Published in proceedings of ECCV'02, part IV, pages 171-178, 2002.

A Robust PCA Algorithm for Building Representations from Panoramic Images*

Danijel Skočaj[1], Horst Bischof[2], and Aleš Leonardis[1]

[1] Faculty of Computer and Information Science
University of Ljubljana, Slovenia
{danijels,alesl}@fri.uni-lj.si
[2] Inst. for Computer Graphics and Vision
Graz University of Technology, Austria
bischof@icg.tu-graz.ac.at

Abstract. Appearance-based modeling of objects and scenes using PCA has been successfully applied in many recognition tasks. Robust methods which have made the recognition stage less susceptible to outliers, occlusions, and varying illumination have further enlarged the domain of applicability. However, much less research has been done in achieving robustness in the learning stage. In this paper, we propose a novel robust PCA method for obtaining a consistent subspace representation in the presence of outlying pixels in the training images. The method is based on the EM algorithm for estimation of principal subspaces in the presence of missing data. By treating the outlying points as missing pixels, we arrive at a robust PCA representation. We demonstrate experimentally that the proposed method is efficient. In addition, we apply the method to a set of panoramic images to build a representation that enables surveillance and view-based mobile robot localization.

1 Introduction

Appearance-based modeling of objects and scenes using subspace representations has become very popular in the vision community. Most of the approaches have used Principle Component Analysis (PCA) for building efficient representations and for subsequent recognition. The approach has led to a variety of successful applications, e.g., human face recognition [22, 2], visual inspection [25], visual positioning and tracking of robot manipulators [15], illumination planning [14], mobile robot localization [11], and background modeling [17]. However, the standard way to perform recognition, based on projections, is prone to errors in the case of non-Gaussian noise, e.g., occlusions, varying illumination conditions, and cluttered background in the input images. Therefore, different authors have proposed robust procedures [18, 3, 12] to obtain reliable recognition also in these cases.

* D. S. and A. L. acknowledge the support from the Ministry of Education, Science, and Sport of Republic of Slovenia (Research Program 506). H. B. was supported by the K plus Competence Center ADVANCED COMPUTER VISION.

However, if the training images are taken under non-ideal conditions, the obtained representations encompass various non-desirable effects, which cannot be overcome at the recognition stage. Let us consider, for example, a case of a view-based localization of a mobile platform using a set of panoramic images to represent the environment [8, 9, 1, 11]. Due to the wide field-of-view, it is almost impossible to obtain training images without outliers (e.g., people moving around). This problem gets even more pronounced in the cases when the training has to be performed "in-vivo", i.e., when the environment can not be specifically tailored for the training phase. This clearly indicates that we need a method to perform *robust training* in order to obtain parametric representations insensitive to these effects. More specifically, we need a procedure which is able to detect inconsistencies in the input data, eliminate them, and then calculate the representation from the consistent data only. In the case of representations based on the PCA, this requires a novel way of calculating the eigenimages from a subset of data points.

In this paper we propose a novel robust PCA method to obtain a consistent subspace representation in the presence of outlying pixels in the training images. The method is based on the EM algorithm [19, 21], which enables the calculation of the eigenspaces, i.e., maximum likelihood solution of PCA, in the case of missing data. The fact that we can calculate the PCA on a subset of pixels in the input images, makes it possible to remove the outliers and treat them as missing pixels, arriving at a robust PCA representation. The outliers are determined by a consistency measure over the set of training images. We demonstrate experimentally that the proposed method is computationally efficient and compare its results to a recently proposed method [5]. In addition, we apply the method to a set of panoramic images to build a representation that would enable surveillance and view-based mobile robot localization, where it is essential to have a robust training algorithm which is insensitive to people moving in front of the sensor.

The paper is organized as follows: In the next section we briefly review the related work. In section 3 we present our robust PCA method. Then we show the performance of the method on several experiments. Finally, we summarize the paper and outline some work in progress.

2 Related Work

A variety of methods have been proposed to perform PCA on a set of data. For an overview see, for example, [6, 16]. Recently, several new algorithms for PCA have been proposed. Roweis [19] proposed an Expectation Maximization (EM) algorithm for estimating principal axes. In [21], Tipping and Bishop proposed a Probabilistic PCA algorithm, based on an isotropic error model. They have also introduced an iterative EM algorithm for estimating the parameters. The algorithm has also been used in a mixture model of principal component analyzers. Of particular interest is that these algorithms can be used in the case of missing data (i.e., when we know which pixels are missing) to calculate a maximum likelihood solution for PCA. The major drawback of all these methods is that they are not robust, because they are basically based on the least squares minimization.

Several methods for estimation of principal axes in the presence of missing data have been proposed. Wiberg [23] has proposed a method based on the weighted least squares technique, which was later extended by Shum et al. [20]. Gabriel and Zamir [7] 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.

Several methods to robustly extract the principal exes in the presence of outliers have also been proposed in the statistical community; see [5] for a nice overview. The major drawback of these methods is that either they rely on the calculation of a "robust" covariance matrix, which is due to the high dimensionality of image data not feasible, or that they discard entire data vectors [24]. In the later case, a whole image would be eliminated just because of a single outlying pixel. For the tasks we envision to tackle, this would mean that no images would be usable since, in general, each of them might contain some outliers.

Only very recently De la Torre and Black [5] have presented a method which is robust and suitable for high dimensional image data. They have presented a robust PCA algorithm based on robust M-estimator. Their formulation yields a high dimensional non-linear optimization problem which has to be solved in an iterative manner. Therefore, the overall computational complexity of the algorithm is very high. They have demonstrated the performance of their algorithm on images contaminated with a few outliers and on temporal coherent image sequences.

The method for performing PCA in the presence of missing pixels, which is presented in this paper, is related to these methods [23, 20]. However, we have paid a special attention to the high-dimensional nature of image data and adapted the algorithms to avoid an ill-posedness of the problem. Together with a procedure for detection of outliers, the proposed method enables efficient and robust subspace learning.

3 Robust PCA Based on EM

Before we present the robust PCA algorithm, we briefly introduce the notation. Let $\mathbf{x}_i = [x_{1i}, \ldots, x_{mi}]^T \in \mathbb{R}^m$ be an individual image represented as a vector, and $X = [\mathbf{x}_1, \ldots \mathbf{x}_n] \in \mathbb{R}^{m \times n}$. To simplify the notation, we assume X to be normalized, having zero mean. The eigenvectors (principal axes) obtained from X are denoted by $\mathbf{e}_i = [e_{1i}, \ldots, e_{mi}]^T \in \mathbb{R}^m$; $E = [\mathbf{e}_1, \ldots \mathbf{e}_n] \in \mathbb{R}^{m \times n}$. The columns of E, i.e., eigenvectors, are arranged in decreasing order with respect to the corresponding eigenvalues. Usually, only k, $k < n$, eigenvectors (those with the largest eigenvalues) are needed to represent $\mathbf{x}_i$ to a sufficient degree of accuracy as a linear combination of eigenvectors $\mathbf{e}_i$:

$$\tilde{\mathbf{x}} = \sum_{i=1}^{k} a_i(\mathbf{x})\mathbf{e}_i = E\mathbf{a} \ , \tag{1}$$

where $\tilde{\mathbf{x}}$ denotes the approximation of $\mathbf{x}$. The entire set of images X can thus be represented as $\tilde{X} = EA$ where $A = [\mathbf{a}_1, \dots \mathbf{a}_n] \in \mathbb{R}^{k \times n}$ consists of coefficient vectors $\mathbf{a}_i = [a_{1i}, \dots, e_{ki}]^T \in \mathbb{R}^k$.

Having an eigenspace encompassing the training images and being given an input image $\mathbf{y}$, recognition occurs as an estimation of the parameters $a_i(\mathbf{y})$. These can be calculated by a standard projection

$$a_i(\mathbf{y}) = \mathbf{e}_i^T \mathbf{y} = \sum_{j=1}^{m} e_{ji} y_j \quad , \quad i = 1 \dots k \quad , \tag{2}$$

or, as a robust procedure [12], by solving a system of linear equations

$$y_{r_i} = \sum_{j=1}^{k} a_j(\mathbf{y}) e_{r_i,j} \quad , \quad i = 1 \dots q \quad , \tag{3}$$

evaluated at $q \geq k$ points $\mathbf{r} = (r_1, \dots r_q)$.

Now we can turn to the robust PCA algorithm. We will first present an algorithm based on EM and Probabilistic PCA (PPCA) which can cope with missing data. Based on that algorithm we will then present an efficient robust PCA algorithm.

3.1 EM Algorithm for PCA

Most of the algorithms for building principal subspaces are based on the eigendecomposition of the covariance matrix of the input data. However, there exist other approaches, for instance a probabilistic approach, where PCA can be considered as a limiting case of a linear Gaussian model, when the noise is infinitesimally small and equal in all directions [19]. From this observation one can derive an algorithm for calculating principal axes, which is based on the EM (expectation-maximisation) algorithm [4, 19, 21]. This algorithm consists of two steps, E and M, which are sequentially and iteratively executed:

- **E-step**: Estimate coefficients A (unknown states in the context of EM) using computed principal exes E.
- **M-step**: Compute new principal axes E which maximize expected joint likelihood of the estimated coefficients A and the observed images X.

The EM algorithm for PCA proposed by Roweis [19] looks as follows:

- **E-step**: $A = (E^T E)^{-1} E^T X$
- **M-step**: $E = X A^T (A A^T)^{-1}$.

At convergence, the columns of E span the space of the first k principal exes. These vectors are, in general, not orthogonal, but, when desired, they can be orthogonalized later. The PPCA algorithm in [21] is very similar except that an isotropic noise model is used.

The convergence of the algorithm can be checked by looking at a difference in the successive estimates of the average lost variance σ^2 per discarded dimension.

As shown in [21], the maximum likelihood solution for σ^2 is $\sigma^2 = \frac{1}{n-k}\sum_{j=k+1}^{n}\lambda_j$, which corresponds to the sum of the discarded eigenvalues. It can be conveniently calculated by

$$\sigma^2 = \frac{1}{n-k}\left(\mathrm{VAR}(X) - \sum_{j=1}^{k}\lambda_j\right) \quad , \tag{4}$$

where $\mathrm{VAR}(X)$ is the variance of X which is defined as the sum of variances in rows of X. Since in the EM algorithm we do not explicitly calculate eigenvalues at each iteration, we can estimate σ^2 by

$$\sigma^2 = \frac{1}{n-k}\left(\mathrm{VAR}(X) - \mathrm{VAR}(\tilde{X})\right) \quad . \tag{5}$$

Roweis [19] has proposed also a generalized E-step for handling missing data. He has proposed to treat missing data as additional hidden states and to estimate them in the E-step simultaneously with estimating the coefficients by solving a least squares problem. However, in the case of images with considerable amount of missing pixels, this results in an ill-posed problem, which does not produce satisfactory results. In the next subsection we propose a different algorithm for handling missing data in the EM approach, which produces better results.

3.2 PCA in the Presence of Missing Data

It is important to note that in the case when we use all data points, the E-step is equivalent to calculating the coefficients by the solution of the linear system of equations (3). This can be easily seen by noting that $\mathbf{a}_i = (E^TE)^{-1}E^T\mathbf{x}_i = E^\dagger\mathbf{x}_i$. The least squares solution of (3) is equivalent to the pseudo-inverse. A very similar observation holds also for the M-step. Therefore, we can perform the EM-algorithm by iteratively solving the following systems of linear equations:

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

$$x_{ij} = \sum_{p=1}^{k} e_{ip}a_{pj} \quad , \quad i = 1 \ldots m \quad . \tag{6}$$

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

$$x_{ij} = \sum_{p=1}^{k} e_{ip}a_{pj} \quad , \quad j = 1 \ldots n \quad . \tag{7}$$

From these considerations we can also see how to compute the coefficients and principal axes in the case of missing data. We only need to set up Eqs. (6) and (7) with the known data points and compute the coefficients $\mathbf{a}_i$ and the principal axes $\mathbf{e}_i$. Similar solutions, but with different derivations and formulations, are proposed in [20, 23].

Notice that in the case of missing pixels, we can no longer assume that the data is mean normalized, however this can easily be achieved by estimating the mean image over the known pixels.

When dealing with images containing a considerable amount of missing pixels, such a formulation results in an ill-posed problem. To alleviate this problem we impose additional application dependent constraints to the minimization proces. When the images are ordered as it is the case of image sequences, we can augment the algorithm to include also a smoothness prior to enforce that the missing pixels are changing smoothly over time. Thus, in the M-step we minimize the second derivative of the reconstructed missing pixels. If $\mathcal{M}$ is a set containing all missing pixels, the algorithm looks as follows:

- **E-step**: Estimate coefficients in A in the following way: For each image j, $j = 1 \dots n$, solve the following system of linear equations in the least squares sense:
$$x_{ij} = \sum_{p=1}^{k} e_{ip} a_{pj} \ , \ \ i = 1 \dots m \ \mid \ x_{ij} \notin \mathcal{M} \ . \tag{8}$$
- **M-step**: Estimate principal axes in E in the following way: For each pixel i, $i = 1 \dots m$, solve the following system of linear equations in the least squares sense:
$$\begin{aligned} x_{ij} &= \sum_{p=1}^{k} e_{ip} a_{pj} \ , \ \ j = 1 \dots n \ \mid \ x_{ij} \notin \mathcal{M} \\ 0 &= \alpha \sum_{p=1}^{k} e_{ip}(a_{p,j-1} - 2a_{pj} + a_{p,j+1}) \ , \ \ j = 1 \dots n \ \mid \ x_{ij} \in \mathcal{M} \ , \end{aligned} \tag{9}$$
where α is the parameter which weights the influence of the smoothness constraint.

The overall algorithm minimizes the following error function:

$$\mathcal{E} = \sum_{j=1}^{n} \sum_{i \in \mathcal{G}_j} \left(x_{ij} - \sum_{p=1}^{k} e_{ip} a_{pj} \right)^2 + \alpha \sum_{j=1}^{n} \sum_{i \in \mathcal{B}_j} \left(\sum_{p=1}^{k} e_{ip} a''_{pj} \right)^2 \ , \tag{10}$$

where $\mathcal{G}_j$ denotes a set of indices of non-missing pixels in j-th image, while $\mathcal{B}_j$ denotes a set of indices of the corresponding missing pixels.

Alternatively, the principal axes can in the M-step also be obtained by applying the standard PCA using all pixels provided that missing pixels are filled-in. The question is how to optimally fill-in the values of the missing pixels. 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 by the missing pixels. Given the principal subspace E, which models the input data, the optimal location is a point in the missing pixels subspace which is closest to the principal subspace. This point is obtained by replacing missing pixels with the reconstructed values, which are calculated by (1) using the coefficients $\mathbf{a}_i$ estimated in E-step of the current iteration and the principal axes $\mathbf{e}_i$ obtained in the previous iteration. Therefore, the new M-step looks as follows:

- **M-step**: Estimate principal axes in E by applying the standard PCA on X with the reconstructed missing pixels:

$$x_{ij} = \tilde{x}_{ij}\ ,\ i = 1 \dots m\ ,\ j = 1 \dots n \mid x_{ij} \in \mathcal{M} \ \text{ where } \ \tilde{X} = EA\ . \qquad (11)$$

What still remains to be determined is how to calculate an initial solution for E. In fact, we could start with random values as the initial principal axes. A more efficient approach is to calculate the principal axes E from an estimate of A obtained by performing SVD on the inner product matrix S of the input set X estimated from the non-missing pixels:

$$s_{ij} = \frac{1}{|P|} \sum_{p \in P} x_{pi} x_{pj}\ ;\ P = \{p \mid x_{pi} \notin \mathcal{M},\ x_{pj} \notin \mathcal{M}\}\ . \qquad (12)$$

To summarize, the algorithm *missPixPCA* looks as follows:

```
Estimate inner product matrix $S$ from the known data, using (12).
Perform SVD on $S$ yielding eigenvectors as an estimate for $A$.
Perform M-step according to (9) obtaining an estimate for $E$.
**repeat**
   Perform E-step according to (8).
   Perform M-step according to (9) or (11).
   Replace missing pixels in $X$ by reconstruction using (1).
   Calculate $\sigma^2$ from (4) or (5).
**until** change in $\sigma^2$ is small.
```

3.3 Robust PCA

The basic idea of our robust PCA algorithm is to determine the outliers and treat them as missing pixels, and then use the algorithm described above to calculate PCA without these outliers. Based on the estimated principal axes and coefficients we can again determine the outliers and repeat the process. The crucial question is how to determine the outliers. We have investigated two approaches, one giving a global threshold and another giving a local threshold:

1. Global threshold: Depending on the number of principal components we choose to represent the images with, we expect that an average error per image is equal to the discarded variance, which can be computed as $\sigma^2_{img} = \sum_{i=k+1}^{n} \lambda_i$. When we divide this number by the number of pixels, we get an average expected error per pixel $\sigma^2_{pix} = \frac{1}{m} \sum_{i=k+1}^{n} \lambda_i$. We treat all those pixels as outliers whose reconstruction error is larger than σ^2_{pix}, multiplied by a factor, which depends on the expected quantity of outliers in the images. The problem with this method is that it assumes the same variability across the whole image, which is in general not true.

2. Local threshold: De la Torre and Black [5] have proposed a method to compute a local threshold for each pixel based on the Median Absolute Deviation (MAD) of a pixel. They compute for each pixel p: $\sigma_p = \beta(1.4826\mathrm{med}_R(|\mathbf{r}^p - \mathrm{med}_R(|\mathbf{r}^p|)|), \sigma_{\min})$ where med_R is median taken over a region, R, around pixel p, $\sigma_{\min}$ is the MAD over the whole image, and β is a constant factor. The initial error $\mathbf{r}^p$ is calculated using standard PCA on the whole image set. This method has the advantage that it takes into account the variability within an image.

Based on this, we can outline our robust PCA algorithm *robPCA* as follows:

```
repeat
    Compute principal axes and coefficients using standard PCA on $X$.
    Detect outliers using either global or local threshold.
    Treat outliers as missing pixels and perform PCA by using missPixPCA algorithm from subsection 3.2.
    Replace missing pixels in $X$ by reconstruction using estimated $E$ and $A$.
until the outlier set is empty.
```

Usually, only a few iterations (even only a single one) of this algorithm are sufficient for convergence.

4 Experimental Results

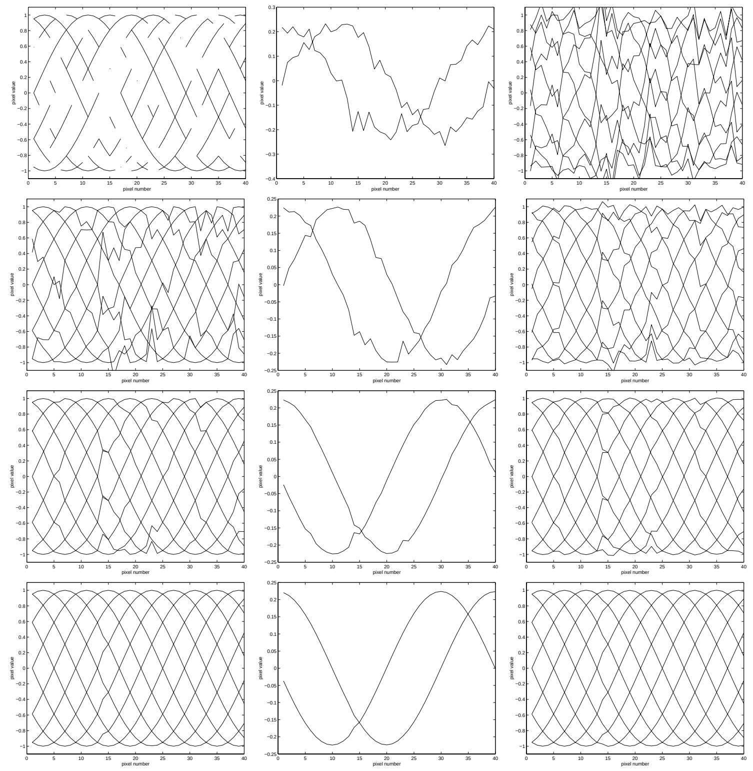
In this section we demonstrate the performance of our proposed algorithm and evaluate it on several examples. First, we illustrate the behavior of the algorithm on a set of 1-D signals. Then we evaluate the results of applying our algorithm to a set of panoramic images. Finally, we compare the results of our approach to the ones obtained by a recently proposed method [5] on their set of input images.

4.1 A 1-D Example

Let us consider a set of 1-D vectors which is formed of 10 shifted harmonic (sinus) functions. Next, we randomly remove 20% of the elements, as depicted in the top row of Fig. 1(a). Now, the goal is to find the optimal principal axes representing these vectors which contain missing data by applying *missPixPCA* algorithm from subsection 3.2. In this experiment, the estimation of the principal axes in the M-step of the EM algorithm was performed by applying the standard PCA on the input data with the reconstructed missing pixels.

Fig. 1 illustrates the performance of the proposed method. First, the initial values of the principal exes are obtained from the covariance matrix estimated from the non-missing values of the input vectors. Then, based on the E-step, we calculate the coefficients and the reconstructed signals. The missing pixels are then replaced with their reconstructed values from the previous iteration.

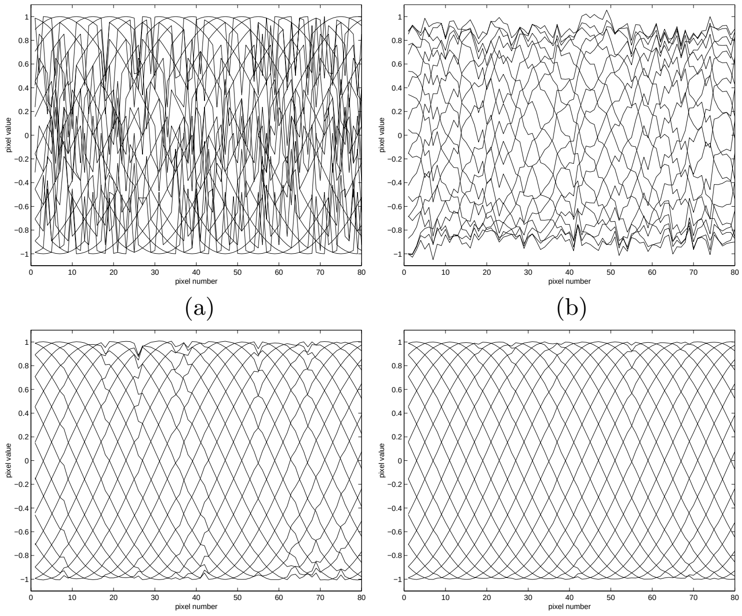
The initialization, the first, the third, and the tenth iteration of the algorithm are shown in subsequent rows of Fig. 1. Note how the functions representing estimated principal axes are getting smoother after every iteration. Finally, we obtain a perfect reconstruction of the input vectors.



(a) (b) (c)

Fig. 1. A 1-D example with missing pixels: (a) input data at each iteration, (b) estimated principal axes, (c) reconstructed vectors.

Fig. 2 illustrates the performance of the algorithm *robPCA* proposed in subsection 3.3. Now, a set of 1-D vectors is formed from 40 shifted sinus functions (only half of them are shown). 20% of the elements are contaminated with random noise (Fig. 2(a)). Figs. 2(b-d) depict the reconstruction after the second step in the first iteration of the proposed algorithm, and the reconstruction at the end of the first, and at the end of the second iteration. One can observe, how initially very noisy signals become more and more regular, converging to the ground truth.



(a) (b) (c) (d)

Fig. 2. A 1-D example with noisy signals: (a) input data, (b) reconstruction after using the standard PCA, (c) reconstruction after the first iteration, (d) reconstruction after the second iteration of *robPCA* algorithm.

4.2 Robust PCA on Panoramic Images

First, we tested the performance of our method on the images with known ground truth. We captured 30 panoramic images of the size 100 × 100 in the faculty hall. Then, we synthetically applied gradual illumination changes and nonlinear illumination changes (a shadow—the vertical “cloud”) to this set of images. In addition, we added, as an outlier area, a square on a randomly chosen position in every image. The goal is to learn the panoramic representation capturing the illumination variations (linear and nonlinear) but discarding the outliers. Since these images were captured in a sequence, they are temporally well correlated. Therefore, we include the smoothness prior in the calculation of the principal axes in the M-step of the EM algorithm.

The results are depicted in Fig. 3. In the images reconstructed from the first principal component obtained with the standard PCA, one can clearly observe that the linear illumination changes are modelled, but not the nonlinear. If the model consists of the first 8 principal axes produced by the standard PCA, then all illumination changes are captured in the reconstructions, however, the model also contains the outliers (squares). On the other hand, using our robust algorithm, one can observe from the reconstructions based on the first 8 principal axes, that all illumination changes are captured in the model, while the outliers are not, which is exactly what we want to achieve.

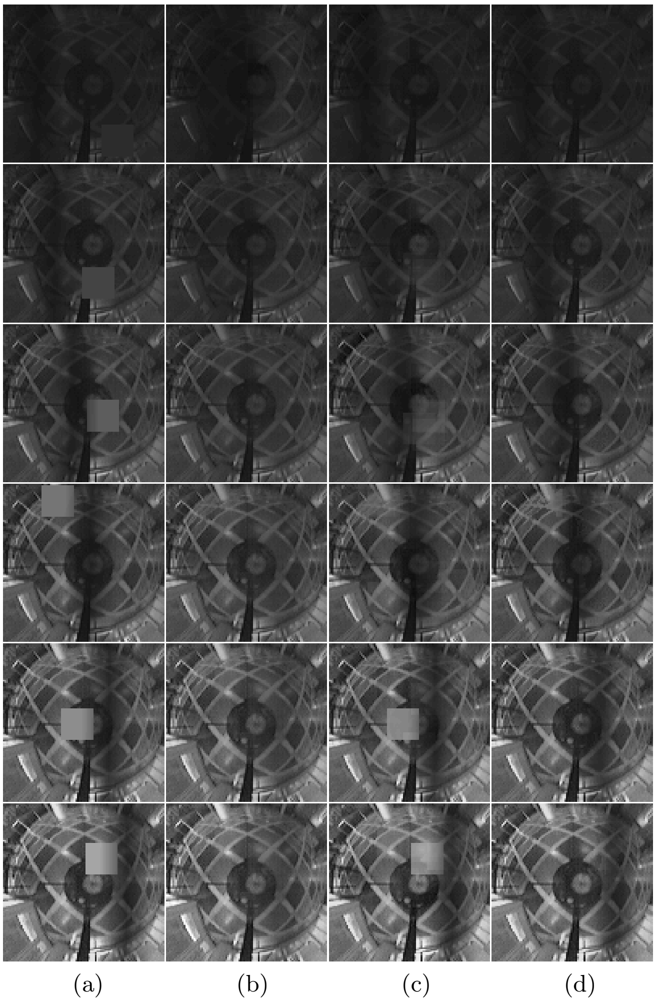


Fig. 3. A comparison of our method with the standard PCA. (a) Input images (every fifth image from the training set). (b) Reconstruction based on the first principle axis (PA) using standard PCA. (c) Reconstruction based on the first 8 PA using standard PCA. (d) Reconstruction based on the first 8 PA using *robust PCA*.

Since the positions of the outliers are known, we can calculate the mean absolute reconstruction error in the outliers (squares) and in the inliers (background). The reconstruction error in the inliers should be as small as possible, while the reconstruction error in the outliers should be large, enabling an efficient detection of outliers. Table 1 compares the reconstruction errors with the errors obtained using optimal principal axes, which were estimated from the data without outliers (ground truth). It is evident that the robust PCA outperforms the standard one since the errors obtained with the proposed algorithm are much closer to the optimal ones.

Table 1. Comparison of the reconstruction errors obtained using the standard and the robust PCA.

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

We also applied the proposed method to a sequence of panoramic images, which were captured at several locations in the laboratory. At each location a sequence of 180 images of the size 100 × 100 was captured, of which 60 were used in the training stage. During the acquisition, we varied the illumination conditions and people were free to walk around in the laboratory. To enable an efficient and robust appearance based localization of a mobile platform, a representation has to be built which parametrically models the laboratory under different illumination conditions but at the same time excludes randomly moving subjects (objects) in the training images.

The results for one sequence are shown in Fig. 4. Since people appear in most of the training images, the standard PCA incorporates them in the representation. Consequently, they appear in the reconstructed images as undesirable "ghost people". In contrast, the images, which were reconstructed based on the output of the robust PCA do not contain these effects since the outliers (people) are eliminated from the representation during the process. Therefore, these images represent solely the appearance of the location under different illuminations. Robust representations obtained at different locations can be combined in an overall appearance-based representation of the laboratory, suitable for mobile robot localization and navigation, or for performing surveillance tasks.

4.3 Comparison with the Previous Work

We performed an experiment where we applied our method on the same image sequence[3] of 256 images of size 120×160 pixels as De la Torre and Black [5], who recently proposed a method for robust learning of appearances based on PCA. Some of the results obtained by our algorithm are presented in Fig. 5. By visually

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

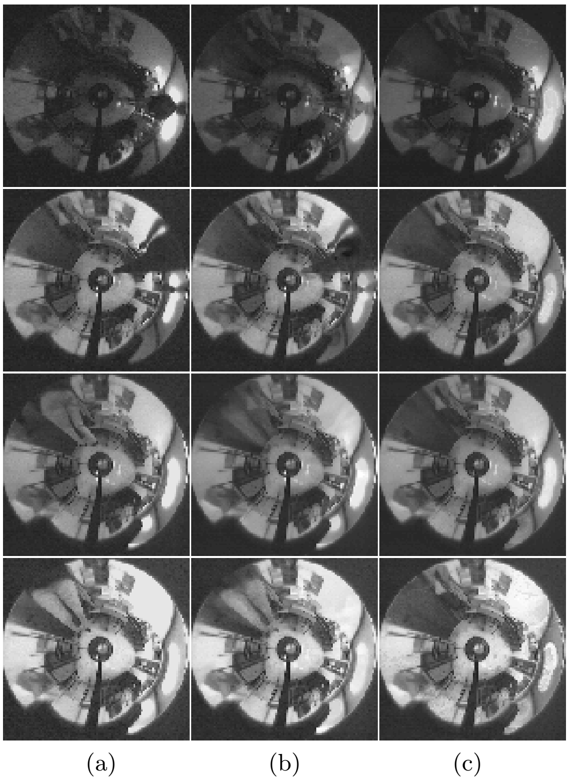


Fig. 4. (a) Four panoramic images from the sequence, (b) their reconstructions based on the standard PCA, and (c) their reconstructions based on the robust PCA.

comparing Fig. 5 with Fig. 8 in [5], we can conclude that both algorithms produce very similar results. However, as reported in [5], their algorithm takes three hours on a 900 MHz Pentium III to produce these results, while our algorithm finishes the task in 19 minutes on a 550 MHz Pentium III.

5 Conclusion

In this paper we presented a novel method which builds robust PCA representations in the presence of outlying pixels in the training images. The method is based on the EM algorithm and its computational efficiency has been experimentally demonstrated. We showed the utility of the method on a set of panoramic images with immediate applications for view-based mobile robot localization and surveillance.

In fact, robust building of representations is an unavoidable step in all realistic learning scenarios when the environment can not be specifically tailored for the training phase. Since the method primarily exploits the temporal coherence

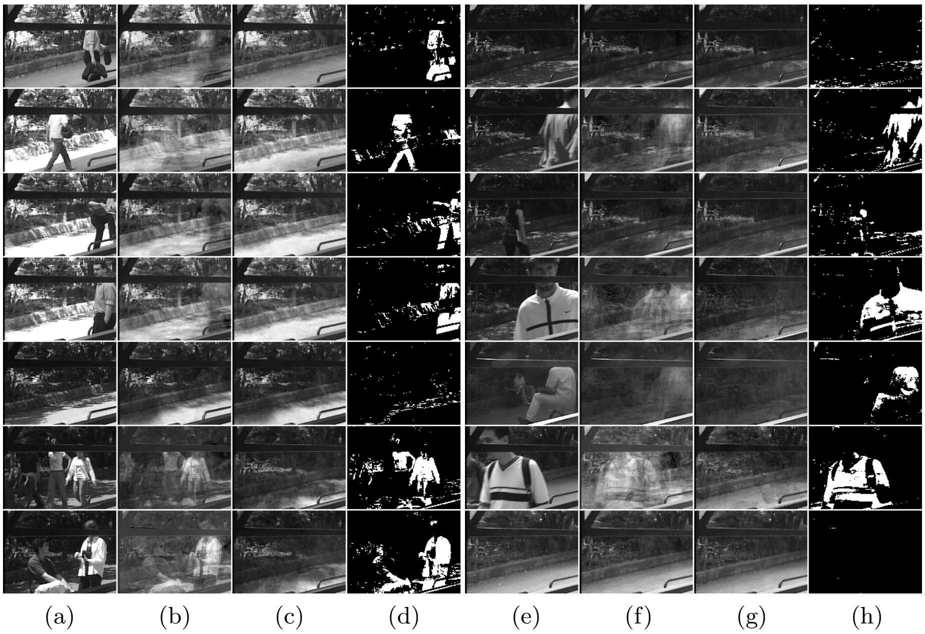


Fig. 5. (a),(e) Original data from [5], (b),(f) standard PCA reconstruction, (c),(g) our robust PCA reconstruction, (d),(h) outliers obtained by our method.

of the input image set to build a consistent subspace representation, it is best suited for temporally well correlated inputs. While in these cases the non-robust initialization, which has been used in the proposed method, does not pose a problem, we are well aware that one issue that needs to be further explored is robust initialization [10].

Future work will also focus on image sets containing several classes of different views of objects and scenes. In such cases, the presented method has to be combined with an unsupervised clustering method of input images, such as for example recently proposed [13], which would result in multiple robust eigenspaces.

References

1. H. Aihara, N. Iwasa, N. Yokoya, and H. Takemura. Memory-based self-localisation using omnidirectional images. In *14th ICPR*, pages 297–299, 1998.
2. D. Beymer and T. Poggio. Face recognition from one example view. In *Proceedings of 5th ICCV'95*, pages 500–507. IEEE Computer Society Press, 1995.
3. M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV96*, pages 329–342. Springer, 1996.

4. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.
5. F. De la Torre and M. Black. Robust principal component analysis for computer vision. In *ICCV01*, pages 362–369. IEEE Computer Society Press, 2001.
6. K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks.* John Wiley and Sons, INC, 1996.
7. K. R. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4):489–498, 1979.
8. J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, 2000.
9. H. Ishiguro and S. Tsuji. Image-based memory of environment. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 634–639, 1996.
10. D. W. Jacobs. Linear Fitting with Missing Data for Structure-from-Motion. *CVIU*, 82(1):57–81, 2001.
11. M. Jogan and A. Leonardis. Robust localization using panoramic view-based recognition. In *15th ICPR*, volume 4, pages 136–139, 2000.
12. A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000.
13. A. Leonardis, H. Bischof, and J. Maver. Multiple eigenspaces. *Pattern Recognition*, in press. 2002.
14. H. Murase and S. K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, 1994.
15. S. K. Nayar, H. Murase, and S. A. Nene. Learning, positioning, and tracking visual appearance. In *IEEE International Conference on Robotics and Automation*, San Diego, May 1994.
16. E. Oja. *Subspace Methods of Pattern Recognition.* Research Studies Press, 1983.
17. N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.
18. R. Rao. Dynamic appearance-based recognition. In *CVPR'97*, pages 540–546. IEEE Computer Society, 1997.
19. S. Roweis. EM algorithms for PCA and SPCA. In *NIPS*, pages 626–632, 1997.
20. H. Shum, K. Ikeuchi, and 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.
21. M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. Technical report, Microsoft Research, 1999.
22. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
23. T. Wiberg. Computation of Principal Components when Data are Missing. *Proc. Second Symp. Computational Statistics*, 229–236, 1976.
24. L. Xu and A. L. Yuille. Robust principal component analysis by self-organising rules based on statistical physics approach. *IEEE Trans. Neural Networks*, 6(1):131–143, 1995.
25. S. Yoshimura and T. Kanade. Fast template matching based on the normalized correlation by using multiresolution eigenimages. In *Proceedings of IROS'94*, pages 2086–2093, 1994.