Robust Principal Component Analysis for Computer Vision

Fernando De la Torre[*] **Michael J. Black**[†]

[*]Departament de Comunicacions i Teoria del Senyal, Escola d'Enginyeria la Salle,
Universitat Ramon LLull, Barcelona 08022, Spain. ftorre@salleURL.edu

[†]Department of Computer Science, Brown University, Box 1910,
Providence, RI 02912, USA. black@cs.brown.edu

Abstract

Principal Component Analysis (PCA) has been widely used for the representation of shape, appearance, and motion. One drawback of typical PCA methods is that they are least squares estimation techniques and hence fail to account for "outliers" which are common in realistic training sets. In computer vision applications, outliers typically occur within a sample (image) due to pixels that are corrupted by noise, alignment errors, or occlusion. We review previous approaches for making PCA robust to outliers and present a new method that uses an intra-sample *outlier process to account for pixel outliers. We develop the theory of Robust Principal Component Analysis (RPCA) and describe a robust M-estimation algorithm for learning linear multivariate representations of high dimensional data such as images. Quantitative comparisons with traditional PCA and previous robust algorithms illustrate the benefits of RPCA when outliers are present. Details of the algorithm are described and a software implementation is being made publically available.*

1 Introduction

Automated learning of low-dimensional linear models from training data has become a standard paradigm in computer vision. Principal Component Analysis (PCA) in particular is a popular technique for parameterizing shape, appearance, and motion [8, 4, 18, 19, 29]. These learned PCA representations have proven useful for solving problems such as face and object recognition, tracking, detection, and background modeling [2, 8, 18, 19, 20].

Typically, the training data for PCA is pre-processed in some way (e.g. faces are aligned [18]) or is generated by some other vision algorithm (e.g. optical flow is computed from training data [4]). As automated learning methods are applied to more realistic problems, and the amount of training data increases, it becomes impractical to manually verify that all the data is "good". In general, training data may contain undesirable artifacts due to occlusion (e.g. a hand in front of a face), illumination (e.g. specular reflections), image noise (e.g. from scanning archival data), or errors from the underlying data generation method (e.g. incorrect optical flow vectors). We view these artifacts as statistical "outliers" [23] and develop a theory of Robust PCA (RPCA) that can be used to construct low-dimensional linear-subspace representations from this noisy data.

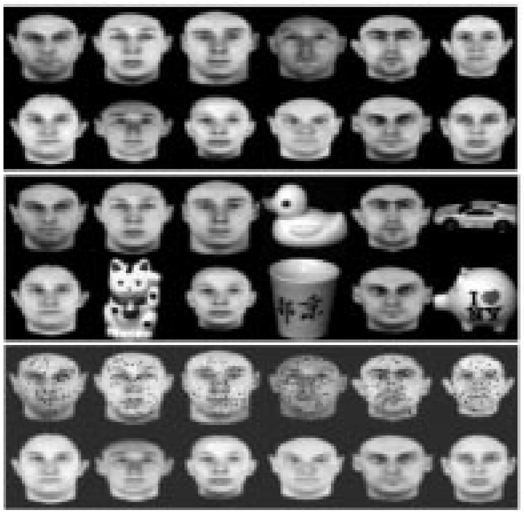

Figure 1: *Top:* A few images from an illustrative training set of 100 images. *Middle:* Training set with *sample outliers*. *Bottom:* Training set with *intra-sample outliers*.

It is commonly known that traditional PCA constructs the rank k subspace approximation to training data that is optimal in a least-squares sense [16]. It is also commonly known that least-squares techniques are not robust in the sense that outlying measurements can arbitrarily skew the solution from the desired solution [14]. In the vision community, previous attempts to make PCA robust [30] have treated entire data samples (i.e. images) as outliers. This approach is appropriate when entire data samples are contaminated as illustrated in Figure 1 (*middle*). As argued above, the more common case in computer vision applica-

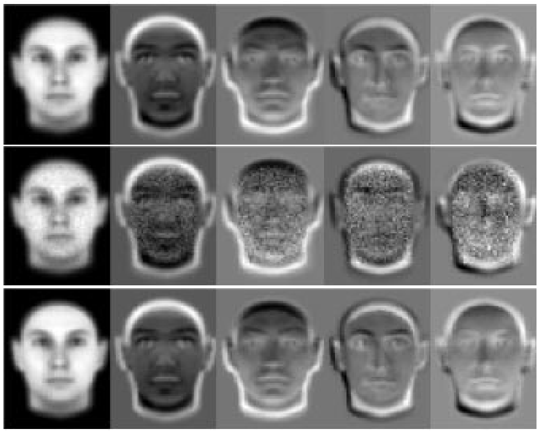

Figure 2: Effect of intra-sample outliers on learned basis images. *Top:* Standard PCA applied to noise-free data. *Middle:* Standard PCA applied to the training set corrupted with intra-sample outliers. *Bottom:* Robust PCA applied to corrupted training data.

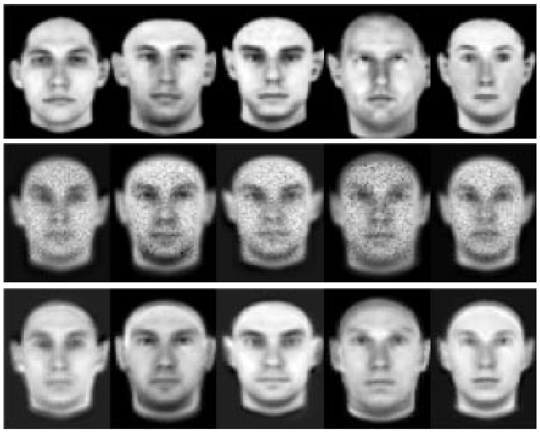

Figure 3: Reconstruction results using subspaces constructed from noisy training data. *Top:* Original, noiseless, test images. *Middle:* Least-squares reconstruction of images with standard PCA basis (MSRE 19.35) . *Bottom:* Reconstructed images using RPCA basis (MSRE 16.54) .

tions involves *intra-sample* outliers which effect some, but not all, of the pixels in a data sample (Figure 1 (*bottom*)).

Figure 2 presents a simple example to illustrate the effect of intra-sample outliers. By accounting for intra-sample outliers, the RPCA method constructs the linear basis shown in Figure 2 (*bottom*) in which the influence of outliers is reduced and the recovered bases are visually similar to those produced with traditional PCA on data without outliers. Figure 3 shows the effect of outliers on the reconstruction of images using the linear subspace. Note how the traditional least-squares method is influenced by the outlying data in the training set. The "mottled" appearance of the least squares method is not present when using the robust technique and the Mean Squared Reconstruction Error (MSRE, defined below) is reduced.

In the following section we review previous work in the statistics, neural-networks, and vision communities that has addressed the robustness of PCA. In particular, we describe the method of Xu and Yuille [30] in detail and quantitatively compare it with our method. We show how PCA can be modified by the introduction of an outlier process [1, 13] that can account for outliers at the pixel level. A robust M-estimation method is derived and details of the algorithm, its complexity, and its convergence properties are described. Like all M-estimation methods, the RPCA formulation has an inherent scale parameter that determines what is considered an outlier. We present a method for estimating this parameter from the data resulting in a fully automatic learning method. Synthetic experiments are used to illustrate how different robust approaches treat outliers. Experiments on natural data show how the RPCA approach can be used to robustly learn a background model in an unsupervised fashion.

2 Previous Work

A full review of PCA applications in computer vision is beyond the scope of this paper. We focus here on the robustness of previous PCA methods. Note that there are two issues of robustness that must be addressed. First, given a learned basis set, Black and Jepson [2] addressed the issue of robustly recovering the coefficients of a linear combination that reconstructs an input image. They did not address the general problem of robustly learning the basis images in the first place. Here we address this more general problem.

2.1 Energy Functions and PCA

PCA is a statistical technique that is useful for dimensionality reduction. Let $\mathbf{D} = [\mathbf{d}_1\ \mathbf{d}_2\ ...\ \mathbf{d}_n] = [\mathbf{d}^1\ \mathbf{d}^2\ ...\ \mathbf{d}^d]^T$ be a matrix $\mathbf{D} \in \Re^{d\times n}$ [1], where each column $\mathbf{d}_i$ is a data sample (or image), n is the number of training images, and d is the number of pixels in each image. We assume that training data is zero mean, otherwise the mean of the entire data set is subtracted from each column $\mathbf{d}_i$. Previous formulations assume the data is zero mean. In the least-squares case, this can be achieved by subtracting the mean from the training data. For robust formulations, the "robust mean" must be explicitly estimated along with the bases.

[1] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{I}$ represents the identity matrix and $\mathbf{1}_m = [1, \cdots, 1]^T$ is a m-tuple of ones. $\mathbf{d}_j$ represents the j-th column of the matrix $\mathbf{D}$ and $\mathbf{d}^j$ is a column vector representing the j-th *row* of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in row i and column j of the matrix $\mathbf{D}$ and the scalar i-th element of a column vector $\mathbf{d}_j$. d_{ji} is the i-th scalar element of the vector $\mathbf{d}^j$. All non-bold letters represent scalar variables. $diag$ is an operator that transforms a vector to a diagonal matrix, or a matrix into a column vector by taking each of its diagonal components. $[\mathbf{D}]^{.-1}$ is an operator that calculates the inverse of each element of a matrix $\mathbf{D}$. $\mathbf{D}_1 \circ \mathbf{D}_2$ denotes the Hadamard (point wise) product between two matrices of equal dimension.

Let the first k principal components of $\mathbf{D}$ be $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_k] \in \Re^{d \times k}$. The columns of $\mathbf{B}$ are the directions of maximum variation within the data. The principal components maximize $\max_{\mathbf{B}} \sum_{i=1}^{n} ||\mathbf{B}^T \mathbf{d}_i||_2^2 = \mathbf{B}^T \mathbf{\Gamma} \mathbf{B}$, with the constraint $\mathbf{B}^T \mathbf{B} = \mathbf{I}$, where $\mathbf{\Gamma} = \mathbf{D}\mathbf{D}^T = \sum_i \mathbf{d}_i \mathbf{d}_i^T$ is the covariance matrix. The columns of $\mathbf{B}$ form an orthonormal basis that spans the principal subspace. If the effective rank of $\mathbf{D}$ is much less than d and we can approximate the column space of $\mathbf{D}$ with $k << d$ principal components. The data $\mathbf{d}_i$ can be approximated by linear combination of the principal components as $\mathbf{d}_i^{rec} = \mathbf{B}\mathbf{B}^T \mathbf{d}_i$ where $\mathbf{B}^T \mathbf{d}_i = \mathbf{c}_i$ are the linear coefficients obtained by projecting the training data onto the principal subspace; that is, $\mathbf{C} = [\mathbf{c}_1 \; \mathbf{c}_2 \ldots \; \mathbf{c}_n] = \mathbf{B}^T \mathbf{D}$.

A method for calculating the principal components that is widely used in the statistics and neural network community [7, 9, 21, 26] formulates PCA as the least-squares estimation of the basis images $\mathbf{B}$ that minimize:

$$\begin{aligned} E_{pca}(\mathbf{B}) &= \sum_{i=1}^{n} e_{pca}(\mathbf{e}_i) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{B}^{\mathbf{T}}\mathbf{d}_i||_2^2 \\ &= \sum_{i=1}^{n} \sum_{p=1}^{d} (d_{pi} - \sum_{j=1}^{k} b_{pj} c_{ji})^2 \qquad (1) \end{aligned}$$

where $c_{ji} = \sum_{t=1}^{d} b_{tj} d_{ti}$, $\mathbf{B}^T \mathbf{B} = \mathbf{I}$, $||.||_2$ denotes the L_2 norm, $\mathbf{e}_i = \mathbf{d}_i - \mathbf{B}\mathbf{B}^T \mathbf{d}_i$ is the reconstruction error vector, and $e_{pca}(\mathbf{e}_i) = \mathbf{e}_i^T \mathbf{e}_i$ is the reconstruction error of $\mathbf{d}_i$.

Alternatively, we can make the linear coefficients an explicit variable and minimize

$$E_{pca_2}(\mathbf{B}, \mathbf{C}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{c}_i||_2^2. \qquad (2)$$

One approach for estimating both the bases, $\mathbf{B}$, and coefficients, $\mathbf{C}$, uses the Expectation Maximization (EM) algorithm [24, 28]. The approach assumes that the data is generated by a random process and computes the subspace spanned by the principal components when the noise becomes infinitesimal and equal in all the directions. In that case, the EM algorithm can be reduced to the following coupled equations:

$$\begin{aligned} \mathbf{B}^T \mathbf{B} \mathbf{C} &= \mathbf{B}^T \mathbf{D} \quad \text{(E-step)}, \qquad (3) \\ \mathbf{B} \mathbf{C} \mathbf{C}^T &= \mathbf{D} \mathbf{C}^T \quad \text{(M-step)}. \qquad (4) \end{aligned}$$

EM alternates between solving for the linear coefficients $\mathbf{C}$ (Expectation step) and solving for the basis $\mathbf{B}$ (Maximization step).

In the context of computer vision, Shum et al. [27] solve the PCA problem with known missing data by minimizing an energy function similar to (2) using a weighted least squares technique that ignores the missing data. The method is used to model a sequence of range images with occlusion and noise and is similar to the method of Gabriel and Zamir [11] described below. Rao [22] has recently proposed a Kalman filter approach for learning the bases $\mathbf{B}$ and the coefficients $\mathbf{C}$ in an incremental fashion. The observation process assumes Gaussian noise and corresponds the error E_{pca_2} above. While the Rao does not use a robust learning method for estimating the $\mathbf{B}$ and $\mathbf{C}$ that minimize E_{pca_2}, like Black and Jepson [2] he does suggest a robust rule for estimating the coefficients $\mathbf{C}$ once the bases $\mathbf{B}$ have been learned.

2.2 Robustifying Principal Component Analysis

The above methods for estimating the principal components are not robust to outliers that are common in training data and that can arbitrarily bias the solution (e.g. Figure 1). This happens because all the energy functions and the covariance matrix are derived from a least-squares (L_2 norm) framework. While the robustness of PCA methods in computer vision has received little attention, the problem has been studied in the statistics [5, 15, 16, 25] and neural networks [17, 30] literature, and several algorithms have been proposed.

One approach replaces the standard estimation of the covariance matrix, $\mathbf{\Gamma}$, with a robust estimator of the covariance matrix [5, 25]. This approach is computationally impractical for high dimensional data such as images. Alternatively, Xu and Yuille [30] have proposed an algorithm that generalizes the energy function (1), by introducing additional binary variables that are zero when a data sample (image) is considered an outlier. They minimize

$$\begin{aligned} E_{xu}(\mathbf{B}, \mathbf{V}) &= \sum_{i=1}^{n} \left[V_i \, ||\mathbf{d}_i - \mathbf{B}\mathbf{B}^{\mathbf{T}}\mathbf{d}_i||_2^2 + \eta(1 - V_i) \right] \\ &= \sum_{i=1}^{n} \left[V_i \big(\sum_{p=1}^{d} (d_{pi} - \sum_{j=1}^{k} b_{pj} c_{ij})^2 \big) + \eta(1 - V_i) \right] \qquad (5) \end{aligned}$$

where $c_{ij} = \sum_{t=1}^{d} b_{tj} d_{ti}$. Each V_i in $\mathbf{V} = [V_1, V_2, ..., V_n]$ is a binary random variable. If $V_i = 1$ the sample $\mathbf{d}_i$ is taken into consideration, otherwise it is equivalent to discarding $\mathbf{d}_i$ as an outlier. The second term in (5) is a penalty term, or prior, which discourages the trivial solution where all V_i are zero. Given $\mathbf{B}$, if the energy, $e_{pca}(\mathbf{e}_i) = ||\mathbf{d}_i - \mathbf{B}\mathbf{B}^{\mathbf{T}}\mathbf{d}_i||_2^2$ is smaller than a threshold η, then the algorithm prefers to set $V_i = 1$ considering the sample $\mathbf{d}_i$ as an inlier and 0 if it is greater than or equal to η.

Minimization of (5) involves a combination of discrete and continuous optimization problems and Xu and Yuille [30] derive a mean field approximation to the problem which, after marginalizing the binary variables, can be solved by minimizing:

$$E_{xu}(\mathbf{B}) = -\sum_{i=1}^{n} \frac{1}{\beta} f_{xu}(\mathbf{e}_i, \beta, \eta) \qquad (6)$$

where $\mathbf{e}_i = \mathbf{d}_i - \mathbf{B}\mathbf{B}^T\mathbf{d}_i$ and where $f_{xu}(\mathbf{e}_i, \beta, \eta) = log(1 + e^{-\beta(e_{pca}(\mathbf{e}_i) - \eta)})$ is a function that is related to robust statistical estimators [1]. The β can be varied as an annealing parameter in an attempt to avoid local minima.

The above techniques are of limited application in computer vision problems as they reject entire images as outliers. In vision applications, outliers typically correspond to small groups of pixels and we seek a method that is robust to this type of outlier yet does not reject the "good" pixels in the data samples. Gabriel and Zamir [11] give a partial solution. They propose a weighted Singular Value Decomposition (SVD) technique that can be used to construct the principal subspace. In their approach, they minimize:

$$E_{gz}(\mathbf{B}, \mathbf{C}) = \sum_{i=1}^{n} \sum_{p=1}^{d} w_{pi}(d_{pi} - (\mathbf{b}^p)^T \mathbf{c}_i)^2 \qquad (7)$$

where, recall, $\mathbf{b}^p$ is a column vector containing the elements of the p-th *row* of $\mathbf{B}$. This effectively puts a weight, w_{pi} on every pixel in the training data. They solve the minimization problem with "criss-cross regressions" which involve iteratively computing dyadic (rank 1) fits using weighted least squares. The approach alternates between solving for $\mathbf{b}^p$ or $\mathbf{c}_i$ while the other is fixed; this is similar to the EM approach [24, 28] but without a probabilistic interpretation.

Gabriel and Odorof [12] note how the quadratic formulation in (1) is not robust to outliers and propose making the rank 1 fitting process in (7) robust. They propose a number of methods to make the criss-cross regressions robust but they apply the approach to very low-dimensional data and their optimization methods do not scale well to very high-dimensional data such as images. In the following section we develop this approach further and give a complete solution that estimates all the parameters of interest.

3 Robust Principal Component Analysis

The approach of Xu and Yuille suffers from three main problems: First, a single "bad" pixel value can make an image lie far enough from the subspace that the entire sample is treated as an outlier (i.e. $V_i = 0$) and has no influence on the estimate of $\mathbf{B}$. Second, Xu and Yuille use a least squares projection of the data $\mathbf{d}_i$ for computing the distance to the subspace; that is, the coefficients which reconstruct the data $\mathbf{d}_i$ are $\mathbf{c}_i = \mathbf{B}^T\mathbf{d}_i$. These reconstruction coefficients can be arbitrarily biased for an outlier. Finally, a binary outlier process is used which either completely rejects or includes a sample. Below we introduce a more general analogue outlier process that has computational advantages and provides a connection to robust M-estimation.

To address these issues we reformulate (5) as

$$E_{rpca}(\mathbf{B}, \mathbf{C}, \boldsymbol{\mu}, \mathbf{L}) = \sum_{i=1}^{n} \sum_{p=1}^{d} \left[L_{pi}(\frac{\tilde{e}_{pi}^2}{\sigma_p^2}) + P(L_{pi}) \right] \quad (8)$$

where $0 \leq L_{pi} \leq 1$ is now an analog outlier process that depends on both images and pixel locations and $P(L_{pi})$ is a penalty function. The error $\tilde{e}_{pi} = d_{pi} - \mu_p - \sum_{j=1}^{k} b_{pj} c_{ji}$ and $\boldsymbol{\sigma} = [\sigma_1 \ \sigma_2 \ ... \ \sigma_d]^T$ specifies a "scale" parameter for each of the d pixel locations.

Observe that we explicitly solve for the mean $\boldsymbol{\mu}$ in the estimation process. In the least-squares formulation the mean can be computed in closed form and can be subtracted from each column of the data matrix $\mathbf{D}$. In the robust case, outliers are defined with respect to the error in the reconstructed images which include the mean. The mean can no longer be computed and first subtracted. Instead it is estimated (robustly) analogously to the other bases.

Also, observe that PCA assumes an isotropic noise model; that is, the noise at each pixel is assumed to be Gaussian ($e_{pi} \sim N(0, \sigma^2)$). In the formulation here we allow the noise to vary for every row of the data ($e_{pi} \sim N(0, \sigma_p^2)$).

Exploiting the relationship between outlier processes and robust statistics [1], minimizing (8) is equivalent to minimizing the following robust energy function:

$$E_{rpca}(\mathbf{B}, \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{n} e_{rpca}(\mathbf{d}_i - \boldsymbol{\mu} - \mathbf{B}\mathbf{c}_i, \boldsymbol{\sigma})$$

$$= \sum_{i=1}^{n} \sum_{p=1}^{d} \rho(d_{pi} - \mu_p - \sum_{j=1}^{k} b_{pj} c_{ji}, \sigma_p) \qquad (9)$$

for a particular class of robust ρ-functions [1], where $e_{rpca}(\mathbf{x}, \boldsymbol{\sigma}) = \sum_{p=1}^{d} \rho(x_p, \sigma_p)$, for $\mathbf{x} = [x_1 \ x_2 \ ... \ x_d]^T$.

Throughout the paper, we use the Geman-McClure error function [10] given by $\rho(x, \sigma_p) = \frac{x^2}{x^2 + \sigma_p^2}$, where σ_p is a parameter that controls the convexity of the robust function and is used for deterministic annealing in the optimization process. This robust ρ-function corresponds to the penalty term $P(L_{pi}) = (\sqrt{L_{pi}} - 1)^2$ in (8) [1]. Details of the method are described below and in the Appendix.

Note that while there are robust methods such as RANSAC and Least Median Squares that are more robust than M-estimation, it is not clear how to apply these methods efficiently to high dimensional problems such as the robust estimation of basis images.

3.1 Quantitative Comparison

In order to better understand how PCA and the method of Xu and Yuille are influenced by intra-sample outliers, we consider the contrived example in Fig. 4 where four face images are shown. The second image is contaminated with one outlying pixel which has 10 times more energy than the sum of the others image pixels. To visualize the large range of pixel magnitudes the log of the image is displayed.

We force each method to explain the data using three basis images. Note that the approach of Xu and Yuille does

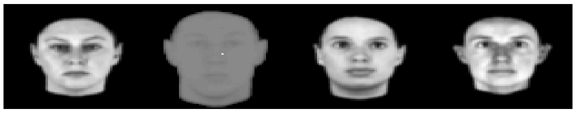

Figure 4: Original training Images. The second one is the log of original image.

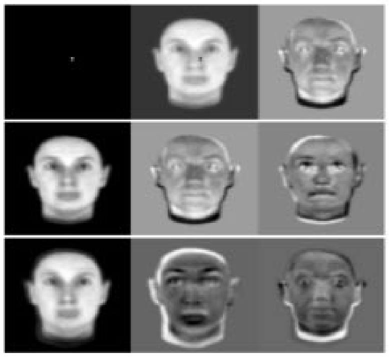

Figure 5: Learned basis images. *Top:* Traditional PCA. *Middle:* Xu and Yuille's method. *Bottom:* RPCA.

not solve for the mean, hence, for a fair comparison we neither solved for nor subtracted the mean for any of the methods. In this case the mean is approximately recovered as one of the bases. In Fig. 5, the three learned bases given by standard PCA, Xu and Yuille's method, and our proposed method are shown. The PCA basis captures the outlier in the second training image as the first principal component since it has the most energy. The other two bases approximately capture the principal subspace spanning the other three images. Xu and Yuille's method, on the other hand, discards the second image for being far from the subspace and uses all three bases to represent the three remaining images. The RPCA method proposed here, constructs a subspace that takes into account all four images while ignoring the single outlying pixel. Hence, we recover three bases to approximate the four images.

In Fig. 6 we project the original images (without outliers) onto the three learned basis sets. PCA "wastes" one of its three basis images on the outlying data and hence has only two basis images to approximate four training images. Xu and Yuille's method ignores all the useful information in image 2 as the result of a single outlier and, hence, is unable to reconstruct that image. Since it uses three basis images to represent the other three images, it can represent them perfectly. The RPCA method provides an approximation of all four images with three basis images. The MSRE (MSRE=$\frac{1}{n}\sum_{i=1}^{n}||\mathbf{d}_i - \boldsymbol{\mu} - \mathbf{B}\mathbf{c}_i||_2^2$) is less for RPCA than for the other methods: RPCA is 7.02, while PCA and Xu and Yuille's method are 18.59 and 9.02 respectively.

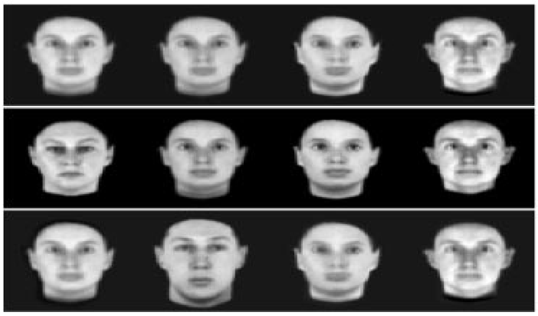

Figure 6: Reconstruction from noiseless images. *Top:* PCA. *Middle:* Xu and Yuille's method. *Bottom:* RPCA

3.2 Computational Issues

We now describe how to robustly compute the mean and the subspace spanned by the first k principal components. We do this without imposing orthogonality between the bases; this can be imposed later if needed [28]. To derive an algorithm for minimizing (9), we can reformulate the robust M-estimation problem as an iteratively re-weighted least-squares problem [6]. However, the computational cost of one iteration of weighted least squares is $\mathcal{O}\big(nk^2d\big)$ for $\mathbf{C}$ and $\mathcal{O}\big(nk^2d\big)$ for $\mathbf{B}$ [6]. Typically $d \gg n \gg k$, and, for example, estimating the bases $\mathbf{B}$ involves computing the solution of d systems of $k \times k$ equations, which for large d is computationally expensive. Rather than directly solving d systems of $k \times k$ equations for $\mathbf{B}$ and n systems of $k \times k$ equations for $\mathbf{C}$, we perform gradient descent with a local quadratic approximation [2] to determine an approximation of the step sizes, to solve for $\mathbf{B}$, $\mathbf{C}$ and $\boldsymbol{\mu}$. The robust learning rules for updating successively $\mathbf{B}$, $\mathbf{C}$ and $\boldsymbol{\mu}$ are as follows:

$$\mathbf{B}^{n+1} = \mathbf{B}^{n} - [\mathbf{H_b}].^{-1} \circ \frac{\partial E_{rpca}}{\partial \mathbf{B}}, \tag{10}$$

$$\mathbf{C}^{n+1} = \mathbf{C}^{n} - [\mathbf{H_c}].^{-1} \circ \frac{\partial E_{rpca}}{\partial \mathbf{C}}, \tag{11}$$

$$\boldsymbol{\mu}^{n+1} = \boldsymbol{\mu}^{n} - [\mathbf{H}_{\boldsymbol{\mu}}].^{-1} \circ \frac{\partial E_{rpca}}{\partial \boldsymbol{\mu}}. \tag{12}$$

The partial derivatives with respect to the parameters are:

$$\frac{\partial E_{rpca}}{\partial \mathbf{B}} = -\boldsymbol{\Psi}(\tilde{\mathbf{E}}, \boldsymbol{\sigma})\mathbf{C}^T \tag{13}$$

$$\frac{\partial E_{rpca}}{\partial \mathbf{C}} = -\mathbf{B}^T\boldsymbol{\Psi}(\tilde{\mathbf{E}}, \boldsymbol{\sigma}) \tag{14}$$

$$\frac{\partial E_{rpca}}{\partial \boldsymbol{\mu}} = -\boldsymbol{\Psi}(\tilde{\mathbf{E}}, \boldsymbol{\sigma})\mathbf{1}_n \tag{15}$$

where $\tilde{\mathbf{E}}$ is the reconstruction error and an estimate of the step size is given by:

$$\mathbf{H_b} = \boldsymbol{\zeta}(\tilde{\mathbf{E}}, \boldsymbol{\sigma})(\mathbf{C} \circ \mathbf{C})^T \qquad \mathbf{h_b}_i = max\ \ diag\Big(\frac{\partial^2 E_{rpca}}{\partial \mathbf{b}_i \partial \mathbf{b}_i^T}\Big)$$

$$\mathbf{H_c} = (\mathbf{B} \circ \mathbf{B})^T \boldsymbol{\zeta}(\tilde{\mathbf{E}}, \boldsymbol{\sigma}) \quad \mathbf{h_c}_i = max \;\; diag\Big(\frac{\partial^2 E_{rpca}}{\partial \mathbf{c}_i \partial \mathbf{c}_i^T} \Big)$$

$$\mathbf{H}_{\boldsymbol{\mu}} = \boldsymbol{\zeta}(\tilde{\mathbf{E}}, \boldsymbol{\sigma}) \mathbf{1}_n \quad \mathbf{h}_{\boldsymbol{\mu}_i} = max \;\; diag\Big(\frac{\partial^2 E_{rpca}}{\partial \boldsymbol{\mu} \partial \boldsymbol{\mu}^T} \Big)$$

where $\frac{\partial E_{rpca}}{\partial \mathbf{B}} \in \Re^{d \times k}$ is the derivative of E_{rpca} with respect to $\mathbf{B}$, and similarly for $\frac{\partial E_{rpca}}{\partial \mathbf{C}} \in \Re^{k \times n}$ and $\frac{\partial E_{rpca}}{\partial \boldsymbol{\mu}} \in \Re^{d \times 1}$. $\boldsymbol{\Psi}(\tilde{\mathbf{E}}, \boldsymbol{\sigma})$ is a matrix that contains the derivatives of the robust function; that is, $\psi(\tilde{e}_{pi}, \sigma_p) = \frac{\partial \rho(\tilde{e}_{pi}, \sigma_p)}{\partial \tilde{e}_{pi}} = \frac{2\tilde{e}_{pi}\sigma_p^2}{(\tilde{e}_{pi}^2 + \sigma_p^2)^2}$. $\mathbf{H_b} \in \Re^{d \times k}$ is a matrix in which every component ij is an upper bound of the second derivative; that is, $h_{ij} \geq \frac{\partial^2 E_{rpca}}{\partial b_{ij}^2}$ and, similarly, $\mathbf{H_c} \in \Re^{n \times k}$ and $\mathbf{H}_{\boldsymbol{\mu}} \in \Re^{d \times 1}$. Each element pi of the matrix $\boldsymbol{\zeta}(\tilde{\mathbf{E}}, \boldsymbol{\sigma}) \in R^{d \times n}$, contains the maximum of the second derivative of the ρ-function; that is $\zeta_{pi} = \max_{\tilde{e}_{pi}} \frac{\partial^2 \rho(\tilde{e}_{pi}, \sigma_p)}{\partial \tilde{e}_{pi}^2} = \frac{2}{\sigma_p^2}$.

Observe that now the computational cost of one iteration of the learning rules (10) or (11) is $\mathcal{O}\big(ndk\big)$. After each update of $\mathbf{B}$, $\mathbf{C}$, or $\boldsymbol{\mu}$, we update the error $\tilde{\mathbf{E}}$. Convergence behavior is described in the appendix.

3.3 Local measure of the scale value

The scale parameter $\boldsymbol{\sigma}$ controls the shape of the robust ρ-function and hence determines what residual errors are treated as outliers. When the the absolute value of the robust error $|\tilde{e}_{pi}|$ is larger than $\frac{\sigma_p}{\sqrt{3}}$, the ρ-function used here begins reducing the influence of the pixel p in image i on the solution. We estimate the scale parameters σ_p for each pixel p automatically using the local Median Absolute Deviation (MAD) [3, 23] of the pixel. The MAD can be viewed as a robust statistical estimate of the standard deviation, and we compute it as:

$$\sigma_p = \beta \max(1.4826 \, \mathrm{med}_R(|\mathbf{e}^p - \mathrm{med}_R(|\mathbf{e}^p|)|), \sigma_{min}) \tag{16}$$

where med_R indicates that the median is taken over a region, R, around pixel p and σ_{min} is the MAD over the whole image [3]. β is a constant factor that sets the outlier σ_p to be between 2 and 2.5 times the estimated standard deviation. For calculating the MAD, we need to have an initial error, $\mathbf{e}^p$, which is obtained as follows: we compute the standard PCA on the data, and calculate the number of bases which preserve the 55% of the energy (E_{pca}). This is achieved when the ratio between the energy of the reconstructed vectors and the original ones is larger than 0.55; that is, $\xi = \frac{\sum_{i=1}^{n} ||\mathbf{B}\mathbf{c}_i||_2^2}{\sum_{i=1}^{n} ||\mathbf{d}_i||_2^2} \geq 0.55$. Observe, that with standard PCA, this ratio can be calculated in terms of eigenvalues of the covariance matrix [9]. With this number of bases we compute the least-squares reconstruction error $\mathbf{E}$ and use that to obtain a robust estimate of $\boldsymbol{\sigma}$.

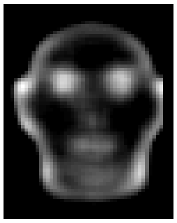

Figure 7: Local σ_p values estimated in 4×4 regions.

Figure 7 shows σ_p for the training set in Fig. 1. Observe how larger values of σ_p are estimated for the eyes, mouth, and boundary of the face. This indicates that there is higher variance in the training set in these regions and larger deviations from the estimated subspace should be required before a training pixel is considered an outlier.

4 Experimental Results

The behavior of RPCA is illustrated with a collection of 256 images (120×160) gathered from a static camera over one day. The first column of Fig. 8, shows example training images; in addition to changes in the illumination of the static background, 45% of the images contain people in various locations. While the people often pass though the view of the camera quickly, they sometimes remain relatively still over multiple frames. We applied standard PCA and RPCA to the training data to build a background model that captures the illumination variation. Such a model is useful for person detection and tracking [20].

The second column of Fig. 8 shows the result of reconstructing each of the illustrated training images using the PCA basis (with 20 basis vectors). The presence of people in the scene effects the recovered illumination of the background and results in ghostly images where the people are poorly reconstructed.

The third column shows the reconstruction obtained with 20 RPCA basis vectors. RPCA is able to capture the illumination changes while ignoring the people. In the fourth column, the outliers are plotted in white. Observe that the outliers primarily correspond to people, specular reflections, and graylevel changes due to the motion of the trees in the background. This model does a better job of accounting for the illumination variation in the scene and provides a basis for person detection. The algorithm takes approximately three of hours on a 900 MHz Pentium III in Matlab.

5 Discussion

While the examples illustrate the benefits of the method, it is worth considering when the algorithm may give unwanted results. Consider, for example, a face database that contains a small fraction of the subjects wearing glasses. In this case, the pixels corresponding to the glasses are likely to be treated as outliers by RPCA. Hence, the learned basis set will not contain these pixels, and it will be impossible to reconstruct images of people wearing glasses. Whether or not this is desirable behavior will depend on the application.

In such a situation, people with or without glasses can be considered as two different classes of objects and it might be more appropriate to robustly learn multiple linear subspaces corresponding to the different classes. By detecting outliers, robust techniques may prove useful for identifying such training sets that contain significant subsets that are not well modeled by the majority of the data and should be separated and represented independently. This is one of the classic advantages of robust techniques for data analysis.

6 Conclusion and Future Work

We have presented a method for robust principal component analysis that can be used for automatic learning of linear models from data that may be contaminated by outliers. The approach extends previous work in the vision community by modeling outliers that typically occur at the pixel level. Furthermore, it extends work in the statistics community by connecting the explicit outlier formulation with robust M-estimation and by developing a fully automatic algorithm that is appropriate for high dimensional data such as images. The method has been tested on natural and synthetic images and shows improved tolerance to outliers when compared with other techniques.

This work can be extended in a variety of ways. We are working on applications for robust Singular Value Decomposition, generalizing to robustly factorizing n-order tensors, on adding spatial coherence to the outliers and on developing a robust minor component analysis (useful when solving Total Least Square problems).

The use of linear models in vision is widespread and increasing. We hope robust techniques like those proposed here will prove useful as linear models are used to represent more realistic data sets. Towards that end an implementation of the method can be downloaded from `http://www.salleURL.edu/˜ftorre`.

Acknowledgments. The first author was partially supported by Catalonian Government grant 2000 BE I200132. We are grateful to Allan Jepson for many discussions on robust learning and PCA. We also thank Niko Troje for providing the face image database. Images from the Columbia database were also used in the examples (http://www.cs.columbia.edu/CAVE/research/softlib/).

References

[1] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 25(19):57–92, 1996.

[2] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of objects using view-based representation. *ECCV*, pp. 329–342, 1996.

[3] M. Black, G. Sapiro, D. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. Im. Proc.*, 7:421–432, 1998.

[4] M. Black, Y. Yacoob, A. Jepson, and D. Fleet. Learning parameterized models of image motion. *CVPR*, pp. 561–567, 1997.

[5] N. Campbell. Multivariate Analysis I: Robust Covariance Estimation. *Applied Statistics*, 29(3):231–2137, 1980.

[6] F. De la Torre and M. Black A Framework for Robust Subspace Learning. *Submitted to IJCV*.

[7] C. Eckart and G.Young. The approximation of one matrix by another of lower rank. *Psychometrika* 1, pp. 211–218, 1936.

[8] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *5th ECCV*, 1998.

[9] K. Diamantaras. *Principal Component Neural Networks (Theory and Applications)*. John Wiley & Sons, 1996.

[10] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute.* LII-4:5, 1987.

[11] K. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics, Vol. 21, pp.*, 21:489–498, 1979.

[12] K. Gabriel and C. Odoroff. Resistant lower rank approximation of matrices. *Data Analysis and Informatics, III.*, 1984.

[13] D. Geiger and R. Pereira. The outlier process. *IEEE Workshop on Neural Networks for Signal Proc.*, pp. 61–69, 1991.

[14] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York., 1986.

[15] P. Huber. *Robust Statistics*. New York: Wiley, 1981.

[16] I. Jolliffe. *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[17] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 4(8):549–562, 1995.

[18] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. *ICCV*, 1995.

[19] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 1(14):5–24, 1995.

[20] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. *ICVS. Gran Canaria, Spain*, Jan. 1999.

[21] E. Oja. A simplified neuron model as principal component analyzer. *J. Mathematical Biology*, (15):267–273, 1982.

[22] R. Rao. An optimal estimation approach to visual perception and learning. *Vision Research*, 39(11):1963–1989, 1999.

[23] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, 1987.

[24] S. Roweis. EM algorithms for PCA and SPCA. *NIPS*, pp. 626–632, 1997.

[25] F. Ruymagaart. A Robust Principal Component Analysis. *J. Multivariate Anal.*, vol. 11, pp. 485–497, 1981.

[26] T. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, (2):459–473, Nov. 1989.

[27] H. Shun, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *PAMI* , 17(9):855–867,1995.

[28] M. Tipping and C. Bishop. Probabilistic principal component analysis. Journal of the Royal Statistical Society B, 61, 611-622, 1999.

[29] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

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

7 Appendix: Implementation Details

In standard PCA, the number of bases is usually selected to preserve some percentage of the energy (E_{pca}). In RPCA this criterion is not straightforward to apply. The robust error, E_{rpca}, (9), depends on $\boldsymbol{\sigma}$ and the number of bases so we can not directly compare energy functions with different scale parameters. Moreover, the energy of the outliers is *confused* with the energy of the signal. We have experimented with different methods for automatically selecting of the number of basis images including the Minimum Descriptor Length criterion and Akaike Information Criterion. However, these model selection methods do not scale well to high dimensional data and require the manual selection of a number of normalization factors. We have exploited more heuristic methods here that work in practice.

We apply standard PCA to the data, and calculate the number of bases that preserve 55% of the energy (E_{pca}). With this number of bases, we apply RPCA, minimizing (9), until convergence. At the end of this process we have a matrix $\mathbf{W}$ that contains the weighting of each pixel in the training data. We detect outliers using this matrix and set the values of $\mathbf{W}$ to 0 if $|w_{pi}| > \frac{\sigma_p}{\sqrt{3}}$ and to w_{pi} otherwise, obtaining $\mathbf{W}^*$. We then incrementally add additional bases and minimize $E(\mathbf{B}, \mathbf{C}, \boldsymbol{\mu}) = ||\mathbf{W}^* \circ (\mathbf{D} - \boldsymbol{\mu}\mathbf{1}_n^T - \mathbf{B}\mathbf{C})||_2^2$ with the same method as before but maintaining constant weights $\mathbf{W}^*$. Each element, w^*_{pi} will be equal to $w^*_{pi} = \psi(\tilde{e}_{pi}, \sigma_p)/\tilde{e}_{pi}$ [6]. We proceed adding bases until the percentage of energy accounted for, ξ, is bigger than 0.9, where $\xi = \frac{\sum_{i=1}^{n} \mathbf{c}_i^T \mathbf{B}^T \mathbf{W}_i^* \mathbf{B} \mathbf{c}_i}{\sum_{i=1}^{n} (\mathbf{d}_i - \boldsymbol{\mu})^T \mathbf{W}_i^* (\mathbf{d}_i - \boldsymbol{\mu})}$.

In general the energy function (9) is non-convex and the minimization method can get trapped in local minima. We make use of a deterministic annealing scheme which helps avoid these local minima [2]. The method begins with $\boldsymbol{\sigma}$ being a large multiple of (16) such that all pixels are inliers. Then $\boldsymbol{\sigma}$ is successively lowered to the value given by (16), reducing the influence of outliers. Several realizations with different initial solutions are performed, and the solution with the lowest minimum error is chosen. Since minimization of (9) is an iterative scheme, an initial guess for the parameters $\mathbf{B}, \mathbf{C}$ and $\boldsymbol{\mu}$ has to be given. The initial guess for the parameters $\mathbf{B}$, is chosen to be the mean of $\mathbf{D}$ plus random Gaussian noise. The convergence of all the trials have given similar energy and visual results.

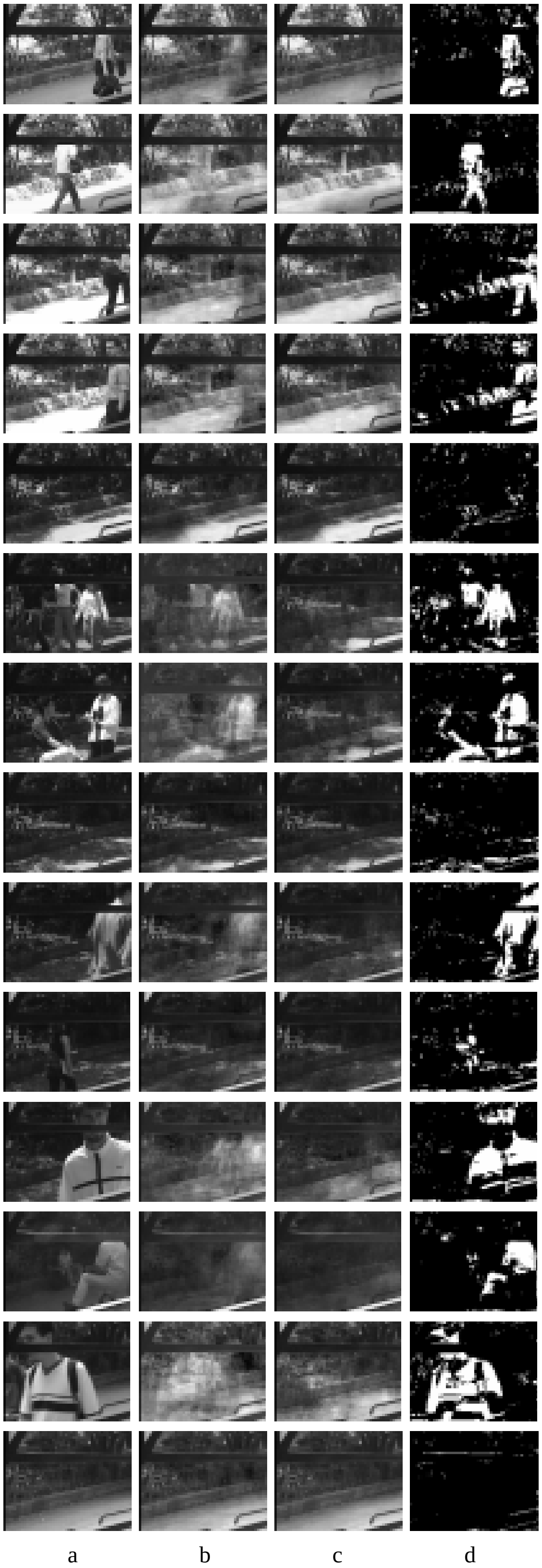

Figure 8: *(a)* Original Data. *(b)* PCA reconstruction. *(c)* RPCA reconstruction. *(d)* Outliers.