

# Incremental PCA for On-line Visual Learning and Recognition \*

Matej Artáč, Matjaž Jogan and Aleš Leonardis  
Faculty of Computer and Information Science,  
University of Ljubljana,  
Tržaška 25, SI-1001 Ljubljana, Slovenia  
{matej.artac, matjaz.jogan, ales.leonardis}@fri.uni-lj.si

## Abstract

*The methods for visual learning that compute a space of eigenvectors by Principal Component Analysis (PCA) traditionally require a batch computation step. Since this leads to potential problems when dealing with large sets of images, several incremental methods for the computation of the eigenvectors have been introduced. However, such learning cannot be considered as an on-line process, since all the images are retained until the final step of computation of space of eigenvectors, when their coefficients in this subspace are computed. In this paper we propose a method that allows for simultaneous learning and recognition. We show that we can keep only the coefficients of the learned images and discard the actual images and still are able to build a model of appearance that is fast to compute and open-ended. We performed extensive experimental testing which showed that the recognition rate and reconstruction accuracy are comparable to those obtained by the batch method.*

## 1 Introduction

Eigenspace-based methods for visual learning and recognition use the Principal Components Analysis (PCA) [4] in order to obtain a set of so-called *eigenvectors*, which span the space of eigenvectors. Images are then represented as points in this subspace, where point coordinates are coefficients obtained by projecting the images onto the space. PCA is usually performed off-line, in a batch mode. More specifically, we first acquire all the training images, compute PCA, and afterwards project the images onto the subspace in order to compute the coefficients.

The drawback of the batch PCA method is that when the image set is large, the first step, i.e., the PCA computa-

tion, becomes prohibitive. Another problem is that, in order to update the subspace of eigenvectors with another image, we have to recompute the whole decomposition from scratch. To overcome these problems, several methods have been introduced that allow for an incremental computation of eigenimages [1, 6, 3]. These methods take the training images sequentially and compute the new set of eigenimages based on the previous space of eigenvectors and the new input image.

Although the eigenimages are computed incrementally, we are still unable to use the model until the training samples are represented in the eigenspace. However, we can project the input training image and discard it immediately after the subspace is updated. The resulting coefficients, in case we do not keep all of the eigenimages, represent only an approximation of the original image. Since these coefficients constantly change in the subsequent iterations of incremental building, also the representation of the images changes. This may cause the overall eigenspace representation to deteriorate.

In this paper we propose a method that allows for complete incremental learning using the eigenspace approach. We propose to use the incremental PCA algorithm and to project every input image immediately onto the subspace. Each input image is then discarded, and its representation consists only of the corresponding coefficients stored. Therefore, we can immediately use the model for the task at hand, e.g., recognition. In this paper we study how to update the coefficients stored in the subspace in order to bound the overall error of the representation.

In our experiments on large image databases we show that the resulting model is comparable in performance to the model computed with the batch method. Furthermore, the incremental model can easily be improved by re-learning the data.

This paper is organized as follows. In Section 2 we introduce the standard procedure of building the space of eigenvectors and an incremental PCA method. Then we describe our novel approach and explain in details how to apply it. In

\*The authors acknowledge the support from the Ministry of Education, Science, and Sport of Republic of Slovenia (Research Program 506).

Section 3 we present the results of the experiments which show the feasibility of our approach. We summarize the paper in Section 4.

## 2 PCA and incremental PCA

In this section we briefly outline the standard procedure of building the space of eigenvectors from a set of training images and its incremental version.

We represent input images as normalized image vectors  $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$ ;  $i = 1 \dots n$ , where  $m$  is the number of pixels in the image and  $n$  is the number of images. We compute the eigensystem by solving the SVD of the covariance matrix  $C \in \mathbb{R}^{m \times m}$  composed as

$$C = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top, \quad (1)$$

where  $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$  is the mean image vector.

The eigenvectors  $\mathbf{u}_i$ ,  $i = 1 \dots n$  corresponding to non-zero eigenvalues of the covariance matrix span a subspace of a maximum  $m$  dimensions. We can then choose a subset of only  $k$  eigenvectors corresponding to the largest eigenvalues to be included in the model. Each image can thus be optimally approximated in the least-squares sense up to a pre-defined reconstruction error. Every input image  $\mathbf{x}_i$  projects into some point  $\mathbf{a}_i$  in the  $k$ -dimensional subspace, spanned by the selected eigenvectors (eigenimages) [4].

Let us now turn to the incremental version of the algorithm. We assume we have already built a set of eigenvectors  $U = [\mathbf{u}_j]$ ,  $j = 1 \dots p$ , after having used the images  $\mathbf{x}_i$ ,  $i = 1 \dots n$  as an input. The corresponding eigenvalues are  $\lambda = \text{diag}(\Lambda)$  and the mean image is  $\bar{\mathbf{x}}$ . Incremental building requires to update these eigenimages to take into account a new input image  $\mathbf{x}_{n+1}$ .

Here we briefly summarize the method described in [2]. First, we update the mean:

$$\bar{\mathbf{x}}' = \frac{1}{n+1} (n\bar{\mathbf{x}} + \mathbf{x}_{n+1}). \quad (2)$$

We then update the set of eigenvectors by adding a new vector and applying a rotational transformation. In order to do this, we first compute the orthogonal residual vector  $\mathbf{h}_{n+1} = (U\mathbf{a}_{n+1} + \bar{\mathbf{x}}) - \mathbf{x}_{n+1}$  and normalise it to obtain  $\hat{\mathbf{h}}_{n+1} = \frac{\mathbf{h}_{n+1}}{\|\mathbf{h}_{n+1}\|_2}$  for  $\|\mathbf{h}_{n+1}\|_2 > 0$  and  $\hat{\mathbf{h}}_{n+1} = \mathbf{0}$  otherwise. The new matrix of eigenvectors  $U' \in \mathbb{R}^{m \times (k+1)}$  is computed by

$$U' = [U \hat{\mathbf{h}}_{n+1}] R, \quad (3)$$

where  $R \in \mathbb{R}^{(k+1) \times (k+1)}$  is a rotation matrix.  $R$  is the solution of the eigenproblem of the following form:

$$D R = R \Lambda'. \quad (4)$$

We compose  $D \in \mathbb{R}^{(k+1) \times (k+1)}$  as

$$D = \frac{n}{n+1} \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} + \frac{n}{(n+1)^2} \begin{bmatrix} \mathbf{a}\mathbf{a}^\top & \gamma\mathbf{a} \\ \gamma\mathbf{a}^\top & \gamma^2 \end{bmatrix}, \quad (5)$$

where  $\gamma = \hat{\mathbf{h}}_{n+1}^\top (\mathbf{x}_{n+1} - \bar{\mathbf{x}})$  and  $\mathbf{a} = U^\top (\mathbf{x}_{n+1} - \bar{\mathbf{x}})$ .

There are other ways to construct  $D$ . However, only the method described in [2] allows for the updating of mean.

### 2.1 Updating the image representations

To achieve a simultaneous on-line learning and recognition process, at each step of the incremental PCA the resulting model has to contain the points corresponding to images that had been previously included in the representation. Our contribution thus focuses on how to update the coefficients of images during the updating of the subspace without having to retain the original images.

During the process of learning at a discrete time  $n$ , we have learnt  $n$  images  $\mathbf{x}_i$ ,  $i = 1 \dots n$ , which has produced a space of  $k$  eigenvectors  $\mathbf{u}_j$ ,  $j = 1 \dots k$ . The images are represented with coefficient vectors  $\mathbf{a}_{i(n)}$  and can be approximated by  $\hat{\mathbf{x}}_{i(n)} = U\mathbf{a}_{i(n)} + \bar{\mathbf{x}}$ . The values of the coefficient vectors are dependent on the sequence of images added, hence the subscript  $(n)$ , which denotes the discrete timeline of adding the images.

When a new observation  $\mathbf{x}_{n+1}$  arrives, we compute the new mean using (2), we construct the intermediate matrix  $D$  (5), and solve the eigenproblem (4). This produces a new subspace of eigenvectors  $U'$ .

In order to remap the coefficients  $\mathbf{a}_{i(n)}$  into this new subspace, we first compute an auxiliary vector  $\eta$

$$\eta = [U \hat{\mathbf{h}}_{n+1}]^\top (\bar{\mathbf{x}} - \bar{\mathbf{x}}'), \quad (6)$$

which is then used in the computation of all coefficients

$$\mathbf{a}_{i(n+1)} = (R')^\top \begin{bmatrix} \mathbf{a}_{i(n)} \\ 0 \end{bmatrix} + \eta, \quad i = 1 \dots n+1. \quad (7)$$

The above transformations produce a representation with  $k+1$  dimensions. The approximations of previously learnt images  $\hat{\mathbf{x}}_{i(n)}$ ,  $i = 1 \dots n$  and the new observation  $\mathbf{x}_{n+1}$  can be fully reconstructed from this representation. However, due to the *increase* of the dimensionality by one, the representation requires more storage capacity. We can therefore decide to *keep* the dimensionality  $k$  by preserving only the first  $k$  eigenvectors and the corresponding elements of the coefficient vectors.

If we keep a  $k$ -dimensional eigenspace, we discard a certain amount of information. Therefore, we need a criterion for balancing the growing of the eigenspace on the one side and the overall reconstruction error on the other side. In the literature, several criterions have been used, e.g., the fraction of the smallest eigenvalue in the sum of all eigenvalues [2]. However, what we propose is to compute the overall reconstruction error that is caused by keeping a  $k$ -dimensional eigenspace and discarding the eigenvector  $u'_{k+1}$ .

Since  $\lambda'_{k+1}$  represents the variation in the direction of the eigenvector  $u'_{k+1}$ , we can use  $(n+1)\lambda'_{k+1}$  as our criterion value. Based on this value, we decide whether adding  $u'_{k+1}$  significantly improves our representation; if not, we keep only  $k$  eigenvectors. Therefore, if this value exceeds an absolute threshold, we add a new dimension. As we will show, by using this criterion we can keep the overall reconstruction error bounded.

### 3 Experiments

We carried out a set of experiments to test the behaviour of the on-line visual learning. We used two types of input images. As the first set, we used images from the Columbia Object Image Library (COIL-20) [5]. The set consisted of images of 20 objects rotated about their vertical axis, resulting in 72 images per object (Fig. 1(a)). We used these images for estimating the performance of the incremental eigenspace representation for object recognition, which will be explained later.

The second set of images consisted of panoramic views of an indoor environment, as shown in the Fig. 1(b). These images were acquired by a mobile robot equipped with a panoramic camera setup and have been used in our experiments to localize the robot, i.e., to recognize the momentary input image by matching it to the eigenspace model of images acquired earlier in a training stage. Hyperbolic images obtained from the camera are unwrapped to a cylindrical shape, so that we can simulate images in multiple orientations by shifting the pixels row-wise. We therefore generated  $t$  rotated images from one original image, where  $t$  is the number of pixels in a row.

**Reconstruction error** We used the set of panoramic images to test the quality of the eigenspace representation when constructed incrementally by using our method. During the process of building the eigenspace, we monitored the quality of the momentary representation on a subset of training images. At each step of updating the eigenspace, we calculated the reconstruction error for these images by summing the difference between the original image and its reconstruction. Fig. 2 shows the dynamics of this value; one

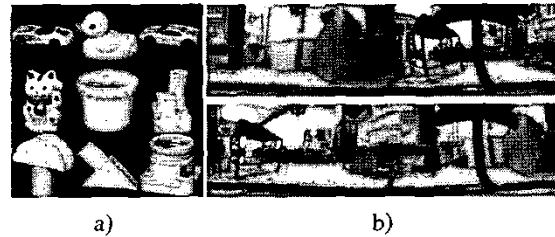


Figure 1. Sample images used for testing: (a) from COIL database and (b) cylindrical panoramic images.

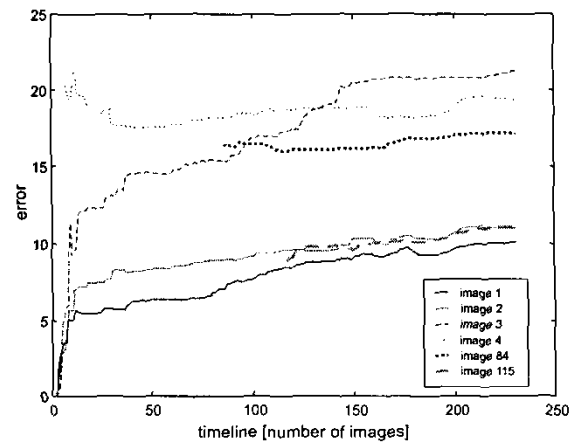


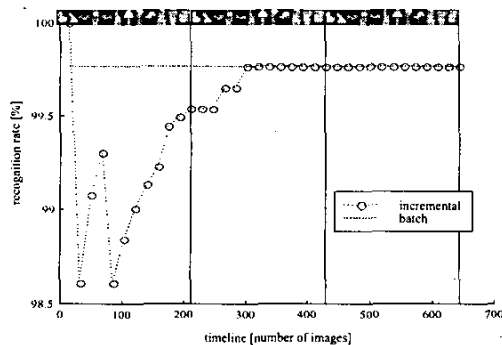
Figure 2. Reconstruction errors for a subset of training panoramic images during the process of learning.

can see from the graph that for all of the images, the reconstruction error is bounded. This indicates that the representation does not deteriorate dramatically during the process of learning.

**Object recognition and re-learning** We used the COIL database for experiments in object recognition. The eigenspace was build from a subset (i.e., every fourth image) of views of each of the objects. The remaining images were then used as a testing set for estimating the recognition ratio and the performance of pose estimation.

While experimenting we came across an interesting issue whether re-learning the same images would improve the quality of the representation. We therefore extended both tests by learning the same sequence of images repetitively three times in a row. On the second and third run we replaced the coefficients in the subspace with the new ones.

We performed object recognition by searching for the



**Figure 3. Recognition rate during the learning and two additional re-learning runs.**

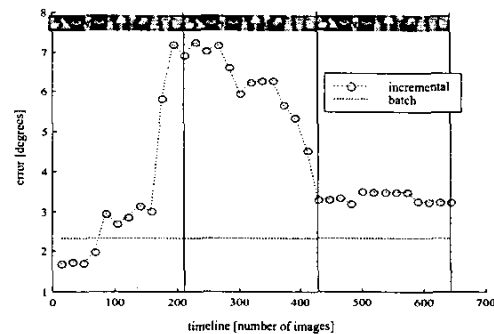
point in the momentary representation that is the closest to the projection of the image depicting the unknown object. Since we wanted to test the performance during the incremental process of building the eigenspace, we had to measure it only for the objects that were already included in the representation. We therefore first added all the training images of one object and then tested the performance on the resulting eigenspace.

At each step we calculated the *recognition ratio* by dividing the number of correct estimations with the number of the test images. We also calculated the *pose estimation error* as the difference between the true pose of the test image and the estimated one. To refine the estimation of the pose, we interpolated the projections representing a particular object. In this way, we obtained a denser spline, representing images at a step of  $2^\circ$ .

The first third of the graph on Fig. 3 depicts the recognition ratio; on top of the graph one can see the object whose images were added in the corresponding time interval. One can see that the batch method performs slightly better, yet the loss of accuracy in the incremental method is negligible.

The vertical line on the same figure denotes the ending of one learning iteration and the beginning of the next iteration. We can see that when we re-learn the same series of input images for the second time, the performance of the incremental method converges to that of the batch method. This result suggests that the subspace can describe the observations better after each iteration of learning.

The graph on Fig. 4 shows the average pose estimation errors during the learning. Due to the interpolation of coefficients, even though the training was performed at a step of  $20^\circ$ , the average estimation error stayed at the fraction of this resolution. Again, the performance of the incremental method is only slightly worse than that of the batch method and it gets better with re-learning.



**Figure 4. Pose estimation error during the learning and two additional re-learning runs.**

## 4 Conclusion

In this paper we introduced a method for on-line visual learning and recognition using the eigenspace approach. With our approach it is possible to use the model during the training stage, which bridges the gap between the learning and the training stage. This is extremely important in applications such as mobile robotics, where appearance of the environment has to be learnt, while the knowledge acquired so far already has to be used for navigation.

Since the model is open-ended, it is always possible to enrich it with new knowledge. In the off-line learning approach, the only way to do that is to build the model from scratch. As we show, it is feasible to keep only the subspace representations of the input images throughout the learning process. We showed that by constantly monitoring the error, our method manages to preserve the important features during the learning, which enables highly accurate recognition (in case of objects) and mobile robot localization.

## References

- [1] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, September 1997.
- [2] P. Hall, D. Marshall, and R. Martin. Incremental eigenanalysis for classification. In *British Machine Vision Conference*, volume 1, pages 286–295, September 1998.
- [3] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *PAMI*, 22(9):1042–1048, 2000.
- [4] H. Murase and S. Nayar. Visual learning and recognition of 3-D objects from appearance. *IJCV*, 14(1):5–24, January 1995.
- [5] S. Nene, S. Nayar, and H. Murase. Columbia object image library: COIL, 1996.
- [6] J. Winkeler, B. S. Manjunath, and S. Chandrasekaran. Subset selection for active object recognition. In *CVPR*, volume 2, pages 511–516. IEEE Computer Society Press, June 1999.