Correspondence

Inductive Robust Principal Component Analysis

Bing-Kun Bao, Guangcan Liu, *Member, IEEE*,
Changsheng Xu, *Senior Member, IEEE*,
and Shuicheng Yan, *Senior Member, IEEE*

***Abstract*— In this paper, we address the error correction problem, that is, to uncover the low-dimensional subspace structure from high-dimensional observations, which are possibly corrupted by errors. When the errors are of Gaussian distribution, principal component analysis (PCA) can find the optimal (in terms of least-square error) low-rank approximation to high-dimensional data. However, the canonical PCA method is known to be extremely fragile to the presence of gross corruptions. Recently, Wright *et al.* established a so-called robust principal component analysis (RPCA) method, which can well handle the grossly corrupted data. However, RPCA is a transductive method and does not handle well the new samples, which are not involved in the training procedure. Given a new datum, RPCA essentially needs to recalculate over all the data, resulting in high computational cost. So, RPCA is inappropriate for the applications that require fast online computation. To overcome this limitation, in this paper, we propose an inductive robust principal component analysis (IRPCA) method. Given a set of training data, unlike RPCA that targets on recovering the original data matrix, IRPCA aims at learning the underlying projection matrix, which can be used to efficiently remove the possible corruptions in any datum. The learning is done by solving a nuclear-norm regularized minimization problem, which is convex and can be solved in polynomial time. Extensive experiments on a benchmark human face dataset and two video surveillance datasets show that IRPCA cannot only be robust to gross corruptions, but also handle the new data well and in an efficient way.**

***Index Terms*— Error correction, robust principal component analysis (PCA), subspace learning.**

I. Introduction

Data that arise from real applications, such as computer vision and document analysis, are often characterized by low-rank subspaces, leading to the challenging problem of exploring the low-rank subspace structures of the data [1]–[6]. Generally, a fundamental problem in this topic is to handle the data containing errors, i.e., the *error correction* problem described as follows.

Problem 1 (Error Correction): Suppose the data are generated by

$$x = y + e, \quad x \in \Re^d \tag{1}$$

where y lies on a subspace of low-rank and e is the error term. Given any data vector x generated as above, the goal is to identify the original data vector y and the error e.

Statistically, the above problem is equal to exploring the *principal components* (identified by y) of the data, as a low-rank subspace can be well modeled by a degenerate Gaussian distribution [7]. Provided that the errors follow Gaussian distribution with small variance, the widely used principal component analysis (PCA) [8] can well handle the above problem. Given a set of n training data vectors $X = [x_1, x_2, \ldots, x_n]$, with each x_i being generated as in (1), PCA learns a low-rank projection by minimizing the following reconstruction errors:

$$\min_U \|X - UU^T X\|_F^2, \quad \text{s.t. } U^T U = \mathtt{I}_r \tag{2}$$

where $\mathtt{I}_r$ is an identity matrix of size $r \times r$, $\|\cdot\|_F$ denotes the Frobenious norm of a matrix, and r is a parameter that specifies the rank of U (i.e., the number of columns of U). The solution to the above optimization problem can be efficiently found by singular value decomposition (SVD), which is computationally stable. For a given data vector x, its principal components y can be estimated by

$$y = U^*(U^*)^T x \tag{3}$$

where U^* is learnt from a training set X by solving (2), i.e., calculating the SVD of X. Due to the advantages of computational stability and efficiency, PCA has become one of the most widely used tools for error correction. However, in real scenario, the application and performance of PCA are limited due to lack of robustness to gross corruptions, that is, the estimation obtained by PCA could be far away from the underlying true subspace structure in the presence of gross corruptions [9], as illustrated in Fig. 1. To overcome the limitations of PCA, recent years have witnessed a surge of robust principal component analysis (RPCA) methods (e.g., [9]–[15]). In particular, Wright *et al.* [9], [14] recently established a RPCA method which is emerging as a powerful tool for various applications, such as document analysis [16], multimedia [17], and image processing [18]. For a given set of data vectors $X = [x_1, x_2, \ldots, x_n]$, RPCA[1] finds their principal components $Y = [y_1, y_2, \ldots, y_n]$ by solving the following convex optimization problems:

$$\min_{Y,E} \|Y\|_* + \lambda \|E\|_1, \quad \text{s.t. } X = Y + E \tag{4}$$

where $\|\cdot\|_*$ denotes the nuclear-norm, also known as the trace-norm or Ky Fan-norm (sum of the singular values), $\|\cdot\|_1$ is the ℓ_1-norm chosen for characterizing the corruptions possibly existed in data, and $\lambda > 0$ is a parameter. Unlike the small Gaussian noise assumed in PCA, RPCA can well handle the gross corruptions of large magnitude [9], provided that the corruptions are sparse enough (i.e., only a fraction of entries are corrupted). Note here that the sparse assumption is mild, since it is ubiquitous in reality due to occlusion or sensor failure. It is

Manuscript received May 14, 2011; revised February 21, 2012; accepted March 10, 2012. Date of publication April 3, 2012; date of current version July 18, 2012. This work was supported in part by the National Program on Key Basic Research Project 973 Program under Project 2012CB316304, the National Natural Science Foundation of China under Grant 90920303 and Grant 61003161, the China Post-Doctoral Science Foundation under Grant 2011M500430, and the Project Grant NRF2007IDM-IDM002-069 on Life Spaces from the IDM Project Office, Media Development Authority, Singapore. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rafael Molina.

B.-K. Bao and C. Xu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the China-Singapore Institute of Digital Media, 119613, Singapore (e-mail:bingkunbao@gmail.com; csxu@nlpr.ia.ac.cn).

G. Liu is with the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: gcliu@illinois.edu).

S. Yan is with the National University of Singapore, 10000, Singapore (e-mail: eleyans@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2192742

[1]Notice that there are many RPCA methods in existence. For ease of presentation, in this paper, "RPCA" solely refers to the method introduced in [9] and [14].

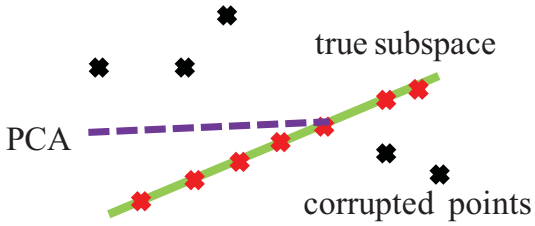


Fig. 1. Illustrating that PCA cannot well handle the data corrupted by gross corruptions.

also necessary, because in *unsupervised* environments, the learning is impractical if all the entries of X have been grossly corrupted. However, it is not easy for RPCA to generalize the learnt model to new data. Let (Y^*, E^*) be the optimal solution to (4), then a possible way for RPCA to be used with test data is as follows: compute the SVD of Y^*, denoted as $Y^* = U^*\Sigma^*V^*$, and use $y = U^*(U^*)^T x$ to deal with any new datum x. This approach (denoted as "RPCA_I"), unfortunately, does not work well, as it cannot even well handle the training data X itself. To be more precise, the produced solution $(U^*(U^*)^T X, X - U^*(U^*)^T X)$ may be not optimal to (4). Evidence is that the error term estimated by $E = X - U^*(U^*)^T X$ is usually not sparse, as verified by our experiments. Hence, RPCA is essentially a transductive method, which is inappropriate for the applications requiring online computation.

Due to the limitations of PCA and RPCA, in this paper, we therefore introduce the so-called inductive robust principal component analysis (IRPCA), which cannot only well handle the gross corruptions (comparable to RPCA), but also own good generalization ability (much better than PCA and RPCA_I). To achieve this, we establish a simple but effective algorithm to learn a *low-rank* projection from a given set of training data. The learning is done by solving a nuclear-norm regularized minimization problem, which is convex and can be solved in polynomial. For any new datum corrupted by gross errors, the learnt projection can efficiently remove the possible corruptions by projecting the datum onto its underlying subspace. Extensive experiments on a benchmark human face dataset and two video surveillance datasets show that IRPCA cannot only be robust to gross corruptions, but also handle well the new data in an efficient way.

II. Related Work

Error correction has a long history (e.g., [9]–[15]). The rather old methods (e.g., [19] and [20]) usually suffer two pitfalls: performance degradation with dimension increase and computational intractability. To be more precise, these methods have breakdown points proportional to the inverse of dimensionality, and hence are useless in the high-dimensional regime we consider. Algorithms with nondiminishing breakdown points, such as projection-pursuit [12] are nonconvex or even combinatorial, and hence computationally intractable (NP-hard) as the size of the problem scales.

Methods based on nuclear-norm minimization to recover low-rank matrices are now standard (e.g., [9], [14], [15], [21], and [22]). In particular, the work done by Wright *et al.* [14], which is referred to as "RPCA" in this paper, has taken the nuclear-norm minimization to address the error correction problem 1. While RPCA is powerful in terms of both theory [9] and applications [16]–[18], as discussed in Section I, it is essentially a transductive method and is inappropriate for the applications requiring online computation. Namely, given a new datum, RPCA essentially needs to recalculate over all the data, resulting in high computational cost.

Recently, Liu *et al.* [6], [23] generalize the concept of RPCA, and establish the so-called low-rank representation (LRR) method. The general formulation of LRR is as follows:

$$\min_{Z,E} \|Z\|_* + \lambda\|E\|_\ell, \quad \text{s.t. } X = AZ + E \tag{5}$$

where A is a "dictionary" that linearly spans the data space, and $\|\cdot\|_\ell$ indicates certain strategy for characterizing the error term E. By choosing the dictionary as $A = \mathtt{I}$ and the regularization strategy as $\|E\|_1$, LRR identifies RPCA. So, LRR is a generalization of RPCA. In [6], it is shown that the problem of *subspace segmentation* [24] can be well addressed by solving the following specific LRR problems:

$$\min_{Z,E} \|Z\|_* + \lambda\|E\|_{2,1}, \quad \text{s.t. } X = XZ + E \tag{6}$$

which is actually to set $A = X$ and use the $\ell_{2,1}$-norm to characterize E in (5). Indeed, the above LRR method can also address the problem of error correction, as illustrated in [6]. However, like RPCA, this method is also a transductive method and cannot process new data efficiently. Namely, for a new datum x given, this method essentially needs to recalculate over all the data (note that each column of X is a data point), resulting in high computational cost. Moreover, the $\ell_{2,1}$-norm is chosen to model outliers and sample-specific corruptions [23], and thus does not fit well the random corruptions addressed in this paper.

III. Robust Error Correction by IRPCA

The core of our IRPCA method is the learning of a low-rank projection, which takes the role of projecting corrupted data vectors onto the underlying subspace. The learning is done by solving a convex optimization problem, as detailed in the following.

A. Pursue Low-Rank Projection by Convex Optimization

As mentioned above, the goal of IRPCA method is to own the advantages of both PCA [8] and RPCA [9], [14], namely, not only can well handle the data contaminated by gross corruptions but also own good generalization ability for processing new data. Without imposing any restriction on the corruptions, generally, it is impracticable to achieve the goal of IRPCA. For example, if the corruptions are very disordered as shown in Fig. 2(a), then there generally exists no simple model for fitting the corruptions. Fortunately, IRPCA is feasible due to the following two facts. First, even if the corruptions are disordered as in Fig. 2(a), there exists a linear projection P_0 that can project any data point x onto the underlying subspace, and thus produce accurate recovery results (although P_0x may not exactly recover y in this case). Second, note that two high-dimensional vectors, which are generated independently, are *almost* orthogonal to each other [25], i.e., the corruptions usually lie inside the null space of the true subspace, as shown in Fig. 2(b). In this case, there exists a linear projection P_0 for strictly removing the gross corruptions of the data. Hence, given a data vector x generated in (1), we could obtain its principal components y (or an approximation of y) by using P_0x, where P_0 is the matrix that identifies the underlying low-rank projection. Provided that the data are of dimension d, the size of P_0 is $d \times d$.

Up to present, the only left issue is to establish an algorithm for learning the low-rank projection P_0 from a given set of training data. According to the low rankness of the projection and the sparsity of the noise, we may learn the low-rank projection $P \in \Re_{d\times d}$ and the principal components Y by solving the following problems:

$$\min_{P,Y} \text{rank}\,(P) + \lambda\|X - PX\|_0, \quad \text{s.t. } Y = PX$$

where $X = [x_1, \ldots, x_n]$ is a data matrix consisting of a set of training vectors, $Y = [y_1, \ldots, y_n]$ represents the principal components of X,

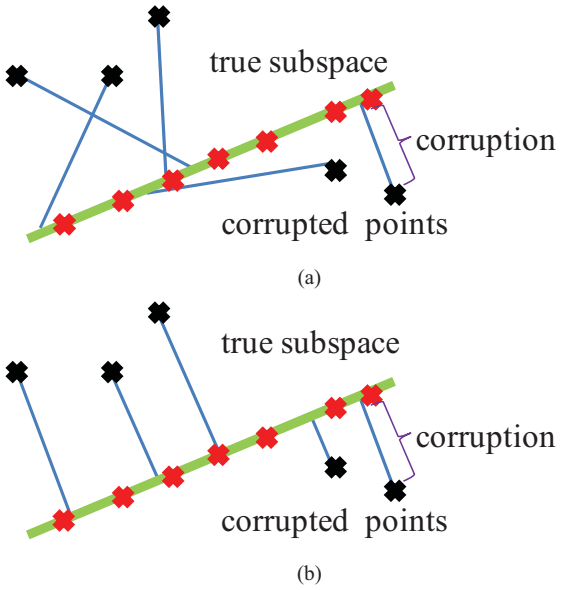


Fig. 2. (a) Corruptions are disordered. (b) Corruptions are orthogonal to the true low-rank subspace.

$\|\cdot\|_0$ denotes the ℓ_0-norm, i.e., the number of nonzero entries of a matrix, and $\lambda > 0$ is a parameter. Since $X - PX$ identifies the noise E, we rewrite the above formulation into the following:

$$\min_{P,E} \text{rank}\,(P) + \lambda\|E\|_0, \quad \text{s.t. } X = PX + E. \tag{7}$$

The above optimization problem is difficult to solve due to the discrete nature of the rank function and the ℓ_0-norm. As a common practice in rank minimization problems, we replace the rank function with the nuclear-norm. Moreover, as suggested by [26], it is adequate to replace the ℓ_0-norm with the ℓ_1-norm. Hence, we could learn the low-rank projection by solving the following *convex* optimization problems:

$$\min_{P,E} \|P\|_* + \lambda\|E\|_1, \quad \text{s.t. } X = PX + E. \tag{8}$$

Suppose P^* is an optimal solution (with respect to the variable P) to the above problem, then for any data vector x given, we could estimate the principal components y and error part e by using P^*x and $x - P^*x$, respectively.

B. Solving the Optimization Problem

Problem (8) can be solved in a similar way as the LRR problem (6). One just needs to transform it into the following equivalent problems:

$$\min_{P,E} \|P^T\|_* + \lambda\|E^T\|_1, \quad \text{s.t. } X^T = X^T P^T + E^T.$$

According to the report in [6], the computational complexity of solving (8) is $O(d^3)$, which is expensive for high-dimensional data. For efficiency, we do not solve (8) directly, but instead to convert (8) into a simpler problem by utilizing the theories established by Liu *et al.* [23]. By Theorem 1 of [23], the transpose of the optimal solution P^* always lies within the subspace spanned by the columns of X. Namely, P^* can be factorized into the product of two matrices: $P^* = L^*(Q^*)^T$, where Q^* is computed by orthogonalizing the columns of X. So, (8) can be equivalently converted into a simpler problem by replacing P with $L(Q^*)^T$

$$\min_{L,E} \|L\|_* + \lambda\|E\|_1, \quad \text{s.t. } X = LA + E \tag{9}$$

Algorithm 1 Solving Problem (9) by Inexact ALM

Input: matrices X and A; parameter λ.

Initialize: $L = J = 0, E = 0, Y_1 = 0, Y_2 = 0, \mu = 10^{-6}, \mu_{\max} = 10^6, \rho = 1.1$, and $\varepsilon = 10^{-8}$.

while not converged **do**

1. fix the others and update J by

$$J = \arg\min \frac{1}{\mu}||J||_* + \frac{1}{2}||J - (Z + \frac{Y_2}{\mu})||_F^2$$

2. fix the others and update L by

$$L = ((X - E)A^T + J + \frac{Y_1 A^T - Y_2}{\mu})(\mathtt{I} + AA^T)^{-1}$$

3. fix the others and update E by

$$E = \arg\min \frac{\lambda}{\mu}||E||_1 + \frac{1}{2}||E - (X - LA + \frac{Y_1}{\mu})||_F^2$$

4. update the multipliers

$$Y_1 = Y_1 + \mu(X - LA - E),$$
$$Y_2 = Y_2 + \mu(Z - J)$$

5. update the parameter μ by $\mu = \min(\rho\mu, \mu_{\max})$

6. check the convergence conditions:

$$||X - LA - E||_\infty < \varepsilon \text{ and } ||Z - J||_\infty < \varepsilon$$

end while

Output: (L, E).

where $A = (Q^*)^T X$. The above problem can be solved in the same way as that for the LRR problem (5). For the competence of presentation, the detailed optimization procedure is outlined in Algorithm 1, which is based on the inexact argument Lagrange multiplier (ALM) [27] method. Note that although Steps 1 and 3 of the algorithm are convex problems, they both have closed-form solutions. Step 1 is solved via the singular value thresholding (SVT) operator [28], while Step 3 is solved via the shrinkage operator [27].

Suppose (L^*, E^*) is an optimal solution to (9), then the optimal solution to (8) can be recovered by $(L^*(Q^*)^T, E^*)$. As the number of the rows of A is at most k ($k = \text{rank}\,(X) \leq \min(d, n)$), the computational complexity of Algorithm 1 is $O(k^3 + k^2 d + knd)$, which is up bounded by $O(n^3 + n^2 d)$ (assume $n \leq d$). So the computation is quite efficient if n (and/or d) is small. Also, the computation cost of IRPCA is mainly spent in the training process. After learning the projection P^* from the training data, it is very efficient for IRPCA to handle the new data. Namely, for any data vector x given, we only need to calculate P^*x.

C. Discussion

1) Connections With LRR: The general LRR formulation (5) can involve our IRPCA formulation (8). By setting $A = X$, choosing $\|E\|_1$ and considering each row of X as a data point, then (5) is identified to (8). So, IRPCA could be regarded as a special example of LRR. Nevertheless, the IRPCA problem explored in this paper has not been explored before. What has been explored in [6] and [23] is the formulation (6), which is designed for subspace segmentation in the presence of outliers and sample-specific corruptions. Although the method based on (6) can be used for error correction [6], like RPCA, this method is also a transductive method and cannot process new data efficiently. Namely, for a new datum x given, this

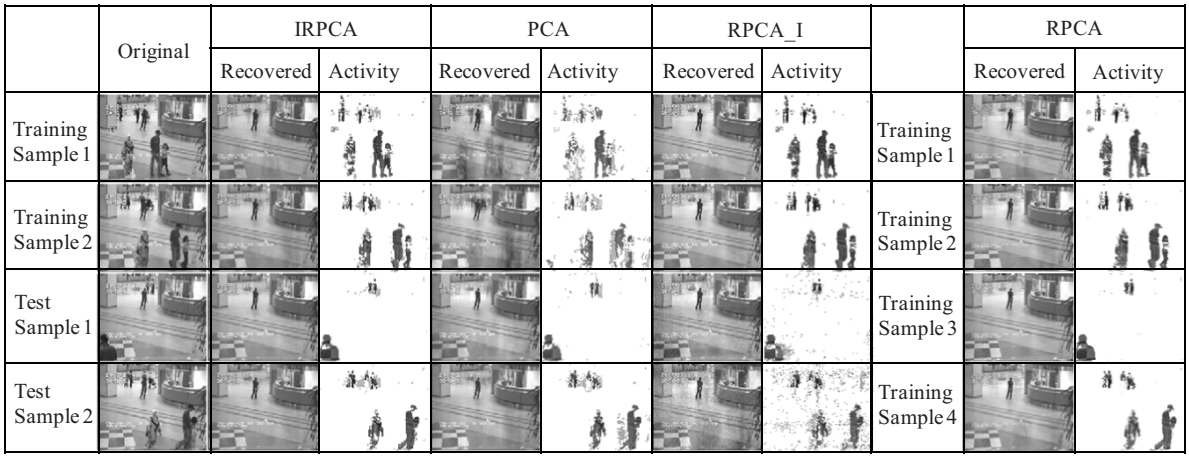


Fig. 3. This set of frames, taken in an airport [30], illustrates the results from IRPCA, PCA, RPCA_I (use RPCA in an inductive way, as presented in the third paragraph of Section I), and RPCA. For IRPCA, PCA, and RPCA_I, the top two frames are selected from the training set, and the bottom two frames are selected from the test set. For RPCA, these four frames are all regarded as training samples.

TABLE I

PRECISION, RECALL, AND F-SCORE FOR ACTIVITIES ON AIRPORT HALL SURVEILLANCE VIDEO [30]

	IRPCA		PCA		RPCA_I		RPCA
	Training	Test	Training	Test	Training	Test	Training
Precision	0.8435	0.8254	0.6802	0.6788	0.9028	0.5625	0.9119
Recall	0.8218	0.8132	0.7534	0.7369	0.8649	0.7773	0.8597
F-score	0.8325	0.8193	0.7149	0.7067	0.8834	0.6527	0.8850

method essentially needs to recalculate over all the data (note that each column of X is a data point), resulting in high computational cost. Moreover, as mentioned, the $\ell_{2,1}$-norm is designed for modeling outliers and sample-specific corruption, but cannot fit well the random corruptions addressed in this paper.

2) Connections With PCA: To see the connections between PCA and our IRPCA, we consider the special case that the data are noiseless, i.e., we consider a simpler version of (8)

$$\min_{P} \|P\|_*, \quad \text{s.t. } X = PX. \tag{10}$$

Here, the noise E is assumed to be zero. Based on the theorems introduced by [23] and [29], the following theorem can be simply proven.

Theorem 1: For any data matrix X given, the minimizer to (10) is unique and has the following closed forms:

$$P^* = UU^T$$

where $X = U\Sigma V^T$ is the skinny SVD of X.

The above theorem illustrates that our IRPCA method identifies PCA when the data are noiseless. Since the proof can be simply finished by making use of the results in [23] and [29], we omit the detailed procedure here.

3) Connections With RPCA: Both RPCA [9] and our IRPCA method can be regarded as special examples of a general framework as follows:

$$\min_{P,E} \|P\|_* + \lambda\|E\|_1, \quad \text{s.t. } X = PA + E$$

which is a simple variation of the general LRR formulation (5) (note that each column X is a data vector). By choosing $A = \mathtt{I}$ (i.e., standard basis), the above formulation corresponds to RPCA. By choosing $A = X$, the above formulation identifies our IRPCA method.

IV. EXPERIMENTS

In this section, we systematically evaluate the proposed method on the applications, including background modeling from video, removing occlusions from face images [9], and face recognition. All the experiments are performed on an Intel Xeon X5570 workstation with 3.0-GHz CPU and 24-GB memory. The code is implemented on MATLAB platform.

A. Background Modeling from Surveillance Video

Applications, such as event detection and human action recognition, need the function of automatically identifying the activities out from the background. For accuracy and reliability, such applications require to estimate a good model, which is not only robust to the background variations, but also can process real-time identification. As discussed in [9], the RPCA method fits well this task. Namely, the matrix composed by stacking the surveillance video frames as columns is naturally of low-ranked, and the activities can be treated as sparse corruptions. So, we illustrate two videos in our experiments. The first one is a sequence of 200 grayscale frames taken in an airport. Each frame is of size 176×144. The second video is taken with several illumination changes. There are in total 250 frames, each of which has a size of 128×160. Compared to the first video, this video has less activities, but the illumination changes more drastically toward the end of the sequence. To obtain a quantitative comparison, we manually quote out the activities and use the F-score to measure the accuracy of segmentation

$$F\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{11}$$

Here, *precision* and *recall* are computed as follows:

$$\text{precision} = \frac{|G \cap T|}{|T|}, \quad \text{recall} = \frac{|G \cap T|}{|G|} \tag{12}$$

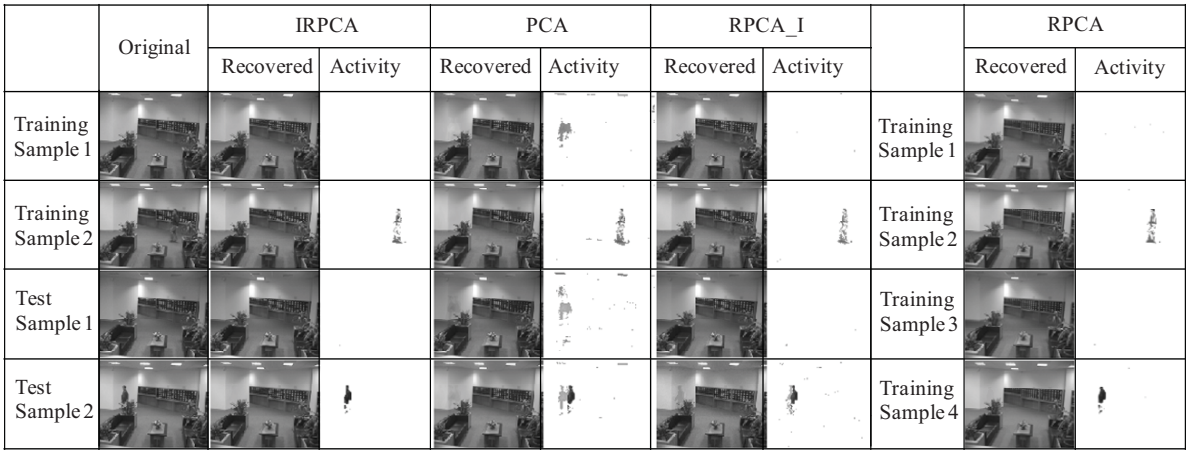


Fig. 4. This set of frames, taken in a lobby with varying illumination [30], illustrates the results from IRPCA, PCA, RPCA_I, and RPCA. For IRPCA, PCA, and RPCA_I, the top two frames are selected from the training set, and the bottom two frames are selected from the test set. For RPCA, these four frames are all regarded as training samples.

Fig. 5. Recovered results from IRPCA, PCA, RPCA_I, and RPCA in the experiment in which 10% pixels are corrupted. We have tuned the parameters of all methods to the best.

where G is the ground truth mask (of the foreground activity), and T is the mask output by an algorithm.

For the first video, we randomly select 100 frames as the training set $X \in \Re^{25344\times 100}$. The remaining 100 frames act as the testing set. The parameters of various algorithms are set as follows: the parameter λ of IRPCA is set as 0.0001; in RPCA, the parameter λ is set as 0.0068 according to [9] and for PCA, the rank of the background subspace is optimally chosen as 11. Fig. 3 shows some experimental results, including the original frames from the video, and the recovered frames and identified activities from IRPCA, PCA, RPCA_I, and RPCA. Table I shows the qualitative comparison results. It can be seen that our IRPCA method is much more effective than PCA in terms of separating activities from backgrounds. While RPCA can produce the most accurate segmentation, it is essentially a transductive method and cannot process the new samples efficiently, namely it needs to solve (4) over all data when a new datum is raised. In contrast, our IRPCA requires much less computational cost in processing new samples, namely it only needs to compute P^*x for processing a new datum x. This offers advantage in the applications requiring fast online computation to identify the activities.

For the second video, we randomly select 100 frames for training, and the remaining 150 frames for testing. In IRPCA, the parameter λ is set as 0.0035. In RPCA, the parameter λ is set as 0.007 according to [9]. In PCA, the rank of the background subspace is optimally chosen as ten. Fig. 4 and Table II shows the performance of IRPCA, PCA,

Fig. 6. Recovered results from IRPCA, PCA, RPCA_I, and RPCA in the experiment in which 30% pixels are corrupted. We have tuned the parameters of all methods to the best.

TABLE II
PRECISION, RECALL, AND F-SCORE FOR ACTIVITIES ON LOBBY SURVEILLANCE VIDEO [30]

	IRPCA		PCA		RPCA_I		RPCA
	Training	Test	Training	Test	Training	Test	Training
Precision	0.9420	0.8873	0.3391	0.2887	0.9329	0.4744	0.9235
Recall	0.7930	0.7562	0.9198	0.8880	0.7828	0.7366	0.7888
F-score	0.8612	0.8165	0.4955	0.4358	0.8531	0.5771	0.8508

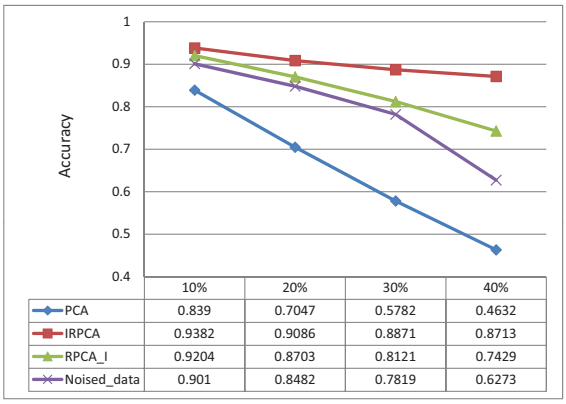
RPCA_I, and RPCA. It can be seen that IRPCA achieves comparable performance to RPCA (in terms of segmentation accuracy), with the advantage of handling the testing frames much more efficiently.

B. Face Image De-Noising and Recognition

We compare our method with PCA and RPCA in recovering the face images from corruptions. The dataset is selected from the CMU PIE face dataset [31], which contains more than 40 000 facial images of 68 people. The images were acquired across different poses, under variable illumination conditions, and with different facial expressions. We choose one near frontal pose C07, which includes 1629 images of 68 people, for experiments. These images are split into two parts: 539 images associated to 68 people are set as the training set and the rest 1090 images associated to the 68 people are set as the testing set.

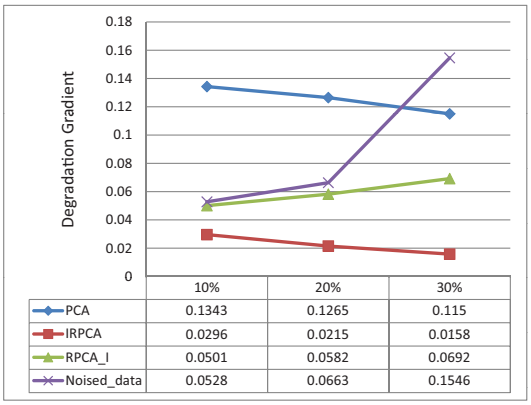
To investigate the robustness of various methods, half of the training set and the whole test set are corrupted by artificial noise. We consider two settings for the noise degree: one is with 10% pixels corrupted and the other is with 30% pixels corrupted. For a pixel chosen to be corrupted, its observed value is computed by adding Gaussian noise with zero mean and standard deviation $\sigma = 70$. For the inductive methods IPCA, PCA, and RPCA_I, only the training set is used to train their models. For RPCA, both the training and test sets are merged as one set to train its model. Figs. 5 and 6 show the results on two different settings of noise. It can be seen that PCA may fail to completely remove the noise (see the fourth column of Figs. 5 and 6). This verifies that PCA does not well resolve gross errors. Whereas, IRPCA can produce much better recoveries (see the third column of Figs. 5 and 6). Compared to RPCA, as mentioned above, the superiority of IRPCA is its generalization ability of handling the new testing images.

For quantitative evaluation, we also investigate the performance of IRPCA under the context of face recognition. Suppose P^* is the optimal projection learnt from a set of training samples. Then for any given data sample x, we could compute its feature vector computed by P^*x for classification. The final classification results are obtained by support vector machine with linear kernel. Since RPCA has no generalization ability of handling new testing data, we only compare IRPCA with PCA and RPCA_I in this experiment. To investigate the robustness of various methods, half of the training samples are corrupted. For each image chosen to be corrupted, its 30% pixels are corrupted by adding Gaussian noise. Fig. 7(a) shows the results on face recognition under different corruption rates in the test data. From these results, we can observe that IRPCA clearly outperforms the other competitors. While the corruption rate is increasing, the classification performance of IRPCA degrades much slower than the others, as shown in Fig. 7(b). Here, the degradation gradient at the ith corruption rate defined as $\text{acc}_i - \text{acc}_{i+1}$, where acc_i and acc_{i+1}



	10%	20%	30%	40%
PCA	0.839	0.7047	0.5782	0.4632
IRPCA	0.9382	0.9086	0.8871	0.8713
RPCA_I	0.9204	0.8703	0.8121	0.7429
Noised_data	0.901	0.8482	0.7819	0.6273

(a)



	10%	20%	30%
PCA	0.1343	0.1265	0.115
IRPCA	0.0296	0.0215	0.0158
RPCA_I	0.0501	0.0582	0.0692
Noised_data	0.0528	0.0663	0.1546

(b)

Fig. 7. Detailed recognition performance on pose, illumination and expression (PIE) dataset. (a) Recognition performance against different corruption rates. (b) Absolute degradation gradient of the recognition performance against different corruption rates.

are the accuracies obtained at ith and $(i+1)$th corruption rates, respectively. These results illustrate that our IRPCA is better than PCA and RPCA_I on handling the corrupted data.

V. Conclusion

The recently established RPCA [9] provides an effective and efficient way to deal with the data corrupted by gross errors. Unfortunately, RPCA suffers the problem of how to handle the new data. This limitation can greatly reduce the practice of RPCA, because the real applications usually require online computation. In this paper, we therefore proposed IRPCA, which aims at well handling the gross corruptions and owning good generalization ability. Given a set of training data, unlike RPCA that target on recovering the original data matrix, IRPCA aims at learning the underlying projection matrix. The learnt projection could map any datum onto its underlying subspace so as to efficiently remove the possible corruptions, i.e., IRPCA can efficiently process the new samples, which were not involved in the training process. The learning was done by solving a nuclear-norm regularized minimization problem, which is convex and can be solved in polynomial time. Extensive experiments well verify the effectiveness and efficiency of our IRPCA method.

References

[1] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[2] F. De la Torre and M. J. Black, "Robust principal component analysis for computer vision," in *Proc. 8th Int. Conf. Comput. Vis.*, vol. 1. Barcelona, Spain, 2001, pp. 362–369.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces versus fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 2002.

[4] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[5] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*, vol. 2. Jun. 2009, pp. 2790–2797.

[6] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 1–8.

[7] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, Sep. 2007.

[8] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosoph. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

[9] E. J. Candes, X. Li, Y. Ma, and J. Wright. (2009, Dec.). *Robust Principal Component Analysis?* [Online]. Available: http://www-stat.stanford.edu/~candes/papers/RobustPCA.pdf

[10] R. A. Maronna, "Robust M-estimators of multivariate location and scatter," *Ann. Stat.*, vol. 4, no. 1, pp. 51–67, 1976.

[11] S. J. Devlin, R. Gnanadesikan, and J. R. Kettenring, "Robust estimation of dispersion matrices and principal components," *J. Amer. Stat. Assoc.*, vol. 76, no. 374, pp. 354–362, 1981.

[12] G. Li and Z. Chen, "Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and Monte Carlo," *J. Amer. Stat. Assoc.*, vol. 80, no. 391, pp. 759–766, 1985.

[13] F. Torre and M. Black, "Robust principal component analysis for computer vision," in *Proc. Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, 2001, pp. 362–369.

[14] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1–9.

[15] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2010, pp. 1–30.

[16] K. Min, Z. Zhang, J. Wright, and Y. Ma, "Decomposing background topics from keywords by principal component pursuit," in *Proc. 19th Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 269–278.

[17] G. Zhu, S. Yan, and Y. Ma, "Image tag refinement toward low-rank, content-tag prior and error sparsity," in *Proc. Int. Conf. Multimedia*, 2010, pp. 461–470.

[18] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma, "TILT: Transform invariant low-rank textures," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 314–328.

[19] V. Barnett, "The ordering of multivariate data," *J. Royal Stat. Soc. Ser. A*, vol. 3, no. 138, pp. 318–344, 1976.

[20] D. M. Titterington, "Estimation of correlation coefficients by ellipsoidal trimming," *Appl. Stat.*, vol. 27, no. 3, pp. 227–234, 1978.

[21] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. Inst. Electr. Electron. Eng.* vol. 98, no. 6, pp. 925–936, Jun. 2010.

[22] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *J. Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[23] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. (2010). *Robust Recovery of Subspace Structures by Low-Rank Representation* [Online]. Available: http://arxiv.org/pdf/1010.2955.pdf

[24] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[25] R. R. Hamming, *Art of Doing Science and Engineering: Learning to Learn*. Boca Raton, FL: CRC Press, 1997, p. 364.

[26] D. L. Donoho, "For most large underdetermined systems of equations, the minimal ℓ^1-norm near-solution approximates the sparsest near-solution," *Commun. Pure Appl. Math.*, vol. 59, no. 7, pp. 907–934, 2006.

[27] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," Dept. Electr. Comput. Eng., UIUC, Urbana, Tech. Rep. UILU-ENG-09-2215, 2009.

[28] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *Soc. Ind. Appl. Math. J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[29] S. Wei and Z. Lin. (2010). *Analysis and Improvement of Low Rank Representation for Subspace Segmentation* [Online]. Available: http://research.microsoft.com/pubs/151340/MSR-TR-2010-177.pdf

[30] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[31] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.