Ukrainian Catholic University

Faculty of Applied Sciences

Data Science Master Programme

Decomposition into Low-Rank and Sparse Matrices in Computer Vision

Linear Algebra final project report

Authors:
Yaroslava Lochman
Yevhen Pozdniakov
(contributed equally, order chosen alphabetically)

January 24, 2019

Abstract

One of the most important problems in video surveillance is to distinguish between the background and the foreground objects. This task is complicated by the presence of anomalies: in busy scenes every frame may contain moving objects, the varying illumination, noise may also cause the difficulties. So the background model needs to be flexible enough to accommodate changes in the scene. Various studies of the problem came up with considering the task as a decomposition into low-rank and sparse matrices, starting from the tremendous work on the Robust Principal Component Analysis (RPCA) with a convex relaxation. The challenges arise when the data matrix increases in size, which is usual in video analysis. Therefore we focus on the one of the recent approaches which is a compromise between complexity and limitations – Robust Orthonormal Subspace Learning (ROSL). The efficiency of the algorithm is demonstrated. We performed different experiments using synthetic noisy data, real-world video and also face images with illumination changes to compare the algorithm with simple and fast Singular Value Decomposition (SVD), and the performance-guaranteed RPCA. The code is available on https://github.com/ylochman/apps-LA-RPCA.

Keywords: Low-rank matrix recovery · robust principal component analysis · robust orthonormal subspace learning · augmented lagrange multipliers · alternating direction method · background modeling · video surveillance.

1 Introduction

The recent rapid growth of data in the different areas of human activities creates a challenge as well as an opportunity to many areas including computer vision. In image/video analysis it is a typical situation when data lies in thousands or even billions dimensions but in fact contains an intrinsic structure e.g. a linear subspace with much less dimensions, and sparse components. Moreover, video is a natural candidate for low-rank modeling, due to the correlation between frames. The simple mathematical formulation is the following hypothesis: given large data matrix X can be decomposed as $A+E$, where A is a low-rank matrix and E is a sparse matrix.

The question is how to obtain such matrices A and E. The classical PCA or SVD helps to find the low-rank approximation A, but is inappropriate for noisy measurements (See Section 7) and the matrix E is inadequate.

A number of approaches to robustifying PCA have been proposed over last decades. The methods use convex and non-convex optimization approches. The very well known and representative approach is described in RPCA [4], the authors proposed a convex relaxation on the settings and used Principal Component Pursuit, that showed excellent results, but turned out to be very computationally expensive – $O(\min(m^2n, mn^2))$ for X being $m \times n$ matrix. Non-convex approach ROSL [11] is investigated as a strong competitor of the previous. Its biggest advantage is computational efficiency – $O(rmn)$, where $r = \operatorname{rank}(A)$. If $r \ll \min\{m, n\}$, ROSL has squared complexity that is much better than cubic complexity in RPCA.

In this report we compare the quality and efficiency of SVD, convex RPCA, and non-convex ROSL algorithms in order to exploit a low-rank structure A from a large data matrix X obtained with corruptions/noise/etc, and show how these algorithms can be used in video and image processing. We describe our experiments on the synthetic and real datasets, that also explains our interest in such methods.

2 Problem setting

2.1 Notations

We denote different norms used in study below. For the $m \times n$ matrix $A = \begin{pmatrix} \mathbf{a}_1 & \dots & \mathbf{a}_m \end{pmatrix}^\top$, $r = \mathrm{rank}(A)$, $\sigma_i(A)$ – singular values of A:

$\|A\|_0$ – a total number of non-zero elements,	$\|A\|_1 = \sum_{i,j} \|a_{ij}\|$ – entry-wise l_1 norm,
$\|A\|_F = \sqrt{\sum_{i=1}^r \sigma_i^2(A)}$ – Frobenius norm,	$\|A\|_* = \sum_{i=1}^r \sigma_i(A)$ – nuclear norm,
$\|A\|_{\text{row-0}}$ – row-0 norm a total number of non-zero rows,	$\|A\|_{\text{row-1}} = \sum_{i=1}^m \|\mathbf{a}_i\|_2$ – row-1 norm

2.2 Problem formulation

Suppose we are given m video frames that have a spatial size $h \times w$. Background / foreground separation is simply finding for every frame 2 corresponding $h \times w$ matrices – a background and a foreground of it, that should add up to the raw frame.

Let the matrix X be constructed by stacking the video frames, preliminary converted to vectors $\in \mathbb{R}^n$, where $n = h \cdot w$. Hence the problem can be formulated in a way of finding the decomposition of a large matrix X as follows: $X = A + E$. As was already mentioned, A has low-rank and corresponds to the backgrounds, E is sparse and represents foreground moving objects, illumination changes, noise etc. A comprehensive analysis of the setting is provided in RPCA [4] resulting in the following optimization problem formulation:

$$\min_{A,E} \ \mathrm{rank}\,(A) + \lambda \|E\|_0, \quad \text{s.t. } X = A + E \tag{1}$$

Thus using such a decomposition one can model the background sequence, that can gradually change over time, by the subspace A, and the moving foreground objects by the sparse outliers E. But this problem is NP-hard, typical solution might involve a search with a combinatorial complexity! So we have to investigate the methods that efficiently recover the low-rank and the sparse matrices.

3 Overview of approaches

We do not know the low-dimensional column and row space of A, not even their dimension. And we do not know the locations of the nonzero entries in E, not even how many there are. This is already a challenge. In any case, to recover a low-rank matrix from data one can use simple PCA based on the single SVD factorization which is very fast. The best in the Frobenius norm rank-k approximation of X can be seen as A, but there is no gurantee on the sparsity of $E = X - A$. The model is not robust to outliers since it doesn't consider its presence. There exists a lot of studies [3] on the efficient solution of (1) which is essentially a way of robustifying PCA. Early natural approaches such as influence function techniques [13], multivariate trimming [6], alternating minimization [8], random sampling techniques [5] had been explored before 2000s, but these techniques were mostly heuristic, they did not have guarantees to work. One of the initial works on RPCA [4] described the problem extensively and showed that it can be provably solved via convex relaxation methods, under some natural conditions on the low rank and sparse components, called Principal Component Pursuit (PCP):

$$\min_{A,E} \ \|A\|_* + \lambda \|E\|_1 \qquad \text{s.t.} \quad X = A + E \tag{2}$$

Althought the method is performance-guaranteed, in practice it's not an efficient algorithm due to the complexity (See Section 6). Besides PCP, the decomposition can be achieved by the Stable PCP [15], Quantized PCP [1], Block based PCP [12] and Local PCP [14]. Among the optimization methods, there may be used the Augmented Lagrange Multiplier Method (ALM) [9], Alternating Direction Method (ADM) [17], Fast Alternating Minimization (FAM) [10]; Iteratively Reweighted Least Squares (IRLS) [7] that requires a good initialization which is non-trivial. Taking into consideration that video analysis requires working with large-scale data, we focus on the recent method that solves equivalent* to (2) non-convex optimization problem with a better convergence rate, called Robust Orthonormal Subspace Learning [11]:

$$\min_{E,D,C} \|C\|_{\text{row-1}} + \lambda \|E\|_1 \qquad \text{s.t.} \quad X = DC + E,\ D^\top D = I_k \tag{3}$$

The authors of ROSL also presented a random sampling algorithm that speeds it up, called ROSL+. Although we consider it to be a potentially useful method in solving the problem, there exists several problems such as it adds more stochasticity (random permutations), more parameters to tune (β_2 and β_3), that's why we focused on the more stable ROSL version.

4 Algorithm analysis

ROSL represents the low-rank $m \times n$ matrix A under an orthonormal subspace as $A = DC$, meaning it is spanned by $m \times k$ matrix $D = \begin{pmatrix}\mathbf{d}_1 & \dots & \mathbf{d}_k\end{pmatrix}$ (where $\{\mathbf{d}_1, \dots, \mathbf{d}_k\}$ is the orthonormal set) with $k \times n$ coefficient matrix $C = \begin{pmatrix}\mathbf{c}_1 & \dots & \mathbf{c}_k\end{pmatrix}^\top$ so that $\mathbf{c}_{ij}$ reflects the contribution of $\mathbf{d}_i$ to the j-th column of A. The dimension k of the subspace is set as $k = \beta_1 r$ ($\beta_1 > 1$ is fixed).

It is clear that $\text{rank}(A) \le \text{rank}(C) \le \|C\|_{\text{row-0}}$, we will use it further. D constrained to be constructed from an orthonormal basis in order:

1. to avoid the vanishing of coefficients C (the subspace bases should be on the unit sphere: $\mathbf{d}_i^\top \mathbf{d}_i = 1\ \forall i$)

2. to eliminate the correlation of columns of D so that the group sparsity $\|C\|_{\text{row-0}}$ could be a valid measure of $\text{rank}(A)$

Thus, ROSL recovers the low-rank matrix A from X by minimizing the number of non-zero rows of C and the regularized sparsity of E:

$$\min_{E,D,C} \|C\|_{\text{row-0}} + \lambda \|E\|_0 \qquad \text{s.t.} \quad X = DC + E,\ D^\top D = I_k$$

Similarly as a well-known sparsity-inducing l_1-norm is an acceptable substitute for the sparsity measure (a total number of non-zero elements), the row-1 norm is a good heuristic for the row-0 norm. That is what ROSL ends up with – it recovers the low-rank matrix A from X by minimizing the row-1 norm of C, and the entry-wise l_1-norm of E, as was formulated in (3). To show that it is a valid non-convex relaxation of the performance-guaranteed RPCA [4] (See Equation (2)), authors investigated the relationship between the group-sparsity-based norm and the rank / nuclear norm of the matrix.

Lemma 1. *Let $A = DC$, $D^\top D = I_n$, C consists of orthogonal rows. Then:*

$$\|A\|_* = \|C\|_{\text{row-1}}$$

*in terms of the same global minimum; see Section 4

Proof. Let $A = USV^\top$ got by SVD. Then D is exactly U and C is $SV^\top$.

$$\|C\|_{\text{row-1}} = \sum_{i=1}^{m} \sqrt{\mathbf{c}_i^\top \mathbf{c}_i} = \sum_{i=1}^{m} \sqrt{\sigma_i^2 \mathbf{v}_i^\top \mathbf{v}_i} = \left| \forall i > r \ \sigma_i = 0 \right| = \sum_{i=1}^{r} \sigma_i = \|A\|_*$$

□

Now without loss of generality suppose $D^\top D = I_n$ for the orthonormal subspace decomposition $A = DC$ ($D \in \mathbb{R}^{m\times n},\ C \in \mathbb{R}^{n\times n}$), and $m \geq n$. With the use of Lemma 1 it has been proven [11] that:

$$\min_{\mathcal{F}_n(C)} \|C\|_{\text{row-0}} = \text{rank}(A), \qquad \min_{\mathcal{F}_n(C)} \|C\|_{\text{row-1}} = \|A\|_*$$

where $\mathcal{F}_k(A) = \{C \mid A = DC,\ D^\top D = I_k\}$.

Using this statement we conclude that $\|C\|_{\text{row-1}} \geq \|A\|_* \quad \forall C \in \mathcal{F}_n(A)$. Now, suppose, with regularization parameter λ RPCA recovers the low-rank matrix $\hat{A}$ just as the ground-truth A. Then

$$\left\|\hat{C}\right\|_{\text{row-1}} + \lambda \left\|X - \hat{A}\right\|_1 \geq \left\|\hat{A}\right\|_* + \lambda \left\|X - \hat{A}\right\|_1 = \|A\|_* + \lambda \|X - A\|_1 \ \ \forall \hat{C} \in \mathcal{F}_k(\hat{A})$$

Hence with the same λ, the optimization problem proposed by ROSL has the same global minimum as by RPCA: $\hat{A} = A,\ \hat{D} = U,\ \hat{C} = SV^\top$ (where $A = USV^\top$ got by SVD).

Next, the equality constraints in (3) are removed by using the augmented Lagrangian multiplier (ALM) [2], resulting in the augmented Lagrangian function:

$$\mathcal{L}(D, C, E, \Lambda, \mu) = \|C\|_{\text{row-1}} + \lambda \|E\|_1 - \langle \Lambda, X - DC - E\rangle - \frac{\mu}{2} \|X - DC - E\|_F^2 \qquad s.t.\ D^\top D = I_k \tag{4}$$

where μ is the over-regularization parameter, Λ is the $m \times n$ ALM matrix.

5 Implementation of ROSL algorithm

The minimization of formulated augmented Lagrangian (4) is solved using inexact Alternating Direction Method (ADM) at the higher scale and inexact Block Coordinate Descent (BCD) at the lower scale. This solver is called inexact ADM/BCD.

5.1 Inexact Alternating Direction Method

The ADM algorithm iterates through the following steps:

Step 1 Solving D and C simultaneously with constraint $DC + E = X + \frac{\Lambda}{\mu}$:

$$(D^{i+1},\ C^{i+1}) \leftarrow \text{argmin}_{D,C} \ \mathcal{L}(D, C, E^i, \Lambda^i, \mu^i)$$

It is a non-convex problem. But updating one matrix while fixing the other is convex. This indicates using BCD (See Section 5.2).

Step 2

$$E^{i+1} \leftarrow \text{argmin}\, \mathcal{L}(D^{i+1}, C^{i+1}, E, \Lambda^i, \mu^i)$$

We can easily update $E^{i+1} = \textbf{Shrink}(X - D^{i+1}C^{i+1} + \frac{\Lambda^i}{\mu^i}, \frac{\lambda}{\mu^i})$, where the shrinkage function $\textbf{Shrink}(X, a) = \max\{|X| - a, 0\} \cdot \text{sgn}(X)$ ("$\cdot$" denotes element-wise multiplication).

Step 3

$$\Lambda^{i+1} \leftarrow \Lambda^i + \mu^i (X - D^{i+1}C^{i+1} - E^{i+1})$$
$$\mu^{i+1} = \rho\mu^i$$

where $\rho > 1$ is a constant.

5.2 Inexact Block Coordinate Descent

The BCD scheme sequentially updates each pair $(\mathbf{d}_t,\ \mathbf{c}_t)$, $1 \le t \le k$ such that $\mathbf{d}_t\mathbf{c}_t^\top$ is a good rank-1 approximation to $R_t^i = X + \frac{\Lambda^i}{\mu^i} - E^i - \sum_{j<t} D_j^{i+1}C_j^{i+1} - \sum_{j>t} D_j^i C_j^i$. Thus, if removing the orthonormal constraint on D, the pair $(D_t,\ C_t)$ can be efficiently updated as follows:

$$D_t^{i+1} = R_t^i C^{i^T}$$

$$C_t^{i+1} = \frac{1}{\|D_t^{i+1}\|_2^2}\mathbf{MagnShrink}(D_t^{i+1^T} R_t^i,\ \frac{1}{\mu^i})$$

where the magnitude shrinkage function $\mathbf{MagnShrink}(X, a) = \max\{\|X\|_2 - a, 0\}/\|X\|_2 X$. The Gram-Schmidt process is used to orthonormalize D_t^{i+1}.

To summarize, we describe numerical process* in the following way. Function `InexactALM_ROSL(X)` takes initial matrix X as a single input parameter. The iteration process inside this function continues until one of two conditions reaches:

1. the number of iterations exceeds maximum possible (by default 10^2 iterations);
2. $\|X - A - E\|_F^2 < \varepsilon$ – the expected accuracy (by default 10^{-6}).

The steps appointed above are performed consequentially in the loop. **Step 1** is implemented inside the function `LowRankDictionaryShrinkage(X)`, which is described above. **Step 2** is implemented in the following lines:

```
Etmp = *X + Z - A;
E = arma::abs(Etmp) - lambda / mu;
E.transform([ ](double val){ return (val > 0.) ? val : 0.;});
E = E  % arma::sign(Etmp)
```

Step 3 is implemented in the following lines:

```
Z = (Z + *X - A - E) / rho;
mu = (mu * rho < mubar) ? mu * rho : mubar
```

Despite the non-convexity, it is shown to exhibit strong convergence behavior, though given random initialization of matrix C. Extensive experiments in [11, 3] and ours as well validated that this solver gives the identical or very close to the RPCA global optimum solution. The solver's pseudo code is provided in the Algorithm 1.

*Further we explain main functions that are consistent with the provided code on github.

Algorithm 1 ROSL Solver

function SHRINK(a, X)
  return $\max\{\text{abs}(X) - a, 0\} \cdot \text{sgn}(X)$

function MAGNSHRINK(a, X)
  if then $\|X\|_2 > 0$
    return $\max\{\|X\|_2 - a, 0\} / \|X\|_2 X$
  else
    return X

function ROSL(X, k, $\lambda > 0$, $\mu > 0$, $\rho > 1$)
  $D^0 = \mathbf{0}_{m\times k}$ $(\ = \begin{bmatrix}\mathbf{d}_1 & \cdots & \mathbf{d}_k\end{bmatrix})$
  $E^0 = \mathbf{0}_{m\times n}, \ \Lambda^0 = \mathbf{0}_{m\times n}$
  $C^0 = rand_{k\times n}$ $(\ = \begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_k\end{bmatrix}^\top)$
  while E not converged **do**
    for $t \in \overline{1,k}$ **do**
      $R \leftarrow X - E^i + \frac{1}{\mu^i}\Lambda^i - \sum_{j<t} \mathbf{d}_j^{i+1}\mathbf{c}_j^{i+1\top} - \sum_{j>t} \mathbf{d}_j^i \mathbf{c}_j^{i\top}$
      $R \leftarrow R - \sum_{j=1}^{t-1} \mathbf{d}_j^{i+1}\mathbf{d}_j^{i+1\top} R$
      $\mathbf{d}_t^{i+1} \leftarrow R\mathbf{c}_t^i$
      $\mathbf{d}_t^{i+1} \leftarrow \frac{\mathbf{d}_t^{i+1}}{\|\mathbf{d}_t^{i+1}\|_2}$
      $\mathbf{c}_t^{i+1} \leftarrow$ **MagnShrink**$(\mathbf{d}_t^{i+1\top} R, \ \frac{1}{\mu^i})$
    for $t \in \overline{1,k}$ **do**
      if $\mathbf{c}_t^{i+1\top}\mathbf{c}_t^{i+1} = 0$ **then**
        del $\mathbf{d}_t^{i+1}, \ \mathbf{c}_t^{i+1}$
        $k \leftarrow k - 1$
    $E^{i+1} =$ **Shrink**$(X - D^{i+1}C^{i+1} + \frac{1}{\mu^i}\Lambda^i, \ \frac{\lambda}{\mu^i})$
    $\Lambda^{i+1} = \Lambda^i + \mu^i(X - D^{i+1}C^{i+1} - E^{i+1})$
    $\mu^{i+1} = \rho\mu^i$
  return $D^{i+1}, C^{i+1}, E^{i+1}$

6 Complexity, parameters, limitations

The dominant computational processes of the algorithm are:

1. left multiplying $m \times n$ residual matrix R by D;
2. right multiplying the latter by C.

Thus, the complexity of ROSL depends on the subspace dimension k. If we set the initial value of k on the same order as r being much smaller than $\min\{m, n\}$, the complexity is quadratic of matrix size: $O(mnk) = O(mnr)$, which is much better than $O(mn\min\{m, n\})$ complexity of RPCA due to multiple iterations of SVD.

The most important parameter for the algorithm is k which is in fact an initial estimate of data dimensionality r and supposed to be quite small. This is a specific parameter that depends on the problem domain, e.g. in our experiments on real data we tuned the parameter up to a value of 5. The regularization parameter on l_1-norm (the sparse error term) λ should also be provided, but empirically an optimal value of 0.01 was derived. Besides, the values of $\rho = 1.5$ and $\mu = 10\lambda / \max\{|X_{ij}|\}$ was retrieved as optimal without a need of change.

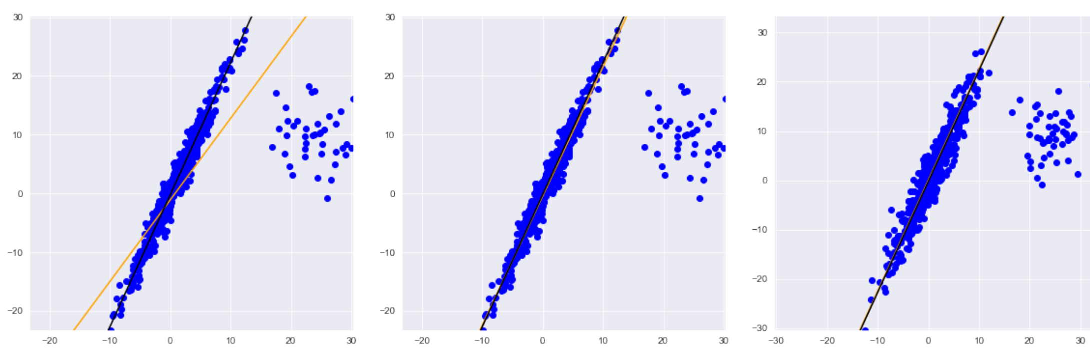
As all the non-convex problems, ROSL has no theoretical guarantee of convergence. However, empirical evidence suggests that ROSL solver has strong convergence behavior and provides a valid solution: $A^{i+1} = D^{i+1}C^{i+1}$ and E^{i+1} for the zero-initialization of matrices E^0, Λ^0, D^0 and random initialization of matrix C^0.

7 Numerical experiments

SVD, RPCA and ROSL are being studied in this section to show why we chose the latter as a main approach. A comparison of considered algorithms on the performance is quantitative, on the precision – both quantitative (synthetic data) and qualitative (real data). For more visual results please refer our repository, different demo files correspond to different data experiments.

7.1 Fitting noisy synthetic data by affine subspace

We generated a set of synthetic observations in 2D-space without loss of generality and constructed data matrix X (m observations, $n = 2$). The true low-rank (in fact, of rank 1) matrix A is a part of the matrix, generated by the dominant 2D-distribution, and the outliers are generated from another "noise distribution". Since we are dealing with an affine subspace A of dimension 1, we represent it as a line with its slope and intercept. In order to compare the algorithms' precision we use the squared error of the slope and intercept that we got from the low-rank matrix A and the ground truth.



(a) SVD (b) RPCA (c) ROSL

Figure 1: An example of synthetic data (blue points), that are modeled with algorithms to be compared – orange line is the low-rank affine subspace.

As shown on the Figure 2 we explored the dependence of the error on the size of the noise added to data. One can see that ROSL sometimes behaves unpredictably bad, the experiments showed that it is a rare stochastic situation. In general, both RPCA and ROSL work very good and are indeed robust.

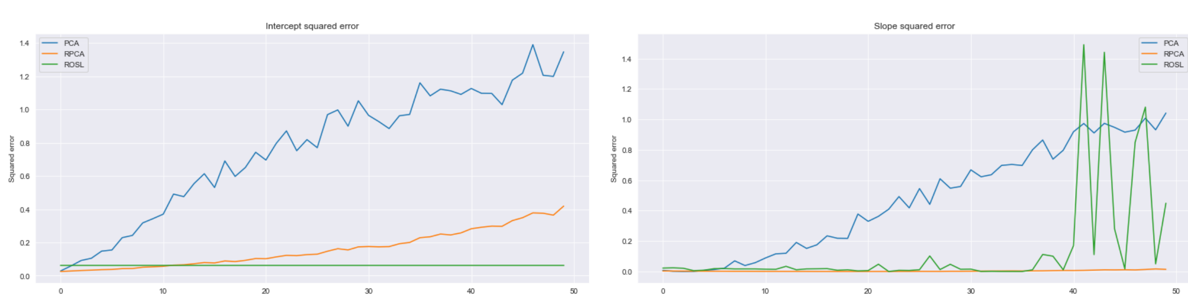


Figure 2: The squared error of the slope and the intercept as a function of the outliers number.

7.2 Background modeling from video surveillance data

We investigate whether the proposed methods can separate the group-sparse moving objects from the low-rank background and compare the time of execution. Here we present one of the several studied datasets – video from highway. This video has a relatively static background, but significant foreground variations. We dealed with a sequence of 340 frames taken from highway, converted to grayscale colormap. The frames have resolution 320×240. We stacked the frames as the columns of data matrix X, experimenting with different spatial sizes (h, w), the number of frames m and the frequency of frames m_{freq} (with maximum possible 76800×340 matrix X), and then decomposed into a low-rank term and a sparse term.

Figure 3 shows the processing of one frame (raw is the first column) with fixed spatial size and number of observations (which is in fact m and n), but different m_{freq} (every frequency correspond to a row). The 2-4 columns present models for the background (BG), 5-7 – for the foreground (FG). Figure 4 shows one processed frame with fixed spatial size and frequency, but different m. We find decreasing spatial size useless here due to the relatively low resolution. All the algorithms try to recover A as the background, and E as the moving cars, although the artifacts appear on all the images.

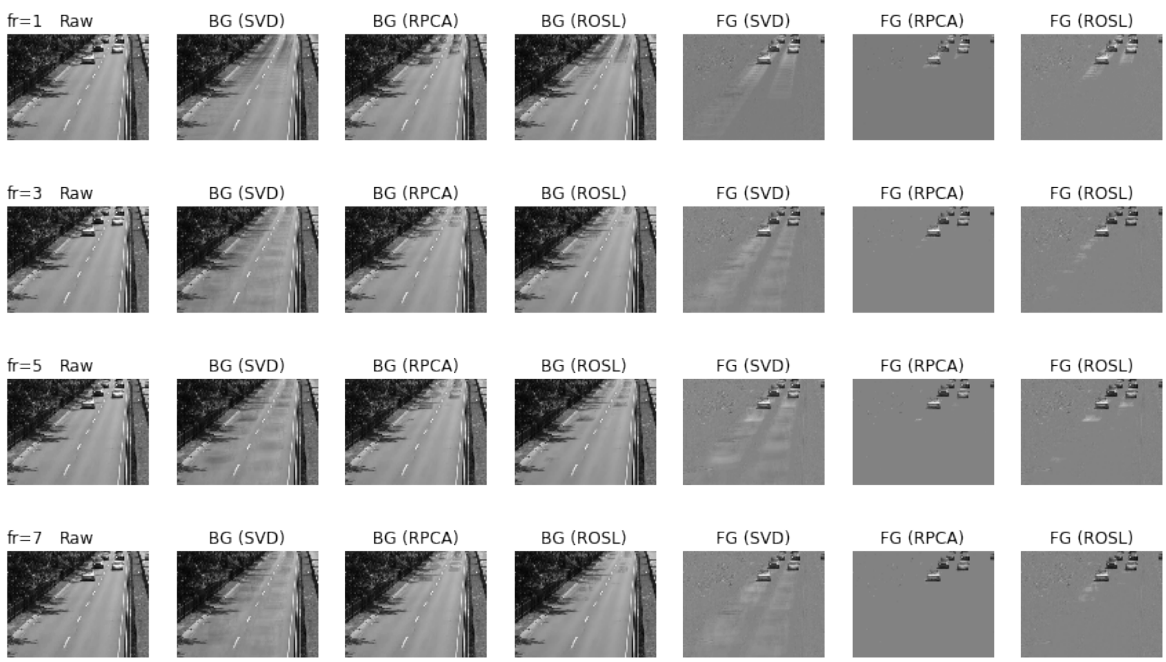


Figure 3: Background / foreground separation for diffeent frequency m_{freq} (e.g. $m_{freq} = 2$ means every 2^{nd} frame), shortly **fr** [$m = 20, \;\; h \times w = 240 \times 320$].

When comparing the results for the background, ROSL seems to have the best visual results. In terms of the foreground, RPCA and ROSL both give almost identical results, though when modeling on small amounts of frames (less than 50) ROSL is noisier, meaning it includes a few of such part that supposed to be in background. We could conclude that it would be better to use RPCA in the foreground object detection if the frame rate is low, but since this algorithm is quite slow, it is not the best solution. On the other hand, for the ≥ 50 fps frequency, which is certainly common, ROSL would be the best choice especially considering real-time processing.

The execution time of the aforementioned experiments is presented in Table 1 and Table 2.

As expected, the simple SVD shows the best performance time but relatively bad visual results. RPCA, opposite, is the worst in terms of time, but with a good quality. And finally,

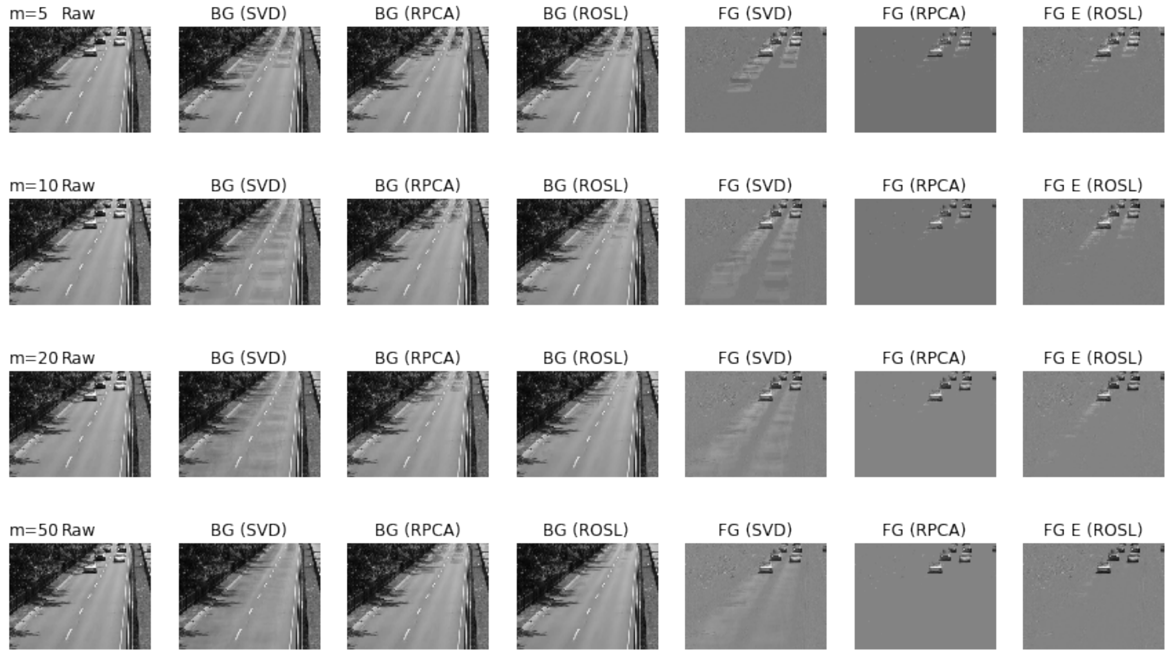


Figure 4: Background / foreground separation for different number of frames m [$m_{freq} = 3$, $h \times w = 240 \times 320$].

m	SVD	RPCA	ROSL
5	0.0126	1.2533	0.3033
10	0.0410	3.6221	0.4944
20	0.1042	9.7495	0.8612
50	0.4338	34.7033	2.3577

Table 1: The execution time in sec for different m [$h \times w = 240 \times 320$, $m_{freq} = 3$]

$h \times w$	SVD	RPCA	ROSL
30×40	0.0007	1.2100	0.0092
60×80	0.0018	4.5908	0.0357
120×160	0.0092	3.7948	0.1025
240×320	0.0343	3.2982	0.4579

Table 2: The execution time in sec for different spatial sizes [$m = 10$, $m_{freq} = 3$]

ROSL has very good modeling results and the processing time as well. So, we conclude that non-convex ROSL is the best algorithm in trade-off problem between quality and efficiency.

7.3 Additional experimental results: illumination changes on video

Face recognition is another problem domain in computer vision where low-dimensional linear models have received a great deal of attention. However, since faces are not perfectly convex, real face images often violate low-rank model, due to cast shadows and specularities. These errors are large in magnitude, but sparse in the spatial domain. It is reasonable to believe that if we have enough images of the same face, RPCA or ROSL will be able to remove these errors.

We experimented with face images taken from the Yale Face database [16]. There is 5 images for the every person under different illumination conditions, each image has resolution 320×243 or 231×195 (if it is centered), we similarly stacked them as columns of 77760×5 or 45045×5 matrix X (respectively). We applied SVD, RPCA and ROSL, and examples of the result are shown on the Figure 5. Mean execution time of every algorithm under this experiments is 0.0078 sec for SVD, 0.4545 sec for RPCA and 0.2798 sec for ROSL, again indicates the highest speed of SVD, the good – of ROSL, and the lowest – of RPCA.

From Figure 5 we can conclude that SVD isn't suitable for removing shadows, specularities

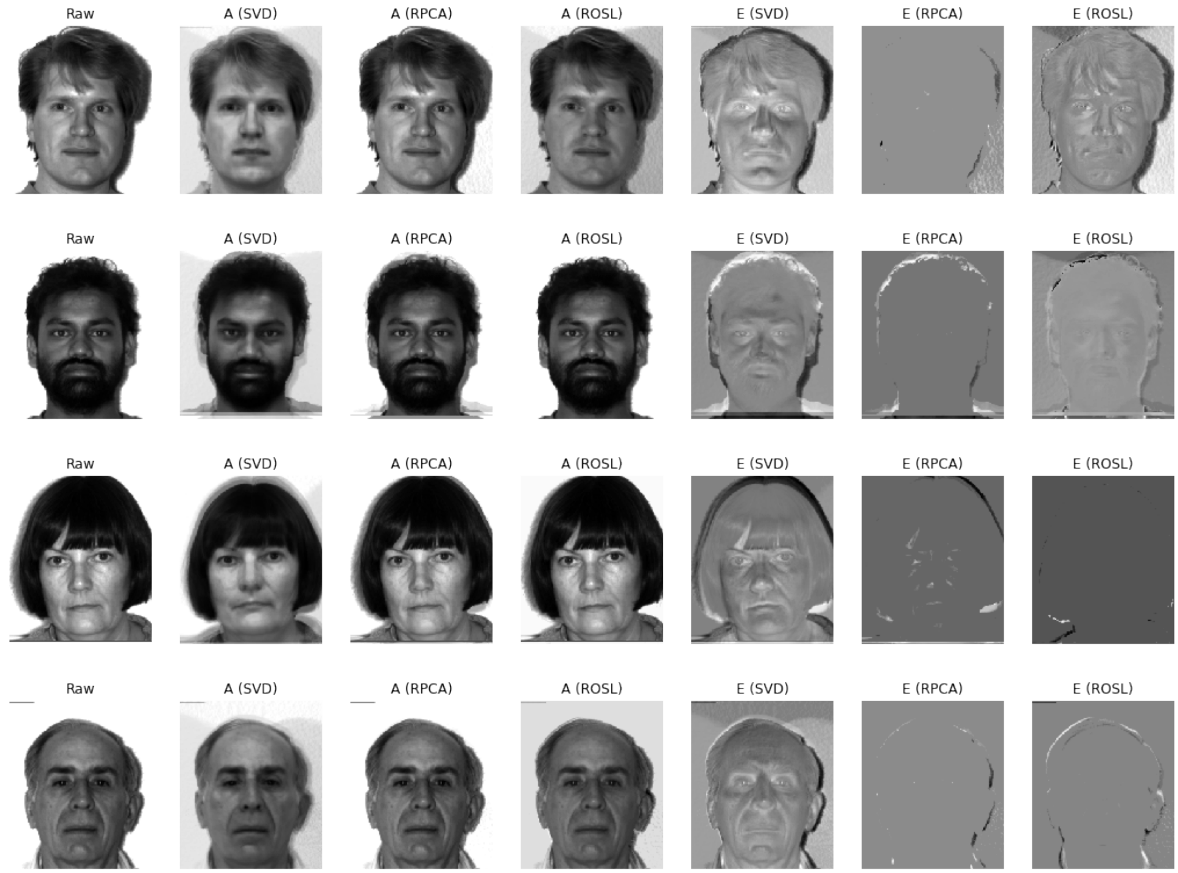


Figure 5: Removing shadows, specularities and saturations from face images using SVD, RPCA, ROSL ($h \times w = 231 \times 195$). A corresponds to the processed image from the low-rank matrix, E is supposed to be w

and saturations from face images. RPCA and ROSL demonstrate similar visual results, but RPCA deals with specularities better. ROSL needs additional investigation. As a result such transition from raw images to those without shadows and specularities may be useful for preprocessing the training data for face recognition, face alignment and tracking under illumination variations.

8 Conclusions

In this report we compared different approaches to decompose large data matrix X as low-rank matrix and sparse matrix. After numerical experiment with synthetic data we came to the conclusion that robust PCA deals with outliers significantly better in comparison with standard PCA. Due to the fact that real data often has corrupted observations, we can conclude that robust PCA is much more applicable in practice.

We run experiments with real video and face images data. Convex RPCA demonstrates good visual results but needs significant calculation time due to the cubic computational complexity of the data matrix size. So in practice RPCA can be used in case of small size of data or when calculation time doesn't matter.

Non-convex ROSL accelerates RPCA and introduces a rank measure that imposes the group sparsity of its coefficients under orthonormal subspace. ROSL demonstrates nice visual effect and high efficiency. It was proved that ROSL has squared computational complexity. But as every non-convex problem, ROSL has no theoretical guarantee of convergence, thus its results could be unpredictable as it's shown on Figure 2. Anyway we can conclude that ROSL is the best algorithm in trade-off problem between quality and efficiency.

So called ROSL+ is a subject of the further investigation. ROSL+ is a faster version of ROSL because it possesses a linear complexity with respect to the matrix size: $O(r^2(m+n))$. We are going to experiment with its parameters and compare the efficiency of ROSL+ with other algorithms.

References

[1] Becker S., Candes E., Grant M.: TFOCS: Flexible First-order Methods for Rank Minimization. Low-rank Matrix Optimization Symposium, SIAM Conference on Optimization (2011)

[2] Bertsekas, D.: Constrained Optimization and Lagrange Multiplier Method. Academic Press (1982)

[3] Bouwmans T., Sobral A., Javed S., Jung S.K., Zahzah E.-H.: Decomposition into Low-rank plus Additive Matrices for Background/Foreground Separation: A Review for a Comparative Evaluation with a Large-Scale Dataset (2016)

[4] Candes, E. J., Li, X., Ma, Y., Wright J.: Robust Principal Component Analysis? Journal of the ACM (2011)

[5] Fischler M., Bolles R.: Random sample consensus: A paradigm for model tting with applications to image analysis and automated cartography. Communications of the ACM, 24:381–385 (1981)

[6] Gnanadesikan R., Kettenring J.: Robust estimates, residuals, and outlier detection with multiresponse data. Biometrics, 28:81–124 (1972)

[7] Guyon C., Bouwmans T., Zahzah E.: Moving Object Detection via Robust Low Rank Matrix Decomposition with IRLS scheme. International Symposium on Visual Computing, ISVC (2012)

[8] Ke Q., Kanade T.: Robust l^1-norm factorization in the presence of outliers and missing data. In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (2005)

[9] Lin Z., ChenM., Wu L., Ma Y.: The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. Journal of Structural Biology. 181(2): 116–27 (2013)

[10] Rodrguez P., Wohlberg B. : Fast Principal Component Pursuit Via Alternating Minimization. IEEE International Conference on Image Processing, ICIP (2013)

[11] Shu X., Porikli F., Ahuja N.: Robust Orthonormal Subspace Learning: Efcient Recovery of Corrupted Low-rank Matrices (2014)

[12] Tang G., Nehorai A.: Robust principal component analysis based on low-rank and block-sparse matrix decomposition. Annual Conference on Information Sciences and Systems, CISS (2011)

[13] Torre F. D. L., Black M.: A framework for robust subspace learning. International Journal on Computer Vision, 54:117–142 (2003)

[14] Wohlberg B., Chartrand R., Theiler J.: Local Principal Component Pursuit for Nonlinear Datasets. International Conference on Acoustics, Speech, and Signal Processing, ICASSP (2012)

[15] Wright J., Peng Y., Ma Y., Ganesh A., Rao S.: Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices by Convex Optimization. Neural Information Processing Systems, NIPS. (2009)

[16] Yale Face Database – Retrieved from: http://vismod.media.mit.edu/vismod/classes/mas622-00/datasets/

[17] Yuan X., Yang J.: Sparse and Low-Rank Matrix Decomposition via Alternating Direction Methods. Optimization Online (2009)