Incremental PCA for On–line Visual Learning and Recognition *

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

The drawback of the batch PCA method is that when the image set is large, the first step, i.e., the PCA computation, 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 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 $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$; $i = 1 \ldots n$, 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

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

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

The eigenvectors $\mathbf{u}_i$, $i = 1 \ldots n$ 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 predefined 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 $U = [\mathbf{u}_j]$, $j = 1 \ldots p$, after having used the images $\mathbf{x}_i$, $i = 1 \ldots n$ as an input. The corresponding eigenvalues are $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$ 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:

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

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

$$U' = \left[U \ \hat{\mathbf{h}}_{n+1} \right] R , \tag{3}$$

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:

$$D\, R = R\, \Lambda' . \tag{4}$$

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

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

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 $\mathbf{x}_i$, $i = 1 \ldots n$, which has produced a space of k eigenvectors $\mathbf{u}_j$, $j = 1 \ldots k$. The images are presented with coefficient vectors $\mathbf{a}_{i(n)}$. The values of the vectors are dependent on the number of images added, hence the subscript (n).

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 $\boldsymbol{\eta}$

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

which is then used in the computation of all coefficients

$$\mathbf{q}_{i(n+1)} = (R')^\top \begin{bmatrix} \mathbf{q}_{i(n)} \\ 0 \end{bmatrix} + \boldsymbol{\eta} , \ i = 1 \ldots n+1. \tag{7}$$

The above transformations produce a representation with $k+1$ dimensions. Approximations $\mathbf{x}_{i(n)}$, $i = 1 \ldots n$ and $\mathbf{x}_{n+1}$ can be fully reconstructed from this representation. However, due to the *increase* of the dimensionality, it requires more storage capacity.

We can therefore decide to *preserve* the dimensionality k by keeping only the first k eigenvectors and the corresponding elements of the coefficient vectors.

Since upon the preservation of the dimensionality a certain amount of information is lost, we need a criterion for making this decision. Authors [2] have used values, such us a fraction of the smallest eigenvalue in the sum of all eigenvalues. We propose to compute the overall increase of the reconstruction error caused by the preservation of the dimensionality. Since $\lambda_{k+1(n+1)}$ represents the variation in the direction of the increased dimension, we can use $(n+1)\lambda_{k+1(n+1)}$ as our criterion value. If this value exceeds an absolute threshold, we add a new dimension.

3 Experiments

We carried out a set of experiments to test the behaviour of the on-line visual learning. We used two sets of input images. The first set consisted of panoramic images of an indoor environment, as shown in the Fig. 1b). 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 during a training process. Hyperbolical images obtained from the camera are unwarped 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. We built the subspace of eigenvectors as described above. The projections of the images in the subspace preserve the original labels.

During the process of building the subspace we monitored the quality of the momentary representation of the training images. We used the reconstruction error measure which is computed by subtracting the reconstructed image from the original one and summing the error. Fig. 2 shows the dynamics of this value for a selected set of individual images. We can see from the results that the error changes in a controlled manner. This indicates the representation does not deteriorate dramatically during the learning process.

The second set of input images used was from the Columbia Object Image Library (COIL-20) [5]. This set consists of images of objects rotated about their vertical axis, making up 72 images per object. Fig. 1a) depicts samples of images used in our experiments.

Each image has a label indicating the object on the image, and the object's pose in degrees. For each set of images we selected a subset (i.e., every fourth image in the series) for the training. We tested the object recognition and orientation estimation of our method.

a) b)

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

An object on a test image is said to be *recognised* when its label describes the same object as the label on the point in the subspace closest to the projection of this image. For each object we count the number of recognised cases. By dividing this number with the number of all test images representing the object, we obtain the *recognition ratio*. Fig. 3 shows the average values for the learnt objects. For this experiment, only the result shown on the first third of the chart is relevant. The points on the chart represent the average recognition ratio for the objects that have been learnt thus far. Batch method at the same level of compression performs better, yet the loss of accuracy is not considerable.

By retrieving the pose value from the label of the closest point in the subspace we perform the *pose estimation*. The *pose estimation error* is the difference between the true pose of the test image (ground truth) and the estimated one. The size of this error depends on the sampling resolution of the training images. In our case we sampled the data at the rotation step of 20°. To refine the pose estimation, we grouped the projections representing a particular object, and we interpolated a spline through each group. In this way we obtained a denser subsampling, i.e., at 2° step.

The first third of the chart on Fig. 4 shows the average pose estimation errors during the learning. The results demonstrate the ability of the methods to *generalise* the knowledge: even though the training was performed at 20° step, 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.

The next issue was whether learning the same images multiple times 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

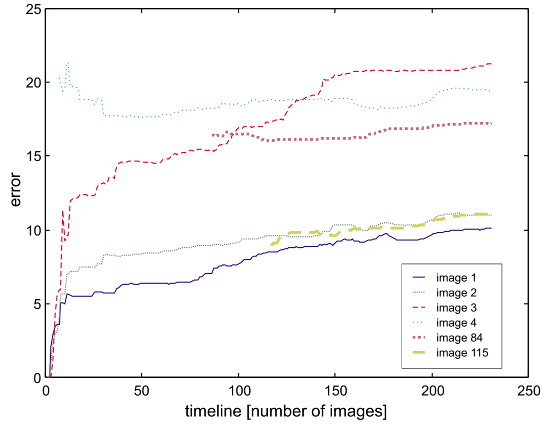

Figure 2. Reconstruction errors for selected images during the learning.

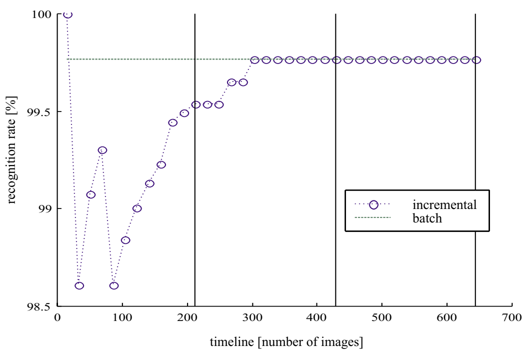

Figure 3. Recognition rate during the learning and two additional relearning runs

ones as they arrived. On Figs. 4 and 3 the vertical lines denote the ending of one learning iteration and the beginning of the next iteration. We can see that when we re-learnt the same series of input images for the second time, both the recognition rate and the pose estimation error converged to a level comparable to the one obtained with the batch method. This result suggests that the subspace can describe the observations better after each iteration of learning, so the loss of accuracy due to a partial representation is smaller on the subsequent iterations.

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

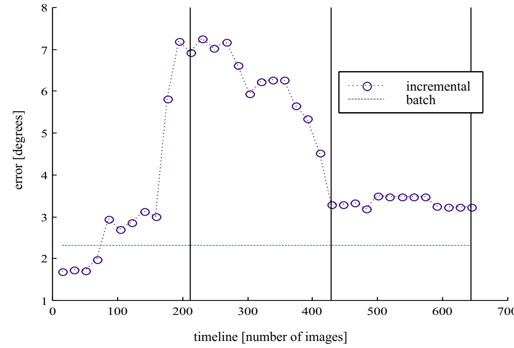

Figure 4. Pose estimation error during the learning and two additional relearning runs

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 our experiments show, it is feasible to keep only the subspace representations of the input images throughout the learning process. Each original image is therefore needed only at the point when we update the space of eigenvectors with the image.

Despite the fact that we are storing and updating only a fraction of full information, by constantly monitoring the error we preserve the significant information. The method manages to preserve the important features during the learning, which enables highly accurate recognition (in case of objects) and mobile robot localisation.

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 eigenalysis 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.