MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Factorized Robust Matrix Completion

Mansour, H.; Tian, D.; Vetro, A.

TR2016-066 June 2016

Abstract

In this book chapter, we discuss a factorization-based approach to robust matrix completion. We begin our discussion with a gauge optimization perspective to robust matrix completion. We then discuss how our approach replaces the solution over the low rank matrix with its low rank factors. In this context, we develop a gauge minimization algorithm and an alternating direction method of multipliers algorithm that take advantage of the factorized matrix decomposition. We then focus on the particular application of video background subtraction, which is the problem of finding moving objects in a video sequence that move independently from the background scene. The segmentation of moving objects helps in analyzing the trajectory of moving targets and in improving the performance of object detection and classification algorithms. In scenes that exhibit camera motion, we first extract the motion vectors from the coded video bitstream and fit the global motion of every frame to a parametric perspective model. The frames are then aligned to match the perspective of the first frame in a group of pictures (GOP) and use our factorized robust matrix completion algorithm to fill in the background pixels that are missing from the individual video frames in the GOP. We also discuss the case where additional depth information is available for the video scene and develop a depth-weighted group-wise PCA algorithm that improves the foreground/background separation by incorporating the depth information into the reconstruction.

Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2016
201 Broadway, Cambridge, Massachusetts 02139

1

Factorized Robust Matrix Completion

Hassan Mansour
Mitsubishi Electric Research Laboratories

Dong Tian
Mitsubishi Electric Research Laboratories

Anthony Vetro
Mitsubishi Electric Research Laboratories

1.1 Introduction

The problem of reconstructing large scale matrices from incomplete and noisy observations has attracted a lot of attention in recent years. Of particular interest is the reconstruction or completion of low-rank matrices, which finds many practical applications in recommender systems, collaborative filtering, system identification, and video surveillance.

The robust matrix reconstruction/completion problem can be formulated as the task of determining a low rank matrix L from observations A that are contaminated by sparse outliers S and noise E. Moreover, the observations A are acquired through a linear operator $\mathcal{P}$ that generates a smaller number of samples than those in L. In the case of matrix completion, the operator $\mathcal{P}$ is a restriction operator that selects a subset Ω of the samples in L. The general observation model is given as follows:

$$A = \mathcal{P}(L) + S + E. \tag{1.1}$$

A natural approach for recovering L from A involves solving the rank minimization problem

$$\min_{L} \operatorname{rank}(L) \;\text{ subject to }\; h(A - \mathcal{P}(L)) \leq \sigma, \tag{1.2}$$

where $h(\cdot)$ is some penalty function suitable for the noise statistics, and σ is a mismatch tolerance. However, problem (1.2) is noncovex and generally difficult to compute even when the function $h(\cdot)$ is convex. Alternatively, Fazel et al. [FHB01] introduced the nuclear norm heuristic as the convex envelope of the rank to replace the rank objective resulting in the nuclear norm minimization problem

$$\min_{L} \|L\|_* \;\text{ subject to }\; h(A - \mathcal{P}(L)) \leq \sigma, \tag{1.3}$$

where the nuclear norm $\|X\|_*$ is equal to the sum of the singular values of the matrix X. The attractiveness of the nuclear norm minimization problem lies in the ability to derive efficient algorithms for solving it [LB09, CCS10, RFP10, AKM+13]. Moreover, Recht et al. [RFP10] derived conditions for which the solution to the nuclear norm minimization problem (1.3) coincides with that of the rank minimization problem (1.2) for the case where $\mathcal{P}$ is a random matrix and $h(\cdot)$ is the Frobenius norm. Recovery guarantees for the case where $\mathcal{P}$ is a matrix completion operator was also extensively studied in [CT10, Rec11]. Another approach (SpaRCS) was proposed in [WSB11] which follows a greedy approach that iteratively estimates the low rank subspace of L as well as the support of S followed by truncated SVD and least squares inversion to compute estimates for L and S.

The choice of the penalty function $h(\cdot)$ determines the sensitivity of the solution of problem (1.3) to the noise model. For example, letting $h(\cdot)$ be the matrix Frobenius norm assumes that there are no sparse outliers S and the error E is Gaussian distributed. In the case where the observations A are contaminated with large sparse outliers, robust penalty functions such as the ℓ_1 norm, Huber or the Student's t penalties have been shown to produce more robust reconstructions [ABF13]. When the penalty function is the ℓ_1 norm, an equivalent formulation to (1.3) can be realized through the stable principal component pursuit (SPCP) problem given by

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{subject to} \quad \|A - \mathcal{P}(L) + S\|_F \leq \sigma_E, \tag{1.4}$$

where λ is a regularization parameter that corresponds to the tolerance σ in (1.3), and σ_E is a bound on the noise level E. The SPCP problem has the advantage of also determining the sparse component S which can be the real target in some applications such as video background subtraction. When $\sigma_E = 0$, the problem is also known as robust principal component analysis (RPCA) or sparse matrix separation.

In this chapter, we discuss a factorization-based approach to robust matrix completion in the context of solving (1.4). We begin our discussion with a gauge optimization perspective to robust matrix completion in Section 1.2. We then discuss in Section 1.3 how our approach replaces the solution over the matrix L with its low rank factors U and V, such that $L = UV^T$, similar to the approach adopted by Aravkin et al. [AKM+13]. In this context, we develop a gauge minimization algorithm and an alternating direction method of multipliers algorithm that take advantage of the factorized matrix decomposition. We then focus in Section 1.4 on the particular application of video background subtraction, which is the problem of finding moving objects in a video sequence that move independently from the background scene. The segmentation of moving objects helps in analyzing the trajectory of moving targets and in improving the performance of object detection and classification algorithms. In scenes that exhibit camera motion, we first extract the motion vectors from the coded video bitstream and fit the global motion of every frame to a parametric perspective model. The frames are then aligned to match the perspective of the first frame in a group of pictures (GOP) and use our factorized robust matrix completion algorithm to fill in the background pixels that are missing from the individual video frames in the GOP. We also discuss the case where additional depth information is available for the video scene and develop a depth-weighted group-wise PCA algorithm that improves the foreground/background separation by incorporating the depth information into the reconstruction. Finally, we conclude the chapter with Section 1.5.

1.1.1 Notation

Throughout the chapter we use upper case letters to refer to matrices or their vectorization interchangeably depending on the context. Consider a matrix $X \in \mathbb{R}^{m \times n}$ of rank $r <$

$\min\{m,n\}$ with singular values $\sigma_i,\ i = 1 \dots r$ indexed in decreasing order. We denote by $\|X\|_F = \sqrt{\sum_{i,j} X_{ij}^2} = \sum_{i=1}^{r} \sigma_i$ and $\|X\|_{\text{op}} = \sigma_1$ the Frobenius norm and the operator norm of a matrix X, respectively. Also, we denote by $\|X\|_1 = \sum_{i,j} |X_{ij}|$ and $\|X\|_\infty = \max_{i,j} |X_{ij}|$ the ℓ_1 norm and the ℓ_∞ norm of the vectorization of the matrix X, respectively. For any two matrices $X, Y \in \mathbb{R}^{m\times n}$, we denote the trace of the inner product by $\langle X, Y\rangle = \text{Tr}\left[X^T Y\right]$.

1.2 Robust Matrix Completion

Consider the case where we are given incomplete measurements A of a data matrix $X_0 \in \mathbb{R}^{m\times n}$ that is composed of the superposition of a low rank matrix L_0 and a sparse outlier matrix S_0, such that, $X_0 = L_0 + S_0$. Let $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ be a restriction operator that selects a subset Ω of size p of the mn samples in X_0. We define the robust matrix completion problem as the problem of finding L_0 from the incomplete measurements $A = \mathcal{P}_\Omega(L_0) + S_0$ using the following ℓ_1-norm constrained nuclear norm minimization problem

$$\min_L \|L\|_* \ \text{ subject to } \ \|A - \mathcal{P}_\Omega(L)\|_1 \le \sigma, \tag{1.5}$$

where the tolerance σ has to be set equal to an upper bound on the ℓ_1-norm of S_0. We may also write problem (1.5) by introducing the variable S such that

$$\begin{aligned} \min_{L,S} \|L\|_* \ \text{ subject to } \quad & A = \mathcal{P}_\Omega(L) + S, \\ & \|S\|_1 \le \sigma. \end{aligned} \tag{1.6}$$

Note that for an appropriate choice of λ, problem is equivalent to the robust PCA problem

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1 \ \text{ subject to } \ A = L + S. \tag{1.7}$$

Problems (1.5) and (1.7) belong to the general class of constrained gauge minimization problems [FMP14] for which a solution framework was developed by van den Berg and Friedlander in [vdBF08, vdBF11] for the ℓ_2 constrained case and later extended by Aravkin et al. [ABF13] for arbitrary convex constraints. A gauge $\kappa(\cdot)$ is a convex, nonnegative, positively homogeneous function that vanishes at the origin. Moreover, we characterize a general gauge optimization problem in L and S as

$$\min_{L,S} \kappa(L,S) \ \text{ subject to } \ h(L,S) \le \sigma, \tag{1.8}$$

where $h(\cdot)$ is a convex function. Note that norms are special cases of gauge functions that are finite everywhere, symmetric, and zero only at the origin [FMP14]. Following the framework of [vdBF11], we can define the value function

$$\phi(\tau) = \min_{L,S} h(L,S) \ \text{ subject to } \ \kappa(L,S) \le \tau, \tag{1.9}$$

and update τ using Newton's method to find τ^* that solves $\phi(\tau) = \sigma$. The solution (L^*, S^*) of $\phi(\tau^*)$ will then be the minimizer of (1.8). The jth Newton update of τ is given by

$$\tau_{j+1} = \tau_j + \frac{\phi(\tau_j) - \sigma}{\phi'(\tau_j)}, \tag{1.10}$$

which requires the evaluation of the derivative $\phi'(\tau)$ with respect to τ. In particular, it was shown in [vdBF11] that $\phi'(\tau) = \kappa^o(h'(L,S))$, where κ^o is the polar of κ, and h' is the derivative of h with respect to L and S.

The framework described above is attractive when the optimization problem defining the value function $\phi(\tau)$ is easy to solve. Consider problem (1.5), the penalty function $h(L,S) = \|A - \mathcal{P}_\Omega(L)\|_1$, which is convex but non-differentiable. On the other hand, the penalty function of problem (1.7) is the Frobenius norm $h(L,S) = \|A - L - S\|_F$ and the Newton step terminates when $\phi(\tau) = 0$. Consequently, [vdBF11] derives the RPCA value function as

$$\phi(\tau) = \min_{L,S} \|A - \mathcal{P}_\Omega L - S\|_F \;\text{ subject to }\; \|L\|_* + \lambda\|S\|_1 \leq \tau, \tag{1.11}$$

where τ is updated according to the Newton step

$$\tau_{j+1} = \tau_j + \frac{\|R_j\|_F}{\max\{\|\mathcal{P}_\Omega^T R_j\|_{\text{op}}, \|\mathcal{P}_\Omega^T R_j\|_\infty/\lambda\}}, \tag{1.12}$$

where $R_j = \frac{A - \mathcal{P}_\Omega(L_j+S_j)}{\|A-\mathcal{P}_\Omega(L_j+S_j)\|_F}$ is the normalized residual matrix at the jth Netwon iteration, and $S_j = \mathcal{P}_\Omega S_j$ since $\mathcal{P}_\Omega$ is a mask. Note here that the denominator $\max\{\|\mathcal{P}_\Omega^T R_j\|_{\text{op}}, \|\mathcal{P}_\Omega^T R_j\|_\infty/\lambda\}$ is the expression for the polar of the gauge function $\kappa(X,Y) = \|X\|_* + \lambda\|Y\|_1$, where $X = Y = \mathcal{P}_\Omega^T R_j$ in this case.

Finally, we point out that it was shown in [CLYW11] that a choice of $\lambda = \hat{n}^{-1/2}$, $\hat{n} := \max\{m,n\}$, is sufficient to guarantee the recovery of L_0 and S_0 with high probability when the $\text{rank}(L_0) \leq C\hat{n}\,(\log\hat{n})^{-2}$ for some constant C that depends on the coherence of the subspace of L_0.

1.3 Factorized Robust Matrix Completion

One drawback of problem (1.7) is that it requires the computation of full (or partial) singular value decompositions of L in every iteration of the algorithm, which could become prohibitively expensive when the dimensions are large. To overcome this problem, we adopt a proxy for the nuclear norm of a rank-r matrix L defined by the following factorization from Lemma 8 in [N.04]

$$\|L\|_* = \inf_{U\in\mathbb{R}^{m,r}, V\in\mathbb{R}^{n,r}} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) \text{ subject to } UV^T = L. \tag{1.13}$$

The nuclear norm proxy has recently been used in standard nuclear norm minimization algorithms [RR13, AKM$^+$13] that scale to very large matrix completion problems.

In this section, we discuss two algorithms that rely on the factorization in (1.13); the first algorithm follows the gauge minimization technique described in section 1.2, whereas the second is an alternating direction method (ADM) that minimizes the augmented Lagrangian of problem (1.7).

1.3.1 Factorized gauge minimization algorithm

The gauge minimization framework discussed in section 1.2 is summarized in Algorithm 1 and requires the following three basic tools:

1. Defining and computing the value function $\phi(\tau)$.
2. Solving the projection onto the gauge constraint $\kappa(\cdot) \leq \tau$.
3. Specifying the polar function κ^o to update τ.

Algorithm 1 Gauge minimization algorithm for factorized robust matrix completion (Gauge-FRMC)

1: **Input** Measurement matrix A, measurement operator $\mathcal{P}_\Omega$, regularization parameter λ, mismatch tolerance σ
2: **Output** Low rank factors U, V, and sparse component S
3: **Initialize** $\tau_0 = 0$, residual signal $R_0 = A$, $j = 0$
4: **while** $\|R_j\|_F > \sigma$ **do**
5: $j = j + 1$
6: Update the gauge constraint τ_j:

$$\tau_j = \tau_{j-1} + \frac{\|R_{j-1}\|_F}{\max\{\|\mathcal{P}_\Omega^T R_{j-1}\|_{\text{op}}, \|\mathcal{P}_\Omega^T R_{j-1}\|_\infty/\lambda\}}$$

7: Compute the value function $\phi(\tau_j)$ by solving for (U, V, S):

$$\phi(\tau_j) = \min_{U,V,S} \|A - \mathcal{P}_\Omega(UV^T + S)\|_F \;\text{ subject to }\; \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \lambda\|S\|_1 \le \tau_j$$

8: Update the residual R_j:

$$R_j = A - \mathcal{P}_\Omega(UV^T + S)$$

9: **end while**

The value function $\phi(\tau)$

The factorization-based counterpart of the value function (1.11) is expressed as follows

$$\phi(\tau) = \min_{U,V,S} \|A - \mathcal{P}_\Omega(UV^T + S)\|_F \;\text{ subject to }\; \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \lambda\|S\|_1 \le \tau. \quad (1.14)$$

The applicability of the factorization (1.13) follows from Theorem 1.1 and Corollary 1.1 proved in [AKM+13] and listed below after specializing to our problem.

THEOREM 1.1 *[AKM+13, Theorem 4.1] Consider an optimization problem of the form*

$$\min_{Z \succeq 0} \quad h(Z) \quad \textit{subject to} \quad \kappa(Z) \le 0, \qquad \operatorname{rank}(Z) \le r, \quad (1.15)$$

where $Z \in \mathbb{R}^{n\times n}$ is positive semidefinite, and h, κ are continuous. Using the change of variable $Z = SS^T$, take $S \in \mathbb{R}^{n\times r}$, and consider the problem

$$\min_{S} \quad h(SS^T) \quad \textit{subject to} \quad \kappa(SS^T) \le 0 \quad (1.16)$$

Let $\bar{Z} = \bar{S}\bar{S}^T$, where $\bar{Z}$ is feasible for (1.15)*. Then $\bar{Z}$ is a local minimum of* (1.15) *if and only if $\bar{S}$ is a local minimum of* (1.16)*.*

COROLLARY 1.1 [AKM+13, Corollary 4.2] Any optimization problem of the form

$$\min_{X} \quad h(X) \quad \text{subject to} \quad \|X\|_* \le \tau, \quad \operatorname{rank}(X) \le r \quad (1.17)$$

where h is continuous, has an equivalent problem in the class of problems (1.15) characterized by Theorem 1.1.

In particular, Corollary 1.1 implies that when the factors U and V have a rank greater than or equal to the true rank of L_0, then a local minimizer of the value function in (1.9) with a least squares objective and the factorization in (1.13) as a constraint coincides with the solution of the same value function when the exact nuclear norm is in the constraint. This result follows from the semidefinite programming (SDP) characterization of the nuclear norm established in [RFP10].

In order to solve (1.14), we use a first-order projected gradient algorithm with a simple line search. The approach is similar to that developed in [vdBF08] without employing the Barzilai-Borwein line-search method. In every iteration of the algorithm, the gradient updates $(\widehat{U}, \widehat{V}, \widehat{S})$ of the variables (U, V, S) are projected onto the gauge constraint. The following section describes an efficient Newton root-finding algorithm for performing this projection.

Projecting onto the gauge constraint

A critical component for the success of the gauge optimization framework is the efficiency of the projection onto the gauge constraint. Therefore, we developed a fast algorithm that can efficiently solve the following projection problem

$$\min_{U,V,S} \frac{1}{2}\|U-\widehat{U}\|_F^2 + \frac{1}{2}\|V-\widehat{V}\|_F^2 + \frac{1}{2}\|S-\widehat{S}\|_F^2 \text{ subject to } \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \lambda\|S\|_1 \leq \tau. \tag{1.18}$$

The projection can be realized as shown in Proposition 1.1 by finding a scalar variable γ using the Newton method described in Algorithm 2. Clearly, the projection is only performed when the current iterates $(\widehat{U}, \widehat{V}, \widehat{S})$ violate the constraint, i.e., when $\frac{1}{2}\left(\|\widehat{U}\|_F^2 + \|\widehat{V}\|_F^2\right) + \lambda\|\widehat{S}\|_1 > \tau$.

PROPOSITION 1.1 The solution to the projection problem (1.18) is given by

$$\begin{aligned} S(\gamma\lambda) &= \mathcal{T}_{\gamma\lambda}(\widehat{S}) \\ U(\gamma) &= \tfrac{1}{\gamma+1}\widehat{U} \\ V(\gamma) &= \tfrac{1}{\gamma+1}\widehat{V}, \end{aligned} \tag{1.19}$$

where $\mathcal{T}_{\gamma\lambda}(\widehat{S}) = \text{sign}(\widehat{S}) \odot \max\{0, |\widehat{S}| - \gamma\lambda\}$ is the soft-thresholding operator, and γ is the scalar that satisfies the inequality $\frac{1}{2(\gamma+1)^2}\left(\|\widehat{U}\|_F^2 + \|\widehat{V}\|_F^2\right) + \lambda\|S(\gamma\lambda)\|_1 \leq \tau$. Moreover, Algorithm 2 specifies a Newton's method for computing γ.

PROOF 1.1 In order to evaluate the projection, we first form the Lagrange dual $\mathcal{L}(U, V, S, \gamma)$ of problem (1.18) with dual variable $\gamma \geq 0$,

$$\begin{aligned} \mathcal{L}(U,V,S,\gamma) &= \inf_{U,V,S} \tfrac{1}{2}\|U-\widehat{U}\|_F^2 + \tfrac{1}{2}\|V-\widehat{V}\|_F^2 + \tfrac{1}{2}\|S-\widehat{S}\|_F^2 \\ &\qquad + \gamma\left(\tfrac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \lambda\|S\|_1 - \tau\right) \\ &= \inf_{S}\left\{\tfrac{1}{2}\|S-\widehat{S}\|_F^2 + \gamma\lambda\|S\|_1\right\} \\ &\quad + \inf_{U,V}\left\{\tfrac{1}{2}\|U-\widehat{U}\|_F^2 + \tfrac{1}{2}\|V-\widehat{V}\|_F^2 + \tfrac{\gamma}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)\right\} - \gamma\tau \end{aligned}$$

The two infimums admit the closed form solutions shown in (1.19), where the expression for S is a soft-thresholding of the signal $\widehat{S}$ with threshold $\gamma\lambda$, and the symbol $\odot$ is an element wise Hadamard product. Consequently, the projection is performed by finding γ that satisfies $\frac{1}{2(\gamma+1)^2}\left(\|\widehat{U}\|_F^2 + \|\widehat{V}\|_F^2\right) + \lambda\|S(\gamma\lambda)\|_1 \leq \tau$.

Denote by T the support of the elements in $\widehat{S}$ with magnitude larger then $\gamma\lambda$, and let $k = |T|$ be the cardinality of the set T. The ℓ_1 norm of the soft-thresholded signal $S(\gamma\lambda)$ is then equal to $\|S(\gamma\lambda)\|_1 = \|\widehat{S}_T\|_1 - \gamma\lambda k$, where $\widehat{S}_T$ is a restriction of the signal $\widehat{S}$ to the support set T. The projection is then achieved using Newton's method to find the root of the function

$$f(\gamma) = \lambda(\|\widehat{S}_T\|_1 - \gamma\lambda k) + \frac{1}{2(\gamma+1)^2}\left(\|\widehat{U}\|_F^2 + \|\widehat{V}\|_F^2\right) - \tau = 0. \tag{1.20}$$

Algorithm 2 Newton's method for solving the factorized RPCA projection

Input Current iterates $\widehat{U}, \widehat{V}, \widehat{S}$, parameter λ, τ
Output Projected iterates U, V, S
Initialize $U = \widehat{U}$, $V = \widehat{V}$, $S = \widehat{S}$, $\gamma = 0$, $C = \left(\|U\|_F^2 + \|V\|_F^2\right)$
while $f(\gamma) > \tau$ **do**
  Compute the gradient: $g(\gamma) = -\lambda^2\|S\|_0 - (\gamma+1)^{-3}C$
  Update γ: $\gamma = \gamma - f(\gamma)/g(\gamma)$
  Soft-threshold the sparse component: $S = \text{sign}(\widehat{S}) \odot \max\{0, |\widehat{S}| - \gamma\lambda\}$
  Scale the low-rank factors: $U = \frac{1}{\gamma+1}\widehat{U}$, $V = \frac{1}{\gamma+1}\widehat{V}$
  Update the function value $f(\gamma) = \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \lambda\|S\|_1 - \tau$.
end while

The polar function

In determining the polar function, we follow the approach in [AKM+13] where the low rank matrix L is evaluated from the current estimates of the factors U and V, and the Newton update is treated in the nuclear norm of L sense as opposed to the Frobenius norm of the factor U and V. The benefit of this approach is that it allows us to reuse the Newton update shown in (1.12) for the standard RPCA problem.

1.3.2 Factorized ADM algorithm

The factorization approach can also be applied to the alternating direction method (ADM) for minimizing the augmented Lagrangian of problem (1.7) resulting in the factorized augmented Lagrangian shown below:

$$\mathcal{L}(U,V,S,Y) = \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \lambda\|S\|_1 + \langle Y, A - \mathcal{P}_\Omega(UV^T + S)\rangle + \frac{\mu}{2}\|A - \mathcal{P}_\Omega(UV^T + S)\|_F^2, \tag{1.21}$$

where $Y \in \mathbb{R}^{m\times n}$ is the dual multiplier, and μ is the augmented Lagrangian smoothing parameter. The above formulation (1.21) adds outlier robustness to the factorization-based ADM method for matrix completion presented in [RFP10] by introducing the ℓ_1 norm of the sparse component S into the augmented Lagrangian. On the other hand, (1.21) improves the stability of the reconstruction compared to the low rank matrix fitting (LMaFit) approach [WYZ12] which also employs low-rank factors by introducing the nuclear norm proxy into the augmented Lagrangian.

Algorithm 3 shows the alternating minimization steps of (ADM-FRMC) for optimizing $\mathcal{L}(U,V,S,Y)$. The non-factorized nuclear norm formulation of ADM algorithms apply a singular value thresholding of L

$$L_{j+1} = \mathcal{D}_{\mu^{-1}}\left(A - S_j - E_j + \mu^{-1}Y_j\right),$$

where E_j is only supported on the complement of the set Ω, and $\mathcal{D}_{\mu^{-1}}$ is the singular value thresholding operator that soft-thresholds the singular values of $A - S_j - E_j + \mu^{-1}Y_j$ by μ^{-1}. On the other hand, Algorithm 3 replaces the singular value thresholding operation with steps 5, 6, and 7 that require the inversion of $r \times r$ matrices. Consequently, when the rank of the factors is relatively small compared to the dimensions m and n, the ADM-FRMC algorithm can result in a significant reduction in computational complexity compared to singular value thresholding based ADM algorithms.

Algorithm 3 Alternating direction method for factorized robust matrix completion (ADM-FRMC).

1: **Input** Measurement matrix A, measurement operator $\mathcal{P}_\Omega$, regularization parameter λ, mismatch tolerance σ, smoothing parameter μ
2: **Output** Low rank factors U, V, and sparse component S
3: **Initialize** $V_0 =$ random $n \times r$ matrix, $Y_0 = 0$, $S_0 = 0$, $E_0 = 0$, $L_0 = 0$, $j = 0$
4: **while** $\|A - L_{j+1} - S_{j+1} - E_{j+1}\|_F > \sigma$ **do**
5: $\quad U_{j+1} = (Y_j + \mu(A - S_j - E_j))\, V_j(I_r + \mu V_j V_j^T)^{-1}$
6: $\quad V_{j+1} = (Y_j + \mu(A - S_j - E_j))^T\, U_{j+1}(I_r + \mu U_{j+1}U_{j+1}^T)^{-1}$
7: $\quad L_{j+1} = U_{j+1}V_{j+1}^T$
8: $\quad E_{j+1} = \mathcal{P}_{\Omega^c}\left[A - L_{j+1} + \mu^{-1}Y_j\right]$
9: $\quad S_{j+1} = \mathcal{P}_\Omega\left[\mathcal{T}_{\lambda\mu^{-1}}\left(A - L_{j+1} + \mu^{-1}Y_j\right)\right]$
10: $\quad Y_{j+1} = Y_j + \mu(A - L_{j+1} - S_{j+1} - E_{j+1})$
11: $\quad j = j + 1$
12: **end while**

1.3.3 Numerical Evaluation

To evaluate the performance of Gauge-FRMC and ADM-FRMC, we plot the reconstruction error versus runtime of the algorithms and compare the performance with respect to PSPG [AGM13] and ADMIP [AI14] that use Lanczos SVDs for fast computation of partial singular value decompositions*. We generate a synthetic data matrix A of size $m \times n$ that is composed of the sum of a rank r matrix L_0, and a sparse matrix S_0 with $mn/5$ non-zero entries. In all test cases, we set the rank of the factors U and V in FRMC equal to $1.2 \times r$ and choose $\lambda = /\sqrt{\max\{m, n\}}$. The tests were run in MATLAB on a 2.5 GHz Intel Core i5 machine. We define the relative error of variables (L_j, S_j) as follows

$$\mathrm{Err}(L_j, S_j) = \frac{\sqrt{\|L_j - L_0\|_F^2 + \|S_j - S_0\|_F^2}}{\sqrt{\|L_0\|_F^2 + \|S_0\|_F^2}}. \tag{1.22}$$

In the first test, we assume the entries of A are fully observed and set $m = n = 500$, and $r = 20$. The location of the non-zero entries of S_0 are randomly chosen from a Bernoulli distribution with probability $\frac{1}{5}$. The magnitudes of the non-zero entires in S_0 are drawn from a standard normal distribution and scaled such that in one case $\|S_0\|_\infty \leq \sqrt{2\|L_0\|_\infty}$ in order to blend the sparse components with the low rank signal, and in another case $\|S_0\|_\infty \leq \sqrt{200\|L_0\|_\infty}$ so that the sparse component constitutes large outliers. The performance evaluation for the two cases above are presented in Fig. 1.1 (a) and (b), respectively.

*ADMIP and PSPG codes available from http://www2.ie.psu.edu/aybat/codes. html

The performance on a larger dataset $m = n = 1500$, $r = 100$ with small outliers is also presented in Fig. 1.1 (c). A zoomed in frame that excludes Gauge-FRMC is shown in Fig. 1.1 (d) to highlight the difference between ADM-FRMC, ADMIP, and PSPG.

The figures show that ADM-FRMC and ADMIP have a comparable performance for both small and large outliers and for the small dataset. As the data size increases, Fig. 1.1 (d) shows that ADM-FRMC converges quickly to a 10^{-3} relative error point in one third of the time it takes ADMIP or PSPG. After that the convergence slows down as is typical of ADM type algorithms. On the other hand, the Gauge-FRMC algorithm is robust to large outliers but converges relatively slowly compared to the other algorithms primarily due to the slow projected gradient steps used to solve the value function $\phi(\tau)$. Using faster solvers for the value function should improve the overall speed of the algorithm. Finally, we point out that the PSPG algorithm fails to recover the correct signal when the variance of the outliers is too large. This could be the result of a bad choice for the smoothing parameter in the algorithm. We also tested the algorithms on the large dataset with large outliers and found the performance to be similar to the small dataset. Therefore, we excluded the corresponding figure from the chapter.

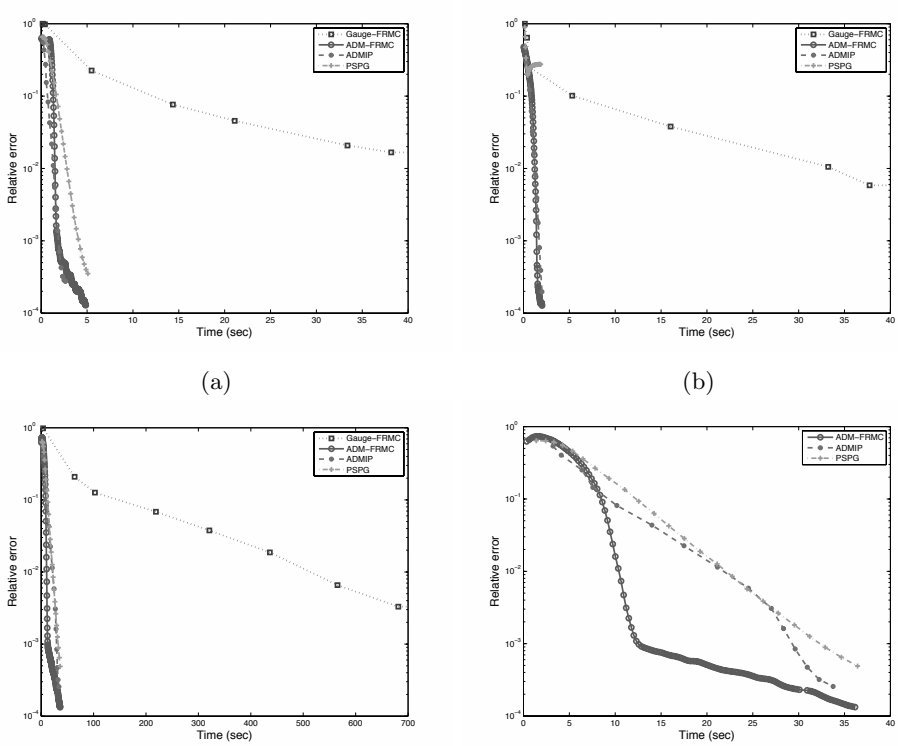
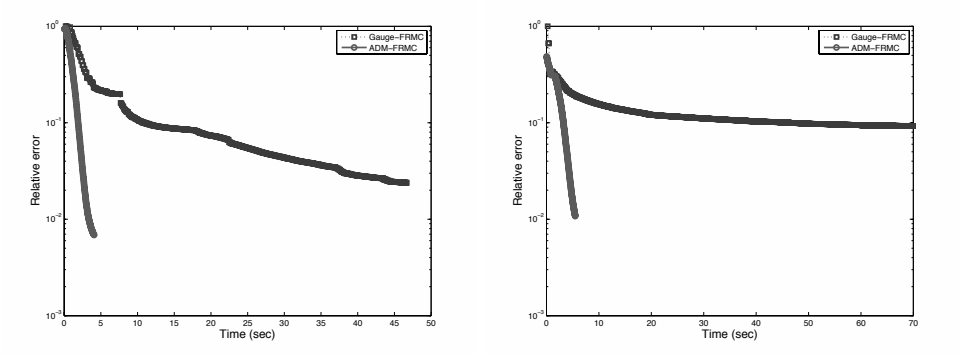


FIGURE 1.1: Performance evaluation on a synthetic dataset of size $m = n = 500$, $r = 20$ with (a) low variance sparse components, and (b) high variance sparse components. (c) Performance on a larger dataset $m = n = 1500$, $r = 100$ for small outliers. (d) Zoomed in version of (c) that excludes Gauge-FRMC to highlight the difference between ADM-FRMC, ADMIP, and PSPG.

In the second test, we only observe 50% of the entries in A and run the experiment on the $m = n = 500$, $r = 20$ data set. We exclude PSPG and ADMIP from the comparison

since the available codes do not support missing data entries. The results for small and large outliers are shown in Fig. 1.2 (a) and (b), respectively. The figures demonstrate that both algorithms correctly solve the problem, however, Gauge-FRMC is significantly slower.



(a) (b)

FIGURE 1.2: Performance evaluation on a synthetic dataset of size $m = n = 500$, $r = 20$ with 50% missing entries for (a) low variance sparse components, and (b) high variance sparse components.

1.4 Application to Video Background Subtraction

Video background subtraction algorithms can be classified into algebraic decomposition techniques [WGRM09, CLYW11, HBL11, HZBT13] and statistical motion flow techniques [SM98,SJK09,EE13,NHLM13]. Algebraic approaches generally model the background scene as occupying a low dimensional subspace. Moving objects in the video scene can then be modeled as sparse outliers that do not occupy the same subspace as the background scene. When the camera is stationary, the low dimensional subspace is also stationary and the video signal can be decomposed into a low rank matrix representing the background pixels as well as a sparse matrix representing the foreground moving objects. In this case, robust PCA has been shown to successfully segment the foreground from the background [WGRM09,CLYW11,WSB11]. When the camera is moving, the low rank structure no longer holds thus requiring adaptive subspace estimation techniques [HZBT13, QV11, MJ15] when the change in the subspace is smooth. However, this is rarely the case in real world videos which exhibit rotation, translation, and zoom among other types of motion. Therefore, a more robust approach performs global motion alignment prior to the matrix decomposition [MV14] when these significant motion distortions occur.

Once the video images are warped and aligned using global motion compensation, the images are vectorized and stacked into a matrix A of size $m \times n$, where m is the number of pixels in the video frame and n is the number of frames in a group of pictures (GOP). The warped images may contain large areas that have no content. Therefore, the problem can be posed as a robust matrix completion problem where a restriction operator $\mathcal{P}_\Omega$ identifies the set of pixels Ω that contains intensity values. Our objective then is to extract a low-rank component L from A that corresponds to the background pixels, and a sparse component S that captures the foreground moving objects in the scene.

1.4.1 Stationary background

For stationary background scenes, we apply the FRMC algorithm directly to the pixel domain, skipping the frame alignment. We test our algorithm on the Shopping Mall video sequence* which is composed of 1000 frames of resolution 320×256. Fig. 1.3 compares the qualitative separation performance of FRMC to that of the state-of-the-art algorithm GRASTA [HBL11]. The FRMC algorithm completes the recovery 7 to 8 times faster than GRASTA and results in a comparable separation quality. For a quantitative comparison, we plot the ROC curves of the two algorithms in Fig. 1.4. The curves show that GRASTA achieves a slightly better accuracy than FRMC, however, the computational cost is considerably higher.

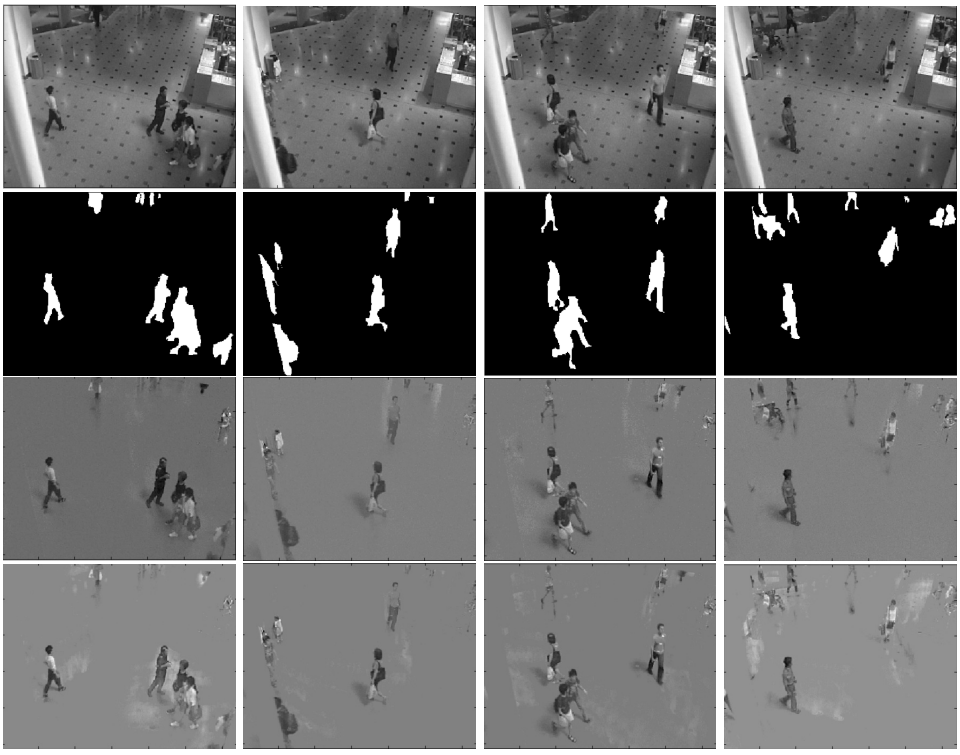


FIGURE 1.3: Background subtraction of four frames from the Shopping Mall sequence. Row one shows the original four frames. Row two shows the ground truth foreground objects. Row three shows the output of the GRASTA algorithm which required 389.7 seconds to complete. Row four shows the output of our FRMC algorithm running in batch mode and completing in 47.1 seconds. ©(2014) IEEE

1.4.2 Depth-weighted Group-wise PCA

In practical image sequences, the foreground objects (sparse components) tend to be clustered both spatially and temporally rather than evenly distributed. This observation led to the introduction of group sparsity into RPCA approaches by [HHM09, DYZ13, JWL13] pushing the sparse component into more structured groups. Our method utilizes the depth

*Available from:
http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

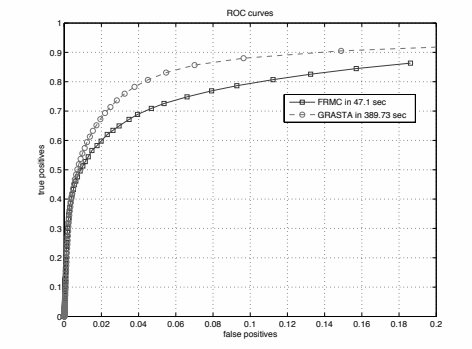


FIGURE 1.4: ROC curves comparing the stationary background subtraction performance between GRASTA and our FRMC algorithm. ©(2014) IEEE

map of the video sequence to define the group structures in a depth-weighted group-wise PCA (DG-PCA) method.

In order to deal with structured sparsity, we replace the l_1-norm in the factorized RPCA problem with a mixed $l_{2,1}$-norm defined as

$$\|S\|_{2,1} = \sum_{g=1}^{s} w_g \|S_g\|_2, \tag{1.23}$$

where S_g is the component corresponding to group g, $g = 1, ..., s$, and w_g's are weights associated to each group. The resulting problem is shown below:

$$\begin{aligned}(U, V, S, Y) = &\arg\min_{U,V,S,Y} \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 + \\ &\lambda\|S\|_{2,1} + \langle Y, A - UV^T - S\rangle + \frac{\mu}{2}\|A - UV^T - S\|_F^2.\end{aligned} \tag{1.24}$$

Algorithm 4 Depth-weighted group-wise PCA (DGPCA) algorithm

Require: Input data A, λ, μ, error tolerance τ, maximum iteration number N, and depth map D

Init: $i = 0$, U_i and $V_i \leftarrow$ random matrix, $G \leftarrow \mathcal{G}(D)$
repeat
  $U_{i+1} = (\mu(A - S_i) + Y_i)V_i(I_r + \mu V_i^T V_i)^{-1}$
  $V_{i+1} = (\mu(A - S_i) + Y_i)^T U_{i+1}(I_r + \mu U_{i+1}^T U_{i+1})^{-1}$
  $S_{i+1,g} = \mathcal{T}_{\lambda/\mu,g}(A_g - U_{i+1,g}V_{i+1,g}^T + \mu^{-1}Y_{i,g})$
  $E = A - U_{i+1}V_{i+1}^T - S_{i+1}$
  $Y_{i+1} = Y_i + \mu E$
  $i = i + 1$
until $i \geq N$ or $\|E\|_F \leq \tau$
return U, V, S, i and $\|E\|_F$

Algorithm 4 describes the proposed DG-PCA framework. In order to define pixel groups G using the depth map D, an operator $\mathcal{G}(D)$ segments the depth map into s groups using the following procedure. Suppose the depth level ranges from 0 to 255, a pixel with depth value d will be classified into group $g = \lfloor d/\frac{256}{s} \rfloor + 1$. Consequently, the input data A can be clustered into A_g with $g \in \{1, .., s\}$. Each A_g is composed of elements from A which

is marked into segment g. In the same way, U_g, V_g, and Lagrangian multiplier Y_g are also grouped.

Next, the operator $\mathcal{T}_{\lambda/\mu,g}$ in Algorithm 4 is a group-wise soft-thresholding, as shown below,

$$\mathcal{T}_{\lambda/\mu,g}(e_g) = \max(\|e_g\|_2 - w_g\lambda/\mu, 0)\frac{e_g}{\|e_g\|_2 + \epsilon}, \tag{1.25}$$

where $e_g = A_g - U_g V_g^T + \frac{1}{\mu}Y_g$, and ϵ is a small constant to avoid division by 0, and w_g defines group weights in (1.23). Since a foreground object has higher chances to be closer to the camera, i.e., to have a higher depth value than a background object, we propose the following equation to set group weights,

$$w_g = c^{1-\frac{d_g}{255}}, \tag{1.26}$$

where c is some constant, and d_g is the mean depth value of pixels in group g. w_g is equal to 1 for objects nearest to the camera, $d = 255$, and it is equal to c for objects farthest to the camera, $d = 0$. The choice of c controls the value of the threshold that permits foreground pixels to be selected based on their location in the depth field. Finally, after S_g is calculated for each group g, the sparse component S is obtained by summing up all S_g together.

Note that the above setup favors group structures where the foreground objects are closer to the camera. It is also possible within our framework to define the groups as the sets of pixels that are spatially connected and have a constant depth, or connected pixels where the spatial gradient of the depth is constant.

1.4.3 Global motion parametrization

In videos where the camera itself is moving, applying the FRMC algorithm directly to the video frames fails in segmenting the correct motion since the background itself is non-stationary. A non-stationary background does not live in a low rank subspace, therefore, we can only expect the algorithm to fail. Therefore, we first estimate the global motion parameters in the video in order to compensate for the camera motion. We then align the background and apply the FRMC algorithm to segment the moving objects.

Global motion estimation received a lot of attention from the research community during the development of the MPEG-4 Visual standard [FJ00]. One approach relates the coordinates (x_1, y_1) in a reference image I_1 to the coordinates (x_2, y_2) in a target image I_2 using an 8-parameter homography vector h such that

$$\begin{aligned} x_2 &= \tfrac{h_0+h_2x_1+h_3y_1}{1+h_6x_1+h_7y_1} \\ y_2 &= \tfrac{h_1+h_4x_1+h_5y_1}{1+h_6x_1+h_7y_1}. \end{aligned} \tag{1.27}$$

Given the homography vector $h = [h_0\ h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7]^T$ that relates two images, we can warp the perspective of image I_2 to match that of image I_1, thereby aligning the backgrounds of both images. However, estimating h from the raw pixel domain requires finding point-to-point matches between a subset of the pixels of the two images.

In order to compute h, we propose to use the horizontal and vertical motion vectors (m_x, m_y) that are readily available from the compressed video bitstream or during the encoding process. Here we assume that motion estimation is performed using the previous video frame as the only reference picture. The motion vectors provide relatively accurate point matches between the two images. Note, however, that we are only interested in matching pixels from the moving background. Therefore, we first compute a 32 bin histogram of each of the motion vectors m_x and m_y. Next, we extract a subset Λ of the indices of pixels

that exclude foreground objects in order to capture the motion of the background and fit the homography parameters to the background pixels alone. The homography parameter vector h is computed by solving the following least squares problem:

$$h = \arg\min_{\tilde{h}} \left\| \begin{bmatrix} x_{2\Lambda} \\ y_{2\Lambda} \end{bmatrix} - E\tilde{h} \right\|_2, \tag{1.28}$$

where $x_{2\Lambda} = x_{1\Lambda} + m_{x\Lambda}$, $y_{2\Lambda} = y_{1\Lambda} + m_{x\Lambda}$, and the matrix

$$E = \begin{bmatrix} \mathbf{1} & \mathbf{0} & x_{1\Lambda} & y_{1\Lambda} & \mathbf{0} & \mathbf{0} & -x_{2\Lambda}x_{1\Lambda} & -x_{2\Lambda}y_{1\Lambda} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & x_{1\Lambda} & y_{1\Lambda} & -y_{2\Lambda}x_{1\Lambda} & -y_{2\Lambda}y_{1\Lambda} \end{bmatrix},$$

where the subscript $_\Lambda$ indicates a restriction of the indices to the set Λ.

Next, we align the pictures relative to the perspective of the first frame in a GOP by sequentially warping the pictures using the coordinates of the previously warped frame $\hat{I}_1$ as reference to warp the coordinates of the next frame I_2 by applying (1.27). Finally, we note that due to the camera motion, the warped frames $\hat{I}_2$ generally occupy a larger viewing area relative to the reference frame I_2. Consequently, applying a forward map $f : (x_1, y_1) \to (\hat{x}_2, \hat{y}_2)$ often results in holes in the warped frame. To remedy this problem, we compute the reverse mapping $g : (\hat{x}_2, \hat{y}_2) \to (x_2, y_2)$ as a function of h and warp the frame to obtain $\hat{I}_2(\hat{x}_2, \hat{y}_2) = I_2(g(\hat{x}_2, \hat{y}_2))$. Fig. 1.5 illustrates the global motion compensation procedure applied to frame 26 of the Bus sequence.

For non stationary background sequences, we run our FRMC algorithm with global motion compensation on the reference video sequence Bus composed of 150 CIF resolution (352×288 pixels) frames*. The Bus sequence exhibits translation and zooming out. We use the HEVC test model (HM) 11 reference software* [BHO+13] to encode the sequence and run our FRMC with GME algorithm in batch mode with a batch size of 30 frames. The recovery performance is illustrated in Fig. 1.6. Notice how the recovered background expands and stretches relative to the original frames in order to cover the translation and zoom of the 30 frame GOP. Notice also how stationary foreground objects are successfully classified as part of the background subspace and are excluded from the segmented moving objects.

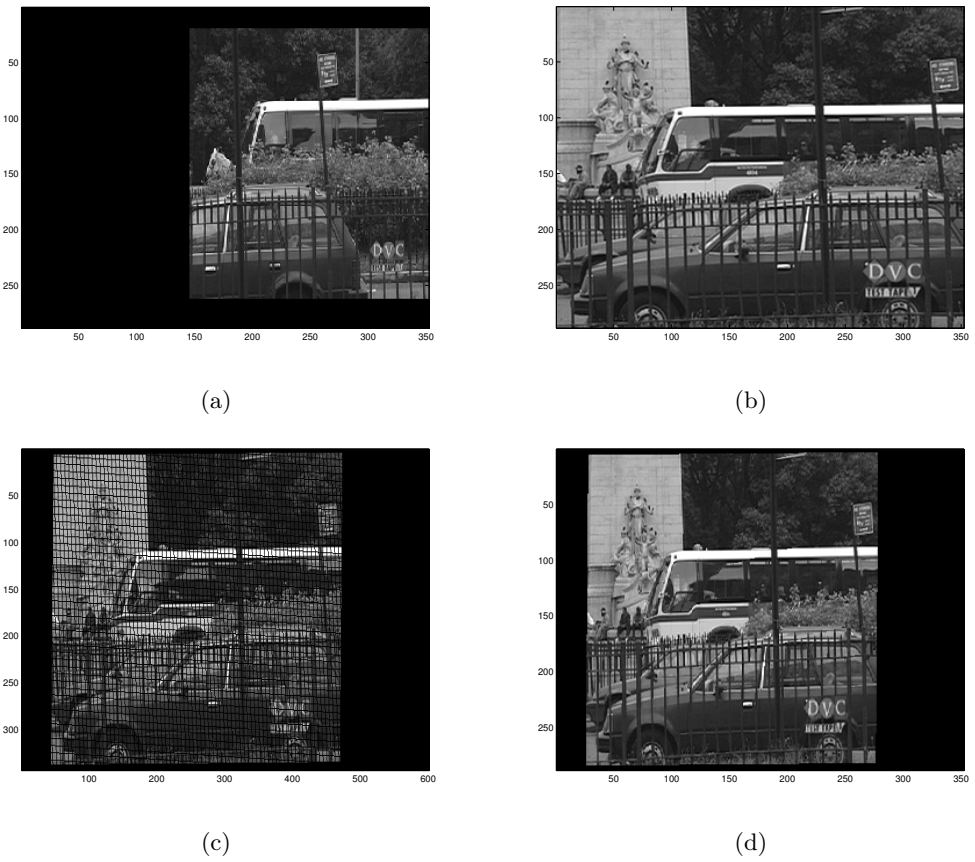
1.4.4 Depth-Enhanced Homography Model

The eight parameter homography model assumes planar motion. However, motion in a video sequence is generally not planar. Therefore, it is still very common to find large motion estimation errors in sequences that have a wide depth range. This could dramatically degrade the detection rate in the separation problem. Therefore, we propose a depth-enhanced homography model. Specifically, 6 new parameters related to depth are added, and we have $h = [h_1, ..., h_8, h_9, ..., h_{14}]^T$. Let z_1 and z_2 stand for the depth of the corresponding pixels, and the proposed depth-enhanced homography model is given as follows,

$$\begin{aligned} x_2 &= \frac{h_1 + h_3 x_1 + h_4 y_1 + h_9 z_1}{1 + h_7 x_1 + h_8 y_1}, \\ y_2 &= \frac{h_2 + h_5 x_1 + h_6 y_1 + h_{10} z_1}{1 + h_7 x_1 + h_8 y_1}, \\ z_2 &= \frac{h_{11} + h_{12} x_1 + h_{13} y_1 + h_{14} z_1}{1 + h_7 x_1 + h_8 y_1}. \end{aligned} \tag{1.29}$$

*Available from: http://trace.eas.asu.edu/yuv/

*Available from: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/



(a) (b)

(c) (d)

FIGURE 1.5: Example of the global motion compensation procedure used to align the backgrounds of images in a GOP. (a) First frame in the GOP aligned and scaled to its relative location. (b) Original frame 26 as input image I_2. (c) Frame 26 warped and aligned as $\hat{I}_2(\hat{x}_2, \hat{y}_2)$, (d) Warped and reverse mapped frame $\hat{I}_2(g(\hat{x}_2, \hat{y}_2))$. ©(2014) IEEE

Note in the above equation, depth value 0 means the object is far from the camera. A larger depth value means that the object is closer to the camera.

To evaluate the performance of the DG-PCA approach, we tested the separation on fr3/walking_rpy sequence from the "dynamic objects" category in the RGB-D benchmark provided by TUM [SEE+12]. The dataset contains dynamic objects with a low- to high-level global motion.

The accompanying depth in the dataset is captured by a Microsoft Kinect sensor and denoted by z. The depth map d is computed from z as follows

$$d = 255 \times \frac{\frac{1}{z} - \frac{1}{z_{\text{far}}}}{\frac{1}{z_{\text{near}}} - \frac{1}{z_{\text{far}}}}. \tag{1.30}$$

where z_{near} and z_{far} denote the nearest and farthest depth extracted from the raw depth data z.

In order to perform FG/BG separation, two consecutive video frames are first aligned using global motion compensation with and without the depth-enhancement. The aligned frames are then processed using FRMC and DG-PCA to separate the background L

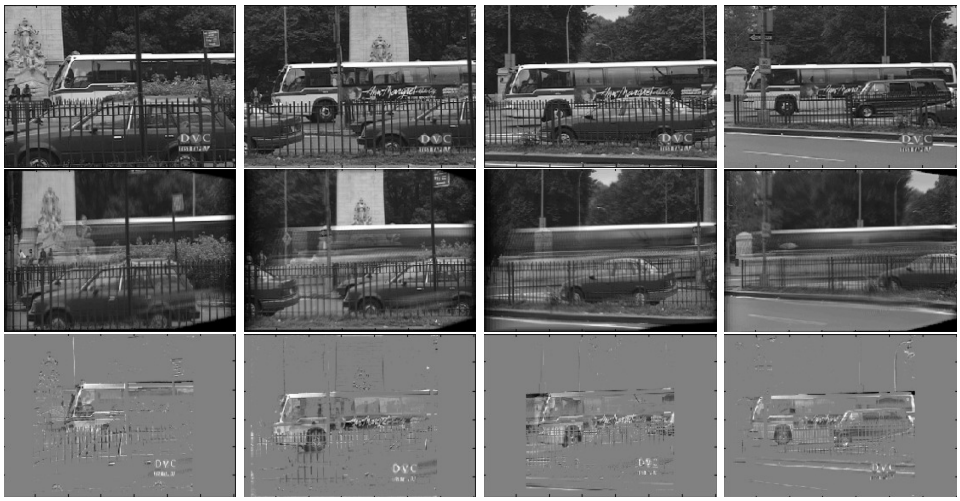


FIGURE 1.6: Background subtraction of four frames from the Bus sequence. Row one shows the original four frames. Row two shows the motion aligned and FRMC separated background relative to a 30 frame GOP. Row three shows the motion aligned and FRMC separated foreground. The total runtime for global motion compensation and background subtraction of 150 frames took 19.8 seconds. ©(2014) IEEE

from the foreground S. The rank of the background is set equal to 2. We used sequence fr3/walking_static with minor camera motion to tune the algorithm parameters and then run tests on the other three sequences with higher motion. We set the parameter $\lambda = 0.05(\|e\|_2/\sqrt{\text{size}(A)}) \times \mu$, where a constant 0.05 is selected empirically to limit the iteration step for a finer background subtraction. When updating the group-wise sparse component in Algorithm 4, we use $\lambda_g = \lambda\sqrt{\text{size}(A_i)}$ instead of the image level λ. This scaling in λ_g ensures the dependence on the size of the group since the thresholding operation is applied to the ℓ_2 norm of the group instead of the magnitudes of individual pixels. Moreover, we set $c = 10$ in (1.26). We set the number of groups $s = 32$ since we found to significant difference in the performance of the algorithm when s was varied in the range $[16, 32]$. We also denoised the depth-based grouping map G using a 5×5 median filter.

Fig. 1.7 shows 5 snapshots across fr3/walking_rpy with 910 frames at VGA resolution which has the greatest global motion in the dataset. The figures show that the two DG-PCA methods (row 4 and 5) produce a much cleaner foreground segmentation compared to FRMC (row 3). For example, in the third snapshot, the person walking at a further distance behind the office partition can also be detected successfully by DG-PCA. Comparing DG-PCA without depth-enhanced global MC (row 4) and with depth-enhanced global MC (row 5), shows that the depth-enhanced homography model helps improve the motion alignment compared to the conventional homography model.

1.5 Conclusion

We developed a factorization-based approach to solving the robust matrix completion that replaces the solution over the low rank matrix with its low rank factors. We showed how the factorization approach can be applied in a gauge optimization framework resulting in the Gauge-FRMC algorithm, and in an alternating direction method (ADM) of multiplier framework resulting in the ADM-FRMC algorithm. Performance evaluation of the two algorithms showed that while both algorithms correctly solve the robust matrix completion

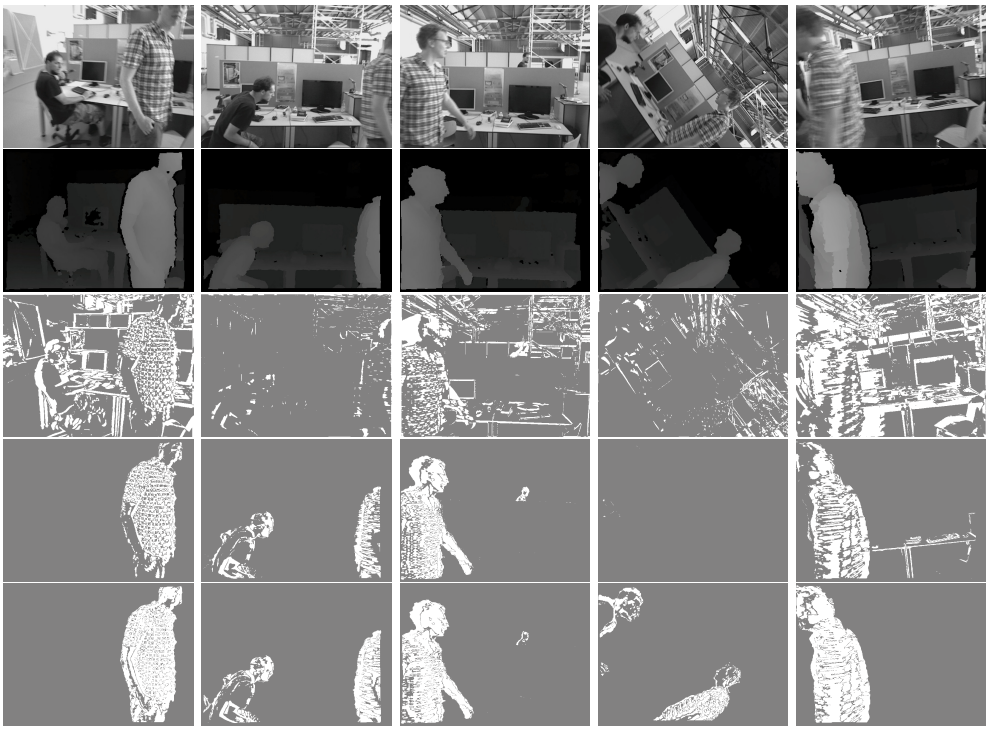


FIGURE 1.7: Performance evaluation. Row 1: color images. Row 2: depth maps. Row 3: Factorized RPCA. Row 4: DG-PCA w/o depth-refined global MC. Row 5: DG-PCA with depth-refined global MC.

problem, the ADM approach enjoys faster convergence than the gauge optimization approach. Moreover, the speed of convergence and accuracy of the ADM-FRMC algorithms matches and in some cases exceeds that of state of the art algorithms. The main bottleneck for the gauge minimization approach comes from the use of a first-order projected gradient step to evaluate the value function.

In the second part of the chapter, we focused on video background subtraction as an application to factorized robust matrix completion. With the help of motion vector information available from a coded video bitstream, we showed that our framework is capable of subtracting the background from stationary and moving camera sequences. We also extended our model to incorporate scene depth information by assigning group structures to the sparse data outliers corresponding to foreground objects. Finally, we demonstrated that incorporating depth information into the problem formulation, we were able to improve the foreground/background separation.

References

1. A. Y. Aravkin, J. Burke, and M. P. Friedlander. Variational properties of value functions. 23(3):1689–1717, 2013.
2. N. Aybat, D. Goldfarb, and S. Ma. Efficient algorithms for robust and stable principal component pursuit. *accepted in Computational Optimization and Applications*, 2013.
3. N. S. Aybat and G. Iyengar. An alternating direction method with increasing penalty for

stable principal component pursuit. *submitted to Computational Optimization and Applications, http://arxiv.org/abs/1309.6553*, 2014.
4. A.Y. Aravkin, R. Kumar, H. Mansour, B. Recht, and F.J. Herrmann. A robust svd free approach to matrix completion, with applications to interpolation of large scale data. *preprint. http://arxiv.org/abs/1302.4886*, 2013.
5. B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, Y. K. Wang, and T. Wiegand. *High Efficiency Video Coding (HEVC) text specification draft 10*. JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Jan. 2013.
6. Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, March 2010.
7. E. J. Candès, X. Li, Ma Y., and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.
8. E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theor.*, 56(5):2053–2080, May 2010.
9. Wei Deng, Wotao Yin, and Yin Zhang. Group sparse optimization by alternating direction method. In *SPIE Optical Engineering+ Applications*, pages 88580R–88580R. International Society for Optics and Photonics, 2013.
10. A. Elqursh and A. M. Elgammal. Online moving camera background subtraction. In *ECCV*, 2013.
11. Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *In Proceedings of the 2001 American Control Conference*, pages 4734–4739, 2001.
12. Dufaux F. and Konrad J. Efficient, robust, and fast global motion estimation for video coding. *IEEE Transactions on Image Processing*, 9(3):497–501, 2000.
13. M. P. Friedlander, I. Macêdo, and T. K. Pong. Gauge optimization and duality. *SIAM Journal on Optimization*, 24(4):1999–2022, 2014.
14. J. He, L. Balzano, and J. C. S. Lui. Online robust subspace tracking from partial information. *preprint, http://arxiv.org/abs/1109.3827*, 2011.
15. Junzhou Huang, Xiaolei Huang, and Dimitris Metaxas. Learning with dynamic group sparsity. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 64–71. IEEE, 2009.
16. J. He, D. Zhang, L. Balzano, and T. Tao. Iterative grassmannian optimization for robust image alignment. *preprint, http://arxiv.org/abs/1306.0404*, 2013.
17. Zhangjian Ji, Weiqiang Wang, and Ke Lv. Foreground detection utilizing structured sparse model via l1, 2 mixed norms. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 2286–2291. IEEE, 2013.
18. Kiryung Lee and Yoram Bresler. Admira: Atomic decomposition for minimum rank approximation. *CoRR*, abs/0905.0044, 2009.
19. H. Mansour and X. Jiang. A robust online subspace estimation and tracking algorithm. In *to appear in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.
20. H. Mansour and A. Vetro. Video background subtraction using semi-supervised robust matrix completion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6528–6532. IEEE, 2014.
21. Srebro N. *Learning with matrix factorizations*. PhD thesis, Cambridge, MA, USA, 2004. AAI0807530.
22. M. Narayana, A. Hanson, and E. Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *ICCV*, 2013.
23. Chenlu Qiu and Namrata Vaswani. Reprocs: A missing link between recursive robust pca and recursive sparse recovery in large but correlated noise. *CoRR*, abs/1106.3286,

2011.

24. Benjamin Recht. A simpler approach to matrix completion. *J. Mach. Learn. Res.*, 12:3413–3430, December 2011.
25. B. Recht, M Fazel, and P.A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.
26. B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 2013.
27. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
28. Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, 2009.
29. J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998.
30. E. van den Berg and M Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.
31. Ewout van den Berg and Michael P. Friedlander. Sparse optimization with least-squares constraints. *SIAM J. Optimization*, 21(4):1201–1229, 2011.
32. J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in Neural Information Processing Systems 22*, 2009.
33. A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk. Sparcs: Recovering low-rank and sparse matrices from compressive measurements. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1089–1097. 2011.
34. Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.