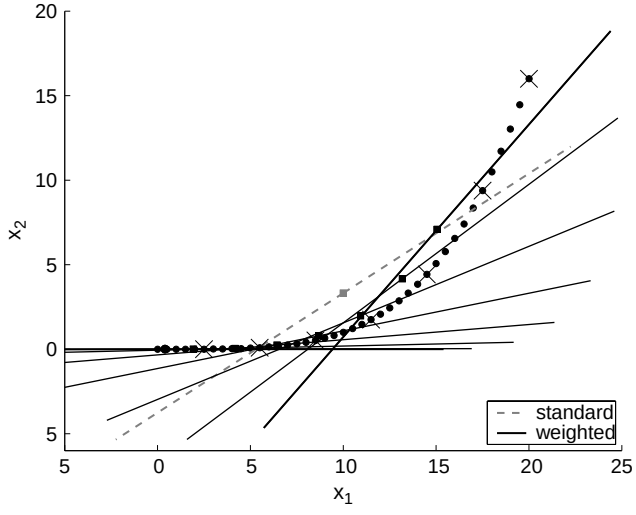

Fig. 2. Weighted incremental learning.

[4] The left superscript is used to distinguish between temporal (${}^{t}\mathbf{w}$) and spatial (${}^{s}\mathbf{w}$) weights.

of the learning sequence, the weighted mean vector is closer to the points at the end of the point sequence, since the weights of these points have higher values. The principal axis is oriented in such a direction that enables superior reconstruction of these points.

4.2. Spatial weights

Spatial weights control the influence of individual pixels within an image. Therefore, if a part of an image is not reliable or important for the estimation of principal components, its influence should be diminished by decreasing the weight of the corresponding pixels.

Incorporating spatial weights into the process of incremental learning is more complex. After the current eigenspace is updated with a new input image, this image is discarded and only its low-dimensional representation is preserved. Therefore, in the later stages we cannot associate weights to individual pixels. This can be done only during the update.

Let us assume that the weights range from 0 to 1. If a weight is set to 1, it means that the corresponding pixel is fully reliable and should be used as is. If a weight is set to 0, it means that the value of the corresponding pixel is irrelevant or erroneous. We can recover an approximate value of this pixel by considering the knowledge acquired from the previous images. By setting the weight between 0 and 1, we can balance between the influence of the value yielded by the current model and the influence of the pixel value of the input image.

We can achieve this by adding a preprocessing step to the update algorithm. First, we calculate the coefficients of the new image $\mathbf{x}$ by using the weighted method. Instead of using the standard projection, the coefficients a_j are obtained by solving an over-determined system of linear equations

$$\sqrt{{}^{s}w_i}\,x_i = \sqrt{{}^{s}w_i}\sum_{j=1}^{k} a_j u_{ij}, \quad i = 1 \ldots M \tag{3}$$

in the least squares sense. By reconstructing the coefficients we obtain the reconstructed image $\mathbf{y}$, which contains pixel values yielded by the current model. By blending images $\mathbf{x}$ and $\mathbf{y}$, considering spatial weights by using the following equation

$$x_i^{new} = {}^{s}w_i x_i + (1 - {}^{s}w_i) y_i, \quad i = 1 \ldots M, \tag{4}$$

we obtain the image which is then used for updating the current eigenspace. In this way, a selective influence of pixels is enabled also in the incremental framework.

4.3. Missing pixels

In the real-world applications, it is often the case that not all data is available. The values of some pixels are missing or they are totally non-reliable. Such pixels are referred to as *missing pixels*. Estimation of the principal subspace in the presence of missing pixels can be regarded as a special case of spatially weighted PCA where the weights of missing pixels are set to zero.

The blending step in the algorithm for weighted incremental learning reduces to the imputation of missing pixels. Before the current eigenspace is updated with the new image, the missing pixels have to be optimally filled in. Since, not all 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 with the values of the known pixels. Given the current principal subspace $\mathbf{U}^{(n)}$, which models the input data seen so far, the optimal location is the point in the missing pixels subspace which is closest to the principal subspace. This point is obtained by filling-in the missing pixels with the reconstructed values, which are calculated from the coefficients estimated from the known pixels only. Since, this coefficients reflect the novel information in the new image contained in the known pixels, we may assume that the prediction in the missing pixels will be fine as well. Such an improved image is the best approximation of the correct image that we can obtain from the information contained in the known pixels and in the current eigenspace.

Thus, the new image $\mathbf{x}$ if first projected into the current principal subspace $\mathbf{U}^{(n)}$ by solving a system of linear equations (3) arising from non-missing pixels. The obtained coefficient vector $\mathbf{a}$ is then reconstructed and the values in the reconstructed image are used for filling-in the missing pixels. The resulting image is then used for updating the current eigenspace.

A practically equivalent rule for imputation of missing pixels was proposed also by Brand in the context of incremental singular value decomposition [15]. As shown in [15], such a rule for imputation of missing pixels minimizes the distance of the vector representing a new image to the current subspace and maximizes the concentration of the variance in the top singular values. Consequently, such imputation rule minimizes the rank of the updated SVD guaranteeing parsimonious model of the data.

4.4. Robust approach

The developed method for subspace learning from incomplete data can be further extended in a method for robust learning. In the robust framework, the positions of 'bad' pixels are not known in advance, however, we are aware that images may contain outliers. We treat as outliers all pixels, which are not consistent with the information contained in other images. Since, at each step we have a current model of the object or scene seen so far, we can detect outliers in the new image and treat them as missing pixels.

This is achieved by projecting the new image into the current eigenspace in a robust manner. Instead of a simple projection, a robust procedure based on subsampling and hypothesize-and-select paradigm is used [35]. Coefficients are obtained mainly from inliers, thus their reconstructions tend to the correct values in outliers as well. Consequently, the reconstruction error in outliers is large, which makes their detection easier. Therefore, to make the incremental learning robust, we first detect outliers in a new image and replace their values with reconstructed values, which are good approximations of the correct values. Such an improved outlier-free

image is then used for updating the eigenspace. Providing that the outliers are detected during the learning process using the robust procedure, the obtained eigenspace is robust as well.

We can refer to this procedure as a 'hard' robust algorithm, since the pixels, which are detected as outliers, are replaced with reconstructed values, while the remaining pixels stay intact. An alternative 'soft' approach is to weight each pixel according to its reliability, which is determined with respect to the reconstruction error. The new image $\mathbf{x}$ is thus projected into the current eigenspace using the simple (and fast) standard projection and the obtained coefficients are used for reconstruction ($\mathbf{y}$). The obtained reconstruction error yields the spatial weights (e.g. ${}^{s}w_i = 1/(|x_i - y_i| + 1)$), which are then used by the weighted algorithm to update the current principal subspace.

To demonstrate the behavior of the robust incremental algorithm, we significantly changed the values of the second coordinate of five points in our 2D example. Fig. 3 shows that when the non-robust incremental method is used, these outlying points pull the origin in a wrong direction and incorrectly orient the estimated principal axis. On the other hand, the robust method sequentially detects the outlying coordinate values, replaces these values with their reconstructions (shown as circles) and updates the eigenspace accordingly. At the end, the principal axis obtained using this approach is very close to the optimal one.

An important advantage of such *incremental* method is that it processes only one image at each step, while the iterative batch robust methods process all images at each iteration. For that reason, the incremental method is significantly faster and enables robust learning from a large number of training images. Since, the model is being incrementally updated with new images, this method is very suitable for on-line applications as well.

On the other hand, it suffers (like all incremental methods) from a potential danger of error propagation. If the initial eigenspaces, built in the early stages of the learning process, encompass only a limited number of appearances of an object or a scene, then all the pixels in the subsequent images, which

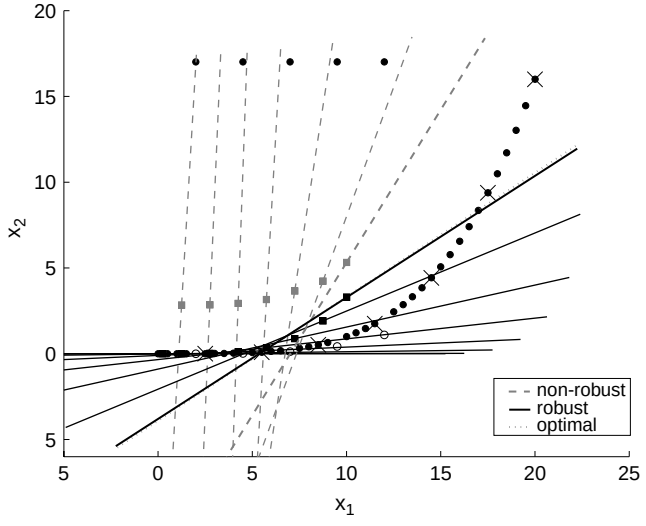

Fig. 3. Robust incremental learning.

significantly differ from the appearances of the first images, are considered as outliers and no novel information is added to the model. This particularly holds true for the 'hard' robust version of the updating algorithm. Therefore, the initial eigenspace, which is built in the beginning of the learning process, should be reliable and stable. It should roughly model heterogeneous appearances of an object or a scene and it should be obtained from a set of pixels containing as few outliers as possible. When the model encompasses a sufficient number of appearances it becomes more stable and this is no longer a problem [42].

5. Experimental results

5.1. Incremental PCA

Principal component analysis in its standard batch form is optimal in the sense of the squared reconstruction error. Thus, its incremental version necessarily degrades the results. But, how severe is this degradation? Are the results still usable? What additional factors influence the results? To clarify these issues and to evaluate the proposed algorithm we will explore the following questions:

- How much does the incremental method degrade the results in comparison with the batch method?
- How does discarding of training images influence the results?
- How does the order of the training images influence the results?

To answer these questions, we performed several experiments. First, we built eigenspaces of various dimensions from 720 images of twenty objects from the COIL database (Fig. 4a). Such number of training images can be processed using the batch method allowing a fair comparison. Fig. 4b depicts the mean squared reconstruction errors (MSRE) of the images reconstructed from the coefficients obtained by projecting the training images into the eigenspaces, which were built using the batch method (in the plots indicated as *batch*) and the proposed incremental method (*incXseq*). The results are very similar; MSRE obtained using the incremental method is only 3.1% worse on the average. The curve *incAseq* represents reconstruction errors of images obtained from the coefficients, which were calculated at that time instant, when the particular image was added to the model and then maintained throughout the process of incremental learning. Using this approach, an image can be discarded immediately after the model is updated. As one can observe, the squared reconstruction errors are still quite similar. In this case, the degradation of the results is 8.6% on the average.

In the first experiment, the images were coming into the learning process in a sorted order, i.e. first all images of the first object, then all images of the second object, and so on. In the second experiment, we changed this sequence by giving the training images to the learning process in a random order. Thus, the eigenspace in the early stage of the learning process

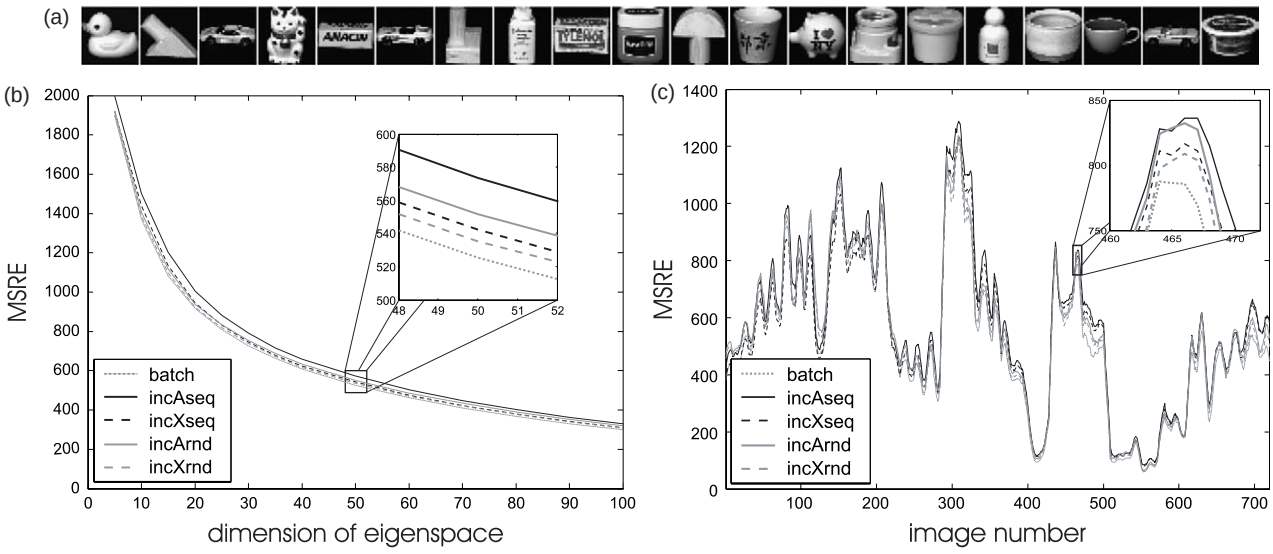

Fig. 4. (a) Training images. MSRE produced by the batch and four versions of the incremental approach: (b) for various dimensions of the eigenspace; (c) for dimension 50.

already encompassed images of several objects. Therefore, it was a good approximation of the final eigenspace. The incoming training images in the later stages were just refining the current eigenspace. Consequently, the results have improved. MSRE produced by *incArnd* and *incXrnd* approaches, are only 3.1 and 1.3% worse than the results of the batch method, respectively.

Fig. 4c shows the MSRE of all 720 images for the dimension of the eigenspace 50. One can observe, that the curves representing the incremental approach follow the curve produced by the batch method very closely without large deviations over the whole sequence of images.

All the results clearly indicate that the incremental method is almost as effective as the batch. In all experiments, the *squared* reconstruction error degraded for less than 10%, which means that the coefficients are still estimated well enough for most applications. It is also evident that the sequential order influenced the results. What really matters is the order of the training images in the early stages of the learning process. To obtain very good results, these images should be heterogeneous, encompassing different objects and views. This assures that the evolving eigenspace is rich and comprehensive enough in the beginning of the learning process already and that it is not specialized for representing a specific object only. In this way, the eigenspace can be adapted to the images of all objects more effectively.

5.2. Incremental weighted method

Then, we put on each image a weight, which was proportional to the second power of the image index, giving more influence to the objects and the images at the end of the image sequence. The results of *incremental and temporally weighted* method are depicted in Fig. 5. The reconstruction errors of the incremental weighted method (*WincA*, *WincX*) do not differ significantly from the results of the batch-weighted method (*Wbatch*). And certainly, the results of the batch and the incremental weighted methods are better than the results of the standard methods for images with larger weights. This is also reflected in better-weighted squared reconstruction errors as presented in Table 1.

Next, we present the results of the proposed *incremental method for learning from incomplete data* to improve the results of the *visually based localization* of a mobile robot [43]. In the learning stage, the representation of the environment (our lab in this case) is built from panoramic images taken from several locations. We can simulate the in-plane robot rotation by shifting cylindrical panoramic images and generating spinning images [44]. Three such views of two locations are depicted in Fig. 6a and b. We thus obtain all necessary views of the environment, which are used for building the representation using PCA. Later, in the localization stage, a novel image is taken and projected into the eigenspace. The location of the

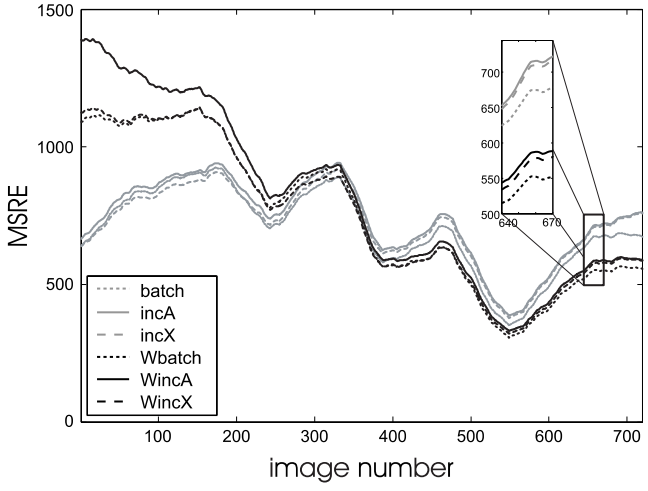

Fig. 5. Reconstruction errors of batch and incremental standard and weighted methods.

Table 1
Weighted reconstruction errors of batch and incremental methods

batch	*incA*	*incX*	*Wbatch*	*WincA*	*WincX*
617	658	648	554	583	565

robot is determined by searching for the closest projected training image.

However, due to the construction of the panoramic sensor, not only the environment is captured in the image, but also the holder of the panoramic mirror (the dark vertical bar in Fig. 6a and b) and the surface of the robot (lower part of the images). If the robot is oriented differently in the localization stage, the holder appears in a different position in the image, which makes the test image less similar to the correct training image and the localization can fail.

The proposed method offers a solution to this problem. Since we know, that the holder is not a part of the environment, we can mask it out during the learning, and learn only the parts, which belong to the environment. We can achieve this by using the incremental method for learning from incomplete data considering undesirable parts as missing pixels (see Fig. 6c).

In the learning stage, the robot was moving from one part of the lab to the other. In the localization stage, the robot returned to the starting position following approximately the same path in the opposite direction. The results are presented in Fig. 7. The gray levels represent coefficient errors; i.e. the distances between the projections of the test images (given in the x-axis) and the projections of the training image (y-axis). Since, the path of the robot was approximately the same as in the learning stage, we expect that the coefficient error would be minimal on the diagonal of the error matrix. Since, the standard approach incorporated in the representation also the vertical holder, which was in a different image position in the localization stage, the results of the standard method are not very good (the diagonal of the error matrix in Fig. 7a is very indistinct). In contrast, the proposed method did not incorporate the holder into the representation of the environment. Consequently, the values around the diagonal in Fig. 7b are significantly smaller, which makes the localization much more accurate and reliable.

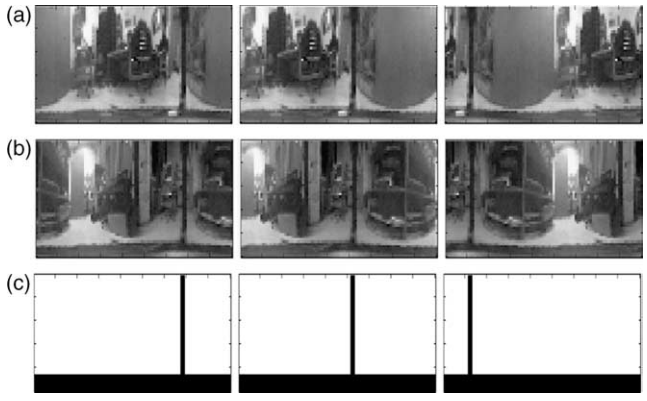

Fig. 6. (a and b) Spinning images from two locations. (c) Weights.

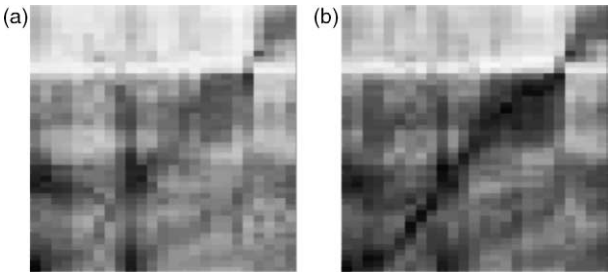

Fig. 7. Coefficient errors using (a) standard and (b) proposed method.

5.3. Incremental robust method

We will demonstrate the performance of the robust incremental algorithm on the problem of the background modeling. The goal of the background modeling is to build a model of the background by detecting and discarding the objects (foreground) in a sequence of images [37,38,45]. Due to its incremental nature and simplicity, the proposed incremental PCA is very well suited for solving such type of problems.

In order to obtain quantitative results of the proposed robust incremental method, we first tested its performance on the images with *known ground truth*. We synthetically applied gradual illumination changes and non-linear illumination changes (a shadow—the vertical 'cloud') to a set of 100 images. In addition, we added, as an outlier area, a square on a randomly chosen position in 80% of the images (see Fig. 8a, the first row). The goal was to learn the representation capturing the illumination variations (linear and non-linear) but discarding the outliers.

We tested several approaches to exhibit some properties of the proposed method. The results are given using two measures. The first measure is the mean squared reconstruction error of the reconstructed outlier-free (ground truth) images (Table 2). Besides MSRE, a precision/recall curve is given for each method in Fig. 8b. In addition, some reconstructed training images are visualized in Fig. 8a (rows 2–5).

First, we applied the standard batch method on ground truth images, i.e. training images without outliers (in table and plot indicated as *batchOnGT*), which produced optimal results. Then, we applied the standard batch method (*batchStd*) on the training images containing outliers, which generated poor results, since the standard method is sensitive to occlusions. Next, we applied the robust batch method [38] (*batchRob*), which produced better results. However, since significant occlusions were present in the training images, the results were still not satisfactory.

Then, we tested the proposed robust incremental method. First, we applied this method under the assumption that the occlusions were known and regarded as missing pixels (*robIncKnownOL*). The results are excellent; they are very close to the optimal ones. This means that the algorithm for updating the eigenspace works fine even if some data in the input images are missing and that the efficiency of the robust incremental algorithm mainly depends on the ability to detect outliers. It turns out that this ability significantly depends on

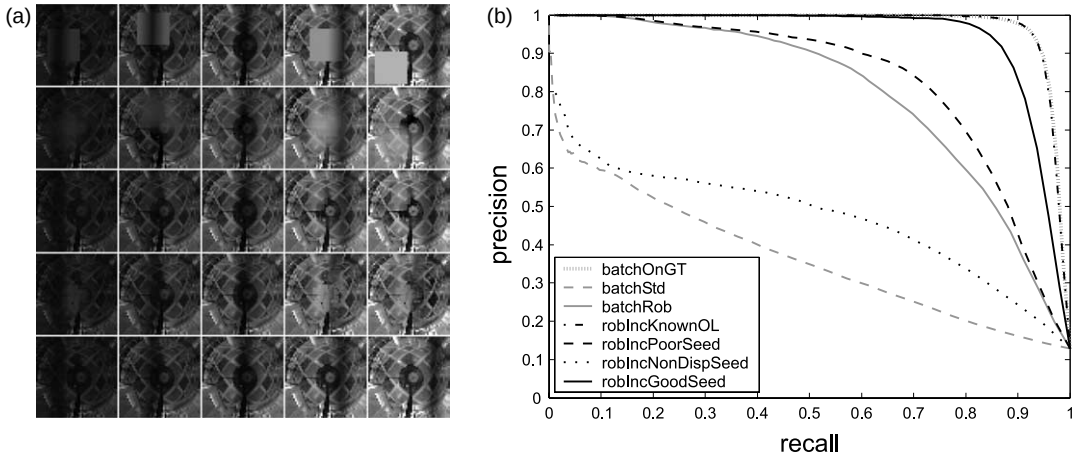

Fig. 8. (a) From top to bottom: training images, reconstructions using *batchStd*, *batchRob*, *robIncNonDispSeed*, and *robIncGoodSeed* approaches, respectively. (b) Precision/recall curves.

the initial stage of the learning process. If the seed (the initial eigenspace, which is used for the initialization of the incremental algorithm) is not reliable and is affected by occlusions (*robIncPoorSeed*), the results of the proposed 'hard' robust incremental method are not very good. If the seed is too small and is built from the training images, which are not dispersed over the whole image sequence (*robIncNonDispSeed*), the results are very poor. To demonstrate this, we built a seed using a few images from the first half of the image sequence. Consequently, the first half of the images were reconstructed well, however, the images from the end of the sequence were reconstructed poorly. Since, not even a rough appearance of these images was encompassed in the initial eigenspace, all the changes in these images were considered to be outliers and were not added to the representation. For this reason, the vertical cloud was not modeled correctly as can be observed in the fourth row of Fig. 8a. At last, we built the seed from the images with the lowest reconstruction errors (images without outliers), which were evenly dispersed over the whole image sequence (*robIncGoodSeed*). This approach produced excellent results, which are rather close to the optimal ones. This indicates that when the eigenspace, which is being updated, is stable enough, i.e. roughly encompassing different views of objects or scenes, the outliers in the training images are successfully detected and correctly reconstructed.

Then, we performed the experiments on the real-world PETS'2001 training sequences.[5] Six images from one sequence are depicted in Fig. 9a. The goal was to detect pedestrians, cars, and bikers, which are crossing the scene and to adapt the background model accordingly. We built the eigenbackground model consisting of eight eigenvectors. The backgrounds estimated at six time steps of the modeling process (i.e. the reconstructed training images, which were processed in those moments) obtained using three different approaches are presented in Fig. 9b–d.

[5] The images are publicly available on http://www.visualsurveillance.org/PETS2001.

First, we applied the proposed non-robust incremental method. One could expect that the outliers (pedestrians and cars), which significantly differ from the background, are considered as noise and are modeled with the eigenvectors corresponding to small eigenvalues and as such are not included in the principal subspace representations. However, this is not true in general; one can observe that the cars are still included in the background model in the third and fourth image in Fig. 9b.

Then, we applied the 'hard' robust method. This method successfully detected the pedestrians and cars, reconstructed their values and excluded them from the representation (Fig. 9c). In the subsequence of images around the images presented in the third and the fourth columns in Fig. 9, one car leaves the scene and another car parks in the spare lot. This changes are detected as 'foreground' and do not affect the background model. Using the 'hard' robust procedure, the background adapts only to smooth changes, which are not detected as outliers.

If a more flexible model is required, we can use the temporally weighted 'soft' robust method. In this case, the outliers are only down-weighted and are not completely replaced. As a consequence, they are not included in the model, if they appear only for a short period of time, however, if they appear for a longer period, they are gradually incorporated in the model of the background. This is evident from the last two images in Fig. 9d. The car, which has left the scene is not a part of the background any more, while the new car, which has parked in the spare lot, has been integrated into the current background. In this way, the eigenbackground

Table 2
MSRE obtained using different learning methods and seeds

batch			robInc			
OnGT	Std	Rob	KnownOL	PoorSeed	Non DispSeed	GoodSeed
1.7	61.1	29.8	2.0	21.2	166.0	2.9

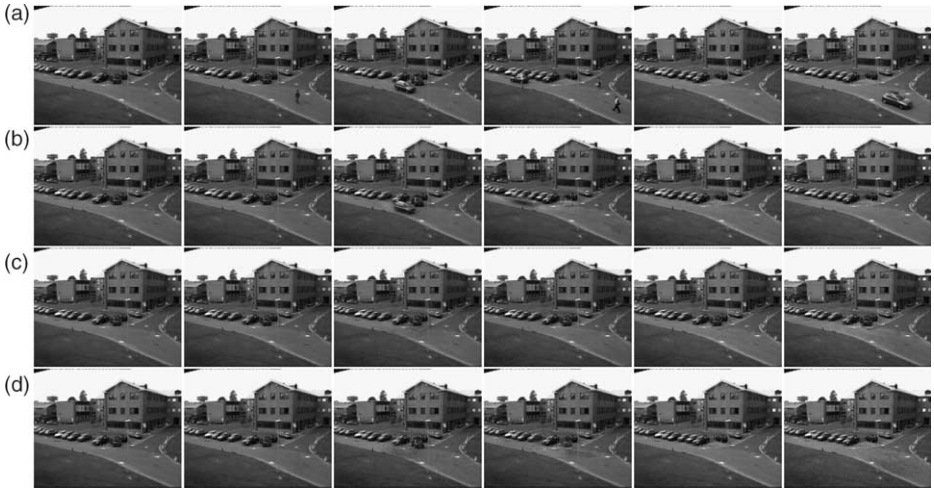

Fig. 9. (a) Six input images from PETS'2001 training sequence. Background extracted by (b) non-robust IPCA, (c) 'hard' robust IPCA, and (d) temporally weighted 'soft' robust IPCA.

model can be more adaptive, accommodating to the current appearance of the scene.

6. Conclusion

Learning is a fundamental capability of any cognitive vision system. In order to enable efficient operation of a cognitive agent in a real-world environment, visual learning has to be a continuous and robust process. Learning should be an incremental, open-ended, life-long process, which keeps continuously updating the representations by adapting them to the changes in the changing world. At the same time, this process should also be robust; it should be able to filter out undesirable input signals and to update the representations using relevant data only. It is important that regardless of the type of representation employed, a cognitive vision system should allow incremental and robust learning. After discussing these requirements in general in the beginning of this article, we then focused on the subspace-based representations, which in their original form do not allow continuous nor robust learning. To overcome these shortcomings, we extended the standard PCA approach.

We proposed a novel subspace method for weighted and robust incremental learning. The proposed incremental algorithm for PCA has the same general advantages over the batch method as other previously reported incremental approaches: it is significantly faster when the number of training images is high, and it enables updating the current eigenspace to allow for on-line learning. In addition, there are two advantageous features that make our method fundamentally distinct. Firstly, our method maintains the coefficients throughout the process of learning, thus the original images can be discarded immediately after the update. For some applications with a limited amount of memory resources (e.g. mobile platforms, wearable computing), this may be the only option. Using other methods, the images have to be kept in the memory until the end of the learning process, if we want to obtain their representations in the final eigenspace. And secondly, since our method maintains the coefficients of all images, it can be advanced into a weighted method, which considers an arbitrary temporal weight at each image at every step. Furthermore, the proposed weighted method also handles spatial weights, which can be set for each pixel in every image separately. Finally, by adding the robust preprocessing step, the method is suited for visual learning in non-ideal training conditions as well. Due to its incremental nature, this method for robust learning of eigenspaces is significantly faster than previously proposed batch methods.

The method is suitable for continuous on-line learning, where the model adapts to input images as they arrive. The algorithm is flexible, since it is able to treat each pixel and each image differently. Therefore, more recent (or more reliable, or more informative, or more noticeable) images can have a stronger influence on the model then others. The principles of short-term and long-term memory, forgetting, and re-learning can be implemented and investigated. These topics are the subject of our ongoing research along with applying these principles to other types of representations.

Acknowledgements

This research has been supported in part by the following funds: Research program Computer Vision P2-0214 (RS), EU FP6-004250-IP project CoSy, EU FP6-511051-2 project MOBVIS, CONEX project, and SI-A project.

References

[1] D. Vernon et al. A research roadmap of cognitive vision. Technical report, ECVision: European research network for cognitive computer vision systems, Available from: http://www.ecvision.org/research_planning/Research_Roadmap.htm, February 2005.

[2] G. Granlund, Cognitive vision background and research issues. Linköping University. Available from: http://www.ecvision.org/research_planning/CVResearchIssues.pdf, November 2002.
[3] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for visual object class recognition. In ECCV 2000, p. 18-32, 2000.
[4] B. Schiele, J.L. Crowley, Recognition without correspondence using multidimensional receptive field histograms, IJCV 36 (1) (2000) 31–50.
[5] D.G. Lowe, Local feature view clustering for 3D object recognition, In CVPR , p. I (682–688) 2001.
[6] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In ECCV 2002, p. 113–130, 2002.
[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In CVPR 2003, p. II: 264–271, 2003.
[8] B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In DAGM 2004, pp. 145–153, Aug. 2004.
[9] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In ECCV 2004, vol. I, pp. 40–54, May 2004.
[10] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In ECCV 2004, vol. II, pp. 71–84, May 2004.
[11] S.K. Nayar, H. Murase, A. Nene, Parametric appearance representation, Early Visual Learning, pp. (1996) 131–160.
[12] H. Hotelling, Analysis of a complex of statistical variables into principal components, Journal of Educational Psychology 24 (1933) 417–441.
[13] H. Murakami, V. Kumar, Efficient calculation of primary images from a set of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (5) (1982) 511–515.
[14] S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkeler, H. Zhang, An eigenspace update algorithm for image analysis, Graphical Models and Image Processing 59 (5) (1997) 321–332.
[15] M. Brand, Incremental singular value decomposition of uncertain data with missing values, In ECCV 2002, vol. I, pp. 707–720 1334 (2002).
[16] J.R. Bunch, P. Nielsen, Updating the singular value decomposition, Numerische Mathematik 31 (1978) 111–129.
[17] M. Gu, S.T. Eisenstat, A stable and fast algorithm for updating the singular value decomposition. Tech. report YALEU/DCS/RR-966, Department of Computer Science, Yale University, New Haven, 1993.
[18] P. Hall, D. Marshall, and R. Martin. Incremental eigenanalysis for classification. In British Machine Vision Conference, vol. 1, pp. 286–295, September 1998.
[19] P. Hall, D. Marshall, R. Martin, Merging and spliting eigenspace models *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1048 2000.
[20] T. Wiberg, Computation of principal components when data are missing, In Proc. Second Symp. Computational Statistics, pp. 229–236, 1976.
[21] H. Shum, K. Ikeuchi, R. Reddy, Principal component analysis with missing data and its application to polyhedral object modeling, IEEE *Trans. on Pattern Analysis and Machine Intelligence*, 17(9):854–867, 1995.
[22] K. Gabriel, S. Zamir, Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(21):489-498 1979.
[23] H. Sidenbladh, F. de la Torre, J. Black, A framework for modeling the appearance of 3D articulated figures, In AFGR00 368–375, 2000.
[24] F. De la Torre, M.J. Black, A framework for robust subspace learning, IJCV 54 (1) (2003) 117–142.
[25] Y. Li, On incremental and robust subspace learning, Pattern recognition 37 (2004) 1509–1518.
[26] X. Liu and T. Chen. Shot boundary detection using temporal statistics modeling. In: ICASSP 2002, Orlando, FL, USA, May 2002.
[27] A. Levy, M. Lindenbaum, Sequential karhunen-loeve basis extraction and its application to images, IEEE Trans. on Image Processing, 9:1371–1374, June 2000.
[28] A.P. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In CVPR 1994, pp. 84–91, July 1994.
[29] K. Ohba, K. Ikeuchi, Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects, IEEE Trans Pattern Analysis and Machine Intelligence 19 (9) (1997) 1043–1048.
[30] H. Murase, K. Nayar, Image spotting of 3D objects using parametric eigenspace representation, In SCIA95 325–332,1995.
[31] J.L. Edwards, H. Murase, Coarse-to-fine adaptive masks for appearance matching of occluded scenes, MVA 10 (5–6) (1998) 232–242.
[32] 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.
[33] R. Dayhot, P. Charbonnier, F. Heitz, Robust visual recognition of color images, CVPR2000 685–690 (2000).
[34] R.P.N. Rao, Dynamic appearance-based recognition, In CVPR , pp. 540–546, 1977.
[35] A. Leonardis, H. Bischof, Robust recognition using eigenimages, Computer Vision and Image Understanding 78 (2000) 99–118.
[36] L. Xu, A. Yuille, Robust principal component analysis by self-organizing rules based on statistical physics approach, IEEE Trans. Neural Networks, 6(1):131–143, 1995.
[37] F. De la Torre, J. Black, Robust principal component analysis for computer vision, ICCV, 362–369, 2001.
[38] D. Skočaj, H. Bischof, and A. Leonardis. A robust PCA algorithm for building representations from panoramic images. In ECCV 2002, vol. IV, pp. 761–775, 2002.
[39] H. Aanæs, R. Fisker, K. Åström, J.M. Carstensen, Robust factorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (9) (2002) 1215–1225.
[40] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In ICCV 2003, II:1494–1501, 2003.
[41] M. Artač, M. Jogan, and A. Leonardis. Incremental PCA for on-line visual learning and recognition. In ICPR 2002, vol. 3, pp. 781–784, 2002.
[42] D. Skočaj. Robust subspace approaches to visual learning and recognition. PhD thesis, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, 2003.
[43] M. Artač, M. Jogan, and A. Leonardis. Mobile robot localization using an incremental eigenspace model. In ICRA 2002, pp. 1025–1030, 2002.
[44] M. Jogan, A. Leonardis, (2000) Robust localization using the eigenspace of spinning-images, IEEE Workshop on Omnidirectional Vision, 37–44.
[45] N.M. Oliver, B. Rosario, A.P. Pentland, A bayesian computer vision system for modeling human interactions, IEEE Trans. Pattern Anal. Machine Intelligence 22 (8) (2000) 831–843.