Book title goes here

by

Author Name

Contents

Part I: This is a Part

1 Robust Orthonormal Subspace Learning (ROSL) for Efficient Low-rank Recovery *Xianbiao Shu, Fatih Porikli, and Narendra Ahuja* **1**-1

I

This is a Part

1 Robust Orthonormal Subspace Learning (ROSL) for Efficient Low-rank Recovery *Xianbiao Shu, Fatih Porikli, and Narendra Ahuja* **1**-1
Introduction • Robust Orthonormal Subspace Learning • Fast Algorithm for ROSL • Acceleration by Random Sampling • Experimental Results • Conclusion • Acknowledgement

1

Robust Orthonormal Subspace Learning (ROSL) for Efficient Low-rank Recovery

Xianbiao Shu
University of Illinois at Urbana-Champaign

Fatih Porikli
Australian National University / NICTA

Narendra Ahuja
University of Illinois at Urbana-Champaign

1.1 Introduction **1**-1
1.2 Robust Orthonormal Subspace Learning **1**-3
Group Sparsity under Orthonormal Subspace • Bound of Group Sparsity under Orthonormal Subspace • A General Framework of Robust Low-Rank Recovery Approaches
1.3 Fast Algorithm for ROSL **1**-6
Alternating Direction Method • Block Coordinate Descent • Computational Complexity
1.4 Acceleration by Random Sampling **1**-8
Random Sampling in ROSL+
1.5 Experimental Results **1**-9
1.6 Conclusion **1**-12
1.7 Acknowledgement **1**-12
References **1**-12

1.1 Introduction

Recovering intrinsic low-rank structure of data from corrupted observations is widely used in many machine learning, data mining, and computer vision tasks. In general, low-rank recovery methods can be grouped into two categories; convex nuclear-norm based formulations and non-convex matrix factorization approaches. Convex nuclear-norm based methods are guaranteed to attain global minimum with cubic computational complexity. While non-convex matrix factorization methods suffer from possible convergence to local minima, they are computationally more efficient with quadratic complexity. Motivated by seizing the favorable aspects of these methods, this chapter presents a computationally efficient low-rank recovery method called Robust Orthonormal Subspace Learning (ROSL) that utilizes a novel rank measure on the data matrix to impose group sparsity structure on its coefficients under an orthonormal subspace. This rank measure is proven to be lower bounded by the the same global minimum as the nuclear norm and is demonstrated experimentally to converge to its global minimum with high probability. This chapter also describes a fast version (ROSL+) empowered by random sampling, which further decreases the computational complexity from quadratic to linear.

Convex nuclear norm based methods, such as Robust PCA (RPCA, also called PCP [CLMW11]) and Sparse Low-Rank Matrix Decomposition (SLRMD) [YY09], em-

ploy the nuclear norm as a surrogate for the highly non-convex rank minimization problem [RFP07]. RPCA has been shown to be a convex problem with performance guarantee [CLMW11]. It assumes the observation matrix $X \in \mathbb{R}^{m\times n}$ is generated by the addition of a low-rank matrix A (rank: $r \ll \min\{m,n\}$) and a sparse matrix E. Suppose Singular Value Decomposition (SVD) of A is denoted as $A = USV^T$, where S is a diagonal matrix with singular values $S_i, 1 \leq i \leq \min\{m,n\}$) on the diagonal, RPCA recovers the low-rank matrix A from the corrupted observation X as follows:

$$\min_{A,E} \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad A + E = X \tag{1.1}$$

where nuclear norm $\|A\|_* = \sum_{i=1}^{n} S_i$.

Despite its superior results, RPCA is computationally expensive with $\mathcal{O}(\min(m^2n, mn^2))$ complexity due to multiple iterations of SVD for large-scale problems. Reducing the number of iterations is a possible remedy [TSSK10], yet the computational load is dominated by SVD itself. Instead of full SVD, partial RPCA [LCM10] computes κ ($r < \kappa$) major singular values, thus it has $\mathcal{O}(\kappa mn)$ complexity. Nevertheless, partial RPCA requires a proper way to preset the optimal value of κ. GoDec [ZT11] uses bilateral random projection to accelerate the low-rank approximation in RPCA. Similarly, RP-RPCA [MDYY11] applies random projection P on A (i.e., $A' = PA$) and then minimizes the rank of A'. However, rank minimization using randomized SVD is unstable and might be even slower than RPCA since it remands conducting SVD on many different projected matrices A' at each iteration.

As an alternative, non-convex matrix factorization approaches including RMF [KK05] and LMaFit [SWZ11] have been proposed for fast low-rank recovery. Instead of minimizing the rank of A, these approaches represent A under some predetermined rank subspaces (spanned by $D \in \mathbb{R}^{m\times k}$) as $A = D\alpha$, where coefficients $\alpha \in \mathbb{R}^{k\times n}$ and $r < k \ll \min(m,n)$. Due to its SVD-free property, these non-convex matrix factorization techniques are computationally preferable to RPCA. Still, their quadratic complexity $\mathcal{O}(kmn)$ is prohibitive for large-scale low-rank recovery. Besides, they need an accurate initial rank estimate that is not easy to obtain itself.

Inspired by the group sparsity structure in sparse coding [TVW05, YL06, MBP$^+$09, BPSS09, HZM09], ROSL solves the rank minimization problem of a matrix A by imposing a group sparsity constraint on its coefficients α under an orthonormal subspace spanned by orthonormal bases D. The intuition is that, given the subspace representation $A = D\alpha$, the rank of A is upper bounded by the number of non-zero rows of α. ROSL can be regarded as a non-convex relaxation of RPCA by replacing the nuclear norm with this rank heuristic. Firstly, this relaxation enables the employment of efficient sparse coding algorithms in low-rank recovery, therefore ROSL has only $\mathcal{O}(rmn)$ ($r < \kappa, k$) complexity, much faster than RPCA. In addition, by imposing this rank heuristic, ROSL is able to seek the most compact orthonormal subspace that represents the low-rank matrix A without requiring accurate rank estimate (unlike RMF and LMaFit). Furthermore, this rank heuristic is proven to be lower bounded by the nuclear norm, which means that ROSL has the same global minimum as RPCA.

An efficient ROSL solver is also presented in this chapter. This solver incorporates a block coordinate descent (BCD) algorithm into an inexact alternating decision method (ADM). Despite its non-convexity, this solver is shown to exhibit strong convergence behavior, given random initialization. Experimental results validate that the solution obtained by this solver is identical or very close to the global minimum of RPCA.

As another contribution, a random sampling algorithm is introduced to further speed up ROSL such that ROSL+ has linear complexity $\mathcal{O}(r^2(m+n))$. Similar sampling based frameworks for RPCA can be found in DFC [MTJ11] and L1 filtering [LLSG14]. Although these methods follow the same idea, i.e. Nyström method [WS00, KMT09, TR10], ROSL+

addresses a different problem of accelerating orthogonal subspace learning. In addition, ROSL+ elucidates a key point in Nyström method, how to estimate multiple submatrices, which is omitted by DFC.

This chapter is organized as follows. Section 1.2 presents the proposed method (ROSL). Section 1.3 develops its efficient solver. Section 1.4 provides its accelerated version (ROSL+). Section 1.5 presents experimental results. Section 1.7 adds the acknowledgement and Section 1.6 gives the concluding remarks.

1.2 Robust Orthonormal Subspace Learning

Similar to RPCA, ROSL assumes that the observation $X \in \mathbb{R}^{m\times n}$ is generated by the addition of a low-rank matrix A (rank: $r \ll \min\{m,n\}$) and a sparse outlier matrix E as shown in Figure 1.1. Different from RPCA that uses the principal subspace, ROSL represents the low-rank matrix A under an ordinary orthonormal subspace (spanned by $D = [D_1, D_2..., D_k] \in \mathbb{R}^{m\times k}$), denoted as $A = D\alpha$, where coefficients $\alpha = [\alpha_1; \alpha_2; ...; \alpha_k] \in \mathbb{R}^{k\times N}$ and α_i specifies the contribution of D_i to each column of A. The dimension k of the subspace is set as $k = \beta_1 r (\beta_1 > 1$ is a constant).

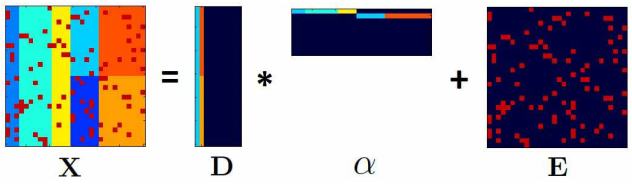


FIGURE 1.1 Illustration of the observation model $X = A + E = D\alpha + E$ in ROSL.

1.2.1 Group Sparsity under Orthonormal Subspace

ROSL rank minimization formulation replaces the nuclear norm used in RPCA. Although the Frobenius-norm regularization is a valid substitute for nuclear norm, as shown in Lemma 1.1, it fails to recover the low-rank matrix without rank estimate.

THEOREM 1.1 $\|A\|_* = \min_{D,\alpha} \frac{1}{2}(\|D\|_F^2 + \|\alpha\|_F^2) \quad s.t. \quad A = D\alpha$ *[FHB01, SRJ05].*

Motivated by the group sparsity [TVW05, YL06, MBP+09, BPSS09, HZM09], ROSL represents A under some vector subspace D and constraints the rank of A by imposing the group sparsity of its coefficients α. Its main idea is that, given $A = D\alpha$, the rank of A, or exactly α, is upper bounded by the number of non-zero rows of α, i.e. $\|\alpha\|_{\text{row-0}}$. In order to avoid the vanishing of coefficients α, the subspace bases are constrained to be on the unit sphere, i.e., $D_i^T D_i = 1, \forall i$. To further enable the group sparsity of α is a valid measure of rank (A), we should eliminate the correlation of columns of D by constraining it to be orthonormal, i.e., $D^T D = I_k$, where I_k is an identity matrix. Thus, ROSL recovers the low-rank matrix A from X by minimizing the number of non-zero rows of α, and the sparsity of E as follows:

$$\min_{E,D,\alpha} \|\alpha\|_{\text{row-0}} + \lambda\|E\|_0 \quad \text{s.t.} D\alpha + E = X, D^T D = I_k, \forall i \tag{1.2}$$

THEOREM 1.2 $\|A\|_* = \|\alpha\|_{\text{row-1}}$, *when* $A = D\alpha, D^T D = I_k$ *and* α *consists of orthogonal rows.*

It is well known that sparsity-inducing ℓ_1-norm is an acceptable substitute for the sparsity measure (i.e., ℓ_0-norm). Similarly, the row-1 norm, which is defined as $\|\alpha\|_{\text{row-1}} = \sum_{i=1}^{k} \|\alpha_i\|_2$, is a good heuristic for the row sparsity (i.e., row-0 norm). Actually, it is easy to reach the conclusion that the nuclear norm $\|A\|_*$ is equal to the group sparsity $\|\alpha\|_{\text{row-1}}$ under orthonormal subspace D, where $A = D\alpha$, if rows of α are orthogonal, as stated in Lemma 1.2. In this case, the subspace bases $D = U$ and coefficients $\alpha = SV^T$, where $A = USV^T$ by SVD. For the computational efficiency, ROSL removes this orthogonal constraint on α and recover the low-rank matrix A from X by minimizing the row-1 norm of α, and the ℓ_1-norm of E.

$$\min_{E,D,\alpha} \|\alpha\|_{\text{row-1}} + \lambda\|E\|_1 \quad \text{s.t.} D\alpha + E = X, D^T D = I_k, \forall i \tag{1.3}$$

1.2.2 Bound of Group Sparsity under Orthonormal Subspace

To show ROSL is a valid non-convex relaxation of the performance-guaranteed RPCA, we investigate the relationship between the group-sparsity-based rank formulation with matrix rank/nuclear norm.

PROPOSITION 1.1 Consider a thin matrix $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$, its SVD and orthonormal subspace decomposition are respectively denoted as $A = USV^T$ and $A = D\alpha$, where $D \in \mathbb{R}^{m\times n}, \alpha \in \mathbb{R}^{n\times n}$ and $D^T D = I_n$ without loss of generality. The minima of row-0 group sparsity and row-1 group sparsity of A under orthonormal subspace are respectively $\text{rank}(A)$ and nuclear norm $\|A\|_*$:

$$\text{(P1.1)} \qquad \min_{D\alpha=A, D^T D=I_n} \|\alpha\|_{\text{row-0}} = \text{rank}(A) \tag{1.4}$$

$$\text{(P1.2)} \qquad \min_{D\alpha=A, D^T D=I_n} \|\alpha\|_{\text{row-1}} = \|A\|_* \tag{1.5}$$

Proof of (P1.1) It is straightforward that the rank of A, where $A = D\alpha$, should not be larger than the dimension of α, resulting in that $\|\alpha\|_{\text{row-0}} \geq \text{rank}(\alpha) \geq \text{rank}(A)$. Thus, the row-0 norm of α under orthonormal subspace D is lower bounded by the rank of A.

Proof of (P1.2) This part can be restated as: $\|\alpha\|_{\text{row-1}} = \sum_{i=1}^{n} \|\alpha_i\|_2$, will reach its minimum $\|A\|_*$, when the orthonormal bases are equal to the principal components, i.e., $D = U$, where $A = USV^T$ by SVD. For simplicity of proof, we ignore other trivial solutions—the variations (column-wise permutation or $\pm$ column vectors) of U. Since both D and U are orthonormal bases, we reach the relationship, $D = U\Omega$ and $\alpha = \Omega^T SV^T$, where Ω is a rotation matrix $(\Omega^T\Omega = I_n, \det(\Omega) = 1)$. Here, we introduce a decreasing sequence of non-negative numbers $\sigma_i, 1 \leq i \leq n$ such that $S_i = \sigma_i, 1 \leq i \leq n$. To validate (P1.2), we need prove that the following relation holds for any Ω (the equality holds when Ω is the identity matrix).

$$\|\alpha\|_{\text{row-1}} = \|\Omega^T SV^T\|_{\text{row-1}} \geq \sum_{i=1}^{n} S_i = \|A\|_* \tag{1.6}$$

1. We begin with the special case that all the singular values are identical. Specifically, we decrease the singular values such that $\forall i \in \{1, ..., n\}$, $S_i = \sigma_n$, where σ_n is the last number in the decreasing sequence $\sigma_i, 1 \leq i \leq n$. Since each row of the rotation matrix Ω is a unit vector, we reach the following relationship:

$$\|\alpha\|_{\text{row-1}} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{n} \Omega_{ij}^2 S_i^2} = n\sigma_n = \sum_{i=1}^{n} S_i = \|A\|_* \tag{1.7}$$

2. Then, we try to prove that $\|\alpha\|_{\text{row-1}} \geq \|A\|_*$ still holds in the general case, i.e., $S_i = \sigma_i, 1 \leq i \leq n$. We can transform the special case above into the general case by $n-1$ steps, among which the t-th step is increasing the top $n-t$ singular values ($S_i, 1 \leq i \leq n-t$) from σ_{n-t+1} to σ_{n-t}. When increasing $S_i, 1 \leq i \leq n-1$ from σ_n to σ_{n-1} in the first step, the partial derivative of $\|\alpha\|_{\text{row-1}}$ with respect to S_i is calculated as follows:

$$\frac{\partial \|\alpha\|_{\text{row-1}}}{\partial S_i} = \sum_{j=1}^{n} \frac{\Omega_{ij}^2}{\sqrt{\sum_{t=1}^{n-1} \Omega_{tj}^2 + \Omega_{nj}^2 (S_n^2 / S_i^2)}} \tag{1.8}$$

Since $S_n \leq S_i, 1 \leq i \leq n-1$ and $\sum_{t=1}^{n} \Omega_{tj}^2 = 1$, we reach the following relationship:

$$\frac{\partial \|\alpha\|_{\text{row-1}}}{\partial S_i} \geq \sum_{j=1}^{n} \Omega_{ij}^2 = 1 = \frac{\partial \|A\|_*}{\partial S_i} \tag{1.9}$$

Thus, $\|\alpha\|_{\text{row-1}} \geq \|A\|_*$ holds when increasing $S_i, 1 \leq i \leq n-1$ in the first step. In the same way, we can prove that $\|\alpha\|_{\text{row-1}} \geq \|A\|_*$ holds in the following $n-2$ steps.

3. In sum, $\|\alpha\|_{\text{row-1}} \geq \|A\|_*$ in the general case where singular values S_i are not identical, i.e., $S_i = \sigma_i, \forall i \in \{1, ..., n\}$.

According to Proposition 1.1, the minimum of row-1 group sparsity under orthonormal subspace is the nuclear norm, i.e., $\|\alpha\|_{\text{row-1}} \geq \|A\|_*$, where $A = D\alpha$ and $D^T D = I_k$. Suppose, at weight λ, RPCA recovers the low-rank matrix as its ground truth A^*, i.e., $\widehat{A} = A^*$, then, $\|\widehat{\alpha}\|_{\text{row-1}} + \lambda\|X - \widehat{A}\|_1 \geq \|\widehat{A}\|_* + \lambda\|X - \widehat{A}\|_1 \geq \|A^*\|_* + \lambda\|X - A^*\|_1$ holds for any $(\widehat{A}, \widehat{D}, \widehat{\alpha})_{\widehat{A}=\widehat{D}\widehat{\alpha}, \widehat{D}^T\widehat{D}=I_k}$. In sum, at the weight λ, ROSL has the same global minimum $(\widehat{A} = A^*, \widehat{D} = U, \widehat{\alpha} = SV^T)$ as RPCA, where $A^* = USV^T$ by SVD.

1.2.3 A General Framework of Robust Low-Rank Recovery Approaches

TABLE 1.1 A general framework of robust low-rank recovery approaches. Given a corrupted low-rank matrix $X = A + E$, $A \in \mathbb{R}^{m \times n}$ $(m > n)$ and its projected version A' can be represented as $A = D\alpha$ and $A' = D'\alpha'$. All approaches follow the same framework; minimizing the sparsity and rank measures under some constraints, where I_n and Δ_n respectively denote identity and diagonal matrices.

Approaches	RPCA/SLRMD	RP-RPCA	RMF	LMaFit	ROSL
Sparsity Measure	$\|E\|_1$	$\|E\|_1$	$\|E\|_1$	$\|E\|_1$	$\|E\|_1$
Rank Measure	$\|\alpha\|_{\text{row-1}}$	$\|\alpha'\|_{\text{row-1}}$	$\|D\|_F^2 + \|\alpha\|_F^2$	N/A	$\|\alpha\|_{\text{row-1}}$
Constraints	$D^T D = I_n$ $\alpha^T \alpha = \Delta_n$	$D'^T D' = I_n$ $\alpha'^T \alpha' = \Delta_n$	N/A	N/A	$D^T D = I_n$

To better comparison of our ROSL with other existing approaches, we present a general framework of robust low-rank recovery approaches, as shown in Table 1.1. All the low-rank

recovery methods listed in the table utilizes the ℓ_1-norm to the sparsity measure and different low-rank measures. RPCA and its variant RP-RPCA use nuclear norm, which is equivalent to the groups sparsity under orthonormal subspace, with the constraint orthogonal coefficients. RMF uses the Frobenius-norm regularization as low-rank measure. LMaFit has no low-rank measure. To recover the low-rank structure, our ROSL seeks the groups sparsity under orthonormal subspace, without the constraint orthogonal coefficients.

ROSL can be considered to be a compromise between RPCA and ordinary matrix factorization methods (e.g. RMF and LMaFit). On one hand, ROSL improves upon RMF and LMaFit by seeking the group sparsity of A under orthonormal subspace D. This helps it to recover the low-rank structure of X without requiring accurate rank estimate. On the other hand, ROSL is a non-convex relaxation of RPCA by replacing nuclear norm $\|A\|_*$ with the group sparsity $\|\alpha\|_{\text{row-1}}$ under orthonormal subspace. As stated in Lemma 1.2, the nuclear norm $\|A\|_*$ is equal to the group sparsity $\|\alpha\|_{\text{row-1}}$ under orthonormal subspace D, where $A = D\alpha$, if rows of α are orthogonal. By removing the orthogonality constraint on α, ROSL can efficiently solve the low-rank recovery problem by sparse coding algorithms without requiring multiple iterations of SVD. To better comparison of our ROSL with other existing approaches, we present a general framework of robust low-rank recovery approaches, as shown in Table 1.1.

1.3 Fast Algorithm for ROSL

In this section an efficient algorithm is presented to solve the ROSL problem in Eq. (1.3).

Algorithm 1 ROSL Solver by inexact ADM/BCD

Require: $X \in \mathbb{R}^{m\times n}$, k, λ.
Ensure: D, α, E
1: $E^0 = Y^0 = \text{zeros}(m,n); D^0 = \text{zeros}(m,k); \alpha^0 = \text{rand}(k,n); \mu^0 > 0; \rho > 1; i = 0;$
2: **while** E not converged **do**
3: **for** $t = 1 \to k$ **do**
4: Compute the t-th residual: $R_t^i = X - E^i + Y^i/\mu^i - \sum_{j<t} D_j^{i+1}\alpha_j^{i+1} - \sum_{j>t} D_j^i\alpha_j^i$;
5: Orthogonalization:
$R_t^i = R_t^i - \sum_{j=1}^{t-1} D_j^{i+1}(D_j^{i+1})^T R_t^i$;
6: Update: $D_t^{i+1} = R_t^i {\alpha_t^i}^T$;
$D_t^{i+1} = D_t^{i+1}/(\|{D_t^{i+1}}^T\|_2)$;
7: Update:$\alpha_t^{i+1} = \overline{\mathbb{S}}_{1/\mu^i}({D_t^{i+1}}^T R_t^i)$;
8: **end for**
9: Prune: for $t = 1 \to k$, delete $(D_t^{i+1}, \alpha_t^{i+1})$ and set $k = k-1$, if $\|\alpha_t^{i+1}\|_2^2 = 0$;
10: Update: $E^{i+1} = \mathbb{S}_{\lambda/\mu^i}(X - D^{i+1}\alpha^{i+1} + Y^i/\mu^i)$;
11: Update: $Y^{i+1} = Y^i + \mu^i(X - D^{i+1}\alpha^{i+1} - E^{i+1}); \mu^{i+1} = \rho\mu^i; i = i+1;$
12: **end while**

1.3.1 Alternating Direction Method

Similar to [LCM10], we apply the augmented Lagrange multiplier (ALM) [Ber82] to remove the equality constraint $X = D\alpha + E$ in Eq. (1.3). Its augmented Lagrangian function is

written as:

$$\mathcal{L}(D,\alpha,E,Y,\mu) = \|\alpha\|_{\text{row-1}} + \lambda\|E\|_1 + Y(X - D\alpha - E) + \frac{\mu}{2}\|X - D\alpha - E\|_F^2 \quad \text{s.t.} \quad D^T D = I_k \tag{1.10}$$

where μ is the over-regularization parameter and Y is the Lagrange multiplier. We solve the above Lagrange function by inexact alternating direction method (ADM), which iterates through the following three steps:

1. Solve $(D^{i+1}, \alpha^{i+1}) = \arg\min \mathcal{L}(D, \alpha, E^i, Y^i, \mu^i)$.
2. Solve $E^{i+1} = \arg\min \mathcal{L}(D^{i+1}, \alpha^{i+1}, E, Y^i, \mu^i)$.
3. Update $Y^{i+1} = Y^i + \mu^i(X - D^{i+1}\alpha^{i+1} - E^{i+1}), \mu^{i+1} = \rho\mu^i$, where $\rho > 1$ is a constant.

In the first step, solving D and α simultaneously with constraint $D\alpha + E = X + \frac{Y}{\mu}$ is a non-convex problem. Fortunately, the sub-problem—updating one matrix when fixing the other one is convex. This indicates solving D and α using coordinate descent method. In the second step, we can easily update $E^{i+1} = \mathbb{S}_{\lambda/\mu^i}(X - D^{i+1}\alpha^{i+1} + \frac{Y^i}{\mu^i})$, where shrinkage function $\mathbb{S}_a(X) = \max\{\text{abs}(X) - a, 0\} \cdot \text{sign}(X)$ and "$\cdot$" denotes element-wise multiplication.

1.3.2 Block Coordinate Descent

Motivated by group sparse coding [BPSS09], we apply block coordinate descent (BCD) to solve D and α in the first step of ADM. Suppose the subspace bases $D = [D_1, ..., D_t, ..., D_k]$ and $\alpha = [\alpha_1; ...; \alpha_t; ...; \alpha_k]$, the BCD scheme sequentially updates the pair (D_t, α_t), by leaving all the other indices intact. In this way, it allows shrinking the group sparsity $\|\alpha\|_{\text{row-1}}$ under the orthonormal subspace D, while sequentially updating (D_t, α_t). In addition, it obtains new subspace bases and coefficients that best fit the constraint $A = D\alpha$ and thus achieves higher convergence rate, as explained in [AEB06, GL10]. The BCD scheme sequentially updates each pair $(D_t, \alpha_t), 1 \le t \le k$ such that $D_t\alpha_t$ is a good rank-1 approximation to R_t^i, where the residual is defined as $R_t^i = X + \frac{Y^i}{\mu^i} - E^i - \sum_{j<t} D_j^{i+1}\alpha_j^{i+1} - \sum_{j>t} D_j^i\alpha_j^i$. Thus, if removing the orthonormal constraint on D, the pair (D_t, α_t) can be efficiently updated as follows:

$$D_t^{i+1} = R_t^i {\alpha^i}^T \tag{1.11}$$

$$\alpha_t^{i+1} = \frac{1}{\|D_t^{i+1}\|_2^2}\overline{\mathbb{S}}_{1/\mu^i}({D_t^{i+1}}^T R_t^i) \tag{1.12}$$

where $\overline{\mathbb{S}}_a(X)$ is the magnitude shrinkage function defined as $\overline{\mathbb{S}}_a(X) = \max\{\|X\|_2 - a, 0\}X/\|X\|_2$. Due to the space limit, we refer the readers to [BPSS09] for the detailed induction of Eq. (1.12).

When taking into account the orthonormal subspace, we need to orthonormalize D_t^{i+1} by the Gram-Schmidt process. As shown in Algorithm 1, the new D_t^{i+1} is obtained via three steps: (1) project R_t^i onto the null space of $[D_1, ..., D_{t-1}]$, (2) update D_t^{i+1} as Eq. (1.11) and (3) then project it onto the unit sphere by normalization.

Above BCD scheme attempts to keep sequentially fitting the rank-1 subspaces $(D_t^{i+1}\alpha_t^{i+1})$ to the objective $X + \frac{Y^i}{\mu^i} = D^{i+1}\alpha^{i+1} + E^i$, until the fitted subspace is canceled by magnitude shrinkage, i.e., $\|\alpha_t^{i+1}\|_2 = 0$. To improve the computational efficiency, we shrink the subspace dimension k by pruning the zero pairs, for they will stay zero in the next iteration.

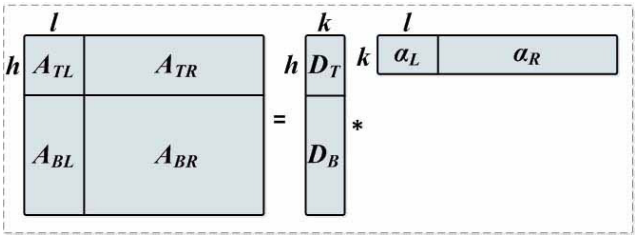


FIGURE 1.2 Decomposition of the low-rank matrix $A \in \mathbb{R}^{m \times n}$.

It is possible to run many rounds of BCD to solve D^{i+1} and α^{i+1} exactly in the first step of ADM. In practice, updating $(D_t^{i+1}, \alpha_t^{i+1})$, $1 \leq t \leq k$ once at each round of ADM is shown to be sufficient for the inexact ADM algorithm to converge to a valid solution (D^{i+1}, α^{i+1} and E^{i+1}) to Eq. (1.3).

As shown in Algorithm 1, ROSL can be solved using inexact ADM at the higher scale and inexact BCD at the lower scale. To the best of our knowledge, there is no established convergence theory, either for ADM algorithms applied to non-convex problems with more than two groups of variables [SWZ11], or for BCD algorithms applied to sparse coding [AEB06, BPSS09]. As all 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: $A^{i+1} = D^{i+1}\alpha^{i+1}$ and E^{i+1}, when the initialize E^0, Y^0 and D^0 as zero matrices, as well as α^0 as a random matrix.

1.3.3 Computational Complexity

Compared with RPCA, which has cubic complexity of $\mathcal{O}(\min(m^2n, mn^2)))$, ROSL is much more efficient, when the matrix rank $r \ll \min(m, n)$. Its dominant computational processes are (1) left multiplying the residual matrix $R \in \mathbb{R}^{m \times n}$ by D, and (2) right multiplying it by α. Thus, the complexity of ROSL depends on the subspace dimension k. If we set the initial value of k as several times larger than r (i.e., r and k are on the same order, being much smaller than m and n), ROSL has the quadratic complexity of matrix size, i.e., $\mathcal{O}(mnk)$ or $\mathcal{O}(mnr)$.

1.4 Acceleration by Random Sampling

Inspired by Nyström method [WS00, KMT09, TR10], we present a random sampling scheme to further speed up ROSL such that its fast version (ROSL+) has linear complexity with respect to the matrix size.

1.4.1 Random Sampling in ROSL+

As shown in Fig. 1.2, the low-rank matrix $A \in \mathbb{R}^{m \times n}$ is first permuted column-wisely and row-wisely, and then divided into four sub-matrices ($A_{TL} \in \mathbb{R}^{h \times l}$, A_{TR}, A_{BL} and A_{BR}). Accordingly, top sub-matrix A_T and left sub-matrix A_L are respectively defined as $A_T = [A_{TL}, A_{TR}]$ and $A_L = [A_{TL}; A_{BL}]$. The same permutation and division are done on X and E. As shown in Fig. 1.2, subspace bases D is divided into $D_T \in \mathbb{R}^{h \times k}$ and D_B, as well as coefficients α is divided into $\alpha_L \in \mathbb{R}^{k \times l}$ and α_R, such that

$$A = \begin{bmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{bmatrix} = \begin{bmatrix} D_T \\ D_B \end{bmatrix} [\alpha_L \quad \alpha_R] \tag{1.13}$$

Nyström method is initially used for large dense matrix approximation [KMT09], and extended to speed up RPCA in DFC [MTJ11]. Suppose $\text{rank}(A_{TL}) = \text{rank}(A) = r$, instead of recovering the full low-rank matrix A, DFC first recovers its sub-matrices and then approximates $\widehat{A}$ as:

$$\widehat{A} = \widehat{A_L}(\widehat{A_{TL}})^+\widehat{A_T} \tag{1.14}$$

where "+" denotes pseudo-inverse. However, DFC does not describe how to estimate the top-left submatrix.

Here, we investigate this specific issue and further simplify Nyström method in the framework of robust subspace learning. An intuitive solution would be independently recovering all three sub-matrices. But this requires exhaustively tuning different parameters λ, which eventually prevents from achieving high accuracy. The feasible way is that ROSL+ directly recovers the left sub-matrix and the top submatrix, i.e., $\widehat{A_L} = \widehat{D}\widehat{\alpha_L}$ and $\widehat{A_T} = \widehat{D_T}\widehat{\alpha}$, and then approximates $\widehat{A_{TL}}$ by the left sub-matrix of $\widehat{A_T}$. Thus, the low-rank matrix A can be reconstructed as follows:

$$\widehat{A} = \widehat{A_L}((\widehat{A_T})_L)^+\widehat{A_T} = \widehat{D}\widehat{\alpha_L}((\widehat{\alpha})_L)^+\widehat{\alpha} \tag{1.15}$$

where $(X)_L$ denotes the left sub-matrix of X. Actually, when $\text{rank}(A_{TL}) = \text{rank}(A)$ holds, $\widehat{\alpha_L}$ recovered from the left observation matrix X_L is a good approximation to, or exactly equal to, $(\widehat{\alpha})_L$ recovered from the top observation matrix X_T. The same relationship exists between $(\widehat{D})_T$ and $\widehat{D_T}$, where $(\widehat{D})_T$ denotes the top sub-matrix of $\widehat{D}$. Thus, we can further simplify ROSL+ as

$$\widehat{A} = \widehat{D}\widehat{\alpha} \tag{1.16}$$

where $\widehat{D}$ and $\widehat{\alpha}$ is respectively recovered from X_L and X_T in the following two simple steps.

1. Solve $\widehat{D}$ and $\widehat{\alpha_L}$ by applying ROSL on X_L:

$$\min_{D,\alpha_L,E_L} \|\alpha_L\|_{\text{row-1}} + \lambda\|E_L\|_1 \;\; \text{s.t.} \;\; \begin{matrix} X_L = D\alpha_L + E_L \\ D^TD = I_k \end{matrix} \tag{1.17}$$

2. Solve $\widehat{\alpha}$ by minimizing $\|X_T - \widehat{D_T}\alpha\|_1$ by fixing $\widehat{D_T}$ as $(\widehat{D})_T$.

In other words, ROSL+ first recovers $\widehat{D}$ from the left sub-matrix X_L (complexity: $\mathcal{O}(mlr)$), and then solve $\widehat{\alpha}$ by minimizing the ℓ_1-norm of $X_T - \widehat{D_T}\alpha$ (complexity: $\mathcal{O}(nhr)$). Thus, the complexity of ROSL+ is $\mathcal{O}(r(ml+nh))$. When the matrix rank r is much smaller that its size, i.e., $r \ll \min(m,n)$, the sample number can be set as $l = \beta_2 r$ and $h = \beta_3 r$, where β_2 and β_3 are constants larger than 1. In this case, ROSL+ has the linear complexity of the matrix size, i.e., $\mathcal{O}(r^2(m+n))$.

1.5 Experimental Results

We present several experiments to evaluate the performance of ROSL and ROSL+ including (1) simulation on a corrupted synthetic low-rank matrix of varying dimension, (2) visual low-rank recovery on real data for background subtraction. Note that, ROSL algorithm is implemented in MATLAB without using any advanced tools unlike some other methods we compare. All the experimental results are executed on an Intel W3530 CPU and 6GB memory. For simplicity, we set the sample number $h = l$ for ROSL+ and other sampling-based methods we tested.

Similar to [MDYY11], a square low-rank matrix $A \in \mathbb{R}^{m\times m}$ is synthesized as a product of a $m \times r$ matrix and a $r \times m$ matrix (r is set to be 10), whose entries obey the normal

TABLE 1.2 Evaluation of ROSL, ROSL+ and the existing low-rank recovery approaches on synthetic $m \times m$ low-rank matrices with rank $r = 10$.

	m=500		m=1000		m=2000		m=4000		m=8000		iter
	MAE	**Time**	MAE	**Time**	MAE	**Time**	MAE	**Time**	MAE	**Time**	
RPCA	2.8E-6	**2.51**	1.0E-6	**12.7**	5.7E-7	**112**	1.2E-6	**981**	N/A	**N/A**	18∼20
Partial RPCA	2.2E-6	**1.44**	1.1E-6	**5.6**	7.6E-7	**24.4**	5.3E-7	**161**	6.7E-7	**802**	18∼20
RP-RPCA	3.0E-2	**5.9**	3.7E-1	**23.7**	4.2E-1	**110**	7.7E-1	**669**	1.6E+0	**3951**	300
LMaFit	5.3E-1	**6.9**	3.8E-1	**28.7**	1.8E-1	**116**	3.4E-2	**442**	5.0E-3	**1750**	300
ROSL	6.3E-6	**0.78**	6.1E-6	**2.83**	2.2E-6	**12.8**	9.8E-6	**41.8**	2.2E-6	**214**	16∼17
ROSL-Nys1	2.4E-0	**0.42**	2.6E-0	**0.89**	2.3E-0	**1.56**	3.0E-0	**3.78**	2.8E-0	**9.0**	18∼20
ROSL-Nys2	4.8E-5	**0.42**	5.4E-5	**0.89**	5.0E-5	**1.56**	4.3E-5	**3.77**	4.6E-5	**8.9**	18∼20
ROSL+	2.9E-5	**0.31**	3.1E-5	**0.65**	3.3E-5	**1.1**	2.7E-5	**2.5**	2.2E-5	**5.6**	18∼20

The Mean of Absolute Error (MAE) between A and $\widehat{A}$ is used to gauge the recovery accuracy. The iterations (rounds of ADM) and the total running time (seconds) are reported. Note: aEb denotes $a \times 10^b$. Parameters are set up as: (1) λ is best tuned for each method. (2) The dimension of D is initialized as $k = 30$. (3) The stop criterion is $\|X - A^{i+1} - E^{i+1}\|_F / \|X\|_F \leq 10^{-6}$. (4) The maximum iteration number (iter) is set to be 300. (5) The sample number $l = h = 100$.

distribution. Then, the corrupted data X is generated by the addition of A and a sparse matrix $E \in \mathbb{R}^{m \times m}$ (10% of its entries are non-zero and drawn from the uniform distribution on [-50, 50]).

On this synthetic data, we evaluate the recovery accuracy and efficiency of ROSL, compared with RPCA, RP-RPCA and LMaFit (advanced version of RMF). As shown in Table 1.2, ROSL is much faster than these methods without compromising the recovery accuracy. The original RPCA using full SVD is computationally costly and is almost infeasible when the matrix size $m = 8000$. Even partial RPCA [LCM10] is consistently 4 times slower than ROSL and also requires a proper way to update κ. Although random projection helps reduce the computation of a single SVD, many iterations of SVD are needed to be conducted on different projected matrices. Thus, the total computation of RP-RPCA is costly and its recovery accuracy is low (Table 1.2). In the ideal case that the matrix rank is known, LMaFit has the same accuracy and complexity as ROSL. However, since it is unable to minimize the matrix rank, it fails to obtain accurate low-rank matrix recovery without exactly setting $k = r$. On this synthetic data (rank $r = 10$) in Table 1.2, LMaFit converges very slowly and fails to obtain accurate recovery at $k = 30$, which is true even at $k = 14$.

To evaluate the performance of ROSL+, we apply the generalized Nyström method (employed in DFC) to ROSL, called ROSL-Nys. Since the performance of ROSL-Nys highly depends on how to recover A_{TL}, we present two different variants of ROSL-Nys, i.e., ROSL-Nys1 recovering sub-matrices (A_{TL}, A_T and A_L) independently, and ROSL-Nys2 recovering A_{TL} by left sub-matrix of A_T. Actually, DFC also employed another column sampling method. But it requires recovering multiple (i.e., $\frac{n}{l}$) sub-matrices (size:$m \times l$) and thus has quadratic complexity, much slower than ROSL+ (linear complexity). As shown in Table 1.2, RPCA-Nys1 fails to obtain accurate recovery. The reason is that tuning a common weight λ cannot guarantee the optimality of three subproblems—estimating A_L, A_T and A_{TL}. Both the computational complexity and recovery accuracy of ROSL+ are on the same order of that of ROSL-Nys2, and are slightly ($1.5 \sim 2$ times) better that the latter. This better performance is due to that ROSL+ consists of only one time ROSL and one time linear regression.

In addition, we evaluate the stability and convergence rate of ROSL/ROSL+ on the same synthetic matrix by varying the initial rank k, weight λ or submatrix size l. Firstly, we observed that the recovery accuracy and convergence rate of ROSL are not sensitive to selection of k, as long as $k > r$. As shown in Fig. 1.3, $\forall k \in [20, 100]$, the subspace dimension recovered by ROSL at $\lambda = 0.03$ fast converges to the rank $r = 10$ and the high accuracy (MAE $\approx 10^{-6}$) is achieved. Secondly, ROSL produces accurate low-rank recovery at any

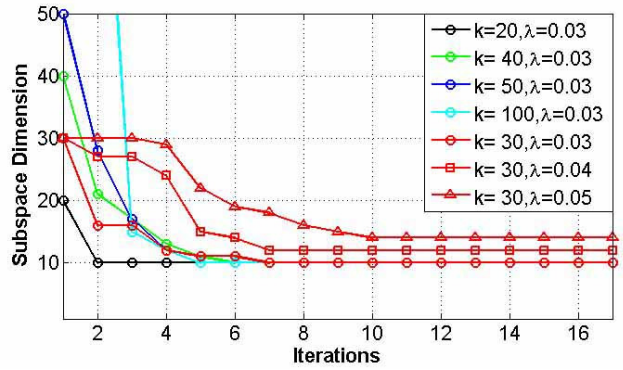


FIGURE 1.3 Convergence rate of ROSL. At the fixed $\lambda = 0.03$, the recovered subspace dimension always converges to $r = 10$ in less than 7 iterations **regardless** of the initial value of k, which indicates the ROSL solver is robust and very stable. The recovered subspace dimension increases as λ increases from 0.03 to 0.05. MAE $\approx 10^{-6}$ at all cases above.

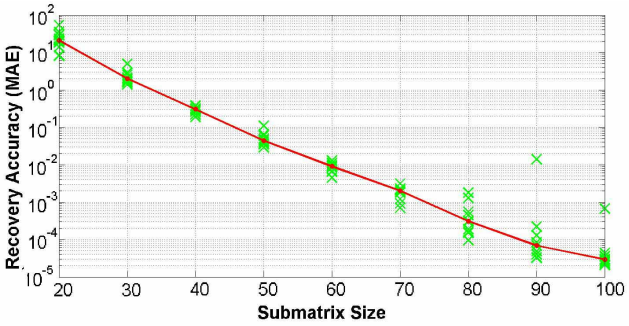


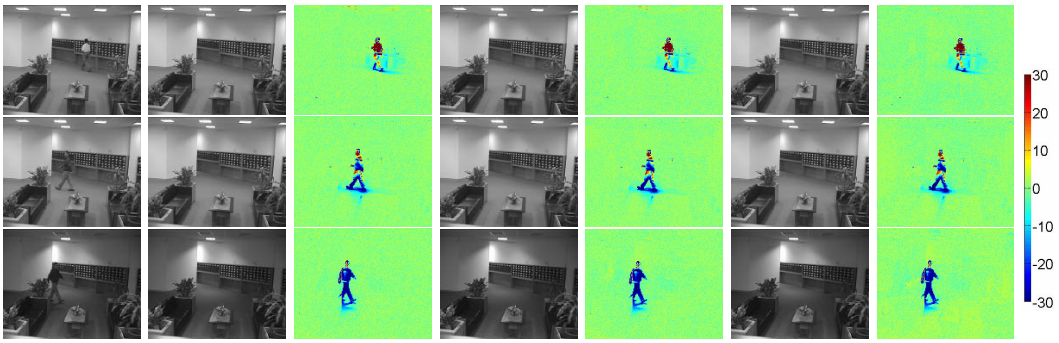
FIGURE 1.4 Recovery accuracy (MAE) of ROSL+ on synthetic data ($m = 1000$, $r = 10$, $k = 30$). For each l, the recovery errors (MAE) of ROSL+ in 10 different random-sampling trials are shown in green (their median in red). The recovery error (MAE) of ROSL+ decreases exponentially with the increase of l. These tests also indicate that ROSL+ gets the same global solution as RPCA in almost all cases.

weight $\lambda \in [0.03, 0.05]$ and the recovered subspace dimension consistently increases with λ. ROSL recovers the 14-dimension orthonormal subspace when $\lambda = 0.05$ and obtains accurate recovery (MAE$\approx 10^{-6}$). Thirdly, at the fixed sub-matrix size l, the recovery accuracy of ROSL+ is relatively stable in different random sampling trials. As the submatrix size l increases, the recovery error (MAE) of ROSL+ decreases exponentially and reaches as low as 3×10^{-5} when $l = 10r = 100$ (Fig. 1.4). This result is in line with the failure probability δ of rank(A_{TL})=rank(A) that exponentially decreases with the increase of l.

To compare the recovery accuracy of ROSL/ROSL+ with that of RPCA, we evaluate them on two standard visual data sets, Yale-B face images and the lobby background subtraction video, similar to [CLMW11]. From each video, we build an observation matrix X by vectorizing each frame as one column, and respectively recover the low-rank component A from X by ROSL and RPCA.

In the lobby video, both ROSL and ROSL+ exactly recover the same (accurate) foreground objects and background components as RPCA at much faster speeds (ROSL: **10**×, ROSL+: **92**×) as shown in Fig. 1.5.

In the face image experiments, the non-diffusive component E detected by ROSL is almost the same as that by RPCA (Fig. 1.6). The results of ROSL+ are very close to those of ROSL and thus not included in Fig. 1.6, due to the space limit. Note that, the lobby



(a) (b) (c) (d) (e) (f) (g)

FIGURE 1.5 Comparison of RPCA, ROSL($k = 10$) and ROSL+($l = 50$) in background modeling on the lobby video (size: 160×128, 1060 frames). (a) Original images. Backgrounds recovered by (b) RPCA, (d) ROSL, and (f) ROSL+. Foregrounds recovered by (c) RPCA, (e) ROSL, and (g) ROSL+. ROSL (time: 34.6s) and ROSL+ (time: 3.61s) are significantly ($10\times$, $92\times$) faster than RPCA (time: 334s) while generating almost identical results.

video is a thin matrix ($20480\times$ 1060) and the efficiency improvement of ROSL/ROSL+ is expected to be even higher for large-scale square matrices. Such matrices are common in typical applications, e.g., in video summarization (10^5 images of 10^6 pixels) and in face recognition (10^6 images of 10^6 pixels).

1.6 Conclusion

A Robust Orthonormal Subspace Learning (ROSL) approach is presented for efficient robust low-rank recovery. This approach accelerates the state-of-the-art method, i.e., RPCA, by replacing the nuclear norm on the low-rank matrix by a light-weight measure—the group sparsity of its coefficients under orthonormal subspace. This enables using fast sparse coding algorithms to solve the robust low-rank recovery problem at the quadratic complexity of matrix size. This novel rank measure is proven to be lower-bounded by the nuclear norm and thus ROSL has the same global optima as RPCA. In addition, a random sampling algorithm is introduced to further speed up ROSL such that ROSL+ has linear complexity of the matrix size. Experimental results on the synthetic and real data show that ROSL and ROSL+ achieve the state-of-the-art efficiency at the same level of recovery accuracy.

1.7 Acknowledgement

The support of Mitsubishi Electric Research Lab (MERL) and the support of National Science Foundation under grant IIS 11-44227 are gratefully acknowledged.

References

1. M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
2. D. Bertsekas, editor. *Constrained optimization and Lagrange multiplier method*. Academic Press, 1982.
3. S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. In *NIPS*, volume 22, pages 82–89, 2009.

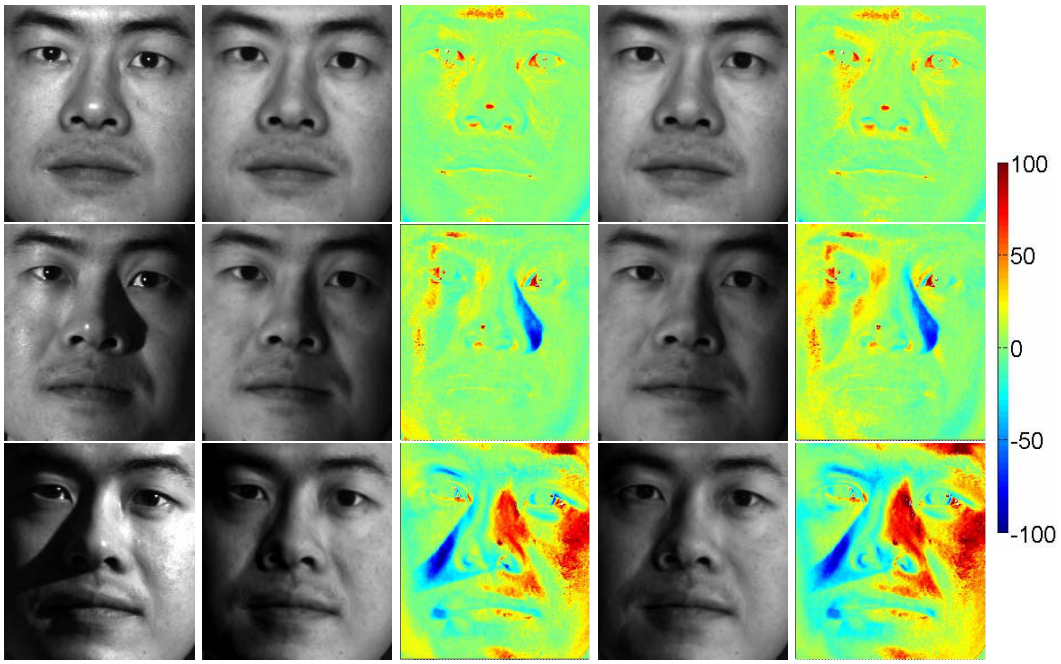


FIGURE 1.6 Visual evaluation of ROSL and RPCA on face images (168×192, 55 frames) under varying illuminations. There is no significant difference between ROSL and RPCA. (a) Original images, diffusive component recovered by (b) RPCA and (d) by ROSL. Non-diffusive component (shadow/specularity) by (c) RPCA (time: 12.16s) and (e) by ROSL (time: 5.85s).

4. E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):article 11, 2011.
5. M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of American Control Conference*, volume 42, pages 115–142, 2001.
6. K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, 2010.
7. J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *ICML*, 2009.
8. Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, volume 1, pages 739–746, 2005.
9. S. Kumar, M. Mohri, and A. Talwalkar. On sampling-based approximate spectral decomposition. In *ICML*, 2009.
10. Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical Report Technical Report UILU-ENG-09-2214, UIUC, 2010.
11. R. Liu, Z. Lin, Z. Su, and J. Gao. Solving principal component pursuit in linear time via l1 filtering. *Neurocomputing*, 2014.
12. J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for iamge restoration. In *ICCV*, 2009.
13. Y. Mu, J. Dong, X. Yuan, and S. Yan. Accelerated low-rank visual recovery by random

projection. In *CVPR*, 2011.
14. L. Mackey, A. Talwakar, and M. Jordan. Divide-and-conquer matrix factorization. In *NIPS*, 2011.
15. B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *Arxiv preprint:0706.4138*, 2007.
16. N. Srebro, J.D.M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *NIPS*, volume 17, pages 1329–1336, 2005.
17. Y. Shen, Z. Wen, and Y. Zhang. Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. Technical Report TR11-02, Rice University, 2011.
18. A. Talwalkar and A. Rostamizadeh. Matrix coherence and the nystrom method. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, 2010.
19. R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. A fast augmented lagrangian algorithm for learning low-rank matrices. In *ICML*, 2010.
20. B. Turlach, W. Venables, and S. Wright. Simultaneous variable selction. *Technometrics*, 47:349–363, 2005.
21. C. Williams and M. Seeger. Using the nystrom method to speed up the kernel machines. In *NIPS*, 2000.
22. M. Yuan and Y. Lin. Model selection and estimation in regresion with grouped variables. *Journal of the Royal Statistical Society. Series B*, 68:49–67, 2006.
23. X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. Technical Report Technical report, Hong Kong Baptist University, 2009.
24. T. Zhou and D. Tao. GoDec: randomized low-rank and sparse matrix decomposition in noisy case. In *ICML*, 2011.