# Robust Algorithms for Low-Rank and Sparse Matrix Models

by

Brian E. Moore

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2018

Doctoral Committee:

Associate Professor Raj Rao Nadakuditi, Chair
Professor Jeffrey A. Fessler
Professor Alfred O. Hero III
Assistant Professor Shuheng Zhou

Brian E. Moore

brimoor@umich.edu

ORCID iD: 0000-0001-7914-1794

For Meriah.

# ACKNOWLEDGEMENTS

I would first like to thank my advisor Prof. Raj Nadakuditi. Thank you for your guidance, your support, and your endless supply of creative research ideas. You gave me direction when I needed it, empowered me to pursue projects when the opportunities arose, and exposed me to many interesting concepts that I would not have encountered otherwise. I admire your enthusiasm for your work and your "why not?" mentality; this mindset has rubbed off on me—both in research and life—and compelled me to pursue my passions with confidence. I am especially grateful for our flexible working relationship that allowed me to work remotely over the last year as I moved to be with my wife. It was a pleasure to have been in your group, and I look forward to continued work on our untapped projects!

I would also like to thank the rest of my committee—Prof. Jeff Fessler, Prof. Al Hero, and Prof. Shuheng Zhou—for their mentorship and invaluable feedback throughout my dissertation work. Prof. Fessler, thank you for adopting me as a de facto member of your research group. Your passion for your work is infectious, and I greatly admire your unique blend of professionalism and attentiveness to your students. I learned much from you about optimization, and I appreciate your input on our numerous collaborations. You have significantly and positively impacted this thesis. Prof. Hero, thank you as well for your valuable guidance and feedback on my work. Your vast knowledge of signal processing and data science was a great resource. I even learned from you during my qualification exam when you were supposed to be evaluating me! Prof. Zhou, thank you for your flexibility and willingness to join my committee. I particularly enjoyed our conversations about statistical signal estimation, which provided some valuable context for our work on robust PCA theory.

Thank you to Sai, Andrew, Chen, and Raj Tejas for our enjoyable and fruitful collaborations. Working with you was truly a pleasure, and I am proud of the research we produced together. Thank you also to Curtis, Nick, Arvind, Madison, Gopal, David, and the other members of the Nadakuditi and Fessler research groups for the many stimulating discussions about research, coursework, current events, philosophy, logic puzzles, and general socializing that made our time in the office enjoyable.

Lastly—and most importantly—thank you to my wife Meriah for your unconditional love and support. Thank you for enduring a temporary long distance relationship as we pursued our educations; I am thrilled for our future together. This work is dedicated to you.

# TABLE OF CONTENTS

**CHAPTER**

# LIST OF FIGURES

# LIST OF TABLES

**Table**

# LIST OF APPENDICES

**Appendix**

# ABSTRACT

Data in statistical signal processing problems is often inherently matrix-valued, and a natural first step in working with such data is to impose a model with structure that captures the distinctive features of the underlying data. Under the right model, one can design algorithms that can reliably tease weak signals out of highly corrupted data. In this thesis, we study two important classes of matrix structure: low-rankness and sparsity. In particular, we focus on robust principal component analysis (PCA) models that decompose data into the sum of low-rank and sparse (in an appropriate sense) components. Robust PCA models are popular because they are useful models for data in practice and because efficient algorithms exist for solving them.

This thesis focuses on developing new robust PCA algorithms that advance the state-of-the-art in several key respects. First, we develop a theoretical understanding of the effect of outliers on PCA and the extent to which one can reliably reject outliers from corrupted data using thresholding schemes. We apply these insights and other recent results from low-rank matrix estimation to design robust PCA algorithms with improved low-rank models that are well-suited for processing highly corrupted data. On the sparse modeling front, we use sparse signal models like spatial continuity and dictionary learning to develop new methods with important adaptive representational capabilities. We also propose efficient algorithms for implementing our methods, including an extension of our dictionary learning algorithms to the online or sequential data setting. The underlying theme of our work is to combine ideas from low-rank and sparse modeling in novel ways to design robust algorithms that produce accurate reconstructions from highly undersampled or corrupted data. We consider a variety of application domains for our methods, including foreground-background separation, photometric stereo, and inverse problems such as video inpainting and dynamic magnetic resonance imaging.

# CHAPTER I

# Introduction

Data in statistical signal processing problems is often inherently matrix-valued. For example, in the canonical Netflix problem, one is interested in completing a large, highly undersampled matrix whose rows represent users, columns represent movies, and entries represent movie ratings. A natural first step in working with matrix-valued data is to impose some structure to make the desired task (estimation, detection, etc.) tractable. In this thesis, we focus on two important classes of matrix structure: low-rankness and sparsity.

## 1.1 Low-Rank and Sparse Matrix Models

Low-rank matrices arise in statistical signal processing problems for many reasons. In practice, low-rankness allows for dimensionality reduction, which is often an essential preprocessing step when working with high dimensional data. Low-rankness is also important from a theoretical perspective because it implies that the data has some inherent redundancy that can be leveraged to reliably tease weak signals out of highly corrupted data. Perhaps the most algorithm for low-rank models is principal component analysis (PCA). In PCA, one estimates the latent low-rank structure of a high-dimensional dataset by computing the subspace spanned by the first few singular vectors of the data. Obtaining an accurate estimate of the underlying subspace is critically important to the success of subsequent inferential tasks. Although PCA is stable in the presence of relatively small noise, it is well-known that even a few large outliers in the data can cause PCA to breakdown completely. In this thesis, we contribute both theoretical understanding of the breakdown of PCA in the presence of outliers and algorithms to avoid this breakdown in practice.

Sparsity is another fundamental property of many datasets. Data may exhibit sparsity in many forms. It may simply contain few non-zero elements, i.e., have sparse support; it may exhibit spatial or temporal continuity, i.e., be sparse in a total variation sense;, or it may be sparse with respect to a more general fixed or adaptive transformation, i.e., as in dictionary learning. Each of these sparsity models will play an important role in this thesis. Depending on the application, the sparsity of a dataset can be an asset or a liability. For example, in conventional PCA, sparse corruptions are a nuisance that conspire to destroy the subspace estimate. However, in other applications—such as foreground-background separation and dynamic medical imaging, which we will consider in this thesis—sparsity may capture the critical dynamic features of the dataset that have physical meaning and importance.

Recently there has been great interest in methods that decompose data into low-rank and sparse (in an appropriate sense) components. These so-called robust PCA models are popular because they are useful models for data in practice and because simple algorithms exist for solving them. The bulk of this thesis is dedicated to developing new robust PCA algorithms that advance the state-of-the-art in several key aspects. On the low-rank front, we apply theoretical results from low-rank matrix estimation to design robust PCA algorithms with improved low-rank models. On the sparsity front, we develop new methods that exploit sparse signal models like spatial continuity and adaptive transform sparsity to achieve best-in-class results on practical problems in computer vision and inverse problems. The underlying theme of this work is to combine ideas from low-rank and sparse modeling in novel ways to design robust algorithms that produce accurate reconstructions from highly undersampled or corrupted data.

## 1.2 Contributions

The rest of this thesis is organized as follows. Chapter II briefly provides some common groundwork and motivation for our investigation, but the subsequent chapters are intended to be mostly self-contained.

In Chapter II, we present some background on the problem of estimating a low-rank matrix corrupted by noise. This fundamental problem underlies all of the robust PCA methods discussed in this thesis, because each algorithm uses an alternating minimization scheme where one step of the problem can be thought of as a low-rank matrix denoising step. We review the prevailing low-rank estimation methods in the literature, and then we present some recent theoretical results from random matrix theory on optimal low-rank matrix estimation that culminates in OptShrink, a recent data-driven low-rank matrix estimator that we employ throughout this thesis.

In Chapter III, we study the robust PCA problem of reliably recovering a low-rank signal matrix from a signal-plus-noise-plus-outliers matrix. We begin by analytically characterizing the effect of outliers on the data matrix, and we discuss why recent classical robust PCA algorithms will produce suboptimal low-rank matrix estimates in the presence of noise. Then we propose a new robust PCA algorithm that leverages OptShrink to improve low-rank matrix estimation quality. We demonstrates the state-of-the-art performance of our proposed method on a background subtraction task from computer vision and highly accelerated dynamic magnetic resonance imaging (MRI) reconstruction. This chapter is based on [6, 7].

In Chapter IV, we extend our work on background subtraction from Chapter III to the general case of foreground-background separation on freely moving camera video with dense and sparse corruptions. We propose a method that can produce a panoramic background component that automatically stitches together corrupted data from partially overlapping frames to reconstruct the full field of view, and we use a weighted total variation framework that enables our method to reliably decouple the true foreground of the video from sparse corruptions. We perform extensive numerical experiments on both corrupted static and moving camera video that demonstrate the state-of-the-art performance of our proposed method compared to existing methods both in terms of foreground and background estimation accuracy. This chapter is based on [8, 9].

Next, we take a theoretical aside and consider the problem of recovering a low-rank matrix corrupted by random noise and outliers in Chapter V. Motivated by the sparse estimation literature, we consider outlier rejection schemes that apply hard or soft thresholding, respectively, to the elements of the data matrix. We analyze the accuracy of the low-rank matrix estimated by applying PCA to the outlier-rejected data by comparing it to an oracle estimator that replaces the known outlier-corrupted entries of the data matrix with zeros. Our analysis reveals a surprising result: in the dense outlier regime, the hard thresholding-based estimator achieves oracle accuracy while the soft thresolding-based estimator breaks down completely. This is an interesting result because in the context of sparse signal estimation, hard and soft thresholding both exhibit similar performance. This chapter is based on [10, 11].

In Chapter VI, we shift our focus to sparse signal models based on adaptive dictionary learning. Traditional dictionary learning problems are non-convex and NP-hard, and the usual alternating minimization approaches for learning are often expensive and lack convergence guarantees. In this chapter, we investigate efficient methods for learning synthesis dictionaries with low-rank atoms. We propose a block coordinate descent algorithm for our dictionary learning model that involves efficient updates, and we provide a convergence analysis of the proposed method. Finally, we provide numerical experiments that demonstrate the usefulness of our schemes for highly accelerated dynamic MRI reconstruction and video inpainting. This chapter is based on [12].

We extend our structured dictionary learning framework to the online setting in Chapter VII. In particular, we adapt our model from Chapter VI to process streaming images from a dynamic image sequence in minibathces. At each step, we jointly estimate the underlying images, a dictionary that adapts to all previous data, and the associated sparse coefficients of the model. Our proposed online algorithm involves efficient memory usage and simple and efficient updates of the images, low-rank atoms, and sparse coefficients. Our numerical experiments demonstrate the compelling performance of our algorithm in inverse problem settings, including video reconstruction from noisy, subsampled pixels and highly accelerated dynamic MRI reconstruction. This chapter is based on [13–15].

In Chapter VIII, we integrate our previous work on robust PCA and dictionary learning models into a single low-rank and adaptive sparse framework for highly accelerated dynamic imaging applications. Our model decomposes the temporal image sequence into a low-rank component and a component whose spatiotemporal (3D) patches are sparse in an adaptive dictionary domain. We investigate various formulations and efficient methods for jointly estimating the underlying dynamic signal components and the spatiotemporal dictionary from limited measurements. Our numerical experiments once again demonstrate the promising performance our proposed methods for highly accelerated dynamic MRI reconstruction. This chapter is based on [16, 17].

We return to computer vision in Chapter IX, where we apply adaptive dictionary learning models to the problem of robust photometric stereo. Photometric stereo is a method for reconstructing the normal vectors of an object from a set of images of the object under varying lighting conditions. Classical photometric stereo relies on a diffuse surface model that cannot handle objects with complex reflectance patterns, and it is sensitive to non-idealities in the images. In this chapter, we leverage our dictionary learning models from Chapter VI to develop three new models for photometric stereo that are robust to corruptions in the images. Specifically, we propose a preprocessing step that utilizes dictionary learning to denoise the images. We also present a model that applies dictionary learning to regularize and reconstruct the normal vectors from the images under the classic Lambertian reflectance model. We then generalize the latter model to explicitly model non-Lambertian objects. This chapter is based on [18, 19].

Finally, in Chapter X, we apply our adaptive dictionary learning framework the problem of robustly reconstructing a surface from imperfect estimates of its normal vectors. Our model simultaneously integrates the gradient fields while sparsely representing the spatial patches of the reconstructed surface in an adaptive dictionary domain. We show that our formulation learns the underlying structure of the surface, effectively acting as an adaptive regularizer that enforces a smoothness constraint on the reconstructed surface. We revisit the photometric stereo problem from Chapter IX by applying our algorithm to robustly reconstruct a surface from photometric stereo normal vectors, which completes the story of performing robust surface reconstruction from possibly corrupted images of an object. This chapter is based on [20].

# CHAPTER II

# Background

This chapter provides a brief background on low-rank matrix models and introduces some recent results on low-rank matrix estimation from the random matrix theory literature that will play an important role throughout this thesis.[1]

## 2.1 Low-Rank Matrix Models

Suppose we have an arbitrary matrix $\widetilde{X}$ that contains—in a vague sense for now—a low-rank matrix $L$ of known rank $r$. One of the most basic estimators of $L$ is the truncated singular value decomposition (TSVD) of $\widetilde{X}$:

$$\mathbf{TSVD}_r(\widetilde{X}) \;:=\; \sum_{i=1}^{r} \widetilde{\sigma}_i \widetilde{u}_i \widetilde{v}_i^H, \tag{2.1}$$

where $\widetilde{X} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^H$ is the SVD of $\widetilde{X}$ with singular values $\{\widetilde{\sigma}_i\}$. The TSVD has many interpretations. For example, it is closely related to the ubiquitous principal component analysis (PCA) [21, 22], where one computes the rank-$r$ subspace in which the data $\widetilde{X}$ has maximum variance. Alternatively, the well-known Eckart-Young theorem [23] asserts that $\mathbf{TSVD}_r(\widetilde{X})$ is the closest rank-$r$ matrix to $\widetilde{X}$ in the Frobenius norm sense. In other words, it is the solution to the rank-constrained optimization problem

$$\begin{aligned} \min_{X} \quad & \|\widetilde{X} - X\|_F^2 \\ \text{s.t.} \quad & \mathbf{rank}(X) \leq r. \end{aligned} \tag{2.2}$$

Thanks in part to the recent explosion of convex optimization, another popular

---

[1] Random matrix theory is a fascinating and deeply rooted area of mathematics. Here we present a brief selection of results that are relevant to this thesis, but each topic merits considerable further attention from an interested reader.

tool for estimating $L$ is the singular value thresholding (SVT) estimator [24]:

$$\mathbf{SVT}_\tau(\widetilde{X}) := \sum_i (\widetilde{\sigma}_i - \tau)_+ \widetilde{u}_i \widetilde{v}_i^H, \tag{2.3}$$

where $\tau > 0$ is a chosen parameter and $(y)_+ := \max(y, 0)$. The SVT estimator arises as the solution to the convex optimization problem [25]:

$$\arg\min_X \ \frac{1}{2}\|\widetilde{X} - X\|_F^2 + \tau\|X\|_\star, \tag{2.4}$$

where $\|X\|_\star = \sum_i \sigma_i(X)$ is the nuclear norm (sum of singular values) of $X$. The nuclear norm can be interpreted as the tightest convex relaxation of the rank penalty $\mathbf{rank}(X)$, and this fact is often invoked when convex relaxations of a nonconvex problem like (2.2) are proposed and solved in practice.

The TSVD and SVT estimators are two of the many possible methods for estimating a low-rank matrix from noisy observations. However, a natural question to ask is what is the quality of these estimators? And, in particular, is there an *optimal* strategy for estimating a low-rank matrix buried in noise? In order to formulate these questions as well-defined problems, one can adopt a random matrix theoretic framework where the matrix $\widetilde{X}$ is modeled as the sum of a deterministic low-rank matrix $L$ and a noise matrix $X$ whose elements are random variables. It turns out that, in this random matrix setting, one can in fact derive a provably optimal method (OptShrink) [26] for estimating $L$ from an observation $\widetilde{X}$. We describe this estimator in Section 2.3, but first we review some relevant results from random matrix theory literature on the singular values and vectors of perturbations of low-rank matrices.

## 2.2 Random Perturbations of Low-Rank Matrices

Consider the random matrix model

$$\widetilde{X}_n = L_n + X_n, \tag{2.5}$$

where $\widetilde{X}_n$ is an $m \times n$ observed data matrix, $X_n$ is additive random noise matrix, and $L_n = \sum_{i=1}^r \theta_i u_i v_i^H$ is a deterministic rank-$r$ matrix with singular values $\theta_i$ and singular vectors $\{u_i, v_i\}$, respectively. We denote by $\widetilde{X}_n = \sum_k \widetilde{\sigma}_k \widetilde{u}_k \widetilde{v}_k^H$ the SVD of $\widetilde{X}_n$ with singular values $\widetilde{\sigma_k}$ and singular vectors $\{\widetilde{u}_k, \widetilde{v}_k\}$, respectively.

Intuitively, if the noise $X_n$ is relatively "weak", one expects the leading $r$ singular values and vectors of $\widetilde{X}_n$ to be relatively close to the corresponding components of $L$, while one expects the singular vectors to become uncorrelated as the relative strength of the noise increases. Theorems II.1 and II.2 formalize this intuition in an asymptotic regime as $m, n \to \infty$.

**Theorem II.1.** *(Singular Value Phase Transition [27]). Fix a sequence of rank-r matrices $L_n$ with non-zero singular values $\theta_1, \dots, \theta_r$, a constant $c \in (0, 1]$, and suppose that $X_n$ is drawn from a random noise model whose empirical singular value density $\mu_{X_n}$ converges almost surely weakly as $m, n \to \infty$ such that $m/n \to c \in (0, 1]$ to a non-random probability measure $\mu_X$ supported on a single interval $[a, b]$. In addition, suppose that the extreme singular values of $X_n$ converge almost surely to the endpoints of the spectral support. Then, the extreme singular values of $\widetilde{X}_n$ exhibit the following asymptotic behavior. For each $1 \le i \le r$:*

$$\widetilde{\sigma}_i \xrightarrow{a.s.} \begin{cases} D_{\mu_X}^{-1}(1/\theta_i^2) & \text{if } \theta_i^2 > 1/D_{\mu_X}(b^+) \\ b & \text{otherwise,} \end{cases} \tag{2.6}$$

*where*

$$D_{\mu_X}(z) := \left[ \int \frac{z}{z^2 - t^2} \, d\mu_X(t) \right] \times \left[ \frac{1-c}{z} + c \int \frac{z}{z^2 - t^2} \, d\mu_X(t) \right] \tag{2.7}$$

*is the $D$-transform of the measure $\mu_X$, and*

$$D_{\mu_X}(b^+) := \lim_{z \searrow b} D_{\mu_X}(z). \tag{2.8}$$

**Theorem II.2.** *(Singular Vector Phase Transition [27]). Under the conditions of Theorem II.1, the extreme singular vectors $\{\widetilde{u}_i, \widetilde{v}_i\}$ of $\widetilde{X}_n$ drawn from the model* (2.5) *exhibit the following behavior as $m, n \to \infty$. For each $1 \le i \le r$ such that $\theta_i^2 > 1/D_{\mu_X}(b^+)$, we have*

$$|\langle \widetilde{u}_i, u_i \rangle \langle \widetilde{v}_i, v_i \rangle| \xrightarrow{a.s.} -2 \frac{D_{\mu_X}^{3/2}(\rho_i)}{D'_{\mu_X}(\rho_i)}, \tag{2.9}$$

*where $\rho_i := D_{\mu_X}^{-1}(1/\theta_i^2)$ is the limit of $\widetilde{\sigma}_i$ from Theorem II.1.*

Theorems II.1 and II.2 characterize the asymptotic behavior of the extreme singular values and vectors of $\widetilde{X}_n$ drawn from the model (2.5) in terms of the $D$-transform of the limiting noise singular value distribution. In particular, Theorem II.1 identifies a phase transition phenomenon around a critical point $\kappa := 1/\sqrt{D_{\mu_X}(b^+)}$ that depends only on the limiting singular value noise distribution, $\mu_X$. If $\theta_i > \kappa$, then the $i$th singular value of $\widetilde{X}_n$ will separate from the bulk noise spectrum and converge to a deterministic location $\rho_i = D_{\mu_X}^{-1}(1/\theta_i^2)$ that depends only on the limiting noise distribution $\mu_X$ and the signal strength $\theta_i$. However, if $\theta_i \leq \kappa$, then the $i$th largest singular value of $\widetilde{X}_n$ remains in the bulk noise spectrum.

Associated with each leading singular value, Theorem II.2 asserts that the perturbed singular vectors $\widetilde{u}_i$ and $\widetilde{v}_i$ contain a deterministic amount of information about the latent singular vectors $u_i$ and $v_i$. Indeed, one can interpret the quantity $\alpha_i := |\langle \widetilde{u}_i, u_i \rangle \langle \widetilde{v}_i, v_i \rangle| \in [0, 1]$ as a measure of the accuracy of $\widetilde{u}_i$ and $\widetilde{v}_i$ with respect to $u_i$ and $v_i$, since $\alpha_i = 1$ if and only if $\widetilde{u}_i = u_i$ and $\widetilde{v}_i = v_i$, and $\alpha_i = 0$ when either pair of singular vectors are orthogonal.

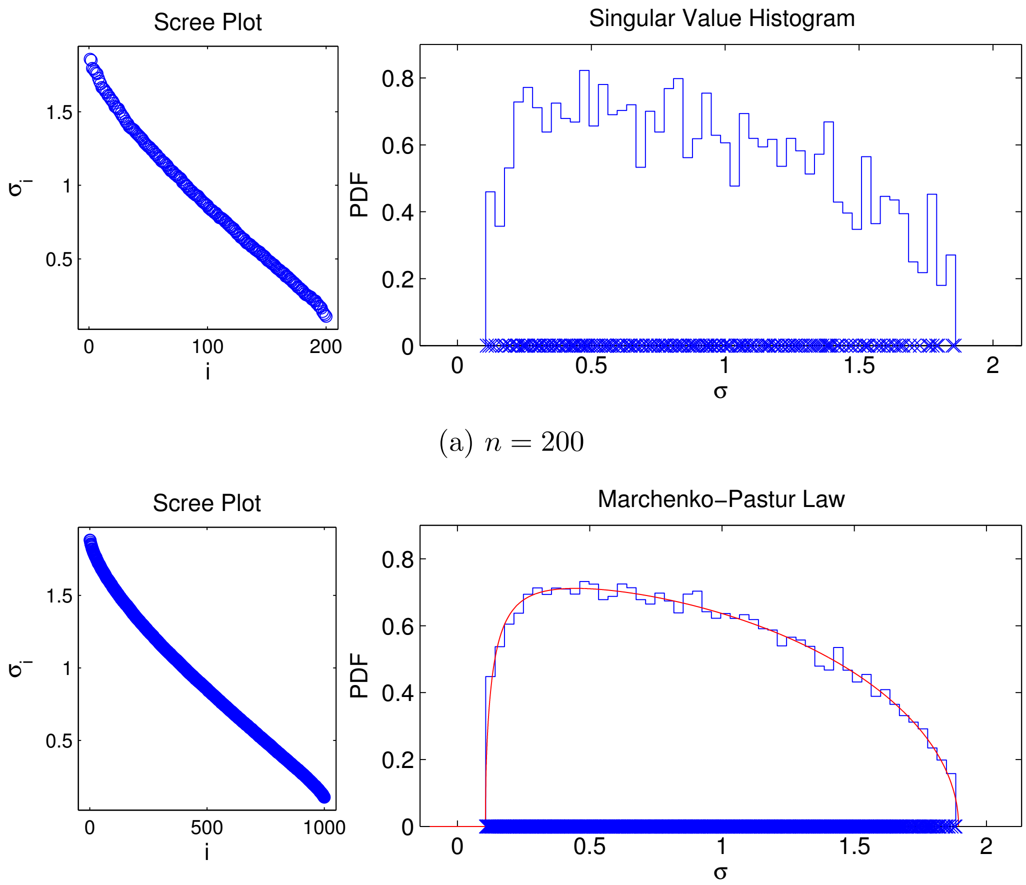
Theorems II.1 and II.2 assume that the singular value spectrum of the noise matrix $X_n$ converges to a non-random probability measure $\mu_X$. Importantly, this condition is satisfied by a wide class of noise models [27–29]. For example, consider the setting where $[X_n]_{ij}$ are i.i.d. with zero mean, variance $\tau^2/m$, and bounded higher order moments. It is known that the spectral density of $X_n$ converges almost surely to the Marcenko-Pastur law [30]

$$\mathrm{d}\mu_X(t) = \frac{\sqrt{(b^2 - t^2)(t^2 - a^2)}}{\pi c \tau^2 t}, \quad t \in [a,\ b], \tag{2.10}$$

where $a = \tau(1 - \sqrt{c})$ and $b = \tau(1 + \sqrt{c})$. Figure 2.1 shows the empirical singular value distributions of two i.i.d. random Gaussian matrices, one with $n = 200$ and the other with $n = 1000$. Clearly the empirical singular value distribution is converging to the Marcenko-Pastur law predicted by (2.10).

Figure 2.2 shows the singular value spectrum of a matrix $\widetilde{X}_n$ drawn from the model (2.5) with $r = 3$ and $\{\theta_1,\ \theta_2,\ \theta_3\} = \{4,\ 3,\ 2\}$. The noise $X_n$ is drawn from the Marcenko-Pastur law with $\tau = 1$. In this case, one can show that the critical point is $\kappa = 1$. All three signals are above the critical point, so Theorem II.1 predicts that the leading 3 singular values of $\widetilde{X}_n$ will separate from the bulk spectrum and converge asymptotically to the locations $\rho_i = D_{\mu_X}^{-1}(1/\theta_i^2)$, for $i = 1, 2, 3$. Figure 2.2 corroborates this result. Conversely, Figure 2.3 shows a different realization of the same model where the third signal is now $\theta_3 = 0.95$. In this case, $\theta_3 < \kappa$, so Theorem II.1 predicts that $\widetilde{\sigma}_3$ will not separate from the bulk spectrum. Figure 2.3 again corroborates this result.

Note that, although the theoretical results in this section are asymptotic in nature, Figures 2.1-2.3 demonstrate an important observation: random matrices of even modest sizes often closely follow their limiting behavior. This observation is important in practice. Indeed, it suggests that it is reasonable to design algorithms for matrix



(a) $n = 200$

(b) $n = 1000$

Figure 2.1: Singular values of a $cn \times n$ matrix with i.i.d. Gaussian entries for two values of $n$. Left: singular value scree plots. Right: empirical singular value historgrams. The red curve denotes the limiting Marcenko-Pastur law (2.10).

models that are based on asymptotic results, which can be expected to reasonably approximate the statistics of the empirical data. We now return to the problem of optimally estimating a low-rank matrix corrupted by random noise.

## 2.3 Optimal Low-Rank Matrix Estimation

Recall the low-rank plus noise model from (2.5):

$$\widetilde{X}_n = L_n + X_n, \tag{2.11}$$

where $L_n = \sum_{i=1}^{r} \theta_i u_i v_i^H$ is an unknown low-rank matrix of (known) rank $r$ with singular values $\theta_i$ and singular vectors $u_i$ and $v_i$, and $X_n$ is an additive noise matrix.

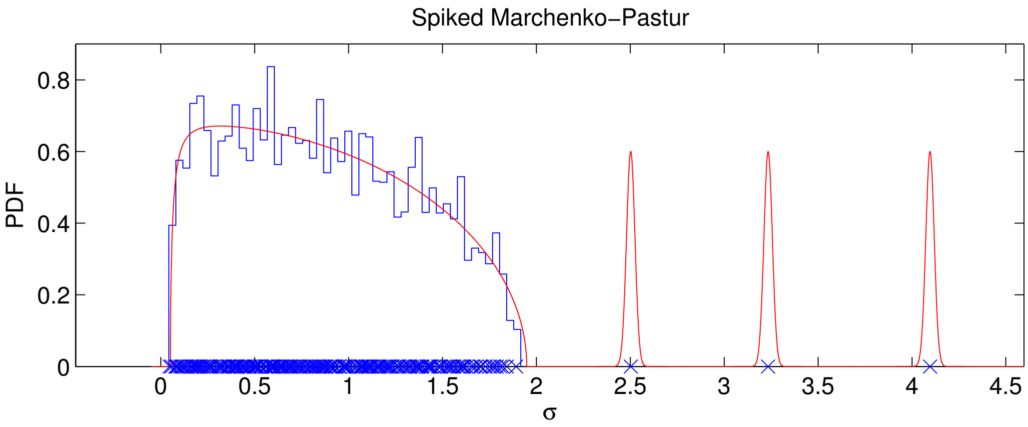


Figure 2.2: Singular value spectrum of $\widetilde{X}_n$ drawn from the model (2.5) with $r = 3$ and $\{\theta_1,\ \theta_2,\ \theta_3\} = \{4,\ 3,\ 2\}$. The noise $X_n$ is drawn from the Marcenko-Pastur law with $\tau = 1$. The blue X's denote the singular values of $\widetilde{X}_n$, the blue curve denotes their empirical histogram, and the red curve is the limiting spectrum predicted by Theorem II.1. All three signals are above the critical point $\kappa = 1$, so the location of the extreme singular values are asymptotically given by $\rho_i = D_{\mu_X}^{-1}(1/\theta_i^2)$, for $i = 1, 2, 3$.

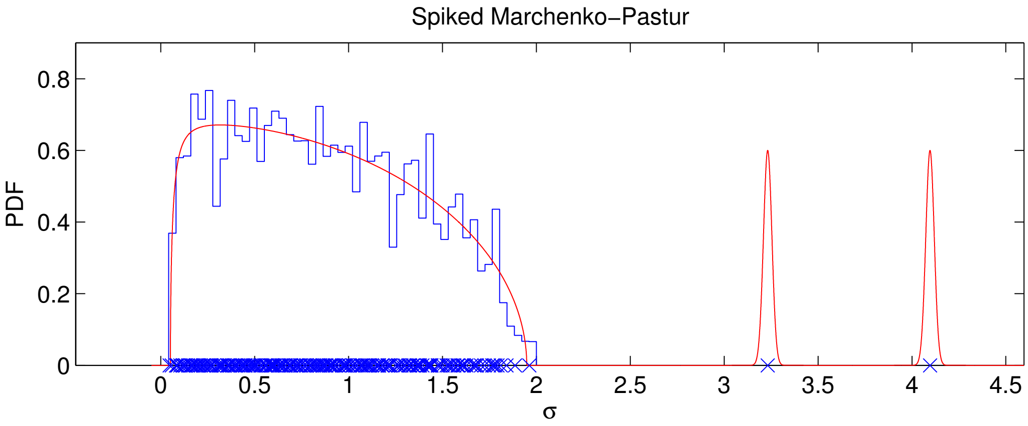


Figure 2.3: Singular value spectrum of $\widetilde{X}_n$ drawn from the model (2.5) with $r = 3$ and $\{\theta_1,\ \theta_2,\ \theta_3\} = \{4,\ 3,\ 0.95\}$. The noise $X_n$ is drawn from the Marcenko-Pastur law with $\tau = 1$. The blue X's denote the singular values of $\widetilde{X}_n$, the blue curve denotes their empirical histogram, and the red curve is the limiting spectrum predicted by Theorem II.1. The third signal $\theta_3 = 0.95$ is less than the critical point $\kappa = 1$, so it does not separate from the bulk spectrum.

### 2.3.1 Oracle Denoising Problem

Suppose that we are interested in producing an estimate of $L_n$ given an instance of $\widetilde{X}_n$. One way to formulate this problem is the oracle denoising problem

$$w^\star = \underset{[w_1,\ \ldots,\ w_r]^T \in \mathbb{R}^r}{\arg\min} \quad \Big\| \sum_{i=1}^{r} \theta_i u_i v_i^H - \sum_{i=1}^{r} w_i \widetilde{u}_i \widetilde{v}_i^H \Big\|_F, \tag{2.12}$$

where $\widetilde{\sigma}_i$ are the singular values and $\{\widetilde{u}_i, \widetilde{v}_i\}$ are the singular vectors of $\widetilde{X}_n$.

Problem (2.12) seeks the best approximation of the latent low-rank signal matrix $L_n$ by an optimally weighted combination of estimates of its left and right singular vectors. We refer to (2.12) as an oracle problem because it implicitly depends on the latent low-rank matrix $L_n$. Nonetheless, note that the TSVD of rank $r$ and SVT are both feasible points for (2.12). Indeed, the truncated SVD corresponds to choosing weights $w_i = \widetilde{\sigma}_i \mathbb{1}\{i \leq r\}$ and SVT with parameter $\tau \geq \widetilde{\sigma}_{r+1}$ corresponds to $w_i = (\widetilde{\sigma}_i - \tau)_+$. However, (2.12) can be solved in closed-form [26], yielding the expression

$$w_i^\star = \sum_{j=1}^{r} \theta_j (\widetilde{u}_i^H u_j)(\widetilde{v}_i^H v_j), \quad i = 1, \ldots, r. \tag{2.13}$$

Of course, (2.13) cannot be computed in practice because it depends on the latent low-rank singular vectors $u_i$ and $v_i$ that we would like to estimate, but it gives insight into the properties of the optimal weights $w^\star$. Indeed, when $\widetilde{u}_i$ and $\widetilde{v}_i$ are good estimates of $u_i$ and $v_i$, respectively, we expect $\widetilde{u}_i^H u_i$ and $\widetilde{v}_i^H v_i$ to be close to 1. Consequently, from (2.13), we expect $w_i^\star \approx \theta_i$. Conversely, when $\widetilde{u}_i$ and $\widetilde{v}_i$ are poor estimates of $u_i$ and $v_i$, respectively, we expect $\widetilde{u}_i^H u_i$ and $v_i^H \widetilde{v}_i$ to be closer to 0 and $w_i^\star < \theta_i$. In other words, (2.13) shows that the optimal singular value shrinkage is inversely proportional to the accuracy of the estimated principal subspaces. As a special case, if $\theta_i \to \infty$, then clearly $\widetilde{u}_i^H u_i \to 1$ and $v_i^H \widetilde{v}_i \to 1$, so the optimal weights $w_i^\star$ must have the property that the absolute shrinkage vanishes as $\theta_i \to \infty$. This shows that, the SVT estimator, which applies a constant shrinkage to each singular value of its input, will necessarily produce suboptimal low-rank estimates in general. See [26] for more details.

Note that the constituent quantities $\{\theta_j,\ \widetilde{u}_i^H u_j,\ \widetilde{v}_i^H v_j\}$ of the solution (2.13) to (2.12) are exactly of the form analyzed in Section 2.2. Therefore, while we cannot compute (2.13) in practice, we can obtain asymptotic expressions for them in the large matrix limit when $X_n$ is a suitable random matrix (e.g., an i.i.d. random matrix). The following theorem [26] formalizes this observation.

**Theorem II.3.** *(Optimal Low-Rank Matrix Estimation [26]). Suppose that $(X_n)_{ij}$ are i.i.d. random variables with zero-mean, variance $\sigma^2$, and bounded higher order moments, and suppose that $\theta_1 > \theta_2 > \ldots > \theta_r > \sigma$. Then, as $m, n \to \infty$ such that $m/n \to c \in (0, \infty)$, we have that*

$$w_i^\star + 2\,\frac{D_{\mu_{\widetilde{X}}}(\widetilde{\sigma}_i)}{D'_{\mu_{\widetilde{X}}}(\widetilde{\sigma}_i)} \xrightarrow{a.s.} 0 \qquad \textit{for} \;\; i = 1, \ldots, r, \tag{2.14}$$

*where*

$$\mu_{\widetilde{X}}(t) = \frac{1}{q-r} \sum_{i=r+1}^{q} \delta\left(t - \widetilde{\sigma}_i\right), \tag{2.15}$$

*with* $q = \min(m, n)$ *is the empirical singular value density of* $\widetilde{X}_n$ *and* $D_{\mu_{\widetilde{X}}}$ *is the D-transform* (2.7) *of the measure* $\mu_{\widetilde{X}}$.

Theorem II.3 establishes that the weights $w_i^\star$—the solution to the oracle denoising problem (2.12)—converge in the large matrix limit to a certain non-random integral transformation of the limiting noise distribution $\mu_{\widetilde{X}}$.

### 2.3.2 Data-Driven OptShrink Estimator

In practice, Theorem II.3 suggests the following data-driven OptShrink estimator [26], defined for a given matrix $Y \in \mathbb{C}^{m \times n}$ and rank $r$ as

$$\textbf{OptShrink}_r(Y) = \sum_{i=1}^{r} \left( -2 \, \frac{D_{\mu_Y}(\sigma_i)}{D'_{\mu_Y}(\sigma_i)} \right) u_i v_i^H, \tag{2.16}$$

where $Y = U \Sigma V^H$ is the SVD of $Y$ with singular values $\sigma_i$ and

$$\mu_Y(t) = \frac{1}{q-r} \sum_{i=r+1}^{q} \delta\left(t - \sigma_i\right) \tag{2.17}$$

is the empirical mass function of the noise-only singular values of $Y$ with $q = \min(m, n)$. Equation 2.16 approximates the optimal shrinkage from Theorem II.3 by plugging in the empirical distribution of the noise-only (non-leading) singular values, $\mu_Y$, in place of the limiting distribution, $\mu_X$, to which the empirical distribution is converging. By Theorem II.3, $\textbf{OptShrink}_r(\widetilde{X})$ asymptotically solves the oracle denoising problem (2.12).

OptShrink has a single parameter $r \in \mathbb{N}$ that directly specifies the rank of its output matrix. Rather than applying a constant shrinkage to each singular value of the input matrix as in SVT, the OptShrink estimator partitions the singular values of its input matrix into *signals* $\{\sigma_1, \ldots, \sigma_r\}$ and *noise* $\{\sigma_{r+1}, \ldots, \sigma_q\}$ and uses the empirical mass function of the noise singular values to estimate the optimal (nonlinear, in general) shrinkage (2.14) to apply to each signal singular value. See [26, 27] for additional detail and intuition.

### 2.3.3 Computational Cost

The computational cost of OptShrink is the cost of computing a full SVD[2] plus the $O(r(m+n))$ computations required to compute the $D$-transform terms in (2.16), which reduce to summations for the choice of $\mu_Y$ in (2.17).

---

[2] In practice, one need only compute the singular values $\sigma_1, \ldots, \sigma_q$ and the leading $r$ singular vectors of $Y$.

## CHAPTER III

# Improved Robust PCA Using Optimal Data-Driven Singular Value Shrinkage

## 3.1 Introduction

Principal component analysis (PCA) is a powerful technique for uncovering latent low-rank structure in high dimensional datasets. It is ubiquitous in statistical signal processing theory and practice and is the first step in many inferential procedures for detection, estimation and classification. It is well-known, however, that PCA is brittle in the sense that relatively few outliers can severely degrade the quality of low-rank components estimated from noisy data. This, in turn, degrades the performance of inferential tasks that utilize these estimated low-rank components. Robust PCA aims to mitigate such problems by producing the best (with respect to squared error) low-rank estimates that are robust to outlier contamination.

Recent breakthroughs [1, 31–33] have established that one can reliably recover a low-rank matrix in the presence of outliers by solving a convex optimization problem of the form

$$\begin{aligned} \min_{L,S} \quad & \|L\|_\star + \lambda \|S\|_1 \\ \text{s.t.} \quad & Y = L + S, \end{aligned} \tag{3.1}$$

where $Y$ is the observed data matrix, $\|L\|_\star$ is the the nuclear norm (sum of singular values) of the low-rank component $L$, and $\|S\|_1$ is the elementwise $\ell_1$ norm of the sparse component $S$. Indeed, sufficient conditions on $L$ and $S$ are given in [31, 32] to guarantee that the solution to (3.1) will exactly recover the low-rank and sparse components of the noiseless model $Y = L + S$. However, much less is known about the noisy setting—when $Y$ is also corrupted by dense noise—except the unsurprising fact that one cannot expect error-free recovery. There is no theoretical reason to expect that a convex optimization-based model like (3.1) that was designed for the noiseless setting will also be optimal in the noisy setting.

In [26] it is shown that, in the noisy but outlier-free setting, the low-rank components produced by solving any convex optimization problem are provably suboptimal. Indeed, [26] shows that the OptShrink estimator (described in Chapter II of this thesis) provably outperforms convex optimization-based methods for low-rank matrix denoising. In this chapter, our goal is to apply these insights from low-rank matrix estimation in the context of performing robust PCA on noisy data.

### 3.1.1 Contributions

We first motivate the need for robust PCA algorithms by providing a first-principles analysis of the effect of outliers on the singular vectors of a noisy low-rank plus sparse matrix. Our analysis demonstrates that PCA is robust to noise but highly sensitive to even relatively few outliers in the data matrix. We then propose a new alternating minimization algorithm for robust PCA that uses the OptShrink estimator to improve the quality of the estimated low-rank component. Our proposed method is suitable for application in any inverse problem setting. Unlike existing methods, our algorithm does not correspond to a convex objective; however, we observe that it behaves well in practice. In particular, we demonstrate that our proposed method outperforms conventional robust PCA methods both in terms of quantitative reconstruction accuracy and qualitative interpretability of the components for two diverse applications: background subtraction and dynamic magnetic resonance imaging (MRI) reconstruction from highly undersampled measurements.

### 3.1.2 Organization

This chapter is organized as follows. In Section 3.2, we formulate our robust PCA problem, and, in Section 3.3, we analytically characterize the effect of outliers on the singular vectors of the observed matrix. We describe the conventional convex optimization-based approach to robust PCA in Section 3.4, and we propose an improved algorithm in Section 3.5. In Section 3.6, we provide numerical experiments that demonstrate the promising performance of our method compared to existing robust PCA methods. Finally, we summarize our findings in Section 3.7.

## 3.2 Problem Formulation

Consider the setting where an $m \times n$ observed signal-plus-noise-plus-outliers matrix $Y$ is modeled as

$$Y = \underbrace{\sum_{i=1}^{r} \theta_i u_i v_i^H}_{=:L} + S + X, \tag{3.2}$$

where, without loss of generality, we assume $m \leq n$. In (3.2), $L$ represents the rank-$r$ low-rank signal matrix that we are interested in reliably recovering, where $u_i$ and $v_i$ are the left and right singular vectors associated with singular value $\theta_i$. The matrix $S$ is modeled as

$$S_{ij} = \begin{cases} Q_{ij} & \text{with probability } p_s \\ 0 & \text{with probability } 1 - p_s, \end{cases}$$

where $Q_{ij}$ are elements drawn from an unknown distribution $q$ with zero-mean, variance $\sigma_q^2$, and bounded higher order moments. The matrix $S$ represents a sparse matrix of outliers (relative to $L$). We assume that the outlier probability $p_s \gg \log n/n$ to avoid any pathologies related to the sparsity pattern of $S$ interfering with the singular vectors of $Y$ [34]. The matrix $X$ has elements that are independently and identically distributed with zero mean, variance $\sigma^2/n$, and bounded higher order moments. In (3.2), we assume the outliers are sparse with respect to the standard Euclidean basis. If they are sparse with respect to some other basis (e.g., Fourier or wavelet), we can, without loss of generality, assume that (3.2) holds after an appropriate sparsifying transformation has been applied to the vectorized elements of the observed matrix.

## 3.3 Motivation for Robust PCA

Our goal is to estimate, as accurately as possible, the low-rank component $L$ from the matrix $Y$ under the model (3.2). This objective is complicated by the presence of the outlier matrix $S$. Indeed, let

$$Y = \sum_{i=1}^{m} \widetilde{\sigma}_i \widetilde{u}_i \widetilde{v}_i^H \tag{3.3}$$

be the SVD of $Y$. The following theorem extends results from random matrix theory [27] to quantify the degradation incurred when estimating the singular vectors of $L$ in the presence of outliers.

**Theorem III.1.** *Assume that the singular vectors of $L$ satisfy a low-coherence condition, i.e.,*

$$\max_{i=1,\dots r} \|u_i\|_\infty \leq C_u \frac{\log^{\eta_u} m}{\sqrt{m}}, \qquad \max_{i=1,\dots r} \|v_i\|_\infty \leq C_v \frac{\log^{\eta_v} n}{\sqrt{n}}$$

*for some universal constants $C_u, C_v, \eta_u, \eta_v > 0$, and suppose that $\theta_1 > \ldots > \theta_r > 0$ in* (3.2) *for some fixed $r > 0$. Then, as $m, n \to \infty$ such that $m/n \to c \in (0, 1]$, we have*

$$|\langle u_i, \widetilde{u_i}\rangle|^2 \xrightarrow{a.s.} \begin{cases} 1 - \dfrac{c\left(1+\bar{\theta}_i^2\right)}{\bar{\theta}_i^2\left(\bar{\theta}_i^2 + c\right)} & \text{if } \bar{\theta}_i > c^{1/4} \\ 0 & \text{if } \bar{\theta}_i \leq c^{1/4}, \end{cases} \tag{3.4}$$

*and*

$$|\langle v_i, \widetilde{v_i}\rangle|^2 \xrightarrow{a.s.} \begin{cases} 1 - \dfrac{\left(c+\bar{\theta}_i^2\right)}{\bar{\theta}_i^2\left(\bar{\theta}_i^2 + 1\right)} & \text{if } \bar{\theta}_i > c^{1/4} \\ 0 & \text{if } \bar{\theta}_i \leq c^{1/4}, \end{cases} \tag{3.5}$$

*where*

$$\bar{\theta}_i = \lim_{n\to\infty} \frac{\theta_i}{\sqrt{\sigma^2 + n\, p_s\, \sigma_q^2}}. \tag{3.6}$$

*Proof.* Model (3.2) is equivalent to the low-rank plus noise model $Y = L + \bar{X}$, where $\bar{X}_{ij} = S_{ij} + X_{ij}$ are independent zero-mean random variables with variance $\bar{\sigma}^2/n$, where $\bar{\sigma}^2 := \sigma^2 + n\, p_s\, \sigma_q^2$. The results follows from Theorem 2.9 and Section 3.1 of [27]. □

Theorem III.1 brings into sharp focus the detrimental effect of sparse outliers on the estimation of low-rank components. For example, suppose that $\sigma_q^2 = O(1)$ and $p_s = O(1)$. Then, by (3.6), $\bar{\theta}_i = 0$ so that $|\langle u_i, \widetilde{u_i}\rangle|^2 \to 0$ and $|\langle v_i, \widetilde{v_i}\rangle|^2 \to 0$ as $n \to \infty$, irrespective of the magnitude of $\theta_i$ relative to the noise variance $\sigma^2$. Consequently, the singular vectors of $Y$ will be poor estimates of the singular vectors of $L$. In contrast, when $p_s = 0$ so that we are in the outlier-free setting and $\theta_i/\sigma \gg c^{1/4}$, we can expect the singular vectors of $Y$ to be good estimates of the singular vectors of $L$.

More generally, from Theorem III.1 and (3.6), we conclude that the singular vectors of $Y$ will be very poor estimates of the singular vectors of $L$ whenever $\sigma_q^2 = O(1)$ and $np_s \to \infty$. The latter condition includes the few-outlier setting when $p_s = O(\log n/n)$, so that an average of just $O(n \log n)$ corrupted entries out of the $mn$ total entries will suffice to severely degrade the eigenstructure of the matrix $Y$. This motivates the necessity of robust PCA methods for reliably extracting low-rank structure in the presence of outliers.

## 3.4 Convex Optimization-Based Robust PCA

The predominant approach for performing robust PCA has been to adopt a low-rank regularization term involving the nuclear norm $\|\cdot\|_\star$ and a sparse regularization term involving the $\ell_1$ norm $\|\cdot\|_1$. One recently proposed [1] method for low-rank plus sparse matrix decomposition in the presence of noise is to solve the convex optimization problem

$$\{\hat{L},\ \hat{S}\} = \arg\min_{L,S}\ \frac{1}{2}\|Y - \mathcal{A}(L+S)\|_F^2 + \lambda_L\|L\|_\star + \lambda_S\|TS\|_1, \tag{3.7}$$

where $\mathcal{A}$ is a linear operator, $Y$ is the problem data, and $T$ is a unitary sparsifying transformation (e.g., wavelet/Fourier basis). Note that (3.7) is an extension of (3.2) to the inverse problem setting that allows for the matrix $S$ to be sparse in a transformed domain (w.r.t. $T$) and for the observations $Y$ to be related to the underlying components $L$ and $S$ through a sensing operator $\mathcal{A}$.

One approach to solving (3.7) is the proximal gradient method [25,35], an iterative algorithm where, at the $k$th iteration, one computes the updates

$$\begin{aligned} L^{k+1} &= \mathbf{SVT}_{t^k\lambda_L}(M^k - S^k) \\ S^{k+1} &= T^H\mathbf{soft}_{t^k\lambda_S}(T(M^k - L^k)) \\ M^{k+1} &= L^{k+1} + S^{k+1} - t^k(\mathcal{A}^\star(\mathcal{A}(L^{k+1} + S^{k+1}) - Y)), \end{aligned} \tag{3.8}$$

with step size $t^k$, where $\mathcal{A}^\star$ denotes the adjoint of $\mathcal{A}$. In (3.8), $\mathbf{SVT}(\cdot)$ is the singular value thresholding (SVT) operator [31], defined for a given $\tau > 0$ as

$$\mathbf{SVT}_\tau(Z) := \sum_i (\sigma_i - \tau)_+ u_i v_i^H, \tag{3.9}$$

where $Z = U\Sigma V^H$ is the SVD of $Z$, $(\cdot)_+ = \max(\cdot, 0)$, and $\mathbf{soft}_\lambda(\cdot)$ is the elementwise soft thresholding operator, defined for a given $\lambda > 0$ as

$$\mathbf{soft}_\lambda(z) := \mathbf{sign}(z)(z - \lambda)_+. \tag{3.10}$$

The attractiveness of the updates (3.8) lies in the fact that the $L$ and $S$ updates have simple closed-form expressions. Moreover, because $\|\cdot\|_\star$ and $\|\cdot\|_1$ are convex, standard convergence results [35, 36] establish the convergence of (3.8) to a solution of (3.7) provided one uses, for example, a constant step size $t = t^k$ that satisfies $t < 1/\|\mathcal{A}\|^2$. However, despite these convergence guarantees, there is no theoretical reason to expect convex optimization-based algorithms for robust PCA to provide *optimal* (in any sense) recovery of low-rank or sparse matrices in the noisy setting.

## 3.5 Proposed Algorithm

One can interpret the iterate $M^k$ from (3.8) as the current estimate of the underlying low-rank plus sparse matrix plus a residual (i.e., noise) term. As such, the argument $M^k - S^k$ of the SVT operator is approximately a low-rank plus noise matrix of the type studied in Chapter II. Therefore, we propose to perform the modified updates

$$\begin{aligned} L^{k+1} &= \mathbf{OptShrink}_r(M^k - S^k) \\ S^{k+1} &= T^H\ \mathbf{soft}_{t^k\lambda_S}(T(M^k - L^k)) \\ M^{k+1} &= L^{k+1} + S^{k+1} - t^k(\mathcal{A}^\star(\mathcal{A}(L^{k+1} + S^{k+1}) - Y)), \end{aligned} \tag{3.11}$$

where we have replaced SVT in the $L$-update of (3.8) with the OptShrink estimator [26]. In (3.11), $\mathbf{OptShrink}(\cdot)$ is the low-rank matrix estimator defined for a given parameter $r > 0$ as

$$\mathbf{OptShrink}_r(Z) = \sum_{i=1}^{r} \left( -2 \frac{D_{\mu_Z}(\sigma_i)}{D'_{\mu_Z}(\sigma_i)} \right) u_i v_i^H, \tag{3.12}$$

where $Z = \sum_i \sigma_i u_i v_i^T$ is the SVD of $Z \in \mathbb{R}^{a \times b}$. In (3.12), the $D$-transform is defined for a given probability measure $\mu$ as

$$D_\mu(z) = \left[ \int \frac{z}{z^2 - t^2}\ \mathrm{d}\mu(t) \right] \times \left[ c \int \frac{z}{z^2 - t^2}\ \mathrm{d}\mu(t) + \frac{1-c}{z} \right], \tag{3.13}$$

where $D'_\mu(z)$ is the derivative of $D_\mu(z)$ with respect to $z$, $c = \min(a,b)/\max(a,b)$, and $\mu_Z(t) = \frac{1}{q-r}\sum_{i=r+1}^{q} \delta(t - \sigma_i)$ is the empirical mass function of the noise-only singular values of $Z$ with $q = \min(a,b)$. Note that the integrals in the $D$-transform terms in (3.12) reduce to summations for this choice of $\mu_Z$, so they can be computed efficiently.

The $\mathbf{OptShrink}_r(Z)$ operator computes the rank $r$ truncated SVD of $Z$ and then applies the shrinkage function defined by the parenthesized term in (3.12) to the leading singular values. We refer to the $D$-transform term as a shrinkage function because it shrinks its argument towards zero [26]. In contrast, the $\mathbf{SVT}_\lambda(Z)$ operator (3.9) applies a constant shrinkage level $\lambda$ to all singular values.

The OptShrink estimator provides two key benefits over SVT. First, it applies a data-driven shrinkage to the singular value spectrum of its argument, the form of which is imputed from the non-leading (noise) singular values. Generically, a smaller shrinkage is applied to larger—and hence more-informative—singular values and a comparatively larger shrinkage to smaller singular values. The effect of this nonlinear shrinkage is to produce an improved estimate of the underlying low-rank matrix embedded in the data [26]. See Chapter II of this thesis for further details. Second, OptShrink has a single parameter $r$ that directly specifies the rank of the output matrix. It is often very natural to set the rank parameter. For example, suppose the columns of $Y$ contain vectorized images of a scene with a static background. In this case, the low-rank component $L$ should ideally be a rank-1 matrix whose columns are repeated (up to scaling) vectorized copies of the static background image. In practice, the registered background may not be perfectly static, but it will still have high temporal correlation, so a small rank ($r = 2, 3, \ldots$) will often suffice. Thus it is often easier to tune the rank parameter $r$ than the analogous real-valued regularization parameter $\lambda_L$ in the SVT updates (3.7).

OptShrink is data-driven and thus does not correspond to the proximal operator of a penalty function $\phi(L)$, so the updates (3.11) are not proximal gradient updates for a cost function like (3.7). Nonetheless, our experiments in Section 3.6 show that the proposed update scheme is numerically stable and yields convergent iterate sequences in practice.

## 3.6 Numerical Experiments

In this section, we demonstrate the performance of the proposed update scheme (3.11) in two applications: background subtraction and dynamic MRI reconstruction from highly undersampled measurements.

### 3.6.1 Background Subtraction

We first demonstrate the performance of the proposed update scheme (3.11) by performing background subtraction on an outlier-corrupted version of the Fountain sequence.[1] The dataset contains $n = 523$ images, each with resolution $m_y \times m_x = 128 \times 160$, of a scene with a fountain in the background and people walking intermittently in the foreground. We arrange the data in an $m \times n$ matrix whose columns contain the $m = m_y m_x$ vectorized pixels of each image. We then generate an ob-

---

[1] Obtained from `perception.i2r.a-star.edu.sg/bk_model/bk_index.html`.

servation matrix $Y$ by adding an outlier matrix $S$ with elements drawn for a given outlier probability $p_s$ as in (3.2) with $Q_{ij} = \pm K$ equiprobably for a given $K > 0$.

In the language of model (3.2), the low-rank component $L$ should capture the static background component of the scene (the fountain), and the sparse component should capture the dynamic foreground and the sparse corruptions.[2] Therefore, we compare the quality of the background estimates ($L$) returned by SVT-based method from (3.8) with the proposed updates from (3.11). Specificially, we package each update scheme into an algorithm by initializing $L_0 = M_0 = Y$ and $S_0 = 0$ and terminating the iterations when $\|M_k - M_{k-1}\|_F < \delta\|M_{k-1}\|_F$ for a given stopping tolerance $\delta > 0$.

Figure 3.1 displays the algorithm outputs for two representative frames of the reconstructions. Figure 3.1c shows that the SVT-based updates with small $\lambda_L$ fail to recover the low-rank background; indeed, $\hat{L}_{\text{svt}}$ was full-rank. On the other hand, Figure 3.1d shows that setting $\lambda_L$ large enough to force the SVT-based updates to return a background estimate $\hat{L}_{\text{svt}}$ with rank-1 is also problematic. Indeed, the SVT-based $L$-update in (3.8) uniformly shrinks all singular values, which, for large $\lambda_L$, results in suboptimal degradation of large singular values. This effect manifests in Figure 3.1d through the dimness of the low-rank components and the leakage of the remaining background intensity into the sparse component. In contrast, the proposed approach successfully isolates the fountain background in its low-rank component.

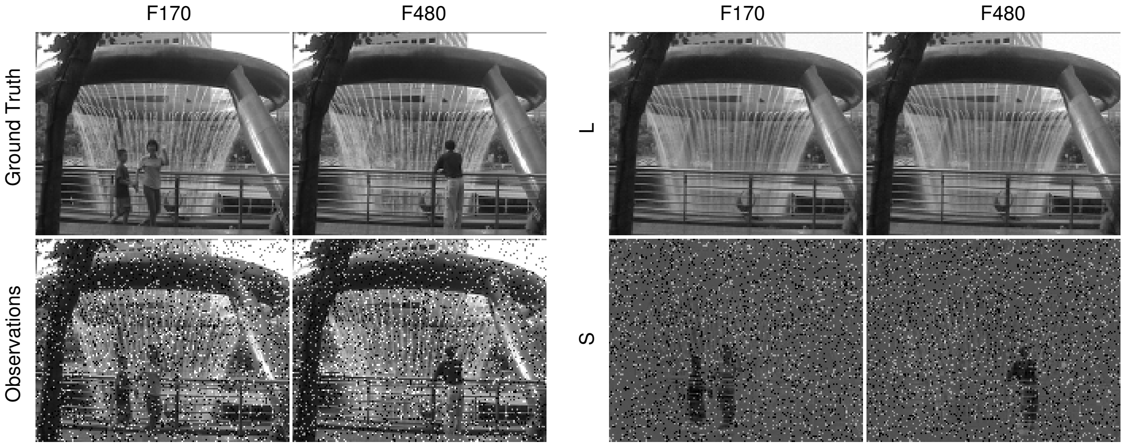
### 3.6.2 Dynamic MRI Reconstruction

Recent advances in dynamic contrast-enhanced MRI (DCE-MRI) algorithms have employed low-rank plus sparse matrix decomposition for joint reconstruction of multicoil data. Such methods model DCE-MRI data as the superposition of low-rank and sparse components because the high spatiotemporal correlations of the static background image are inherently low-rank while the remaining dynamic contrast component is often sparse with respect to an appropriate temporal transformation (e.g., wavelet/Fourier basis).

Specifically, in multicoil DCE-MRI, one acquires data $Y \in \mathbb{C}^{n_d}$ consisting of $k$-space data from $n_c$ coils corresponding to $n_t$ frames of $n_y \times n_x$ images [37]. Given data $Y$, [1] recently proposed to reconstruct the MR frames by solving (3.7), where $\mathcal{A} : \mathbb{C}^{n_p \times n_t} \to \mathbb{C}^{n_d}$ is the linear multicoil encoding operator, which, for Cartesian sampling, incorporates coil sensitivities and performs temporal fast Fourier transform (FFT) operations, and $T$ is a unitary temporal sparsifying transformation

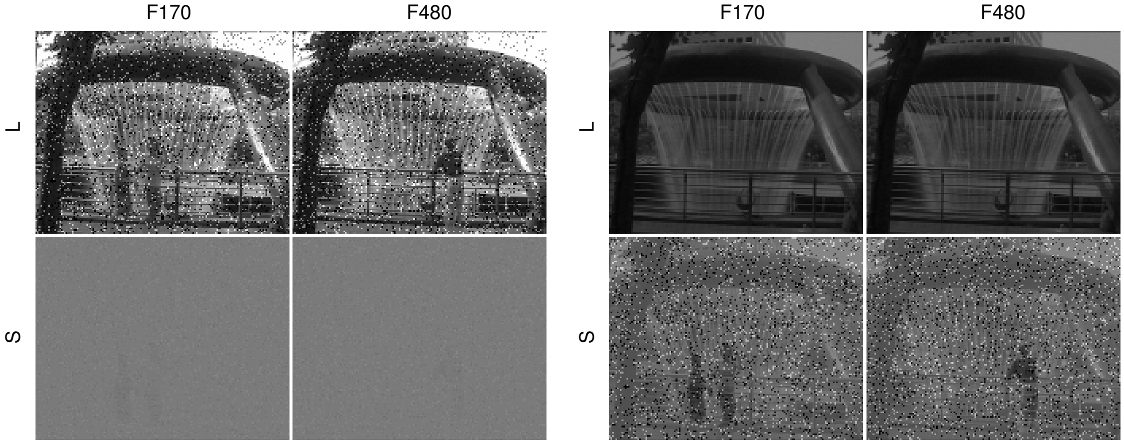
---

[2] We discuss how to separate sparse corruptions from the dynamic foreground in Chapter IV.



(a) Top: ground truth data, $Y$. Bottom: observed data, $\widetilde{X}$.

(b) Proposed approach with $r = 1$ and $\lambda_S = 0.0035$. NRMSE = 9.5%.



(c) SVT-based updates with $\lambda_L = 6.5$ and $\lambda_S = 0.0035$. NRMSE = 38.4%.

(d) SVT-based updates with $\lambda_L = 300$ and $\lambda_S = 0.0035$. NRMSE = 42.3%.

Figure 3.1: Two representative frames from the decompositions produced by the proposed method (3.11) and the SVT-based method (3.8) on the Fountain sequence with parameters $\delta = 0.0025$, $\tau_k = 0.5$, $p_s = 0.15$, and $K = 0.5$. The row labels $L$ and $S$ denote the low-rank and sparse components, respectively, returned by each algorithm. The column labels denote the frame number (i.e., column of $L$ and $S$) that is displayed. Each panel is displayed on the same intensity scale. NRMSE values are reported for the low-rank components using output of the SVT-based updates with $p_s = 0$ as ground truth.

(e.g., wavelet/Fourier transform). Here $n_p := n_y n_x$ and the solutions $\hat{L}, \hat{S} \in \mathbb{C}^{n_p \times n_t}$ to (3.7) are matrices whose columns are vectorized images corresponding to the low-rank and sparse, respectively, components of the frames. The output of the algorithm is

$$\hat{X} := \hat{L} + \hat{S}, \tag{3.14}$$

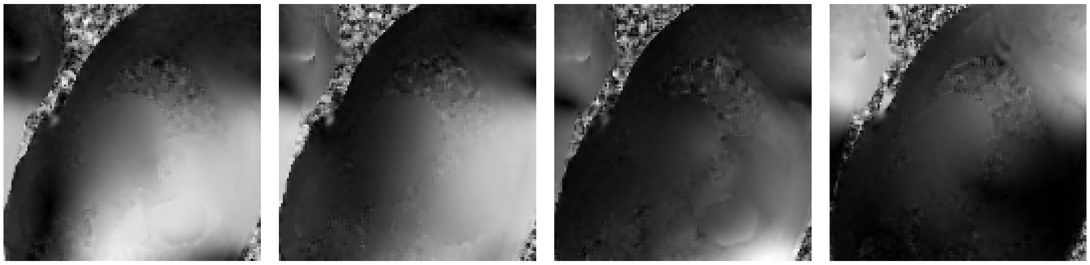


Figure 3.2: Example coil sensitivities for the cardiac perfusion data from [1].

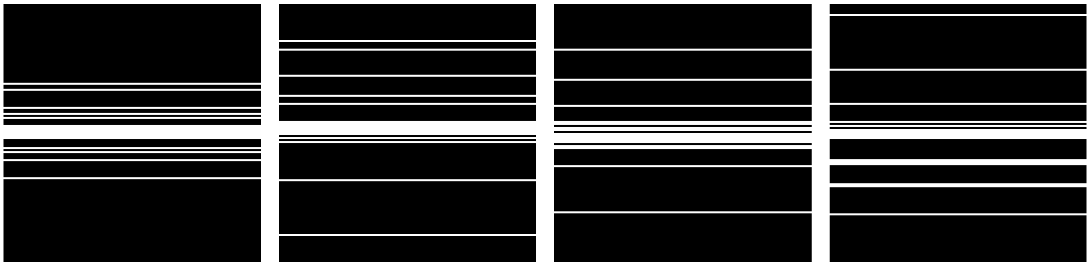


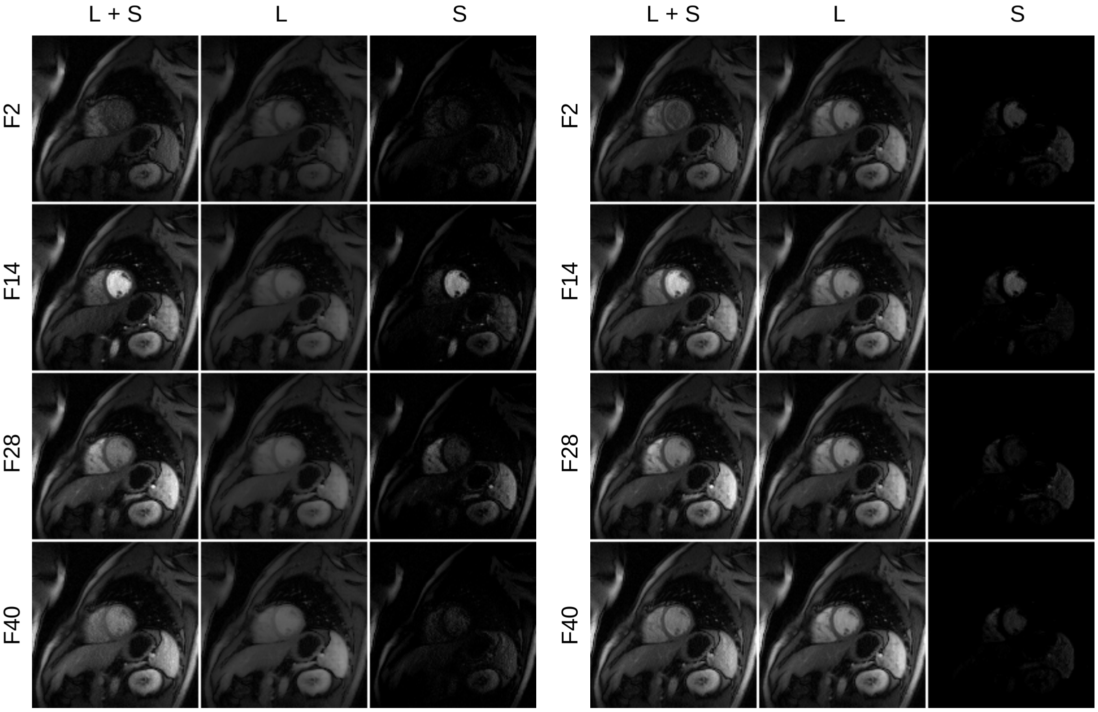
Figure 3.3: Example $k$-space sampling masks for the cardiac perfusion data from [1].

a matrix whose columns contain the vectorized reconstructed MR frames.

Having mapped the DCE-MRI reconstruction problem to the language of (3.7), we can now compare our proposed approach from (3.11) with the standard SVT-based formulation. We packaged each update scheme into an algorithm by initializing $L_0 = M_0 = Y$ and $S_0 = 0$ and terminating when $\|M_k - M_{k-1}\|_F < \delta \|M_{k-1}\|_F$ for a given stopping tolerance $\delta > 0$. In all simulations we set $\delta = 0.0025$ and used constant step size $t_k = 1$. The updates (3.8) require tuning the low-rank penalty parameter $\lambda_L$ and the sparsity penalty parameter $\lambda_S$, while our proposed updates (3.11) requires tuning the rank parameter $r$ and the sparsity penalty parameter $\lambda_S$. In the sequel, all reported results were obtained by tuning the parameters of each algorithm to yield the best results with respect to the performance metric of interest during each test.

#### 3.6.2.1 Cardiac Perfusion Dataset #1

We first consider the cardiac perfusion data from [1]. The data $Y$ contains multicoil $k$-space data from $n_c = 12$ coils and $n_t = 40$ frames, each with resolution $n_y \times n_x = 128 \times 128$. The data was acquired via Cartesian sampling and was retrospectively downsampled by a factor of eight with variable density $k_y - t$ sampling. The associated encoding operator $\mathcal{A}$ incorporates the coil sensitivities and performs temporal FFT operations. See [1] for additional technical acquisition specifications. Figure 3.2 shows four of the coil sensitivity maps embedded in $\mathcal{A}$, and Figure 3.3



(a) SVT-based updates (3.8). (b) Proposed updates (3.11).

Figure 3.4: Four representative frames from the reconstructions on the cardiac perfusion dataset from [1]. The columns of each panel show the final reconstruction $\hat{L} + \hat{S}$, the low-rank component $\hat{L}$, and the sparse component $\hat{S}$, respectively, produced by each method.

shows four of the random $k$-space sampling masks embedded in $\mathcal{A}$.

Figure 3.4 compares the reconstructions produced by each method for a representative subset of the frames with parameters tuned to yield qualitatively good images. In particular, we set $r = 2$. The proposed approach improves the clarity of the myocardial wall, particularly in frames 2 and 40. Comparison of the sparse components produced by the two algorithms suggests that the proposed approach more fully exploits the spatiotemporal correlation of the frame backgrounds. Indeed, more frame-independent body regions were absorbed in the low-rank component of the proposed approach while the resulting sparse components primarily contain traces of the contrast enhancement, which is intuitively pleasing.

A drawback of the SVT operator in the $L$-update of (3.8) is that it uniformly shrinks all singular values. This results in unnecessary degradation of the large singular values of $M^k - S^k$, which contain high signal-to-noise ratio image features. The $L$-update in the proposed updates (3.11) avoids this phenomenon by applying a nonlinear shrinkage that heavily penalizes small singular values while lightly penalizing

| Noise strength ($C$) | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 |
|---|---|---|---|---|---|---|---|---|
| SVT-based updates | 14.5 | 16.8 | 19.3 | 21.6 | 23.0 | 23.7 | 24.1 | 24.9 |
| Proposed updates | **14.1** | **16.5** | **18.1** | **18.9** | **20.0** | **20.9** | **21.7** | **22.5** |

Table 3.1: NRMSE values as percentages for reconstructions of the cardiac perfusion data from [1] as a function of retrospective noise variance $C/n_p^2$. The best NRMSE for each trial is in bold.

large singular values. As a result, our proposed approach yields sharper low-rank images than the analogous SVT-based formulation without sacrificing compressibility, i.e., without increasing the rank of $\hat{L}$ (recall that we set $r = 2$).

To quantitatively compare the performance of the algorithms, we retrospectively added Gaussian noise with variance $C/n_p^2$ over a range of $C$ values to the $k$-space data and measured the resulting normalized root mean squared error (NRMSE) of each algorithm, defined as

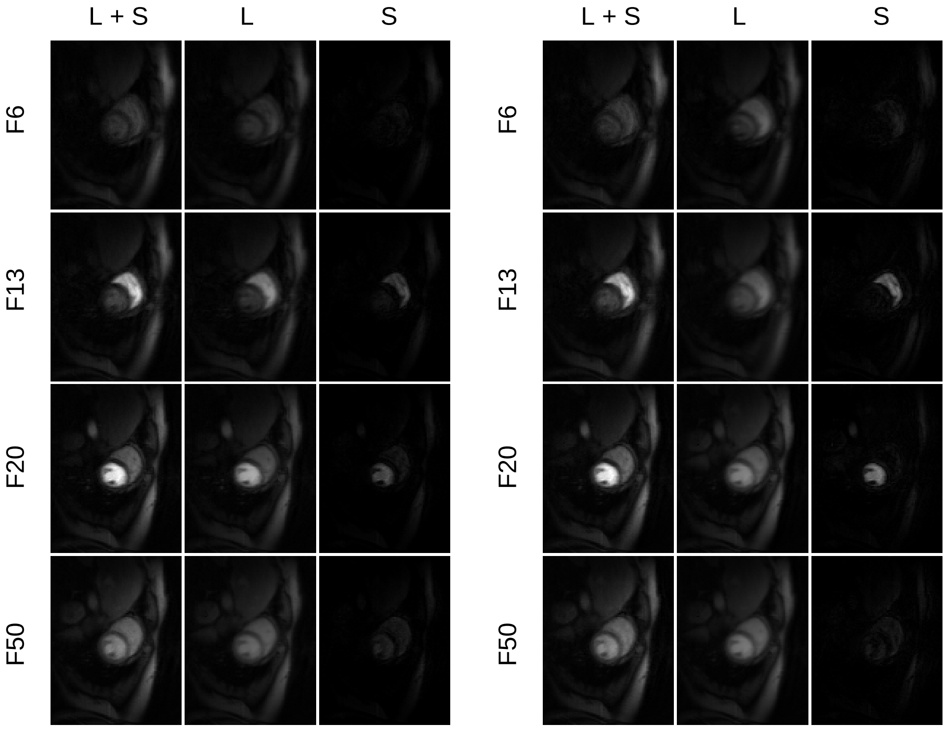
$$\textbf{NRMSE}(\hat{X}) = \frac{\|\hat{X} - X_{\text{true}}\|_F}{\|X_{\text{true}}\|_F}, \tag{3.15}$$

where $\hat{X} = \hat{L} + \hat{S}$ are the reconstructed frames produced by each algorithm, and we used the output of the algorithm for (3.8) on the original data as the ground truth, $X_{\text{true}}$. Table 3.1 shows that the proposed approach produces lower NRMSE over the range of noise variances tested.

#### 3.6.2.2 Cardiac Perfusion Dataset #2

We also consider the cardiac perfusion data from [2, 3]. In this case, we have access to a reference reconstruction $X_{\text{true}}$ computed from fully-sampled data containing $n_t = 70$ frames, each with resolution $n_y \times n_x = 190 \times 90$. We generated $k$-space data $Y$ corresponding to $X_{\text{true}}$ via an encoding operator $\mathcal{A}$ that implements a single-coil, radially sampled MR imaging system with eightfold acceleration. See [2, 3] for additional details about the data acquisition procedure.

Figure 3.5 compares the reconstructions produced by each method for a representative subset of the frames with parameters tuned to yield qualitatively superior images. In particular, we again set $r = 2$. The reconstructed frames are quite similar, but the sparse components recovered by the proposed approach contain more contrast enhancement than those recovered by solving (3.7), which is qualitatively pleasing. The approach based on solving (3.7) absorbed more contrast enhancement into the



(a) SVT-based updates (3.8).

(b) Proposed updates (3.11).

Figure 3.5: Four representative frames from the reconstructions on the cardiac perfusion dataset from [2, 3]. The columns of each panel show the final reconstruction $\hat{L}+\hat{S}$, the low-rank component $\hat{L}$, and the sparse component $\hat{S}$, respectively, produced by each method.

| Noise strength ($C$) | 0 | 1 | 2 | 3 | 4 | 5 | 10 | 15 |
|---|---|---|---|---|---|---|---|---|
| SVT-based updates | 16.5 | 16.6 | 16.9 | 17.5 | 18.3 | 19.2 | 23.8 | 27.3 |
| Proposed updates | **15.6** | **15.8** | **16.2** | **17.1** | **18.3** | **19.2** | **23.5** | **26.5** |

Table 3.2: NRMSE values in percent for reconstructions of the cardiac perfusion data from [2, 3] as a function of retrospective noise variance $C/n_p^2$. The best NRMSE for each setting is in bold.

low-rank components. Table 3.2 compares the NRMSE with respect to $X_{\text{true}}$ attained by both approaches when Gaussian noise with variance $C/n_p^2$ is retrospectively added to the $k$-space data. The proposed approach produces lower NRMSE over the range of noise variances tested.

## 3.7 Conclusions

In this chapter, we proposed a new robust PCA algorithm that utilizes an optimal low-rank matrix estimator (OptShrink) to estimate the low-rank model component, and we demonstrated the application of our algorithm to background subtraction and dynamic MRI reconstruction from highly undersampled measurements. Our numerical experiments show that the proposed method compares favorably to existing methods in both qualitative reconstruction quality and quantitative robustness to noise. Intuitively, our algorithm preserves the quality of high signal-to-noise ratio components of the data without sacrificing compressibility, while also producing sparse components with fewer temporally static elements than the existing methods. Applying our proposed method to other inverse problem settings in which robust PCA are suitable is an interesting area for future work.

# CHAPTER IV

# Panoramic Robust PCA for Foreground-Background Separation on Noisy, Free-Motion Camera Video

## 4.1 Introduction

Principle component analysis (PCA) is an important method in signal processing and statistics for uncovering latent low-rank structure in high dimensional datasets. In turn, low-rank structure is an important model in computer vision because the high temporal correlation of video naturally admits a low-rank representation. Although PCA is stable in the presence of relatively small noise, it is well-known that even a few large outliers in the data can cause PCA to breakdown completely.

To mitigate the breakdown of PCA, robust PCA algorithms have recently been proposed that seek to decompose a data matrix into a low-rank component and a sparse component. Recent works [31, 32, 38, 39] have established that one can exactly recover the low-rank and sparse components of a matrix $Y$ under some mild assumptions in the noiseless setting by solving a convex optimization problem of the form

$$\begin{aligned} \min_{L,S} \quad & \|L\|_\star + \lambda \|S\|_1 \\ \text{s.t.} \quad & Y = L + S, \end{aligned} \tag{4.1}$$

where $\|L\|_\star$ is the nuclear norm (sum of singular values) of the low-rank component and $\|S\|_1$ is the elementwise $\ell_1$ norm of the sparse component. Simple alternating algorithms exist [31] for solving (4.1), which has lead to widespread adoption of robust PCA methods in practice.

Robust PCA has found many applications in computer vision problems. For example, in [40] a robust PCA-based method is developed to learn low-rank textures from corrupted two-dimensional (2D) images of a 3D scene. Or in [41] robust PCA is used to align a batch of linearly correlated images in the presence of gross corruptions such as occlusions. Other applications of robust PCA in vision applications include subspace segmentation and feature extraction [42] and robust subspace clustering [43].

In this work, we focus on another key problem in computer vision: foreground-background separation. Specifically, we consider robust foreground-background separation, where one decomposes a scene into a static background component and a dynamic foreground component in the presence of corruptions. Such decompositions are valuable in vision applications because the components contain useful information for subsequent processing. For example, the foreground component is useful for motion detection [44], object recognition [45], moving object detection [46, 47] and video coding [48]. The background component can also be useful in applications such as background subtraction [49, 50], where one estimates a background model of a scene and then discriminates moving objects by subtracting the model from new frames. The paper [46] provides an overview of robust PCA methods for video surveillance applications.

### 4.1.1 Background

There has been substantial work on foreground-background separation. For example, in [49] the authors propose a non-parametric model for background subtraction, and a probabilistic background model for tracking applications is developed in [51]. Alternatively, supervised approaches like GMM [52] learn a model of the background from labeled training data. Other lines of research have focused on performing background subtraction when the background is known to contain dynamic elements. Examples include a motion-based model [53] that utilizes adaptive kernel density estimation and an online autoregressive model [54] for modeling and subtracting dynamic backgrounds from scenes. In [55] a robust Kalman filter-based approach is developed to segment foreground objects from dynamic textured backgrounds. The paper [50] surveys classic background subtraction methods.

More recently, robust PCA methods have been proposed [31, 32, 56, 57] that decompose video into a low-rank component containing the background and a spatially sparse component that captures the foreground of the scene. Typically the original robust PCA problem (4.1) is extended to the noisy case by relaxing the equality constraint to an inequality constraint, e.g., as in [33], or by adding a data fidelity term and solving an unconstrained problem. Of particular interest in this work is the robust PCA formulation proposed in [1], which, although it was presented for the specific application of dynamic medical imaging, proposes a robust PCA framework applicable for general inverse problems. We refer to this model as the RPCA method.

Although standard sparsity-based foreground models are effective in the noiseless scenario, they are unable to distinguish foreground from sparse corruptions. In this context, models employing total variation (TV) have been proposed to model the spatial continuity of the foreground of a scene [56, 58]. Recently, the TVRPCA method [59] was proposed to separate dynamic background from moving objects using TV-based regularization, which demonstrates that TV-based models can effectively distinguish foreground from sparse corruptions.

Another important class of foreground-background separation models are those that can handle dynamic scenes arising, e.g., from moving camera video. In such cases, the background of the raw video may not be low-rank, so care is required to map the problem to an appropriate model that recovers low-rank structure. One approach to moving camera video is to adopt an online learning framework where batches of frames are sequentially processed and the foreground-background model is sequentially updated based on the latest batch. One popular approach is GRASTA [60, 61], which models the background as a subspace on the Grassmannian manifold and develops an iterative algorithm for tracking the low-rank subspace. A recent online method for low-rank and sparse decomposition is REProCS [62, 63]. Other methods use parameteric models to estimate the transformations that describe the motion in the scene. A robust PCA-based model was proposed in [64] that iteratively estimates the decomposition along with the parameters of an affine transformation model, but this approach considers only the intersection (common view) of the scene. The state-of-the-art method in this area is DECOLOR [65], which employs a robust PCA model with $\ell_0$-based regularization and a Markov random field model to jointly estimate the dynamic background and the support of the foreground.

### 4.1.2 Contributions

In this chapter, we propose a robust foreground-background separation method based on the robust PCA framework that can decompose a corrupted video with freely moving camera into a panoramic low-rank background component and a smooth foreground component. Our algorithm proceeds by registering the frames of the raw video to a common reference perspective and then minimizing a modified robust PCA cost that accounts for the unobserved data resulting from the partially overlapping views of the registered frames.

Our proposed method advances the state-of-the-art in several key aspects. First, our method produces a panoramic background component that spans the entire field of view, whereas existing parametric models typically only estimate the subspace spanning the intersection of the views. This panoramic property is useful because it allows one to produce a denoised version of the entire moving camera video. Our background model also employs an improved low-matrix estimator (OptShrink) [26] that has been shown to yield superior subspace estimate in practice compared to singular value thresholding-based approaches [6, 16]. Second, our method separates the dynamic foreground of a scene from sparse corruptions using TV regularization; our numerical experiments indicate that our formulation produces more accurate foreground estimates compared to existing TV-based methods. We account for the deforming view in the registered frames by considering a weighted total variation penalty that omits differences involving unobserved pixels, and we propose an efficient algorithm for minimizing this objective.

In this chapter, we perform extensive numerical experiments comparing our proposed method to state-of-the art methods in both the static and moving camera settings. In our numerical experiments, we consider multiple corruption models, including dense noise, sparse outliers, and missing data. In the latter case, we extend both our method and the state-of-the-art methods to model missing data to facilitate this numerical study. We also improve the computational efficiency of the total variation-related components of our proposed method.

### 4.1.3 Organization

The chapter is organized as follows. In Section 4.2, we describe our video registration strategy. Section 4.3 formulates our proposed augmented robust PCA model, and we present our algorithm for solving it in Section 4.4. In Section 4.5, we provide extensive numerical experiments that demonstrate the state-of-the-art performance of our method compared to existing methods on static and moving camera video under a variety of corruption models. Finally, Section 4.6 concludes and discusses opportunities for future work.

## 4.2 Video Registration

The vast majority of video data gathered today is captured by moving (e.g., handheld) cameras. To process this data in a robust PCA framework, our approach is to register the raw video—i.e., map the frames to a common reference perspective—and then jointly process the registered data. In this work, we adopt the standard perspec-

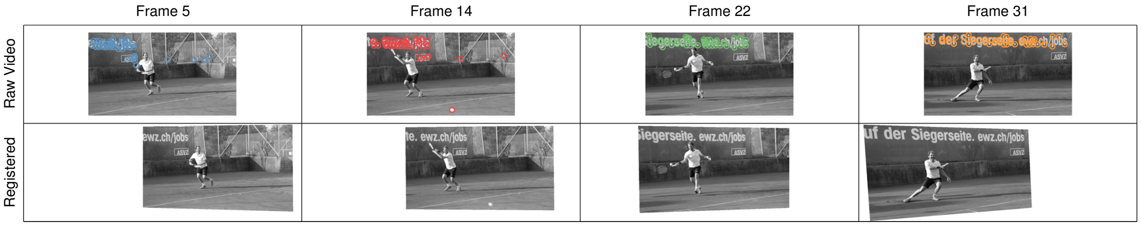


Figure 4.1: The video registration process. The top row depicts raw video frames $F_k$ with SURF features annotated. The bottom row depicts the corresponding registered frames $\widetilde{F}_k$ computed via (4.5). The $k$th column of the mask matrix $M \in \{0,1\}^{mn \times p}$ encodes the support of $\widetilde{F}_k$ within the aggregate view; i.e., $M_{ik} = 0$ for unobserved pixels, which are represented by white space in the registered frames above.

tive projection model [66], which relates different views of a scene via homographic transformations.

### 4.2.1 Registering Two Frames

Consider a point $(x, y)$ in a frame that is known to correspond to a point $(\widetilde{x}, \widetilde{y})$ in another frame. Under a planar surface model, one can relate the points via the projective transformation

$$\kappa \widetilde{p} = H^T p, \tag{4.2}$$

where $\widetilde{p} = [\widetilde{x}, \widetilde{y}, 1]^T$, $p = [x, y, 1]^T$, $\kappa \neq 0$ is an arbitrary scaling constant, and $H \in \mathbb{R}^{3 \times 3}$ with $H_{33} = 1$ is the unknown projective transformation matrix. Given $d > 3$ correspondences $\{(x_i, y_i) \mapsto (\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^d$, one can estimate $H$ in a least squares sense by minimizing [66]

$$\min_h \|Ah\|^2 \quad \text{s.t.} \quad h_9 = 1, \tag{4.3}$$

where $h = \mathbf{vec}(H)$ is the vectorized version of $h$ formed by stacking the columns of $H$ into a vector, $A^T = \left[A_1^T, \ldots, A_d^T\right]$, and

$$A_i = \begin{bmatrix} 0 & p_i^T & -\widetilde{y}_i p_i^T \\ p_i^T & 0 & -\widetilde{x}_i p_i^T \end{bmatrix} \in \mathbb{R}^{2 \times 9}. \tag{4.4}$$

The solution to (4.3) is the smallest right singular vector of $A$, scaled so that the last element is 1.

To estimate $H$ in practice, one must also generate correspondences $(x_i, y_i) \mapsto (\widetilde{x}_i, \widetilde{y}_i)$ between points in the frames. In this work, we adopt the standard procedure [66] of computing SURF features [67] for each frame and then using the RANSAC [68] algorithm to find a robust subset of the correspondences that produce a solution $\hat{H}$

to (4.3) with small cost. Importantly, this robust approach can generate accurate transformations in the presence of corruptions in the raw video.

### 4.2.2 Registering a Video

One can readily extend the two-frame registration procedure from Section 4.2.1 to a video by iteratively constructing homographies $H_k := H_{k \mapsto k+1}$ between frames $k$ and $k+1$ of the video and then composing the homographies to map all $p$ frames to an anchor frame $\widetilde{k} \in \{1, \ldots, p\}$. Here, we choose the middle frame $\widetilde{k} = \lfloor p/2 \rfloor$. Consecutive frames of a video are highly correlated, so the homographies $H_k$ can be computed with high accuracy.

Let $F_1, \ldots, F_p \in \mathbb{R}^{a \times b}$ denote the frames of a moving camera video, and denote by $\mathcal{H}_k := \mathcal{H}_{k \mapsto k+1}$ the linear transformation that applies the projective transformation (4.2) defined by $H_k$ to each pixel of $F_k$. One can register the frames with respect to anchor frame $\widetilde{k}$ by computing

$$\widetilde{F}_k = \begin{cases} (\mathcal{H}_{\widetilde{k}-1} \circ \mathcal{H}_{\widetilde{k}-2} \circ \cdots \circ \mathcal{H}_k)(F_k) & k < \widetilde{k}, \\ F_k & k = \widetilde{k}, \\ (\mathcal{H}_{\widetilde{k}}^{-1} \circ \mathcal{H}_{\widetilde{k}+1}^{-1} \circ \cdots \circ \mathcal{H}_{k-1}^{-1})(F_k) & k > \widetilde{k}, \end{cases} \tag{4.5}$$

for each $k = 1, \ldots, p$. The above procedure yields $\widetilde{F}_1, \ldots, \widetilde{F}_p \in \mathbb{R}^{m \times n}$, a collection of registered frames in a common perspective, where $m$ and $n$ are the height and width of the region defined by the union of the registered frame extents. See Figure 4.1 for a graphical depiction of this procedure applied to a moving camera video.

The registered frames $\widetilde{F}_k$ form a static camera video in the sense that a given coordinate $(\widetilde{F}_k)_{ij}$ now corresponds to the same spatial location for each frame $k$. If the composite projective transformation mapping $F_k$ to $\widetilde{F}_k$ is not the identity transformation, the matrix $\widetilde{F}_k$ will contain some pixels that correspond to locations outside the view of the original frame $F_k$. Without loss of generality, we set such unobserved pixels to zero in $\widetilde{F}_k$.

## 4.3 Problem Formulation

In this section, we describe our proposed robust PCA framework for panoramic foreground-background separation. We first describe our model, discuss our treatment of total variation for moving camera video, and then we present our problem formulation. We discuss our algorithm for solving the proposed problem in Section 4.4.

### 4.3.1 Data Model

Given the registered frames $\widetilde{F}_1, \ldots, \widetilde{F}_p \in \mathbb{R}^{m \times n}$ of a moving camera video computed as in Section 4.2, we construct the data matrix $Y \in \mathbb{R}^{mn \times p}$

$$Y = \left[\mathbf{vec}(\widetilde{F}_1) \ \ldots \ \mathbf{vec}(\widetilde{F}_p)\right] \tag{4.6}$$

whose columns are the vectorized registered frames. Associated with $Y$, we also define the mask matrix $M \in \{0,1\}^{mn \times p}$ whose columns encode the support of the registered frames in the aggregate view. See Figure 4.1 for a graphical depiction.[1]

The representation (4.6) is useful because each row of $Y$ corresponds to a fixed point in space, so we have effectively reduced the moving camera foreground-background separation problem to a static camera problem with incomplete observations (corresponding to the zeros in $M$). Thus, with suitable modifications to account for the missing data, we can readily apply ideas from standard static camera foreground-background separation. In particular, our approach is to model the observed data $Y$ with a decomposition of the form

$$\mathcal{P}_M(Y) = \mathcal{P}_M(L + S + E + N), \tag{4.7}$$

where $\mathcal{P}_M$ denotes the orthogonal projection onto $M$:

$$[\mathcal{P}_M(X)]_{ij} = \begin{cases} X_{ij} & M_{ij} = 1 \\ 0 & M_{ij} = 0. \end{cases} \tag{4.8}$$

In (4.7), $L$ represents the (registered) background of the video, and $S$ represents the foreground. Furthermore, the matrix $E$ captures possible sparse corruptions in the video, and $N$ captures possible dense corruptions. Note that the projection operators in (4.7) exclude unobserved pixels from our model, so we are not attempting to impute the unobserved pixels of the scene; rather we are expressing the moving camera video as a "static" space-time matrix where each row corresponds to a fixed point in space.

Since our data is registered, the background will have high temporal correlation and thus can be well-modeled as a low-rank matrix [59]. In the standard robust PCA

---

[1] When processing video that is known (or modeled) to have a static camera, one can omit the video registration step and directly construct the data matrix $Y$ by vectorizing each frame of the raw video. In this case, the corresponding mask matrix $M$ is the all-ones matrix.

model [31], the foreground component $S$ is modeled as a sparse matrix. However, we are interested in reliably estimating the foreground in the presence of sparse corruptions, so a sparse model for $S$ will be indistinguishable from the sparse corruptions. Instead, we model $S$ as a smoothly-varying matrix and, motivated by the recent TVRPCA method [59], use a total variation-based regularization framework to estimate $S$. In the moving camera setting, we will consider a weighted total variation penalty that avoids penalizing first differences involving unobserved pixels. Motivated by the vast compressed sensing literature [69,70], we model $E$ as a sparse matrix and employ $\ell_1$-based regularization to estimate it. Finally, we model $N$ as a dense noise matrix, and we estimate it by imposing the familiar least-squares-based regularization. We explicitly describe the optimization problem that we employ to learn the model (4.7) in Section 4.3.3.

### 4.3.2 Weighted Total Variation

Total variation regularization is a ubiquitous method in image and video processing for reconstructing signals corrupted by noise [71–73]. In particular, in this work, given a matrix $X \in \mathbb{R}^{mn \times p}$ whose columns contain the vectorized $m \times n$ frames of a video, we consider the weighted anisotropic TV of $X$:

$$\mathbf{TV}(X) = \sum_{ijk} \left( w^x_{ijk} |x_{i+1jk} - x_{ijk}| + w^y_{ijk} |x_{ij+1k} - x_{ijk}| + w^z_{ijk} |x_{ijk+1} - x_{ijk}| \right). \quad (4.9)$$

Here, we use a slight abuse of notation by implicitly referencing the vectorized video $x = \mathbf{vec}(X) \in \mathbb{R}^{mnp}$ in the definition and using $x_{ijk}$ to denote the pixel $(i, j)$ from frame $k$ of the video—i.e., the $(i{+}m(j{-}1),\ k)$ entry of $X$. In (4.9), $w^x_{ijk}$, $w^y_{ijk}$, and $w^z_{ijk}$ are fixed $\{0, 1\}$ weights that omit first differences involving the unobserved pixels that lie outside the extent of the registered frames. These weights can be readily computed from the mask matrix $M$ (see Figure 4.1). We omit the summation indices in (4.9) for brevity, but it should be understood that we are not considering the first differences corresponding to circular boundary conditions in our model (e.g., $x_{1jk} - x_{mjk}$).

Smoothly varying objects with few sharp edges will have low TV, so (4.9) is a good model for the foreground of a video [71–73]. Conversely, sparse corruptions will have very high TV, so it is reasonable to expect that (4.9) will be able to distinguish the foreground from sparse corruptions.

Subsequently, we will refer to (4.9) as three-dimensional (3D) TV because it penalizes both the spatial first differences between neighboring pixels in a given frame and the temporal differences between a given pixel in consecutive frames. Such a model may be appropriate for datasets with high temporal correlation, e.g., due to a slowly moving camera. However, in other cases, it may be preferable to omit the temporal differences from (4.9) by setting $w^z_{ijk} = 0$. We refer to this latter model as 2D TV.

#### 4.3.2.1 Matrix-Vector Representation

When describing our proposed algorithm in Section 4.3.3, it will be convenient for us to express the TV penalty (4.9) using matrix-vector operations.

In the 1D case, one can compute the first differences of $z \in \mathbb{R}^n$ with the matrix-vector product $D_n z$, where $D_n \in \mathbb{R}^{n \times n}$ is the circulant first differences matrix

$$D_n = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & & -1 \end{bmatrix}. \tag{4.10}$$

Note that we include the first difference $[D_n z]_n = z_1 - z_n$ corresponding to circular boundary conditions in this computation, although we omit these circular differences from our TV penalty (4.9). We do this because we will later leverage the fact that $D_n$ is a circulant matrix. Using this notation, we can write the 1D TV penalty as $\mathbf{TV}(z) = \|W D_n z\|_1$, where $W$ is the diagonal matrix with $W_{kk} = 1$ for $k < n$ and $W_{nn} = 0$, which omits the circular boundary difference. In general, one can omit other first differences by setting the corresponding diagonal entry of $W$ to zero.

In the 3D case when $x \in \mathbb{R}^{mnp}$, one can compute the first differences along each dimension of the vectorized $m \times n \times p$ tensor by computing the matrix-vector product $Cx$, where $C \in \mathbb{R}^{3mnp \times mnp}$ is the matrix

$$C = \begin{bmatrix} I_p \otimes I_n \otimes D_m \\ I_p \otimes D_n \otimes I_m \\ D_p \otimes I_n \otimes I_m \end{bmatrix}. \tag{4.11}$$

In (4.11), $I_n$ is the $n \times n$ identity matrix and $\otimes$ denotes the Kronecker product. Again, we have included the first differences corresponding to circular boundary conditions for mathematical convenience so that $C$ is constructed from Kronecker products of circulant matrices. Using this definition, we can write

$$\mathbf{TV}(X) = \|WCx\|_1, \tag{4.12}$$

where $W$ is the diagonal $\{0, 1\}$ matrix that omits first differences involving unobserved pixels and those corresponding to circular boundary conditions. Concretely, one has

$$W = \mathbf{diag}(\mathbf{vec}(w^x), \mathbf{vec}(w^y), \mathbf{vec}(w^z)), \tag{4.13}$$

where $w^x$, $w^y$, and $w^z$ are the $m \times n \times p$ tensors containing the weights from (4.9) and we set $w^x_{mjk} = w^y_{ink} = w^z_{ijp} = 0$ for all $ijk$ to omit the circular boundaries. Here, $\mathbf{vec}(\cdot)$ converts an $m \times n \times p$ tensor into a vector by stacking the columns of each frame into length-$mn$ vectors and then stacking these vectors to form a single length-$mnp$ vector, and $\mathbf{diag}(\cdots)$ constructs a diagonal matrix from the vector formed by concatenating its vector arguments into a single vector. We will rely on the equivalent representation (4.12) of (4.9) when presenting our proposed algorithm in Section 4.3.3.

### 4.3.3 Proposed Optimization Problem

To learn a decomposition of the form (4.7), we propose to solve the augmented robust PCA problem

$$\begin{aligned} \min_{L,S,E,N} \quad & \lambda_L \|L\|_\star + \lambda_S \mathbf{TV}(S) + \lambda_E \|E\|_1 + \tfrac{1}{2}\|N\|_F^2 \\ \text{s.t.} \quad & \mathcal{P}_M(Y) = \mathcal{P}_M(L + S + E + N). \end{aligned} \tag{4.14}$$

Equivalently, one can eliminate matrix $N$ in (4.14) and instead consider the unconstrained problem

$$\min_{L,S,E} \ \tfrac{1}{2}\|\mathcal{P}_M(Y - L - S - E)\|_F^2 + \lambda_L \|L\|_\star + \lambda_S \mathbf{TV}(S) + \lambda_E \|E\|_1. \tag{4.15}$$

Here, $\mathbf{TV}(\cdot)$ is the weighted TV penalty defined in (4.9) and the parameters $\lambda_L, \lambda_S, \lambda_E \geq 0$ are regularization parameters that control the relative contribution of each term to the overall cost. It is well-known that each term in (4.15) is a convex function, so (4.15) is a convex problem.

As discussed in Section 4.3.1, the $L$ component of our model represents the background of the video, which we model as a low-rank matrix and thus regularize with the nuclear norm. The $S$ component represents the foreground, which we model as a smoothly-varying matrix with sharp edges and regularize with the weighted TV penalty. The $E$ component represents sparse corruptions, which we model as a sparse matrix and regularize with the familiar $\ell_1$ penalty. The first term in (4.15) is a data fidelity term that forces the decomposition $L+S+E$ to approximately agree with the data $Y$ at the observed pixel locations encoded by the mask matrix $M$. The choice of $\|\cdot\|_F^2$ for the data fidelity term captures residual dense corruptions in the data.

Our proposed problem (4.15) differs from the recent RPCA [31], TVRPCA [59], and DECOLOR [65] methods in several key ways. First, in the moving camera setting, our frame registration and masking strategy allows us to reconstruct the full field of view of the scene, while DECOLOR only estimates the overlapping (intersection) view. Second, we regularize the foreground component of our model using TV alone, while the TVRPCA method includes both $\ell_1$ and TV-based regularization on its foreground model, which is overly restrictive because the foreground need not be spatially sparse. Finally, our model improves on the standard RPCA model by including the TV-regularized component to disentangle the foreground $S$ from possible sparse corruptions, which are isolated in the $E$ component.

## 4.4 Algorithm and Properties

In this section we derive our algorithm for solving (4.15), present an important modification to the low-rank update, and discuss the properties of our algorithm.

### 4.4.1 Proximal Gradient Updates

We use the proximal gradient method [25] to minimize (4.15). The proximal gradient method is an iterative algorithm for solving problems of the form $f(X)+g(X)$, where $f$ is convex and differentiable and $g$ is convex and has an easily computable proximal operator

$$\mathbf{prox}_g(Y) := \arg\min_X \ \tfrac{1}{2}\|Y-X\|_F^2 + g(X). \tag{4.16}$$

The proximal gradient method prescribes updates of the form

$$X^{k+1} = \mathbf{prox}_{\tau^k g}(X^k - \tau^k \nabla f(X^k)), \tag{4.17}$$

where $\nabla f$ denotes the gradient of $f$ and $\tau^k > 0$ is a chosen step size. It is known [25] that the proximal gradient method converges when a constant step size $\tau^k = \tau < 2/L_{\nabla f}$ is used, where $L_{\nabla f}$ is the Lipschitz constant for $\nabla f$. In fact, the iterates $X^k$ will monotonically decrease the cost when a constant step size $\tau \leq 1/L_{\nabla f}$ is used [25].

To map (4.15) into a suitable form for proximal gradient, we identify $f(L,S,E) = \frac{1}{2}\|\mathcal{P}_M(Y-L-S-E)\|_F^2$ and $g(L,S,E) = \lambda_L\|L\|_\star + \lambda_S\mathbf{TV}(S) + \lambda_E\|E\|_1$, which we regard as functions of the single variable $X = [L\ S\ E]$. Under these definitions, a simple computation shows that $\nabla f = [\nabla f_L\ \nabla f_S\ \nabla f_E]$, where $\nabla f_L(L,S,E) = \nabla f_S(L,S,E) = \nabla f_E(L,S,E) = \mathcal{P}_M(L+S+E-Y)$. Since $g$ is the sum of three functions, its proximal operator (4.16) can be computed separately for each component. Thus our proximal update scheme for (4.15) can be written as

$$\begin{aligned}
U^{k+1} &= \mathcal{P}_M(L^k + S^k + E^k - Y) \\
L^{k+1} &= \mathbf{prox}_{\tau^k\lambda_L\|\cdot\|_\star}(L^k - \tau^k U^{k+1}) \\
S^{k+1} &= \mathbf{prox}_{\tau^k\lambda_S\mathbf{TV}}(S^k - \tau^k U^{k+1}) \\
E^{k+1} &= \mathbf{prox}_{\tau^k\lambda_E\|\cdot\|_1}(E^k - \tau^k U^{k+1}),
\end{aligned} \tag{4.18}$$

where we have introduced the auxiliary variable $U$ for notational convenience. It is straightforward[2] to show that $L_{\nabla f} = 3$, so a constant step size $\tau < 2/3$ suffices to guarantee convergence.

The proximal operators for the $L$ and $E$ updates in (4.18) have simple, closed-form solutions. Indeed, it is well-known that the solution to the nuclear-norm-regularized problem

$$\arg\min_L\ \tfrac{1}{2}\|Z-L\|_F^2 + \lambda\|L\|_\star \tag{4.19}$$

is given by the singular value thresholding operator [24, 31]

$$\mathbf{SVT}_\lambda(Z) := \sum_i (\sigma_i - \lambda)_+ u_i v_i^T, \tag{4.20}$$

where $Z = \sum_i \sigma_i u_i v_i^T$ is the singular value decomposition (SVD) of $Z$, and $(\cdot)_+ = \max(\cdot, 0)$. The solution to the $\ell_1$-regularized problem

$$\arg\min_E\ \tfrac{1}{2}\|Z-E\|_F^2 + \lambda\|E\|_1 \tag{4.21}$$

is given by the elementwise soft thresholding operator [31]

$$\mathbf{soft}_\lambda(z) = \mathbf{sign}(z)(|z| - \lambda)_+. \tag{4.22}$$

---

[2] One can write $f(x) = \|y - [A\ A\ A]x\|_2^2$, where $y = \mathbf{vec}(\mathcal{P}_M(Y))$ are the vectorized observations, $x = [\mathbf{vec}(L)^T\ \mathbf{vec}(S)^T\ \mathbf{vec}(E)^T]^T$ is the concatenation of the vectorized model variables, and $A = \mathbf{diag}(\mathbf{vec}(M))$ is the diagonal $\{0,1\}$ matrix that encodes the observation mask $M$. Therefore $L_{\nabla f} = \|[A\ A\ A]\|^2 = 3\|A\|^2 = 3$.

The proximal operator for the weighted TV penalty (4.9) does not have a closed-form solution in general,[3] so we instead refer to this proximal operator implicitly as the solution to the (weighted) total variation denosing (TVDN) problem

$$\mathbf{TVDN}_\lambda(Z) := \arg\min_S \ \tfrac{1}{2}\|Z - S\|_F^2 + \lambda \mathbf{TV}(S). \tag{4.23}$$

Using the above results and notation, we can express the proximal updates (4.18) as

$$\begin{aligned} U^{k+1} &= \mathcal{P}_M(L^k + S^k + E^k - Y) \\ L^{k+1} &= \mathbf{SVT}_{\tau^k \lambda_L}(L^k - \tau^k U^{k+1}) \\ S^{k+1} &= \mathbf{TVDN}_{\tau^k \lambda_S}(S^k - \tau^k U^{k+1}) \\ E^{k+1} &= \mathbf{soft}_{\tau^k \lambda_E}(E^k - \tau^k U^{k+1}), \end{aligned} \tag{4.24}$$

where it remains to describe how to compute $S^{k+1}$.

### 4.4.2 Total Variation Denoising Updates

Using the notation from Section 4.3.2, we can equivalently express the operator $\mathbf{TVDN}_\lambda(Z)$ as the solution to the vector-valued problem

$$\min_s \ \tfrac{1}{2}\|z - s\|_2^2 + \lambda \|WCs\|_1, \tag{4.25}$$

where $z = \mathbf{vec}(Z)$ and the matrices $W$ and $C$ are defined as in (4.12). We solve (4.25) using the alternating direction method of multipliers (ADMM) [74], a powerful general-purpose method for minimizing convex problems of the form $f(x) + g(x)$ subject to linear equality constraints. To apply ADMM, we perform the variable split $v = Cs$ and write (4.25) as the equivalent constrained problem

$$\begin{aligned} \min_s \ & \tfrac{1}{2}\|z - s\|_2^2 + \lambda \|Wv\|_1 \\ \text{s.t.} \ & Cs - v = 0, \end{aligned} \tag{4.26}$$

---

[3] There is a closed-form solution in the special case of static camera video when circular boundary conditions are allowed in the TV penalty. In this case, the $W$ matrix in (4.13) is the identity matrix and our proposed ADMM updates in Section 4.4.2 in fact converge in one iteration.

which is in the standard form for ADMM.[4] The ADMM updates for (4.26) are

$$\begin{aligned} s^{k+1} &= \arg\min_{s} \; \tfrac{1}{2}\|z - s\|_2^2 + \tfrac{\rho}{2}\|Cs - v^k + u^k\|_2^2 \\ v^{k+1} &= \arg\min_{v} \; \lambda\|Wv\|_1 + \tfrac{\rho}{2}\|Cs^{k+1} - v + u^k\|_2^2 \\ u^{k+1} &= u^k + Cs^{k+1} - v^{k+1} \end{aligned} \tag{4.27}$$

with parameter $\rho > 0$. The $s$ update in (4.27) is a least squares problem with normal equation

$$(I + \rho C^T C)s^{k+1} = z + \rho C^T(v^k - u^k), \tag{4.28}$$

so the solution could in principal be obtained by computing the matrix inverse $(I + \rho C^T C)^{-1}$. However, this matrix has a special block-circulant structure that admits a fast closed-form solution using fast Fourier transforms (FFTs). Indeed, the exact solution can be computed [59, 75] as

$$s^{k+1} = \mathcal{F}_3^{-1}\left[\frac{\mathcal{F}_3(z + \rho C^T(v^k - u^k))}{1 + \rho\mathcal{F}_3(c)}\right], \tag{4.29}$$

where $\mathcal{F}_3 : \mathbb{R}^{mnp} \to \mathbb{R}^{mnp}$ denotes the operator that reshapes its input into an $m \times n \times p$ tensor, computes the 3D Fourier transform, and vectorizes the result; $c$ is the first column of $C^T C$; and division is performed elementwise. The denominator of (4.29) is a constant and can be precomputed.

The vector $c \in \mathbb{R}^{mnp}$ has special structure. Indeed, one can show that

$$c = \mathbf{vec}(|\mathcal{F}_1(d_m)|^2 \circ |\mathcal{F}_1(d_n)|^2 \circ |\mathcal{F}_1(d_p)|^2), \tag{4.30}$$

where $\mathcal{F}_1(\cdot)$ denotes the 1D Fourier transform of a vector; $|\cdot|^2$ denotes elementwise squared-magnitude; the vector $d_n = [-1 \; 0 \; \ldots \; 0 \; 1]^T \in \mathbb{R}^n$ is the first column of (4.10); and $T = a \circ b \circ c$ is the order three tensor sum of vectors $a$, $b$, and $c$—i.e., the tensor with entries $T_{ijk} = a_i + b_j + c_k$.

The $W$ matrix in the $v$-update of (4.27) is a diagonal matrix, so the $v$ update has a simple closed-form solution involving elementwise soft-thresholding with an entry-dependent threshold, which we write as

$$v^{k+1} = \mathbf{soft}(Cx^{k+1} + u^k, \; (\lambda/\rho)w), \tag{4.31}$$

---

[4] Note that we choose the split $v = Cs$ rather than the split $v = WCs$ because the resulting ADMM updates in the former case have efficient closed-form solutions that leverage the block-circulant structure of $C$ (4.11).

where $\mathbf{soft}(x, y) = \mathbf{sign}(x) \odot (x - y)_+$ is interpreted elementwise for vectors and $w$ is the main diagonal of $W$.

### 4.4.3 Improved Low-Rank Update

Motivated by recent work [6,16], we propose to replace the SVT operator in the $L$ update of (4.24) with an improved low-rank matrix estimator (OptShrink) [26] that has been shown to produce superior low-rank components in practice. Our proposed (modified) update scheme thus becomes

$$\begin{aligned} U^{k+1} &= \mathcal{P}_M(L^k + S^k + E^k - Y) \\ L^{k+1} &= \mathbf{OptShrink}_r\left(L^k - \tau^k U^{k+1}\right) \\ S^{k+1} &= \mathbf{TVDN}_{\tau^k \lambda_S}(S^k - \tau^k U^{k+1}) \\ E^{k+1} &= \mathbf{soft}_{\tau^k \lambda_E}(E^k - \tau^k U^{k+1}). \end{aligned} \tag{4.32}$$

In (4.32), $\mathbf{OptShrink}(\cdot)$ is the low-rank matrix estimator defined for a given parameter $r > 0$ as

$$\mathbf{OptShrink}_r(Z) = \sum_{i=1}^{r} \left(-2\frac{D_{\mu_Z}(\sigma_i)}{D'_{\mu_Z}(\sigma_i)}\right) u_i v_i^H, \tag{4.33}$$

where $Z = \sum_i \sigma_i u_i v_i^T$ is the SVD of $Z \in \mathbb{R}^{a \times b}$. In (4.33), the $D$-transform is defined for a given probability measure $\mu$ as

$$D_\mu(z) = \left[\int \frac{z}{z^2 - t^2} \, \mathrm{d}\mu(t)\right] \times \left[c \int \frac{z}{z^2 - t^2} \, \mathrm{d}\mu(t) + \frac{1-c}{z}\right], \tag{4.34}$$

where $D'_\mu(z)$ is the derivative of $D_\mu(z)$ with respect to $z$, $c = \min(a, b)/\max(a, b)$, and $\mu_Z(t) = \frac{1}{q-r}\sum_{i=r+1}^{q} \delta(t - \sigma_i)$ is the empirical mass function of the noise-only singular values of $Z$ with $q = \min(a, b)$. Note that the integrals in the $D$-transform terms in (4.33) reduce to summations for this choice of $\mu_Z$, so they can be computed efficiently.

The $\mathbf{OptShrink}_r(Z)$ operator computes the rank $r$ truncated SVD of $Z$ and then applies the shrinkage function defined by the parenthesized term in (4.33) to the leading singular values. We refer to the $D$-transform term as a shrinkage function because it shrinks its argument towards zero [26]. In contrast, the $\mathbf{SVT}_\lambda(Z)$ operator (4.20) applies a constant shrinkage level $\lambda$ to all singular values.

The OptShrink estimator provides two key benefits over SVT. First, it applies a data-driven shrinkage to the singular value spectrum of its argument, the form of which is imputed from the non-leading (noise) singular values. Generically, a smaller shrinkage is applied to larger—and hence more-informative—singular values and a comparatively larger shrinkage to smaller singular values. The effect of this nonlinear shrinkage is to produce an improved estimate of the underlying low-rank matrix embedded in the data [26]. See Chapter II of this thesis for further details. Second, OptShrink has a single parameter $r$ that directly specifies the rank of the output matrix. In the context of this work, it is very natural to set the rank parameter. Indeed, since our data $Y$ from (4.6) is registered, we can model the background of the registered video as static. In this case, the low-rank component $L$ of our model (4.7) should ideally be a rank-1 matrix whose columns are repeated (up to scaling) vectorized copies of the static background image. In practice, the registered background may not be perfectly static, but it will still have high temporal correlation, so a small rank ($r = 2, 3, \ldots$) will often suffice. In fact, our numerical experiments in Section 4.5 show that the OptShrink-based updates are robust to rank overestimation in the sense that performance degrades slowly as $r$ increases beyond its optimal value.

OptShrink is data-driven and thus does not correspond to the proximal operator of a penalty function $\phi(L)$, so the updates (4.32) are not proximal gradient updates for a cost function like (4.15). Nonetheless, recent alternating minimization schemes involving OptShrink [6,16] have proven to be numerically stable and yield convergent iterate sequences, and our numerical experiments in Section 4.5 corroborate these findings.

### 4.4.4 Accelerated Proximal Gradient Updates

The proximal gradient-based updates (4.32) are perfectly suitable for use in practice. However, one can also consider various accelerated proximal gradient algorithms that include an extrapolation step in (4.17) to improve the convergence rate of the iterates. In this chapter, we adopt the accelerated proximal gradient algorithm due to Nesterov [76] in place of the (standard) proximal gradient updates in (4.32), which have the optimal worst-case convergence rate. Nesterov-accelerated proximal updates have enjoyed fruitful use in practice, e.g., the well-known FISTA algorithm [36]. In any case, we propose to use the following accelerated iterations with OptShrink-based low-rank updates:

$$\begin{aligned}
t^{k+1} &= \frac{1}{2}\Big(1 + \sqrt{1 + 4(t^k)^2}\Big) \\
\overline{L}^{k+1} &= L^k + \frac{t^k - 1}{t^{k+1}}\left(L^k - L^{k-1}\right) \\
\overline{S}^{k+1} &= S^k + \frac{t^k - 1}{t^{k+1}}\left(S^k - S^{k-1}\right) \\
\overline{E}^{k+1} &= E^k + \frac{t^k - 1}{t^{k+1}}\left(E^k - E^{k-1}\right) \\
\overline{U}^{k+1} &= \mathcal{P}_M(\overline{L}^{k+1} + \overline{S}^{k+1} + \overline{E}^{k+1} - Y) \\
L^{k+1} &= \mathbf{OptShrink}_r\left(\overline{L}^{k+1} - \tau^k \overline{U}^{k+1}\right) \\
S^{k+1} &= \mathbf{TVDN}_{\tau^k \lambda_S}(\overline{S}^{k+1} - \tau^k \overline{U}^{k+1}) \\
E^{k+1} &= \mathbf{soft}_{\tau^k \lambda_E}(\overline{E}^{k+1} - \tau^k \overline{U}^{k+1}),
\end{aligned} \tag{4.35}$$

initialized with $t^0 = 0$. If one uses the SVT-based $L$ update from (4.24) instead of the above OptShrink-based $L$ update, then the accelerated proximal gradient updates are guaranteed to minimize the cost (4.15) when a constant step size $0 < \tau \leq 1/L_{\nabla f} = 1/3$ is used [25]. The modified updates (4.35) do not correspond to proximal updates for a fixed cost function, but we nonetheless observe that constant step sizes $0 < \tau \leq 1/3$ work well in practice.

Algorithm 4.2 summarizes the proposed algorithm with accelerated proximal gradient steps and OptShrink-based low-rank updates. Henceforward, we refer to our method as Panoramic Robust PCA (PRPCA).

### 4.4.5 Complexity Analysis

We now analyze the computational complexity of our PRPCA method from Algorithm 4.2. For each outer iteration, the $U$ and $E$ updates require $O(mnp)$ operations, and the cost of computing the $L$ update is $O(m^2n^2p)$—the cost of computing the SVD of a tall $mn \times p$ matrix [77]. Finally, the cost of updating $E$ using the ADMM-based scheme (4.27) is $O(Kmnp\log(mnp))$, where $K$ is the number of ADMM iterations applied and the per-iteration cost is determined by the cost of computing a 3D FFT of an $m \times n \times p$ tensor [78]. Therefore the overall per-iteration cost of our proposed algorithm is dominated by the cost of computing the SVD of a $mn \times p$ matrix, which is the same complexity as RPCA, TVRPCA, and most other robust PCA algorithms involving rank penalties.

In practice, moving camera video magnifies the size of the registered data $Y$

```
PRPCA Algorithm
Inputs: Video frames $F_1, \ldots, F_p$ and parameters $r > 0$, $\lambda_S > 0$, $\lambda_E > 0$, $\tau \leq 1/3$, $\rho > 0$, and $K > 0$
Compute registered frames $\widetilde{F}_1 \ldots \widetilde{F}_p$ via (4.5)
Construct $Y$ and $M$ matrices via (4.6)
Initialization: $U^0 = L^0 = Y$, $S^0 = E^0 = 0$, $t^0 = 0$, and $k = 0$
While not converged:
    1. Compute $t^k$, $\overline{L}^k$, $\overline{S}^k$, $\overline{E}^k$ and $\overline{U}^k$ according to (4.35)
    2. Update $L^k$ and $S^k$ via (4.35)
    3. Update $E^k$ by performing $K$ iterations of (4.27)
    4. $k = k + 1$
End
Outputs: Decomposition $\{L,\ S,\ E\}$
```

Figure 4.2: Summary of the proposed PRPCA algorithm.

processed by our algorithm compared to the data matrices of the other methods. Since the complexity is quadratic in the number of pixels, a twofold increase in pixels (substantial camera motion) would make our algorithm roughly four-times slower than the other methods.

## 4.5 Numerical Experiments

We evaluate the performance of our proposed PRPCA method by comparing to the recent RPCA [31], TVRPCA [59], and DECOLOR [65] methods on corrupted static camera videos. We then demonstrate the ability of our method to process corrupted moving camera videos, a scenario that the other methods cannot handle.

All methods under comparison are foreground-background separation methods, so they have components corresponding to the $L$ (background) and $S$ (foreground) components of our model. To facilitate a direct comparison, we repeat the cost functions of each method from their respective papers here and rename the optimization variables so that the corresponding background and foreground components of each method are denoted by $L$ and $S$, respectively. In each case, we also use the matrix $Y$ to denote the matrix whose columns contain the vectorized frames of the (possibly corrupted) video.[5]

---

[5] Note that the other methods do not employ our frame registration preprocessing step, so here $Y$ contains the vectorized raw video frames.

| Sequence | Proposed | | | RPCA | | | TVRPCA | | | DECOLOR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure |
| Hall | **38.94** | **37.98** | **0.60** | 27.12 | 32.63 | 0.19 | 36.50 | 37.42 | **0.60** | 27.02 | 31.63 | 0.17 |
| Fountain | **39.73** | **35.48** | **0.74** | 26.99 | 32.06 | 0.21 | 36.87 | **35.48** | 0.72 | 26.89 | 30.69 | 0.15 |
| Escalator | **33.15** | **31.56** | **0.72** | 23.45 | 26.27 | 0.35 | 30.91 | 30.96 | 0.69 | 23.27 | 22.17 | 0.25 |
| Water Surface | **42.14** | **36.96** | **0.94** | 22.92 | 31.45 | 0.40 | 40.14 | 36.81 | 0.82 | 22.12 | 20.66 | 0.26 |
| Shopping Mall | **40.26** | 39.83 | **0.74** | 25.06 | 34.62 | 0.31 | 37.43 | **40.88** | 0.73 | 25.01 | 31.42 | 0.26 |
| Average | **38.84** | **36.36** | **0.75** | 25.11 | 31.41 | 0.29 | 36.37 | 36.31 | 0.71 | 24.86 | 27.31 | 0.22 |

Table 4.1: Performance metrics for each method on sequences from the I2R dataset corrupted by 20% outliers.

The RPCA [1, 31] method minimizes the cost

$$\min_{L,S} \ \tfrac{1}{2}\|Y - L - S\|_F^2 + \lambda_L\|L\|_\star + \lambda_S\|S\|_1, \tag{4.36}$$

where $L$ is the low-rank background component and $S$ is the sparse foreground component.

The TVRPCA method minimizes the cost from Equation (7) of [59], which, in our notation, is

$$\begin{aligned} \min_{L,G,E,S} \quad & \|L\|_\star + \lambda_1\|G\|_1 + \lambda_2\|E\|_1 + \lambda_3\mathbf{TV}(S) \\ \text{s.t.} \quad & Y = L + G, \quad G = E + S. \end{aligned} \tag{4.37}$$

In (4.37), $L$ is the low-rank background component and $G$ is a residual matrix that is further decomposed into a smooth foreground component $S$ and a sparse error term $E$. Here, we reuse $\mathbf{TV}(\cdot)$ to denote the standard (unweighted) anisotropic total variation penalty.

The DECOLOR method minimizes the cost from Equation (20) of [65], which, in our notation, is

$$\min_{\tau,L,S} \ \tfrac{1}{2}\|\mathcal{P}_{S^\perp}(Y \circ \tau - L)\|_F^2 + \alpha\|L\|_\star + \beta\|S\|_1 + \gamma\mathbf{TV}(S). \tag{4.38}$$

In (4.38), $L$ is the low-rank (registered) background, $S_{ij} \in \{0, 1\}$ is the (registered) foreground mask, $S^\perp$ is the orthogonal complement of $S$, $\tau$ are the 2D parameteric transforms that register the input frames $Y$, and $\mathbf{TV}(\cdot)$ is again the standard (unweighted) anisotropic total variation penalty. Note that the DECOLOR method directly estimates the support of the foreground. Thus, to display a foreground component for DECOLOR, we plot $(Y - L \circ \tau^{-1}) \odot S$, the difference between the raw video and the estimated background restricted to the support of the estimated foreground mask.

| Sequence | Proposed | | | RPCA | | | TVRPCA | | | DECOLOR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure |
| Hall | **36.66** | **32.72** | 0.58 | 31.80 | 30.14 | 0.30 | 34.64 | 21.83 | **0.59** | 31.65 | 25.14 | 0.56 |
| Fountain | **38.14** | **30.05** | **0.74** | 34.57 | 29.35 | 0.35 | 36.45 | 24.22 | 0.70 | 36.51 | 25.54 | 0.71 |
| Escalator | **32.83** | **26.60** | **0.72** | 29.87 | 25.07 | 0.49 | 31.15 | 22.35 | 0.68 | 25.67 | 23.54 | **0.72** |
| Water Surface | **38.46** | **31.08** | **0.94** | 30.19 | 28.71 | 0.57 | 33.83 | 23.88 | 0.81 | 29.35 | 20.88 | 0.84 |
| Shopping Mall | **37.31** | **35.29** | **0.71** | 32.34 | 31.54 | 0.34 | 35.13 | 24.31 | **0.71** | 32.39 | 30.93 | **0.71** |
| Average | **36.68** | **31.15** | **0.74** | 31.75 | 28.96 | 0.41 | 34.24 | 23.32 | 0.70 | 31.11 | 25.21 | 0.71 |

Table 4.2: Performance metrics for each method on sequences from the I2R dataset corrupted by 30 dB Gaussian noise.

| Sequence | Proposed | | | RPCA | | | TVRPCA | | | DECOLOR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure |
| Hall | **37.25** | **36.58** | 0.58 | 27.64 | 30.75 | 0.27 | 31.02 | 32.58 | 0.35 | 29.69 | 33.17 | **0.65** |
| Fountain | **37.78** | **34.52** | **0.70** | 29.59 | 26.90 | 0.24 | 36.04 | 29.62 | 0.32 | 32.51 | 26.23 | 0.56 |
| Escalator | **30.87** | **28.95** | **0.70** | 21.85 | 23.05 | 0.30 | 24.09 | 24.89 | 0.38 | 23.53 | 24.99 | 0.41 |
| Water Surface | **40.00** | **34.99** | **0.93** | 31.93 | 29.50 | 0.33 | 33.57 | 30.03 | 0.70 | 28.79 | 18.42 | 0.17 |
| Shopping Mall | **37.70** | **39.87** | 0.73 | 28.03 | 32.62 | 0.35 | 31.70 | 34.09 | 0.46 | 29.65 | 34.05 | **0.76** |
| Average | **36.72** | **34.98** | **0.73** | 27.81 | 28.56 | 0.30 | 31.28 | 30.24 | 0.44 | 28.83 | 27.37 | 0.51 |

Table 4.3: Performance metrics for each method on sequences from the I2R dataset corrupted by 70% missing data.

| p | Proposed | | | RPCA | | | TVRPCA | | | DECOLOR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure |
| 10% | **41.48** | **39.37** | **0.60** | 30.35 | 32.67 | 0.27 | 38.38 | 38.98 | **0.60** | 30.28 | 31.54 | 0.29 |
| 20% | **38.94** | **37.98** | **0.60** | 27.12 | 32.63 | 0.19 | 36.50 | 37.42 | **0.60** | 27.02 | 31.63 | 0.17 |
| 30% | **37.69** | **36.21** | **0.59** | 25.40 | 32.39 | 0.15 | 34.94 | 36.08 | 0.58 | 30.27 | 31.54 | 0.29 |
| 40% | **36.49** | **34.73** | **0.58** | 24.26 | 32.03 | 0.13 | 32.51 | 24.13 | 0.57 | 24.13 | 18.50 | 0.07 |
| 50% | **35.84** | **33.73** | **0.57** | 23.57 | 31.49 | 0.12 | 29.85 | 18.11 | 0.49 | 23.47 | 14.61 | 0.07 |
| 60% | **34.93** | **32.38** | **0.56** | 22.87 | 31.36 | 0.10 | 27.98 | 14.65 | 0.35 | 22.79 | 14.13 | 0.07 |

Table 4.4: Performance metrics for each method on the Hall sequence as a function of outlier probability.

| SNR | Proposed | | | RPCA | | | TVRPCA | | | DECOLOR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure |
| 5 dB | **31.78** | **26.15** | **0.52** | 20.85 | 18.55 | 0.07 | 25.20 | 11.29 | 0.08 | 27.98 | 14.30 | 0.07 |
| 10 dB | **32.78** | **27.87** | **0.54** | 23.04 | 23.31 | 0.08 | 26.85 | 13.33 | 0.14 | 28.54 | 14.30 | 0.07 |
| 20 dB | **34.73** | **30.73** | **0.56** | 27.42 | 28.73 | 0.14 | 30.20 | 16.89 | 0.34 | 30.13 | 14.30 | 0.07 |
| 30 dB | **36.66** | **32.72** | 0.58 | 31.80 | 30.14 | 0.30 | 34.64 | 21.83 | **0.59** | 31.65 | 25.14 | 0.56 |
| 40 dB | **39.64** | **33.90** | **0.60** | 36.20 | 31.27 | 0.46 | 37.96 | 25.70 | 0.58 | 36.27 | 31.51 | 0.59 |
| 50 dB | **42.89** | **36.14** | 0.60 | 40.59 | 32.00 | 0.54 | 41.47 | 29.77 | 0.59 | 37.87 | 32.73 | **0.61** |

Table 4.5: Performance metrics for each method on the Hall sequence as a function of SNR (Gaussian noise).

| p | Proposed | | | RPCA | | | TVRPCA | | | DECOLOR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure |
| 60% | **39.01** | **37.79** | 0.59 | 28.33 | 31.19 | 0.33 | 35.44 | 36.01 | 0.50 | 30.35 | 32.57 | 0.64 |
| 70% | **37.25** | **36.58** | 0.58 | 27.64 | 30.75 | 0.27 | 31.02 | 32.58 | 0.35 | 29.69 | 33.17 | 0.65 |
| 80% | **35.69** | **35.43** | 0.58 | 27.13 | 30.00 | 0.20 | 30.32 | 33.06 | 0.07 | 28.26 | 31.47 | 0.23 |
| 90% | **33.30** | **33.40** | **0.55** | 27.45 | 23.43 | 0.08 | 30.26 | 14.50 | 0.07 | 27.13 | 31.19 | 0.11 |

Table 4.6: Performance metrics for each method on the Hall sequence as a function of missing data probability.

### 4.5.1 Static Camera Video

We work with the I2R dataset[6] of static camera sequences. The sequences contain between 523 and 3584 frames, each with a subset of 20 frames that have labeled

---

[6] See http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.

Figure 4.3: A representative frame from the decompositions produced by each method applied to the Hall sequence corrupted by 30 dB Gaussian noise. Left column: observations; $L$: reconstructed background; $S$: reconstructed foreground; $L + S$: reconstructed scene; right column: Hall sequence.

| Sequence | Proposed | | | DECOLOR | | | Baseline (Median Filter) | |
|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR |
| Tennis | **39.39** | **30.13** | **0.76** | - | - | - | 37.33 | 25.09 |
| Paragliding | **42.26** | **33.86** | **0.78** | 26.07 | 18.93 | 0.44 | 41.54 | 29.17 |
| Rollerblade | **41.65** | **28.81** | **0.83** | 28.10 | 19.47 | 0.82 | 38.71 | 23.96 |
| Horsejump | **36.16** | **26.54** | 0.76 | 22.86 | 17.64 | **0.80** | 34.51 | 23.19 |
| Average | **39.86** | **29.84** | **0.78** | 25.68* | 18.68* | 0.69* | 38.02 | 25.35 |

Table 4.7: Performance metrics for each method on sequences from the DAVIS dataset corrupted by 30% outliers. *DECOLOR raises an error when run on the Tennis sequence due to the significant camera motion, so it is omitted.

(ground truth) foreground masks. We run each method on a subset of several hundred (contiguous) frames from each sequence containing 10 labeled frames. To evaluate the robustness of each method, we consider three corruption models: Gaussian noise (dense), salt and pepper outliers (sparse), and missing data (inpainting).

In the missing data case, it is trivial to incorporate the missing data locations in our model: we simply encode them as zeros in the mask matrix $M$. The RPCA, TVRPCA, and DECOLOR objectives as written in (4.36)-(4.38) do not directly sup-

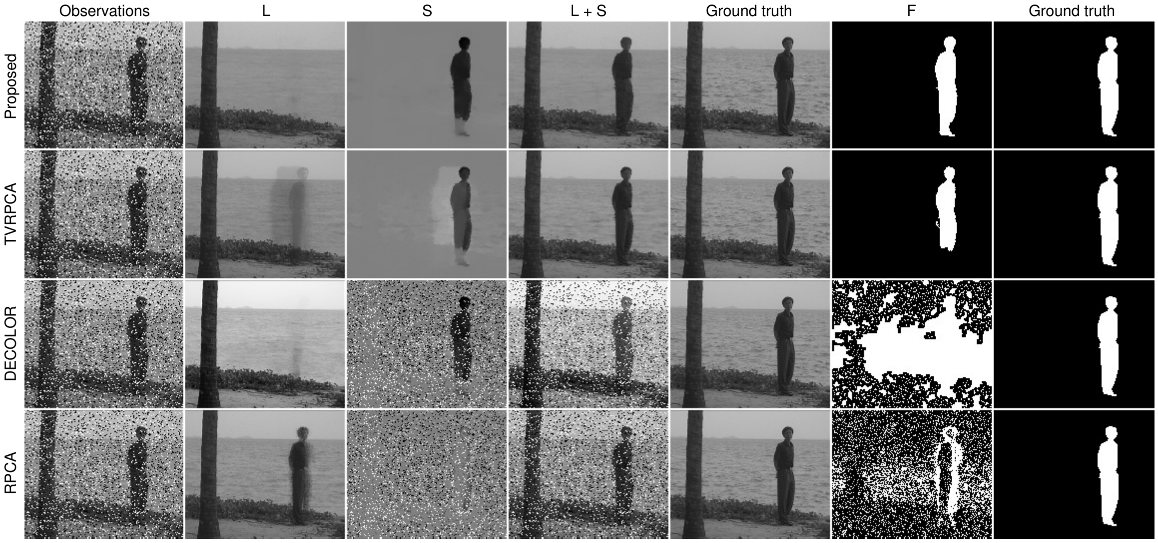


Figure 4.4: A representative frame from the decompositions produced by each method applied to the Water Surface sequence corrupted by 20% outliers. Left column: observations; $L$: reconstructed background; $S$: reconstructed foreground; $L + S$: reconstructed scene; fifth column: Water Surface sequence; $F$: estimated foreground mask; right column: true mask.

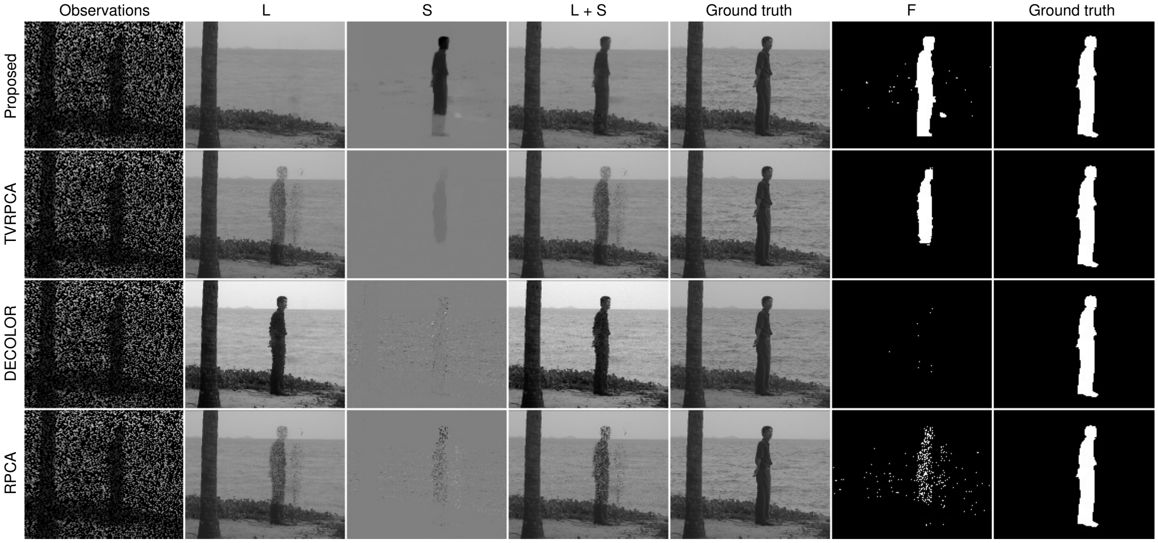


Figure 4.5: A representative frame from the decompositions produced by each method applied to the Water Surface sequence with 70% missing data. Left column: observations; $L$: reconstructed background; $S$: reconstructed foreground; $L + S$: reconstructed scene; Fifth column: original Water Surface sequence; $F$: foreground mask estimated by optimally thresholding $S$; right column: true foreground mask.

port inpainting, but they can be easily modified to do so. See Appendix A for a description of the modified versions of RPCA, TVRPCA, and DECOLOR that we

| Sequence | Proposed | | | DECOLOR | | | Baseline (Wiener Filter) | |
|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR |
| Tennis | **38.61** | **27.86** | **0.75** | 21.94 | 16.87 | 0.36 | 35.40 | 25.56 |
| Paragliding | **40.64** | **32.60** | **0.76** | 28.54 | 19.17 | 0.24 | 37.60 | 27.76 |
| Rollerblade | **38.10** | **28.65** | **0.82** | 28.18 | 17.63 | 0.77 | 37.86 | 23.60 |
| Horsejump | **34.38** | **26.98** | 0.74 | 23.10 | 16.64 | **0.76** | 33.23 | 23.34 |
| Average | **37.93** | **29.02** | **0.77** | 25.44 | 17.58 | 0.53 | 36.02 | 25.07 |

Table 4.8: Performance metrics for each method on sequences from the DAVIS dataset corrupted by 10 dB Poisson noise.

| Sequence | Proposed | | | DECOLOR | | | Baseline (Interpolation) | |
|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR |
| Tennis | **40.50** | **30.86** | **0.77** | 22.38 | 17.76 | 0.37 | 40.02 | 29.86 |
| Paragliding | **43.33** | **34.59** | 0.77 | 27.13 | 18.86 | **0.83** | 42.84 | 32.95 |
| Rollerblade | **42.48** | **29.65** | **0.83** | 24.98 | 19.68 | 0.78 | 41.90 | 27.87 |
| Horsejump | **36.49** | **27.70** | **0.76** | 23.28 | 17.83 | 0.29 | 36.19 | 25.93 |
| Average | **40.70** | **30.70** | **0.78** | 24.44 | 18.53 | 0.57 | 40.24 | 29.15 |

Table 4.9: Performance metrics for each method on sequences from the DAVIS dataset corrupted by 70% missing data.

| p | Proposed | | | DECOLOR | | | Baseline (Median Filter) | |
|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR |
| 10% | **41.88** | **30.98** | **0.76** | - | - | - | 41.44 | 30.54 |
| 20% | **41.04** | **30.46** | **0.76** | - | - | - | 40.55 | 29.61 |
| 30% | **39.39** | **30.13** | **0.76** | - | - | - | 38.58 | 27.36 |
| 40% | **38.21** | **29.82** | **0.75** | - | - | - | 35.76 | 24.21 |
| 50% | **36.86** | **29.14** | **0.73** | - | - | - | 32.72 | 21.04 |

Table 4.10: Performance metrics for each method on the Tennis sequence as a function of outlier probability. DECOLOR raises an error when run on the Tennis sequence due to the significant camera motion, so it produces no decompositions.

| SNR | Proposed | | | DECOLOR | | | Baseline (Wiener Filter) | |
|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR |
| 5 dB | **35.10** | **27.00** | **0.74** | 22.78 | 18.38 | 0.10 | 35.05 | 23.93 |
| 10 dB | **38.61** | **27.86** | **0.75** | 21.94 | 16.87 | 0.36 | 38.32 | 27.11 |
| 15 dB | 40.45 | **30.54** | **0.76** | 21.70 | 17.14 | 0.42 | **41.02** | 29.18 |
| 20 dB | **41.88** | **31.14** | **0.77** | 21.69 | 16.96 | 0.39 | 41.78 | 30.05 |

Table 4.11: Performance metrics for each method on the Tennis sequence as a function of SNR (Poisson noise).

| p | Proposed | | | DECOLOR | | | Baseline (Interpolation) | |
|---|---|---|---|---|---|---|---|---|
| | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR | F-measure | f-PSNR | b-PSNR |
| 60% | **41.61** | **31.33** | **0.77** | 22.09 | 17.35 | 0.36 | 41.23 | 30.60 |
| 70% | **40.50** | **30.86** | **0.77** | 22.38 | 17.76 | 0.37 | 40.02 | 29.86 |
| 80% | **38.74** | **30.28** | **0.75** | 22.48 | 17.47 | 0.43 | 38.35 | 28.84 |
| 90% | **35.78** | **29.22** | **0.74** | 22.95 | 17.54 | 0.38 | 35.67 | 27.18 |

Table 4.12: Performance metrics for each method on the Tennis sequence as a function of missing data probability.

used in our missing data experiments.

To evaluate the denoising capabilities of each method, we measure the peak signal-to-noise ratio of the foreground (f-PSNR) and background (b-PSNR) pixels, respectively, in decibels (dB), using the ground truth foreground masks to distinguish between foreground and background. We also measure the ability of each method to isolate the true foreground by thresholding the foreground component and computing the F-measure of these estimated masks with respect to the labeled masks.[7] Here, F-measure is defined in terms of the precision and recall of the estimated mask as

$$F_{\text{measure}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{4.39}$$

where $F_{\text{measure}} = 1$ corresponds to perfect accuracy.

We run each method for 150 outer iterations, and we optimize the parameters of each algorithm independently for each dataset and performance metric in our tables to show the best possible performance of each method with respect to each metric. For our proposed method, we use 3D TV.

Tables 4.1, 4.2 and 4.3 compare the performance of each method on the I2R sequences corrupted by 20% salt and pepper outliers, Gaussian noise with 30 dB SNR, and 70% missing data, respectively. Tables 4.4, 4.5 and 4.6 show the performance of each method on the Hall sequence as a function of outlier probability, noise SNR, and missing data probability, respectively. Clearly our proposed method performs significantly better than the existing methods in nearly all cases.

Figures 4.3 and 4.4 illustrate the decompositions produced by each method on the Hall sequence corrupted by 30 dB Gaussian noise and the Water Surface sequence corrupted by 20% outliers. The foreground estimates of the RPCA and DECOLOR methods degrade dramatically when outliers are added because they lack the ability to distinguish outliers and other non-idealities from the underlying foreground component. TVPRCA performs better than these methods in the presence of outliers, but its estimated background component contains some residual dense corruptions (cf. Figure 4.3) and foreground artifacts (cf. Figure 4.4) that are not present in the proposed PRPCA method. These results show that our proposed method is better able to uncover the true foreground and background components of corrupted video.

Figure 4.5 illustrates the decompositions produced by each method on the Water Surface sequence corrupted by 70% missing data. The foreground estimates produced by RPCA and DECOLOR are not able to impute the missing foreground pixels

---

[7] For DECOLOR, we use the foreground mask returned by the algorithm.

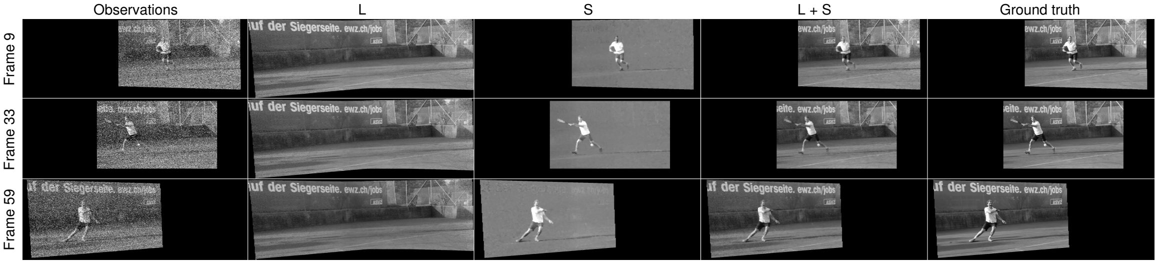


Figure 4.6: Three representative frames from the decomposition produced by the proposed PRPCA method applied to the Tennis sequence corrupted by 30% salt and pepper outliers. Left column: registered observations; $L$: reconstructed registered background; $S$: reconstructed registered foreground; $L + S$: reconstructed registered scene restricted to the current field of view; right column: registered Tennis sequence.

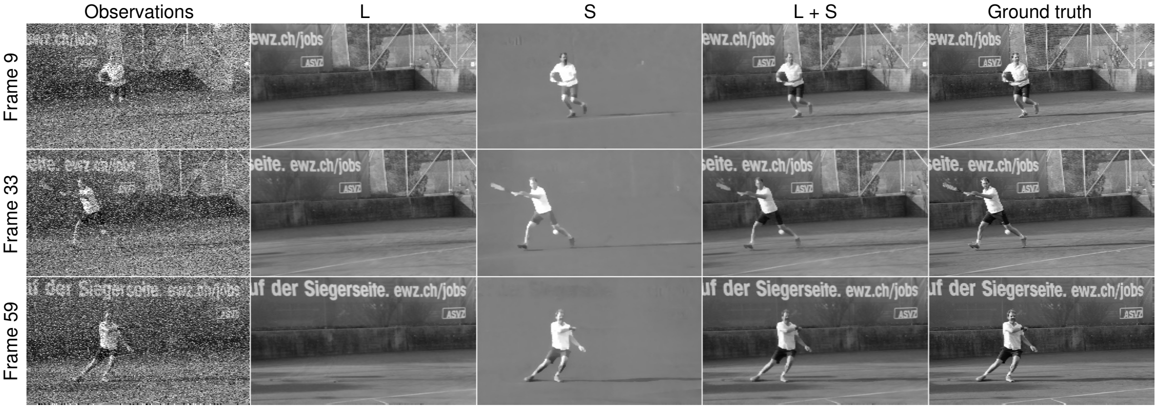


Figure 4.7: The decompositions from Figure 4.6 mapped to the perspective of the original video.

because their models lack a spatial continuity constraint. The TVRPCA method produces a more accurate foreground component, but, as in Figure 4.4, its estimated background component contains some foreground artifacts that are not present in the proposed PRPCA method.

### 4.5.2 Moving Camera Video

We next demonstrate the performance of our proposed PRPCA method on the moving camera sequences from the recent DAVIS benchmark dataset [79].

The RPCA and TVRPCA methods are not suitable for moving camera video, so we only consider the DECOLOR method. As in the static camera case, we consider multiple corruption models: salt and pepper outliers (sparse), Poisson noise (dense), and missing data (inpainting). Although the video registration procedure in Section 4.2 can handle corrupted data, we use the homographies computed from the original videos to isolate the influence of our proposed model (4.7) on reconstruction

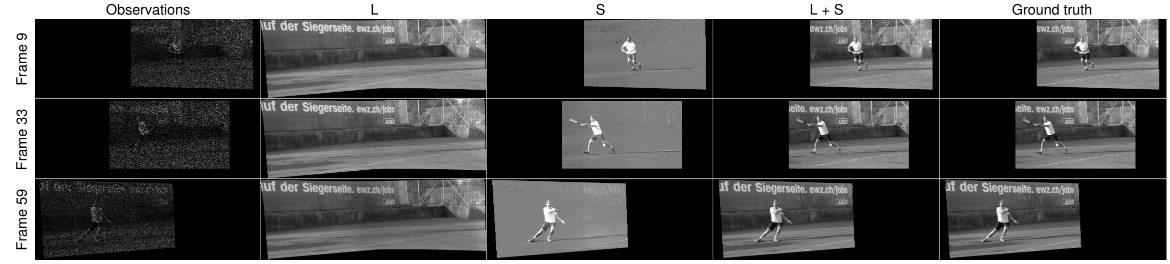


Figure 4.8: Three representative frames from the decomposition produced by the proposed PRPCA method applied to the Tennis sequence corrupted by 70% missing data. Left column: registered observations; $L$: reconstructed registered background; $S$: reconstructed registered foreground; $L + S$: reconstructed registered scene restricted to the current field of view; right column: registered Tennis sequence.

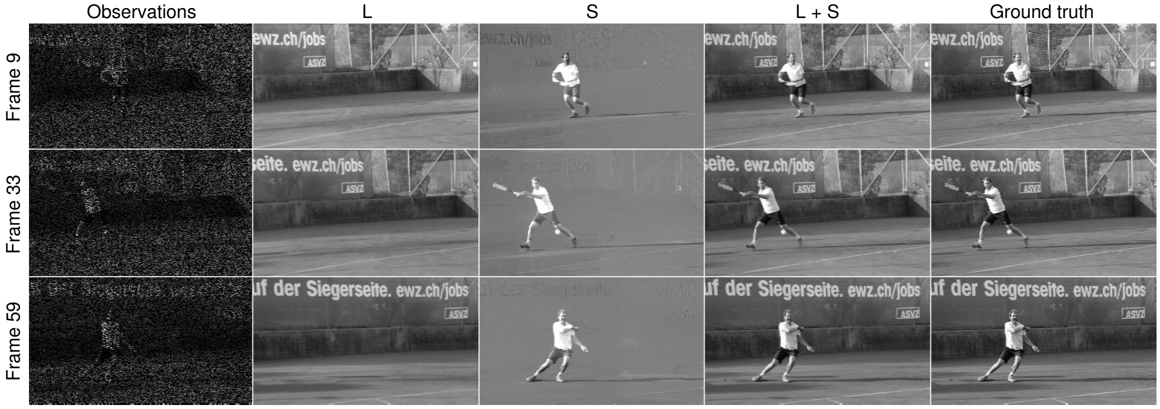
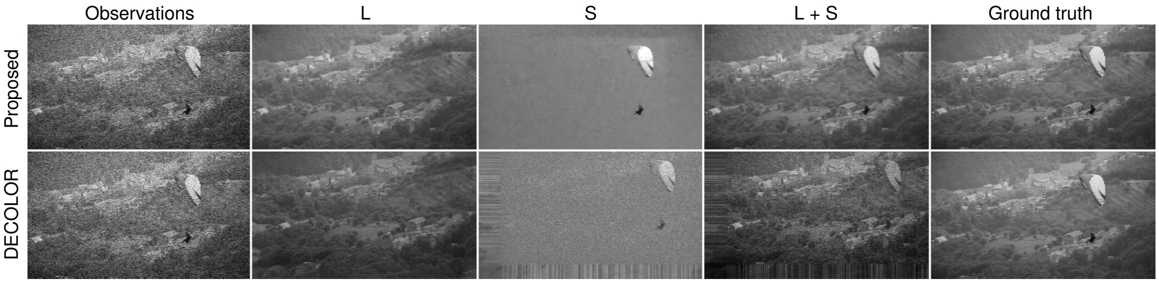


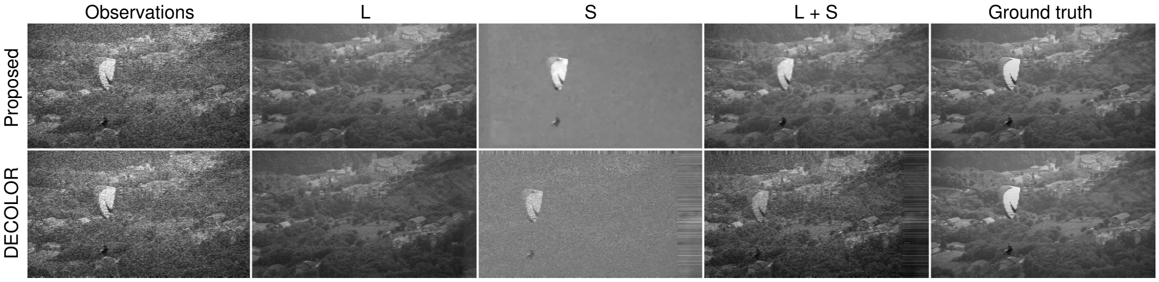
Figure 4.9: The decompositions from Figure 4.8 mapped to the perspective of the original video

quality. We evaluate performance using the same error metrics and parameter tuning strategies from Section 4.5.1. We use 2D TV in our proposed method because the camera motion reduces the temporal continuity of the foreground. To provide an additional benchmark for denoising quality, we also consider the PSNRs produced by the following baseline per-frame denoising methods: median filtering (outlier corruptions), Wiener filtering (Poisson noise corruptions), and cubic interpolation (missing data). Note that these baseline methods are not foreground-background separation strategies, so they have no associated F-measures.

Tables 4.7, 4.8 and 4.9 compare the performance of each method on DAVIS sequences corrupted by 30% salt and pepper outliers, Poisson noise with 10 dB SNR, and 70% missing data, respectively. Tables 4.10, 4.11 and 4.12 show the performance of each method on the Tennis sequence as a function of outlier probability, SNR, and missing data probability, respectively. Our proposed method achieves consistently higher f-PSNR, b-PSNR, and F-measure, which suggests it is well-suited for



(a) Frame 10 of the Paragliding sequence.



(b) Frame 34 of the Paragliding sequence.

Figure 4.10: Two representative frames from decompositions of the Paragliding sequence corrupted by 10 dB Poisson noise. Top row: decomposition produced by the proposed PRPCA method mapped to the perspective of the original video; bottom row: decomposition produced by DECOLOR. Left column: observations; $L$: reconstructed background; $S$: reconstructed foreground; $L+S$: reconstructed scene; right column: Paragliding sequence.

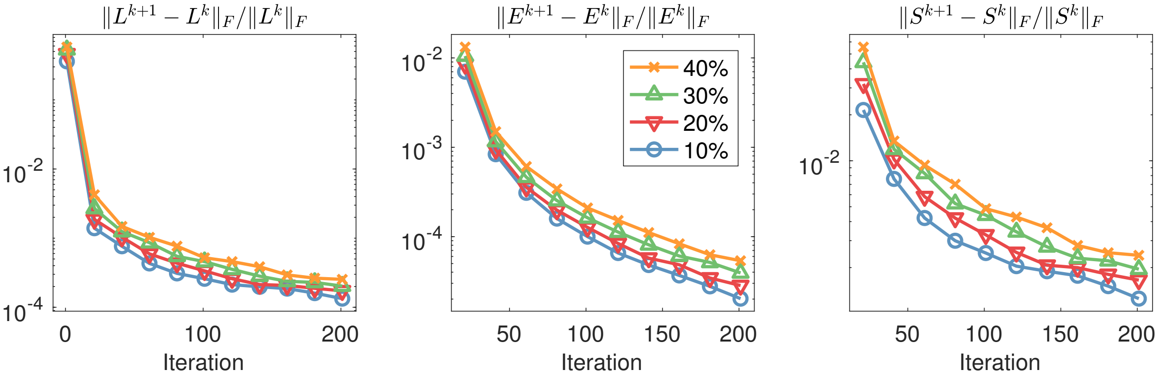


Figure 4.11: Per-iteration convergence of the $L$, $E$, and $S$ components of the proposed PRPCA method on the Fountain sequence corrupted by outliers at various percentages.

processing a variety of corruptions.

Figure 4.6 depicts the decompositions produced by our proposed PRPCA method on the Tennis sequence corrupted by 30% salt and pepper outliers. Note how our proposed method gracefully aggregates the background information from the corrupted

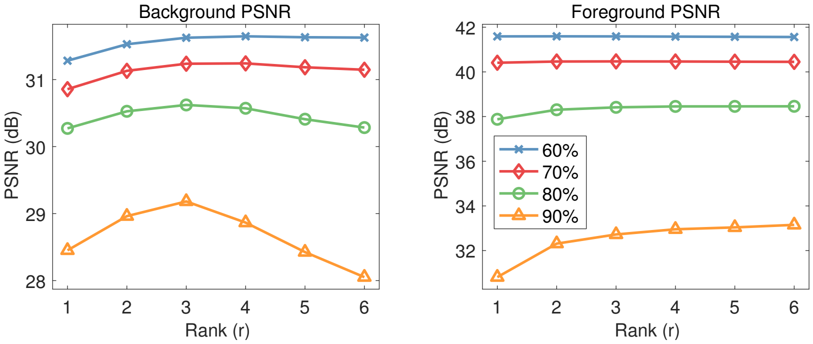


Figure 4.12: Foreground and background PSNRs as a function of OptShrink rank parameter $r$ on the Tennis sequence for various missing data percentages.

frames to produce a clean panoramic estimate $(L)$ of the full field of view. Also, the registered TV-regularized component $(S)$ is able to accurately estimate the dynamic foreground and decouple it from sparse corruptions. None of the methods considered in Section 4.5.1 can produce comparable results. Figure 4.7 shows the decompositions from Figure 4.6 mapped to the perspective of the original video by applying the inverse homographies computed during frame registration. These sequences constitute a direct decomposition of the original moving camera video.

Figures 4.8 and 4.9 show the analogous reconstructions from Figures 4.6 and 4.7, respectively, on the Tennis sequence for the case of 70% missing data. Again, the proposed PRPCA method produces an accurate panoramic decomposition of the scene. In Figure 4.9, the outline of the background text is faintly visible in Frames 9 and 33 of $S$. These artifacts arise from small mismatches in the frame registration process due to violations of the underlying far-field assumption of the frame registration model. This parallax effect captured by $S$ arises so that the reconstructed scene $L+S$ remains faithful to the data $Y$.

Figure 4.10 compares the performance of PRPCA and DECOLOR on the Paragliding sequence corrupted by 10 dB Poisson noise. DECOLOR fails to accurately estimate $L$ and $S$ due to the significant camera motion, while our propsed method consistently produces a high quality decomposition of the dynamic scene from the corrupted video.

### 4.5.3 Algorithm Properties

In this section, we briefly investigate the properties of our PRPCA algorithm as described in Algorithm 4.2. Although the update scheme (4.35) does not correspond to the proximal gradient updates of an explicit cost function that we can track,

Figure 4.11 demonstrates that the $L$, $S$, and $E$ iterates exhibit stable convergence behavior as the iterations progress.

Figure 4.12 plots the f-PSNRs and b-PSNRs produced by our proposed method on the Tennis sequence as a function of rank parameter $r$ at various missing data percentages. This figure shows that our proposed method is quite robust to rank overestimation in the sense that its performance degrades slowly as $r$ increases beyond its optimal value. Intuitively, this behavior is observed because the OptShrink estimator performs a data-driven shrinkage that minimizes the effect of superfluous rank components in $L$.

## 4.6 Conclusions

We proposed a new panoramic robust PCA method for performing robust foreground-background separation on possibly corrupted video with arbitrary camera motion. Our proposed method registers the frames of the raw video, and it utilizes weighted total variation regularization and an improved low-rank matrix estimator (OptShrink) to jointly estimate the foreground and background components of the scene from the registered frames. Our numerical experiments demonstrate that our proposed method is robust to both dense and sparse corruptions of the raw video and produces superior foreground-background separations compared to existing methods. In future work, we plan to investigate the usefulness of the foreground components produced by our method for computer vision tasks like object tracking and activity detection.

## CHAPTER V

# Theoretical Analysis of Low-Rank Matrix Estimation with Thresholding-Based Outlier Rejection

## 5.1 Introduction

Principal component analysis (PCA) [21, 22] is an important tool in statistical signal processing and is often used to tease low-dimensional signals out of high-dimensional data. This dimensionality reduction is an essential first step in many machine learning and inference tasks. Although PCA is inherently robust to noise, it is well-known that outliers can severely corrupt the accuracy of its low-rank subspace estimates, which, in turn, can degrade the performance of subsequent inferential tasks. To remedy the situation, robust PCA algorithms have been recently proposed that aim to jointly estimate the sparse outliers in the data and the underlying low-rank signal.

Several recent results, e.g., [31–33, 38, 80], have shown that convex optimization-based robust PCA algorithms can provably jointly estimate low-rank and sparse components in the noise-free setting. In practice, these algorithms are typically implemented in an alternating fashion where one estimates the sparse component of the data, then estimates the low-rank component of the residual, and then repeats until convergence [81, 82]. Although these optimization-based algorithms can be easily generalized to handle noisy data, there is little theory in place to characterize the low-rank estimation performance in the noisy setting.

Robust PCA algorithms typically use a form of thresholding to estimate outliers in the data. For example, convex optimization-based algorithms typically use soft thresholding

$$\mathbf{soft}_\tau(y) := \mathbf{sign}(y) \max(|y| - \tau, 0), \tag{5.1}$$

which arises as the proximal operator [25] of the $\ell_1$ norm. In turn, the $\ell_1$ norm arises (cf. compressed sensing [70, 83–85]) as the tightest convex relaxation of the $\ell_0$ norm, whose proximal operator is hard thresholding

$$\mathbf{hard}_\tau(y) := y \; \mathbb{1}\{|y| \geq \tau\}. \tag{5.2}$$

The use of hard and soft thresholding for outlier estimation is justified by the seminal work of Donoho and Johnstone [86–90], in which they show that both hard and soft thresholding are within logarithimic factors of minimax optimality for estimating a fixed signal (here, the outliers) corrupted by additive (Gaussian) noise.

### 5.1.1 Contributions

In this chapter, we adopt a first-principles approach to the analysis of robust PCA in the presence of noise. In particular, we address the fundamental issue of the accuracy of the singular vectors of a thresholded low-rank plus noise plus outliers matrix with respect to the underlying low-rank subspace. This analysis is crucial because it sheds light on the extent to which one can use thresholding to make the singular value decomposition (SVD) - the heart of PCA - robust to outliers. Our main contribution is identifying sufficient conditions under which soft or hard thresholding the data and then applying PCA can match the performance of an oracle estimator that replaces the outlier-corrupted data with zeros.

### 5.1.2 Organization

This chapter is organized as follows. In Section 5.2, we introduce our data model, and in Section 5.4 we motivate the study of robust PCA algorithms by analytically characterizing the fundamental limits of PCA in the presence of outliers. We present our oracle and thresholding-based robust PCA estimators in Sections 5.5 and 5.6, and in Section 5.7 we present our main result on sufficient conditions for equivalence of oracle and thresholding-based robust PCA. Finally, we provide empirical validation of our results in Section 5.8, and we connect our results to alternating minimization-based robust PCA algorithms in Section 5.9. Appendices B - E contain the proofs of our results.

## 5.2 Data Model

Consider the setting where an observed low-rank plus noise plus outliers matrix $\widetilde{X} \in \mathbb{R}^{m\times n}$ is modeled as

$$\widetilde{X} = \underbrace{\sum_{i=1}^{r} \theta_i u_i v_i^T}_{=:L} + G + S. \tag{5.3}$$

Here, $L$ is a rank-$r$ matrix that we are interested in reliably recovering with left and right singular vectors $u_i$ and $v_i$, respectively, and singular values $\theta_1 > \theta_2 > \ldots > \theta_r > 0$. The matrix $G$ is an additive noise matrix, where $G_{ij}$ are drawn i.i.d. from a symmetric, zero-mean noise distribution $g$ with $\mathbb{E}g^2 = \sigma^2/n$. The matrix $S$ represents a sparse matrix of outliers, modeled as

$$S_{ij} = \begin{cases} Q_{ij} & \text{with probability } p_s \\ 0 & \text{with probability } 1 - p_s, \end{cases}$$

where $p_s$ denotes the outlier probability, and $Q_{ij}$ are drawn i.i.d. and independent of $G$ from a symmetric, zero-mean distribution $q$ with $\mathbb{E}q^2 = \sigma_q^2$. We assume that the parameters $\{\theta_i\}$, $\sigma^2$, and $\sigma_q^2$ are fixed, and we allow $p_s = p_{s,n}$ to depend on $n$, although we suppress this dependence in our notation for convenience. The following section introduces some technical assumptions that we impose on model (5.3) so that the model is amenable to fruitful analysis.

## 5.3 Assumptions

There is an inherent ambiguity between low-rank and sparse matrices. For example, the matrix $L = e_1 e_1^T$, where $e_1$ is the first canonical basis vector, is perfectly low-rank and sparse: it has rank 1 and only 1 non-zero element. To resolve this ambiguity, one needs to impose additional structure on the low-rank matrix $L$. In this chapter, we make the following incoherence assumption.

**Assumption V.1** (Incoherence). *The singular vectors of $L$ satisfy*

$$\max_{i=1,\ldots,r} ||u_i||_\infty \le C_u \frac{\log^{\eta_u} m}{\sqrt{m}}, \quad \max_{i=1,\ldots,r} ||v_i||_\infty \le C_v \frac{\log^{\eta_v} n}{\sqrt{n}} \tag{5.4}$$

*for some universal constants $C_u, C_v, \eta_u, \eta_v > 0$.*

Assumption V.1 guarantees that $L$ is a dense matrix with $o(1)$ entries.

We also impose some mild technical conditions on the noise and outlier distributions, which will be required in our analysis. Specifically, we assume that the noise distribution $g$ is subgaussian (Assumption V.2), and we assume that the outlier distribution $q$ is locally Lipschitz in the neighborhood of zero (Assumption V.3).

**Assumption V.2** (Subgaussian noise)**.** *The noise distribution $g$ is subgaussian. That is, $\exists \gamma > 0$ such that*

$$\mathbb{P}(|g| \geq \tau) \leq 2\exp(-\gamma n \tau^2) \tag{5.5}$$

*for all $\tau \geq 0$.*

**Assumption V.3** (Lipschitz outliers)**.** *The outlier distribution $q$ is locally Lipschitz in the neighborhood of zero. That is, $\exists C, \delta > 0$ such that*

$$\mathbb{P}(q \in (x, y)) \leq C(y - x) \tag{5.6}$$

*for all subintervals $(x, y) \subseteq (-\delta, \delta)$.*

Assumption V.2 rules out the possibility of heavy-tailed noise distributions, which take large values with super-exponential probability. Intuitively, a heavy-tailed noise distribution would produce corruptions that are indistinguishable from outliers; it is therefore reasonable to limit our study to subgaussian noise distributions, so that our model is identifiable. The class of subgaussian random variables is quite general, including, for example, the Gaussian distribution and all bounded random variables, such as the Rademacher distribution $G_{ij} = \pm\sigma/\sqrt{n}$. Analogously, Assumption V.3 precludes the possibility that the outlier distribution $q$ can have pathological probability masses in the neighborhood of zero, which would correspond to an outlier distribution with "noise like" properties. Our outlier assumption ensures that the model (5.3) is identifiable.

We will also require a lower bound on the outlier probability $p_s$ to avoid any pathological interactions between the sparsity pattern of $S$ and the leading singular vectors of $\widetilde{X}$. Specifically, we assume that the following condition holds.

**Assumption V.4** (The "not-too-sparse" conditon)**.** *Let $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$ be the matrices whose columns are the left and right singular vectors of $L$, respectively, and define*

$$\overline{X} = \frac{G + S}{\sqrt{\sigma^2 + np_s\sigma_q^2}}, \tag{5.7}$$

*an i.i.d. random matrix with variance $1/n$ entries. We assume that $p_s$ is large enough* *such that*

$$\begin{aligned} U^T\left(z^2 I_m - \overline{X}\overline{X}^T\right)^{-1} U &\xrightarrow{a.s.} \left(\int \frac{1}{z^2-t^2}\,d\mu(t)\right) I_r \\ V^T\left(z^2 I_n - \overline{X}^T\overline{X}\right)^{-1} V &\xrightarrow{a.s.} \left(\int \frac{1}{z^2-t^2}\,d\widetilde{\mu}(t)\right) I_r \end{aligned} \tag{5.8}$$

*and*

$$\begin{aligned} U^T\left(z^2 I_m - \overline{X}\overline{X}^T\right)^{-1} \overline{X} V &\xrightarrow{a.s.} 0 \\ V^T\overline{X}^T\left(z^2 I_m - \overline{X}\overline{X}^T\right)^{-1} U &\xrightarrow{a.s.} 0 \end{aligned} \tag{5.9}$$

*uniform in $z$, where $\mu$ is the Marchenko-Pastur distribution [30], and $\widetilde{\mu} := c\mu + (1-c)\delta_0$.*

**Remark V.5.** A constant $p_s = p$ satisfies Assumption V.4, as does $p_s \gg \log^C n/n$ for large enough $C$ [34].

Under Assumptions V.1-V.4, the $L$ component of model (5.3) is a low-rank matrix with elements (roughly) of size $|L_{ij}| = O(\log n \text{ factors}/n)$, $G$ is a dense noise matrix with elements (roughly) of size $|G_{ij}| = O(1/\sqrt{n})$, and $S$ is a relatively sparse matrix whose nonzero elements are (roughly) of size $|S_{ij}| = O(1)$. Thus the outliers are large but infrequent, the noise is dense and moderately sized, and the low-rank matrix is dwarfed by the other components. At the element level, the prospect of estimating the low-rank matrix from such a model seems hopeless; however, the low-rank structure of $L$ implies that there is redundant information spread across the entire observation matrix, and our results in this chapter show that one can indeed reliably estimate the latent low-rank structure by computing the SVD of a carefully constructed outlier-rejected matrix.

Our analysis in Section 5.7 shows that these relative component magnitudes are in fact the most interesting regime under model (5.3). For example, if the low-rank matrix has elements $|L_{ij}| \gg O(\log n \text{ factors}/n)$ or the noise has strength $|G_{ij}| \ll O(1/\sqrt{n})$, then it is trivial to reliably estimate $L$. Conversely, if the noise has strength $|G_{ij}| \gg O(1/\sqrt{n})$, it is impossible to reliably estimate $L$. Interestingly, if $|S_{ij}| \ll O(1)$, the problem becomes *harder*, because it is more difficult to accurately reject the outliers.

## 5.4 Fundamental Limits of PCA

In this section, we motivate the need for an outlier rejction scheme for model (5.3) by theoretically analyzing the fundamental limits of PCA in the presence of outliers.

The following theorem, based on the work of [27], characterizes the asymptotic singular vector accuracy of the raw data $\widetilde{X}$.

**Theorem V.6** (Raw accuracy). *Let*

$$\widetilde{X} = \sum_{i=1}^{m} \widetilde{\theta}_i \widetilde{u}_i \widetilde{v}_i^T \tag{5.10}$$

*be the SVD of $\widetilde{X}$ drawn from model* (5.3)*, and assume that Assumptions V.1-V.4 hold. Then, asymptotically as $m, n \to \infty$ such that $m/n \to c \in (0, 1]$, we have*

$$|\langle u_i, \widetilde{u}_i \rangle|^2 \xrightarrow{a.s.} \begin{cases} 1 - \dfrac{c(1 + \overline{\theta}_i^2)}{\overline{\theta}_i^2(\overline{\theta}_i^2 + c)} & \text{if } \overline{\theta}_i > c^{1/4} \\ 0 & \text{otherwise,} \end{cases} \tag{5.11}$$

*and*

$$|\langle v_i, \widetilde{v}_i \rangle|^2 \xrightarrow{a.s.} \begin{cases} 1 - \dfrac{(c + \overline{\theta}_i^2)}{\overline{\theta}_i^2(\overline{\theta}_i^2 + 1)} & \text{if } \overline{\theta}_i > c^{1/4} \\ 0 & \text{otherwise,} \end{cases} \tag{5.12}$$

*for $i = 1, \ldots, r$, where*

$$\overline{\theta}_i = \lim_{n \to \infty} \frac{\theta_i}{\sqrt{\sigma^2 + n p_s \sigma_q^2}} \tag{5.13}$$

*is the effective signal-to-noise ratio (SNR) of the ith subspace component.*

*Proof.* One can view model (5.3) as $\widetilde{X} = L + \bar{X}$, where $\bar{X}_{ij} = G_{ij} + S_{ij}$ are independent zero-mean random variables with variance $\bar{\sigma}^2/n$, where $\bar{\sigma}^2 := \sigma^2 + n p_s \sigma_q^2$. The results follows from Theorem 2.9 and Section 3.1 of [27]. □

Theorem V.6 brings into sharp focus the detrimental effect of outliers on the singular values of $\widetilde{X}$. Indeed, consider the dense outlier setting where $p_s = O(1)$ so that a constant fraction of the entries of $\widetilde{X}$ are corrupted by outliers. By Theorem V.6, we have $\overline{\theta}_i = 0$ and thus the singular vectors of $\widetilde{X}$ contain asymptotically zero information about the singular vectors of $L$, as their corresponding principal components are asymptotically orthogonal. Moreover, consider the relatively sparse outlier setting where $p_s = O(\log n/n)$ and so that a vanishing fraction of the entries in each row of $\widetilde{X}$ are corrupted by outliers. By Theorem V.6, we again have $\overline{\theta}_i = 0$, and so PCA again breaks down completely. This result definitively motivates the need for robust PCA algorithms to handle outliers.

## 5.5 Oracle Robust PCA

Before considering data-driven methods to remove the outliers from model (5.3), it is instructive to analyze the performance of an oracle outlier rejection scheme. Towards this end, suppose an oracle supplies the outlier support set

$$\mathcal{S}^\star = \{(i,j) : S_{ij} \neq 0\}. \tag{5.14}$$

In this setting, a natural approach is to replace the corrupted entries of $\widetilde{X}$ with zeros, yielding the estimator

$$\widetilde{X}^\star_{ij} = \begin{cases} \widetilde{X}_{ij} & \text{if } (i,j) \notin \mathcal{S}^\star \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

The oracle estimator $\widetilde{X}^\star$ deterministically removes every outlier from the data, so the singular vector accuracy of $\widetilde{X}^\star$ depends only on the extent to which zeroing out elements of $\widetilde{X}$ with probability $p_s$ affects its SVD. The following theorem precisely characterizes the singular vectors of $\widetilde{X}^\star$. The proof is presented in Appendix C.

**Theorem V.7** (Oracle accuarcy)**.** *Assume that Assumptions V.1-V.4 hold. Then, as $m, n \to \infty$ such that $m/n \to c \in (0,1]$, the asymptotic accuracies of the left and right singular vectors of $\widetilde{X}^\star$ are given by* (5.11) *and* (5.12)*, respectively, with effective SNR*

$$\overline{\theta}^\star_i = \lim_{n\to\infty} \frac{\sqrt{1-p_s}\ \theta_i}{\sigma}. \tag{5.16}$$

*Here, it is understood that we reuse $\widetilde{u}_i$ and $\widetilde{v}_i$ in* (5.11) *and* (5.12) *to refer to the singular vectors of the matrix $\widetilde{X}^\star$ in question, not $\widetilde{X}$.*

Theorem V.7 shows that $\widetilde{X}^\star$ is robust to outliers in the sense that its singular vector accuracy degrades gracefully as the outlier probability $p_s$ increases. In fact, when $p_s = o(1)$, the effective SNR of $\widetilde{X}^\star$ is $\overline{\theta}^\star_i = \theta_i/\sigma$, which is precisely the SNR of the raw data $\widetilde{X}$ predicted by Theorem V.6 with $S = 0$, i.e., no outliers.

The key insight of Theorems V.6 and V.7 is that the SVD is inherently robust to missing data but breaks down completely in the presence of large outliers. Intuitively, Assumption V.1 guarantees that $|L_{ij}| = o(1)$, so zeroing an element of $\widetilde{X}$ causes an $o(1)$ corruption of $L$, while an outlier causes an $O(1)$ corruption of $L$.

## 5.6 Thresholding-Based Robust PCA

Now that we understand the performance of an oracle outlier rejection scheme, we are ready to develop a data-driven algorithm. Motivated by the work of Donoho, Johnstone, and others [86–98], we introduce two natural outlier rejection procedures based on element-wise thresholding the elements of $\widetilde{X}$. Specifically, for a given threshold $\tau = \tau_n > 0$, we consider the hard and soft thresholding estimators, respectively, of $S$, defined by

$$\begin{aligned} \widehat{S}_\tau^{\mathrm{HT}} &= \mathbf{hard}_\tau(\widetilde{X}) \\ \widehat{S}_\tau^{\mathrm{ST}} &= \mathbf{soft}_\tau(\widetilde{X}), \end{aligned} \tag{5.17}$$

and the associated residual signal-plus-noise matrices

$$\begin{aligned} \widetilde{X}_\tau^{\mathrm{HT}} &= \widetilde{X} - \widehat{S}_\tau^{\mathrm{HT}} \\ \widetilde{X}_\tau^{\mathrm{ST}} &= \widetilde{X} - \widehat{S}_\tau^{\mathrm{ST}}. \end{aligned} \tag{5.18}$$

One can view $\widetilde{X}_\tau^{\mathrm{HT}}$ and $\widetilde{X}_\tau^{\mathrm{ST}}$ as data-driven approximations of the oracle estimator $\widetilde{X}^\star$. Indeed, if we define

$$\mathcal{S}_\tau = \{(i,j) : |\widetilde{X}_{ij}| \geq \tau\}, \tag{5.19}$$

then we can express $\widetilde{X}_\tau^{\mathrm{HT}}$ as

$$(\widetilde{X}_\tau^{\mathrm{HT}})_{ij} = \begin{cases} \widetilde{X}_{ij} & \text{if } (i,j) \notin \mathcal{S}_\tau \\ 0 & \text{otherwise,} \end{cases} \tag{5.20}$$

and $\widetilde{X}_\tau^{\mathrm{ST}}$ as

$$(\widetilde{X}_\tau^{\mathrm{ST}})_{ij} = \begin{cases} \widetilde{X}_{ij} & \text{if } (i,j) \notin \mathcal{S}_\tau \\ \tau\, \mathbf{sign}(\widetilde{X}_{ij}) & \text{otherwise.} \end{cases} \tag{5.21}$$

From (5.20) we see that $\widetilde{X}_\tau^{\mathrm{HT}}$ is a copy of $\widetilde{X}$ with zeros inserted at indices in $\mathcal{S}_\tau$, which can be viewed as an approximation to the oracle outlier support set $\mathcal{S}^\star$ from (5.14). The same holds for $\widetilde{X}_\tau^{\mathrm{ST}}$, except that $\pm\tau$ values are inserted rather than zeros. Both thresholding-based estimators will remove large outliers from $\widetilde{X}$, so it is plausible to expect that, for an appropriately chosen threshold $\tau$, both $\widetilde{X}_\tau^{\mathrm{HT}}$ and $\widetilde{X}_\tau^{\mathrm{ST}}$ may achieve near-oracle singular vector accuracy. We formalize this observation in Section 5.7.

|  | Sparse outliers<br>$p_s \log^\eta n \to 0$ | Dense outliers<br>$p_s \log^\eta n \to \infty$ |
|---|---|---|
| Hard thresholding | $\overline{\theta}_i^{\text{HT}} = \overline{\theta}_i^\star$ | $\overline{\theta}_i^{\text{HT}} = \overline{\theta}_i^\star$ |
| Soft thresholding | $\overline{\theta}_i^{\text{ST}} = \overline{\theta}_i^\star$ | $\overline{\theta}_i^{\text{ST}} = 0$ |

Table 5.1: Effective SNRs for Theorem V.8

## 5.7 Main Result

We now precisely characterize the leading singular vectors of $\widetilde{X}_\tau^{\text{HT}}$ and $\widetilde{X}_\tau^{\text{ST}}$ for a particular choice of $\tau$ motivated by the thresholding literature. Towards this end, consider a special case of (5.3) where $L = 0$ and our objective is to estimate the sparse matrix $S$ as accurately as possible from an instance $\widetilde{X}$ of (5.3). In this setting, Donoho and Johnstone [86–90] showed that the estimators $\widehat{S}_\tau^{\text{HT}}$ and $\widehat{S}_\tau^{\text{ST}}$ are within logarithmic factors of minimax (w.r.t. the distribution of $S$) mean-squared error optimality for estimating $S$ among all shrinkage estimators, i.e., those estimators $\hat{S}$ for which $|\hat{S}_{ij}| \leq |\widetilde{X}_{ij}|$, when the following threshold is used

$$\tau^\star = \sigma\sqrt{\frac{2\log cn^2}{n}} = O\left(\sqrt{\frac{\log n}{n}}\right). \tag{5.22}$$

In the general setting, $L \neq 0$, but Assumption V.1 guarantees that $|L_{ij}| = o(1)$, so one expects the estimators $\widehat{S}_\tau^{\text{HT}}$ and $\widehat{S}_\tau^{\text{ST}}$ with thresholds $\tau \approx \tau^\star$ to still produce good estimates of $S$, which has $O(1)$ entries. In turn, the residuals $\widetilde{X} - \widehat{S}_\tau^{\text{HT}}$ and $\widetilde{X} - \widehat{S}_\tau^{\text{ST}}$ should be approximately low-rank plus noise matrices whose singular vectors are good estimates of the singular vectors of $L$.

The following theorem formalizes the above argument. The proof is presented in Appendix D for the case of hard thresholding and Appendix E for the case of soft thresholding.

**Theorem V.8** (Accuracy after outlier rejection)**.** *Assume that Assumptions V.1-V.4 hold, and fix a threshold sequence $\tau = \tau_n$ of the form*

$$\tau = C\sqrt{\frac{\log^\eta n}{n}} \tag{5.23}$$

*for some constants $C > 0$ and $\eta > 1$. Then, as $m, n \to \infty$ such that $m/n \to c \in (0, 1]$, the asymptotic accuracies of the left and right singular vectors of $\widetilde{X}_\tau^{HT}$ and*

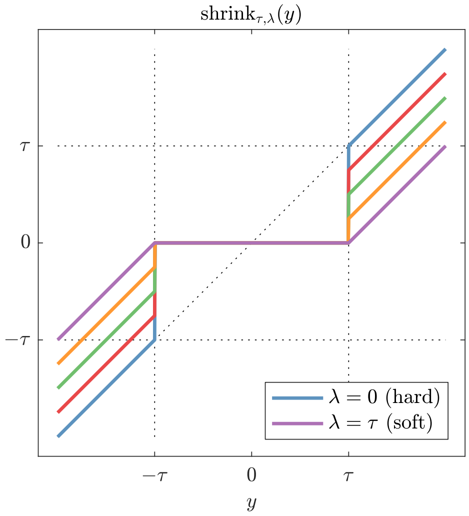


Figure 5.1: Graphical depiction of $\mathbf{shrink}_{\tau,\lambda}$ for various values of $\lambda$. Note that it reduces to hard tresholding when $\lambda = 0$ and reduces to soft thresholding when $\lambda = \tau$.

$\widetilde{X}_\tau^{ST}$ *are given by* (5.11) *and* (5.12) *with effective SNRs given by Table 5.1. Here, it is understood that we reuse* $\widetilde{u}_i$ *and* $\widetilde{v}_i$ *in* (5.11) *and* (5.12) *to refer to the singular vectors of the matrices* $\widetilde{X}_\tau^{HT}$ *and* $\widetilde{X}_\tau^{ST}$ *in question, not* $\widetilde{X}$*.*

**Remark V.9.** More specifically, Appendix D establishes that

$$\overline{\theta}_i^{\mathrm{HT}} = \lim_{n\to\infty} \frac{\sqrt{1-p_s}\,\theta_i}{\sigma} = \overline{\theta}_i^\star \tag{5.24}$$

for any $p_s$, and Appendix E establishes that

$$\overline{\theta}_i^{\mathrm{ST}} = \lim_{n\to\infty} \frac{(1-p_s)\theta_i}{\sqrt{(1-p_s)\sigma^2 + p_s n\tau^2}}. \tag{5.25}$$

**Remark V.10.** Our proof technique can allow $\eta = 1$ in Theorem V.8, but in this case we require a lower bound on $C$ of the form $C > 1/2\gamma$, where $\gamma > 0$ is a constant arising from the analysis. Empirically, we observe that $\eta > 0$ is sufficient, but our current bounding techniques are not sharp enough to establish this result.

**Remark V.11.** Although Theorem V.8 was formulated explicitly in terms of the hard thresholding and soft thresholding-based estimators $\widetilde{X}_\tau^{\mathrm{HT}}$ and $\widetilde{X}_\tau^{\mathrm{ST}}$, our proof technique can be extended to the two-parameter shrinkage function

$$\mathbf{shrink}_{\tau,\lambda}(y) := \mathbf{sign}(y)(|y| - \lambda)\mathbb{1}\{|y| \geq \tau\}, \tag{5.26}$$

with additional parameter $\lambda \in [0,\ \tau]$. Note that $\mathbf{hard}_\tau = \mathbf{shrink}_{\tau,0}$ and $\mathbf{soft}_\tau =$

$\mathbf{shrink}_{\tau,\tau}$, so (5.26) is a generalization of hard and soft thresholding.[1] Figure 5.1 depicts the **shrink** function for various values of $\lambda \in [0,\ \tau]$. Specifically, Theorem V.8 can be extended to show that the asymptotic accuracies of the left and right singular vectors of the estimator

$$\widetilde{X}_{\tau}^{\lambda} := \widetilde{X} - \mathbf{shrink}_{\tau,\lambda}(\widetilde{X}) \tag{5.27}$$

are given by (5.11) and (5.12) with effective SNR

$$\overline{\theta}_i^{\lambda} = \lim_{n\to\infty} \frac{(1-p_s)\theta_i}{\sqrt{(1-p_s)\sigma^2 + p_s n\lambda^2}}. \tag{5.28}$$

Importantly, note from (5.28) that $\overline{\theta}_i^{\lambda}$ is a decreasing function of $\lambda$, so hard thresholding ($\lambda = 0$) is the optimal shrinkage function of the form (5.26) in the sense of maximizing effective SNR. Intuitively, any $\lambda > 0$ makes a residual contribution of $p_s n\lambda^2$ in the denominator of (5.28), which decreases the effective SNR of the outlier-rejected matrix.

Theorem V.8 reveals two interesting phenomena. First, the hard thresholding estimator $\widetilde{X}_{\tau}^{\mathrm{HT}}$ asymptotically attains the performance of the oracle estimator $\widetilde{X}^{\star}$, regardless of outlier probability $p_s$. Intuitively, this result says that hard thresholding is able to reliably remove outliers from $\widetilde{X}$ without degrading the spectrum of the residual matrix.

On the other hand, Theorem V.8 identifies a phase transition in the behavior of soft thresholding. In the sparse outlier regime where $p_s \log^{\eta} n \to 0$, the soft thresholding estimator $\widetilde{X}_{\tau}^{\mathrm{ST}}$ also asymptotically attains oracle performance. However, in the dense outlier regime where the outlier probability $p_s$ is large enough that $p_s \log^{\eta} n \to \infty$, soft thresholding breaks down completely. The dichotomy between the performance of hard and soft thresholding for robust PCA in the dense outlier regime is interesting because no such gap exists for estimating $S$: hard and soft thresholding are both nearly optimal [86–90].

## 5.8 Numerical Validation

In this section, we empirically validate the conclusions of Theorem V.8 by generating synthetic instances of $\widetilde{X}$ from model (5.3) with $r = 1$. We generate $L = \theta uv^T \in \mathbb{R}^{m\times n}$ by taking $v$ to be a vectorized, unit-norm version of the $128 \times 128$ Peppers image, setting $\theta = 2$, and generating $u$ uniformly at random from the unit sphere

---

[1] Perhaps $0 < \lambda < \tau$ should be termed "medium" thresholding.

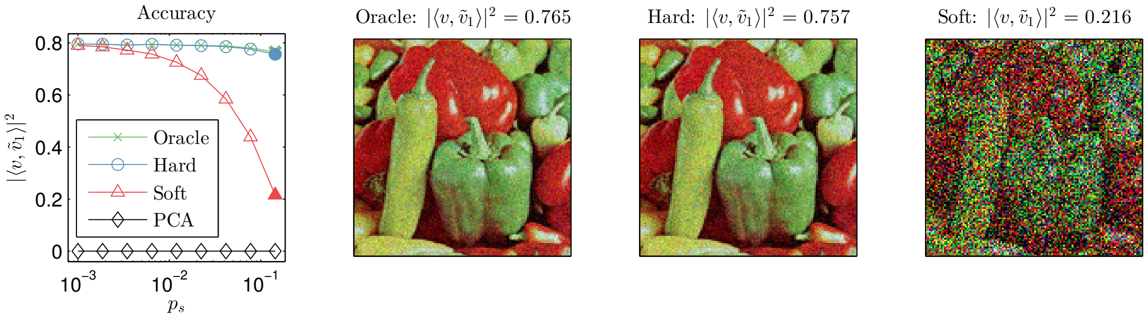


Figure 5.2: Empirical validation of Theorem V.8. The left figure plots the first right singular vector accuracy $|\langle v_1, \widetilde{v}_1\rangle|^2$ of the estimators $\widetilde{X}^\star$ (oracle), $\widetilde{X}_\tau^{\mathrm{HT}}$ (hard), $\widetilde{X}_\tau^{\mathrm{ST}}$ (soft), and $\widetilde{X}$ (PCA) as a function of outlier probability $p_s$. The remaining figures plot (from left to right) the first right singular vectors of $\widetilde{X}^\star$, $\widetilde{X}_\tau^{\mathrm{HT}}$, and $\widetilde{X}_\tau^{\mathrm{ST}}$ reshaped into images for the particular choice of $p_s = 15\%$ denoted by the solid markers in the left figure.

with $m = cn$ and $c = 0.1$. We then populate the noise matrix $G$ with i.i.d. Gaussian random variables with $\sigma = 2$, and we populate the sparse matrix $S$ with i.i.d. Laplace random variables with $\sigma_q = 5$ for a given outlier probability $p_s$. In this setup, the rows of $\widetilde{X}$ contain noisy and outlier corrupted copies of the Peppers image, and we estimate the underlying image by computing the first right singular vectors of the robust PCA estimators from Sections 5.5 and 5.6.

Figure 5.2 compares the accuracy of the first right singular vectors of the three robust PCA estimators $\widetilde{X}^\star$, $\widetilde{X}_\tau^{\mathrm{HT}}$, and $\widetilde{X}_\tau^{\mathrm{ST}}$ as a function of outlier probability $p_s$ with threshold $\tau = \tau^\star$ suggested by (5.22). For reference, we also include the standard PCA estimator that computes the principal components of the raw data $\widetilde{X}$. Consistent with Theorem V.6, standard PCA is highly sensitive to outliers and thus produces nearly orthogonal estimates $\tilde{v}_1$ of $v$ over the range of outlier probabilities $p_s$ tested.

As predicted by Theorem V.8, hard thresholding achieves oracle accuracy over the range of $p_s$ tested while soft thresholding achieves oracle accuracy in the sparse outlier regime and breaks down in the dense outlier regime. Indeed, the reconstructed Peppers images in Figure 5.2 show that soft thresholding peforms significantly worse in the presence of $p_s = 15\%$ outliers.

Intuitively, soft thresholding breaks down in the dense outlier setting because, rather than inserting zeros in the estimated outlier locations as in hard thresholding, the soft thresholding function deposits $\pm\tau$ entries. When there are relatively few outliers, these residuals are not prolific enough to degrade the singular vectors of the data. However, as the outlier density increases, these residual values conspire and eventually dominate the signature of the underlying low-rank matrix.

One might hope that the performance of soft thresholding can be salvaged by choosing a smaller threshold $\tau$. However, while this would decrease the residual magnitudes in $\widetilde{X}_\tau^{\mathrm{ST}}$, the quality of the sparse estimate $\widehat{S}_\tau^{\mathrm{ST}}$ would degrade as the threshold $\tau$ deviates from its (approximately) optimal value $\tau^\star$ from (5.22), and it seems unlikely that intentionally producing a suboptimal sparse estimate $\widehat{S}_\tau^{\mathrm{ST}}$ would improve the singular vector accuracy of the residual matrix $\widetilde{X}_\tau^{\mathrm{ST}}$. Formalizing this conjecture is an interesting future research direction.

## 5.9 Connection to Alternating Minimization

One can interpret the thresholding-based estimators from Section 5.6 as the first iterations of alternating minimization algorithms for certain nonconvex optimization problems. Indeed, consider the optimization problem

$$\begin{aligned} \text{(P0)} \quad \min_{L,S} \quad & \|\widetilde{X} - L - S\|_F^2 + \tau^2 \|S\|_0 \\ \text{s.t.} \quad & \mathbf{rank}(L) \leq r. \end{aligned}$$

One approach to solving (P0) is to adopt a block coordinate descent strategy where one alternatively minimizes with respect to $S$ and $L$ with the other variable held fixed. It is straightforward to show [23, 99] that this strategy yields updates of the form:

$$\begin{aligned} S^{k+1} &= \mathbf{hard}_\tau(\widetilde{X} - L^k) \\ L^{k+1} &= \mathbf{TSVD}_r(\widetilde{X} - S^{k+1}), \end{aligned} \tag{5.29}$$

where $\mathbf{TSVD}_r(Z) = \sum_{i=1}^r \sigma_i u_i v_i^T$ is the rank-$r$ truncated singular value decomposition. There is a close connection between the updates (5.29) and the thresholding-based estimators from Section 5.6. Indeed, suppose one initializes $L^0 = 0$. Then the first iteration of (5.29) yields $S^1 = \mathbf{hard}_\tau(\widetilde{X} - 0) = \widehat{S}_\tau^{\mathrm{HT}}$ and $L^1 = \mathbf{TSVD}_r(\widetilde{X} - S^1) = \mathbf{TSVD}_r(\widetilde{X}_\tau^{\mathrm{HT}})$, which is precisely the hard thresholding-based estimation scheme that we analyzed in Section 5.7. Although (P0) is nonconvex, the block coordinate descent updates (5.29) are guaranteed to monotonically decrease the objective, so the alternating scheme must converge to a critical point of the cost function.

Similarly, if one replaces the $\ell_0$-based regularization in (P0) with $\ell_1$-based regularization, one obtains the related problem

$$\begin{aligned} \text{(P1)} \quad \min_{L,S} \quad & \|\widetilde{X} - L - S\|_F^2 + 2\tau \|S\|_1 \\ \text{s.t.} \quad & \mathbf{rank}(L) \leq r. \end{aligned}$$

It is again straightforward to show that the resulting block coordinate descent updates for (P1) take the form:

$$\begin{aligned} S^{k+1} &= \mathbf{soft}_{\tau}(\widetilde{X} - L^k) \\ L^{k+1} &= \mathbf{TSVD}_r(\widetilde{X} - S^{k+1}). \end{aligned} \tag{5.30}$$

The first iteration of (5.30) yields $S^1 = \widehat{S}_{\tau}^{\mathrm{ST}}$ and $L^1 = \mathbf{TSVD}_r(\widetilde{X}_{\tau}^{\mathrm{ST}})$, which is precisely the soft thresholding-based estimation scheme from Section 5.7. Problem (P1) is still nonconvex due to the rank constraint, but the block coordinate descent updates (5.30) are again guaranteed to converge to a critical point of the cost function.

Theorem V.8 characterizes the asymptotic accuracies of the singular vectors of the low-rank matrices produced after one iteration of the updates (5.29) and (5.30). In particular, it predicts that the first soft thresholding-based update will produce a poor estimate of the underlying low-rank matrix in the dense outlier regime. However, one might expect the accuracy of one or both alternating schemes to improve after subsequent iterations. In particular, one might expect that multiple iterations of (5.30) can salvage the performance of the soft thresholding-based update scheme in the dense outlier regime. However, our empirical experiments indicate that this is not the case.

In our experiments, we generate synthetic rank-1 instances of model (5.3). Specifically, we generate $L = \theta uv^T \in \mathbb{R}^{m \times n}$ with $u$ and $v$ random vectors on the unit sphere with $m = 128$, $n = 1280$, and $\theta = 4$. We populate the noise matrix $G$ with i.i.d. Gaussian random variables by setting $\sigma = 2$, and we populate the sparse matrix $S$ with i.i.d. Laplacian random variables with $\sigma_q = 5$ for a given outlier probability $p_s$. Given an instance $\widetilde{X}$ of this model, we then perform 10 iterations of the alternating updates (5.29) and (5.30) to estimate the underlying signals $u$ and $v$. In each case, we use the regularization parameter (threshold) $\tau = \tau^{\star}$ recommended by (5.22). For reference, we also compare to the oracle alternating scheme that inserts zeros at the known outlier locations during the $S$ updates rather than data-driven thresholding.

Figure 5.3 plots the accuracy $|\langle u^k, u\rangle|^2$ of the first left singular vector of $L^k$ for each update scheme as a function of iteration. The results are averaged over 50 random realizations of the model. Figure 5.3 shows that the performance of each method is determined by the accuracy after the *first iteration*—i.e., subsequent iterations do not improve the accuracy of the estimated singular vectors.[2] Thus, as predicted by

---

[2] In fact, for soft thresholding in the dense outlier regime, performance *degrades* after subsequent iterations.

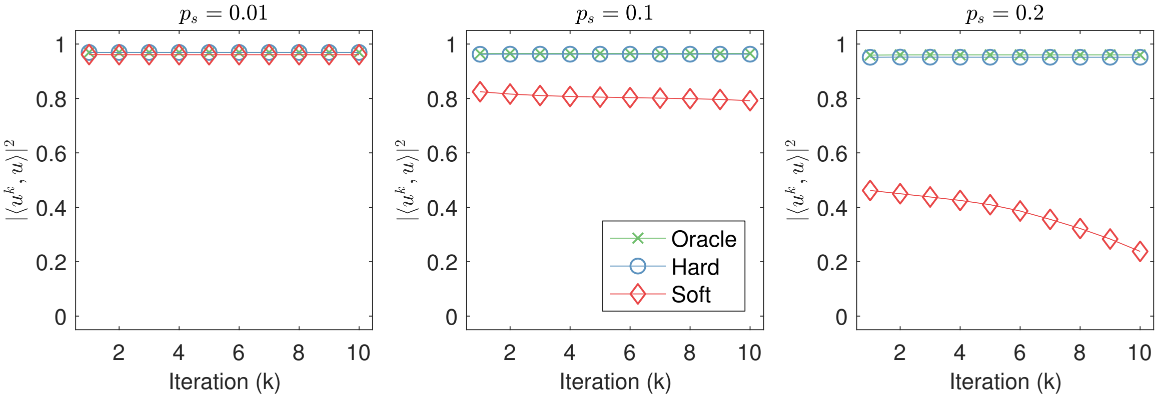


Figure 5.3: Accuracy of the first left singular vector $u^k$ of the low-rank updates $L^k$ from (5.29) and (5.30) as a function of iteration. Each panel corresponds to a different outlier probability $p_s$. The three curves in each panel depict the performance of the variations of alternating minimization where the $S$-updates are performed using hard thesholding (Hard), soft thresholding (Soft), and the oracle sparse estimator (Oracle) that inserts zeros at the known outlier locations.

Theorem V.8, all three update schemes perform equally well in the sparse outlier regime, while the soft thresholding-based updates (5.30) break down in the dense outlier regime.

## 5.10 Conclusions

In this chapter, we studied the problem of recovering a low-rank matrix corrupted by random noise and outliers. In particular, motivated by the sparse estimation literature, we considered outlier rejection schemes that apply hard or soft thresholding, respectively, to the elements of the data matrix. We analyzed the accuracy of the low-rank matrix estimated by applying PCA to the outlier-rejected matrix produced by each thresholding method by comparing it to an oracle estimator that replaces the known outlier-corrupted entries of the data matrix with zeros. Our analysis reveals a surprising result. In the sparse outlier regime, both hard and soft thresholding asymptotically achieve oracle performance. However, in the dense outlier regime, hard thresholding again achieves oracle performance, but soft thresholding does not—in fact, the principal components produced by soft thresholding are asymptotically orthogonal to the latent principal components.

# CHAPTER VI

# Efficient Learning of Dictionaries with Low-Rank Atoms

## 6.1 Introduction

The sparsity of signals and images in a transform domain or dictionary has been extensively exploited in applications such as compression, denoising, and inverse problems in imaging and image processing. In particular, the data-driven adaptation of sparse signal models such as the synthesis model has shown promise in numerous applications [100–103]. Given a set of training signals $\{y_i\}_{i=1}^N$ that are represented as columns of a training matrix $Y \in \mathbb{C}^{n\times N}$, the goal of dictionary learning (DL) is to learn a dictionary $D \in \mathbb{C}^{n\times J}$ and a matrix $X \in \mathbb{C}^{J\times N}$ of sparse codes such that $Y \approx DX$. The DL problem is often formulated as follows [104]:

$$\begin{aligned} \text{(P0)} \quad & \min_{D,X} \ \|Y - DX\|_F^2 \\ & \text{s.t.} \ \ \|x_i\|_0 \le s \ \forall i, \ \ \|d_j\|_2 = 1 \ \forall j, \end{aligned}$$

where $x_i$ and $d_j$ denote the $i$th column of $X$ and the $j$th column (or atom) of $D$ respectively, and $s$ denotes a target sparsity level for each signal. The $\ell_0$ "norm" measures sparsity and counts the number of non-zero entries in a vector. The columns of $D$ are set to unit norm to avoid the scaling ambiguity [105]. Various alternative versions of (P0) exist that replace the $\ell_0$ "norm" with other sparsity-promoting functions, or enforce additional properties on the dictionary [106–108], or enable dictionary learning in an online manner [109].

Dictionary learning algorithms [104, 109–113] typically attempt to solve (P0) or its variants in an alternating manner by performing a sparse coding step (updating $X$) followed by a dictionary update step (updating $D$). Some algorithms also partially update the coefficients in $X$ in the dictionary update step, while a few recent methods attempt to solve for the variables jointly and iteratively [114]. However, (P0) is non-convex and NP-hard, and most popular algorithms such as K-SVD [104] lack proven convergence guarantees, and tend to be computationally expensive. Some recent works [115–118] have studied the convergence of specific DL algorithms (typically making restrictive assumptions such as noiseless data, etc., for their convergence results), but these approaches have not been demonstrated to be advantageous in applications such as inverse problems. Bao et al. [117] find that their method, although a fast proximal scheme, denoises less effectively than K-SVD.

In this chapter, we propose a novel framework for structured dictionary learning. We model the atoms of the dictionary, after reshaping them into matrices, as low-rank. Importantly, imposing the low-rank structure often leads to comparable or improved performance over unstructured dictionary learning methods in practice. We also use an $\ell_0$ sparsity penalty for the coefficients. Although the proposed DL formulation is highly nonconvex, we develop an efficient block coordinate descent algorithm for it and present a convergence analysis for the approach. Our numerical experiments demonstrate the suitability and usefulness of learning low-rank atom dictionaries in applications (inverse problems) involving limited data.

## 6.2 Problem Formulation and Algorithm

This section presents our DL problem formulation with structured (low-rank) atoms and an efficient algorithm for it.

### 6.2.1 Dictionary Learning Problem Formulation

We consider a dictionary learning formulation with a sparsity penalty in this chapter. In particular, we define $C \triangleq X^H$ in (P0), and replace the $\ell_0$ "norm" constraints with an *overall* sparsity penalty $\|X\|_0 \triangleq \sum_{i=1}^{N} \|x_i\|_0 = \|C\|_0 = \sum_{j=1}^{J} \|c_j\|_0$. In addition, we consider a form of structured dictionary learning for images or image patches, wherein we model the columns $d_j \in \mathbb{C}^n$ of the dictionary $D$, after reshaping them into matrices, as low-rank. We refer to this as the DIctioNary with lOw-ranK AToms (DINO-KAT) model. The low-rank model on the reshaped atoms is motivated by our empirical observation that (unstructured) dictionaries learned from image patches often have quickly decaying singular values. When the training matrix $Y$ consists of vectorized versions of $\sqrt{n} \times \sqrt{n}$ (or rectangular) image patches, the dictionary atom vectors are reshaped (by stacking column-wise the vector entries) into similarly sized matrices. Denoting by $\mathcal{R}(\cdot)$ the operator that reshapes an atom into a matrix, our problem formulation for DL is as follows:

$$\text{(P1)} \quad \min_{D,C} \; \|Y - DC^H\|_F^2 + \lambda^2 \|C\|_0$$
$$\text{s.t.} \;\; \mathbf{rank}(\mathcal{R}(d_j)) \leq r, \; \|d_j\|_2 = 1, \; \|c_j\|_\infty \leq L, \; \forall j.$$

Here, $\lambda^2$ with $\lambda > 0$, is a sparsity regularization parameter and $r > 0$ denotes the maximum allowed rank for reshaped atoms.

The objective in (P1) is invariant to joint scaling of any pair $(d_j, c_j)$ as $(\alpha d_j, \alpha^{-1} c_j)$, for $\alpha \neq 0$. Therefore, similar to Problem (P0), the constraint $\|d_j\|_2 = 1$ helps remove this scaling ambiguity. The $\ell_\infty$ constraints in (P1) prevent pathologies that could theoretically arise (e.g., unbounded algorithm iterates) due to the objective being non-coercive [119].[1] In practice, we set $L$ very large, and the constraint is typically inactive.

Unlike the sparsity constraints in (P0), Problem (P1) penalizes the number of non-zeros in the (entire) coefficient matrix, allowing variable sparsity levels across the training signals. For example, in imaging or image processing applications, the dictionary is usually learned on (overlapping) image patches. Patches from different regions of an image typically contain different amounts of information, and thus enforcing a fixed or common sparsity for various patches does not reflect typical image properties (i.e., is restrictive) and usually leads to poor performance in applications.

When $r = \sqrt{n}$ for $\mathcal{R}(d_j) \in \mathbb{C}^{\sqrt{n} \times \sqrt{n}}$, the rank constraints in (P1) are inactive, and Problem (P1) corresponds to an unstructured DL formulation [119]. Structured DINO-KAT models (i.e., with small rank $r \ll \sqrt{n}$) learned using (P1) may be less prone to over-fitting problems in applications involving limited or corrupted data. We demonstrate this through some applications in Section 6.3.

Formulation (P1) is by itself useful for adaptive sparse representation (and eventual compression) of data with structured dictionaries. Learning sparse approximations of the data can also be an effective way to denoise the data. (P1) can also be used as a regularizer in inverse problems [120], where it can enable data-adaptive image reconstructions. We investigate this application extensively in the rest of this thesis.

---

[1] For example, consider a dictionary $D$ that has a (low-rank) column $d_j$ that repeats. The representation $DC^H = \sum_{k=1}^{J} d_k c_k^H$ in (P1) in this case could contain both the terms $d_j c_j^H$ and $-d_j c_j^H$ with $c_j$ that is highly sparse, and the objective would be invariant to (arbitrarily) large scalings of $c_j$. The infinity norm constraint precludes this eventuality.

### 6.2.2 Algorithm and Computational Cost

We propose an iterative block coordinate descent method [121] for (P1) that updates the coefficient columns $c_j$ (of $C$) and atoms $d_j$ (of $D$) sequentially. Specifically, for each $1 \leq j \leq J$, we first solve (P1) with respect to $c_j$, keeping the other variables fixed (the *sparse coding step*). Once $c_j$ is updated, we solve (P1) with respect to $d_j$, keeping all other variables fixed (the *dictionary atom update step*).

#### 6.2.2.1 Sparse Coding Step

Here, we minimize (P1) with respect to $c_j$. This leads to the following problem, where the matrix $E_j \triangleq Y - \sum_{k \neq j} d_k c_k^H$ is computed using the most recent estimates of other atoms and coefficients:

$$\begin{aligned} \min_{c_j} \quad & \|E_j - d_j c_j^H\|_F^2 + \lambda^2 \|c_j\|_0 \\ \text{s.t.} \quad & \|c_j\|_\infty \leq L. \end{aligned} \tag{6.1}$$

Assuming $L > \lambda$, it can be shown by following a proof strategy identical to Proposition 1 in [119] that a global minimizer of (6.1) is

$$\hat{c}_j = \min(|H_\lambda(E_j^H d_j)|,\ L 1_N) \odot e^{j \angle E_j^H d_j}. \tag{6.2}$$

In (6.2), the operator $H_\lambda(\cdot)$ is the elementwise hard thresholding operator

$$[H_\lambda(b)]_i = \begin{cases} 0 & |b_i| < \lambda \\ b_i & |b_i| \geq \lambda \end{cases} \tag{6.3}$$

that sets elements with magnitude less than $\lambda$ to zero, $1_N$ denotes a vector of ones of length $N$, $\odot$ denotes element-wise multiplication, and $\min(\cdot, \cdot)$ denotes element-wise minimum. The term $e^{j\angle c}$ is computed element-wise for vector arguments $c$, with $\angle$ denoting the phase.

#### 6.2.2.2 Dictionary Atom Update Step

In this step, we optimize (P1) with respect to the atom $d_j$, holding other variables fixed. This leads to the problem:

$$\begin{aligned} \min_{d_j} \quad & \|E_j - d_j c_j^H\|_F^2 \\ \text{s.t.} \quad & \mathbf{rank}(\mathcal{R}(d_j)) \leq r,\ \|d_j\|_2 = 1. \end{aligned} \tag{6.4}$$

Proposition VI.1 provides the solution to Problem (6.4). It relies on the full singular value decomposition (SVD) of an appropriate matrix. We assume $\mathcal{R}(d_j) \in \mathbb{C}^{\sqrt{n}\times\sqrt{n}}$, and let $\sigma_i$ denote the $i$th diagonal entry of the diagonal matrix $\Sigma$.

**Proposition VI.1.** *Given $E_j \in \mathbb{C}^{n\times N}$ and $c_j \in \mathbb{C}^N$, let $U_r\Sigma_r V_r^H$ denote an optimal rank-$r$ approximation to $\mathcal{R}(E_jc_j) \in \mathbb{C}^{\sqrt{n}\times\sqrt{n}}$ that is obtained using the $r$ leading singular vectors and singular values of the full SVD $\mathcal{R}(E_jc_j) \triangleq U\Sigma V^H$. Then, a global minimizer in Problem (6.4), upon reshaping, is*

$$\mathcal{R}(\hat{d}_j) = \begin{cases} \dfrac{U_r\Sigma_r V_r^H}{\|\Sigma_r\|_F}, & \text{if } c_j \neq 0 \\ v_1, & \text{if } c_j = 0, \end{cases} \tag{6.5}$$

*where $v_1$ is the reshaped first column of the $n \times n$ identity matrix. The solution is unique if and only if $c_j \neq 0$, and $\sigma_r > \sigma_{r+1}$ or $\sigma_r = 0$.*

*Proof.* First, because $\|d_j\|_2 = 1$, we have

$$\|E_j - d_jc_j^H\|_F^2 = \|E_j\|_F^2 + \|c_j\|_2^2 - 2\,\mathrm{Re}\{d_j^H E_jc_j\}. \tag{6.6}$$

Upon substituting (6.6) into (6.4), Problem (6.4) simplifies to

$$\begin{aligned} \max_{d_j\in\mathbb{C}^n} \quad & \mathrm{Re}\left\{\mathbf{tr}(\mathcal{R}(d_j)^H\mathcal{R}(E_jc_j))\right\} \\ \text{s.t.} \quad & \mathbf{rank}(\mathcal{R}(d_j)) \leq r,\ \|d_j\|_2 = 1. \end{aligned} \tag{6.7}$$

Next, let $\mathcal{R}(d_j) = A\Gamma B^H$, and $\mathcal{R}(E_jc_j) = U\Sigma V^H$ be full SVDs with $\gamma_i$ and $\sigma_i$ denoting the diagonal entries of $\Gamma$ and $\Sigma$, respectively. The problem then becomes

$$\begin{aligned} \max_{\Gamma}\ \max_{A,B} \quad & \mathrm{Re}\left\{\mathbf{tr}(B\Gamma A^HU\Sigma V^H)\right\} \\ \text{s.t.} \quad & \mathbf{rank}(\Gamma) \leq r,\ \|\Gamma\|_F = 1,\ A^HA = B^HB = I. \end{aligned} \tag{6.8}$$

For the inner maximization, we use $\mathrm{Re}\{\mathbf{tr}(B\Gamma A^HU\Sigma V^H)\} \leq \mathbf{tr}(\Gamma\Sigma)$ [122], with the upper bound attained when $A = U$ and $B = V$. The remaining problem with respect to (diagonal) $\Gamma$ is then

$$\begin{aligned} \max_{\{\gamma_i\}} \quad & \sum_{i=1}^{r}\gamma_i\sigma_i \\ \text{s.t.} \quad & \sum_{i=1}^{r}\gamma_i^2 = 1,\ \gamma_j = 0,\ r+1 \leq j \leq \sqrt{n}. \end{aligned} \tag{6.9}$$

Using the Cauchy-Schwarz inequality, $\hat{\gamma}_i = \sigma_i / \sqrt{\sum_{i=1}^{r} \sigma_i^2}$ for $1 \leq i \leq r$, and $\hat{\gamma}_i = 0$ for $r + 1 \leq i \leq \sqrt{n}$ is clearly optimal. The derived solution for the optimal $\mathcal{R}(\hat{d}_j)$ then simply corresponds to a normalized version of the rank-$r$ approximation to $\mathcal{R}(E_j c_j)$. Clearly, the solution in (6.7) is unique if and only if $E_j c_j \neq 0$, and $\sigma_r > \sigma_{r+1}$ or $\sigma_r = \sigma_{r+1} = 0$. Any $d \in \mathbb{C}^n$ satisfying the constraints in (6.7) is a (non-unique) minimizer when $E_j c_j = 0$. In particular $\mathcal{R}(\hat{d}_j) = v_1$ works.

Lastly, to complete the Proposition's proof, we show that $E_j c_j = 0$ in our algorithm if and only if $c_j = 0$. Since $c_j$ here was obtained as a minimizer in the preceding sparse coding step (6.1), we have the following result $\forall\ c \in \mathbb{C}^N$ with $\|c\|_\infty \leq L$ and $\tilde{d}_j$ denoting the $j$th atom in the preceding sparse coding step:

$$\|E_j - \tilde{d}_j c_j^H\|_F^2 + \lambda^2 \|c_j\|_0 \leq \|E_j - \tilde{d}_j c^H\|_F^2 + \lambda^2 \|c\|_0. \tag{6.10}$$

If $E_j c_j = 0$, the left hand side above is $\|E_j\|_F^2 + \|c_j\|_2^2 + \lambda^2 \|c_j\|_0$, which is clearly minimal when $c_j = 0$. Thus, when $E_j c_j = 0$, we must have $c_j = 0$. $\square$

#### 6.2.2.3 Computational Cost

The overall block coordinate descent DINO-KAT algorithm involves $J$ sparse coding and dictionary atom update steps in each outer iteration. Assuming $J \propto n$ and $N \gg J, n$, the cost per iteration of the algorithm scales as $O(Nn^2)$. This cost is dominated by various matrix-vector products. The costs of the truncated hard-thresholding (6.2) and low-rank approximation (6.5) steps are negligible. The per-iteration cost for our method is lower than that for learning an $n \times J$ dictionary $D$ in (P0) using K-SVD [104, 123], which scales (with $s \propto n$ and $J \propto n$) as $O(Nn^3)$. Our algorithms also converge quickly in practice and outperform K-SVD in applications [119].

### 6.2.3 Convergence of the DINO-KAT Learning Algorithm

We briefly present results on the convergence behavior of the proposed algorithm. The proofs of the results in this section follow using similar arguments as in the proofs of related results in [119].

The constraints $\mathbf{rank}(\mathcal{R}(d_j)) \leq r$, $\|d_j\|_2 = 1$, and $\|c_j\|_\infty \leq L$ in (P1) can instead be added as penalties in the cost by using barrier functions $\phi(d_j)$, $\chi(d_j)$, and $\psi(c_j)$, respectively, that take the value $+\infty$ when the corresponding constraint is violated, and are zero otherwise. Problem (P1) is then written in unconstrained form with objective

$$
\begin{aligned}
g(C, D) &= g(c_1, ..., c_J,\ d_1, ..., d_J) \\
&= \|Y - DC^H\|_F^2 + \sum_{j=1}^{J} (\lambda^2 \|c_j\|_0 + \phi(d_j) + \chi(d_j) + \psi(c_j)).
\end{aligned}
\tag{6.11}
$$

Our convergence results are as follows. First, we have the following monotonicity and limit consistency result.

**Theorem VI.2.** *Let $\{C^t, D^t\}$ denote the iterate sequence generated by the algorithm with training data $Y \in \mathbb{C}^{n \times N}$ and initial $(C^0, D^0)$. Then, the objective sequence $\{g^t\}$ with $g^t \triangleq g(C^t, D^t)$ is monotone decreasing and converges to a finite value, say $g^* = g^*(C^0, D^0)$. Moreover, the iterate sequence $\{C^t, D^t\}$ is bounded, and all its accumulation points are equivalent in the sense that they achieve the same objective value $g^*$.*

Theorem VI.2 establishes that for each initialization, all the accumulation points of the (bounded) iterate sequence of the algorithm achieve the same value $g^*$ of the objective, and are equivalent. Because the distance between a bounded sequence and its compact set of accumulation points converges to zero, we have the following corollary.

**Corollary VI.3.** *For each $(C^0, D^0)$, the iterate sequence in the algorithm converges to an equivalence class of accumulation points.*

Finally, the following theorem establishes that the iterates in our algorithm converge to the set of critical points [124] (or generalized stationary points) of $g(C, D)$. Here, $\sigma_k$ denotes the $k$th singular value in the full SVD of a (square) matrix.

**Theorem VI.4.** *Let $\{C^t, D^t\}$ denote the bounded iterate sequence in the algorithm with training data $Y$ and initial $(C^0, D^0)$. Suppose each accumulation point $(C, D)$ of the iterate sequence is such that for each $1 \leq j \leq J$ with $E_j \triangleq Y - DC^H + d_j c_j^H$, the vector $E_j^H d_j$ has no entry with magnitude $\lambda$, and $\sigma_r(\mathcal{R}(E_j c_j)) > \sigma_{r+1}(\mathcal{R}(E_j c_j))$ or $\sigma_r(\mathcal{R}(E_j c_j)) = 0$. Then, every accumulation point of the iterate sequence is a critical point of $g(C, D)$. Moreover, the sequences with terms $\|D^t - D^{t-1}\|_F$ and $\|C^t - C^{t-1}\|_F$ respectively, both converge to zero.*

Theorem VI.4 says that $\|D^t - D^{t-1}\|_F \to 0$ and $\|C^t - C^{t-1}\|_F \to 0$, which are necessary but not sufficient conditions for the convergence of the sequences $\{D^t\}$ and $\{C^t\}$. Although Theorem VI.4 assumes simple conditions (e.g., nondegenerate

Figure 6.1: Images: Barbara, Boat, Hill, and a Microscopy image.

singular values) on the accumulation points, we conjecture that these conditions hold for each accumulation point with probability 1 when the training signals are drawn i.i.d. from an absolutely continuous probability measure.

## 6.3 Numerical Experiments

This section presents numerical results illustrating the convergence of the proposed DL method and its application to inverse problems.

### 6.3.1 Convergence Behavior

To study the practical convergence behavior of the proposed algorithm for (P1), we extracted $3 \times 10^4$ training patches of size $8 \times 8$ from randomly chosen locations in the images Barbara, Boat, and Hill shown in Figure 6.1. We used (P1) with $\lambda = 69$ to learn a $64 \times 256$ dictionary for the data, with reshaped atoms of size $8 \times 8$. We set $C^0 = 0$, and $D^0$ to be the overcomplete DCT [100].

Figure 6.2 shows the behavior of the algorithm for various choices of atom rank $r$. The objective in (P1) converged (Figure 6.2(a)) monotonically and quickly over the iterations. The convergence was faster for smaller values of $r$. Figure 6.2(b) shows the normalized sparse representation error (NSRE) $\|Y - DC^H\|_F / \ \|Y\|_F$ for the training data. (The sparsity $\|C\|_0/Nn$ stayed at about 3% during the algorithm iterations for all choices of $r$.) The NSRE improved significantly beyond the first iteration, indicating the success of the proposed DL scheme. Importantly, the NSRE values achieved for small values of $r$ (DINO-KAT cases) are very similar to the value in the full-rank [119] ($r = 8$) case. This suggests that the low-rank model on reshaped dictionary atoms, despite being a constrained model, can effectively model properties of natural images. Lastly, both $\|D^t - D^{t-1}\|_F$ (Figure 6.2(c)) and $\|C^t - C^{t-1}\|_F$ (Figure 6.2(d)) converge towards 0, as predicted by Theorem VI.4, with quicker convergence observed for the low-rank case.

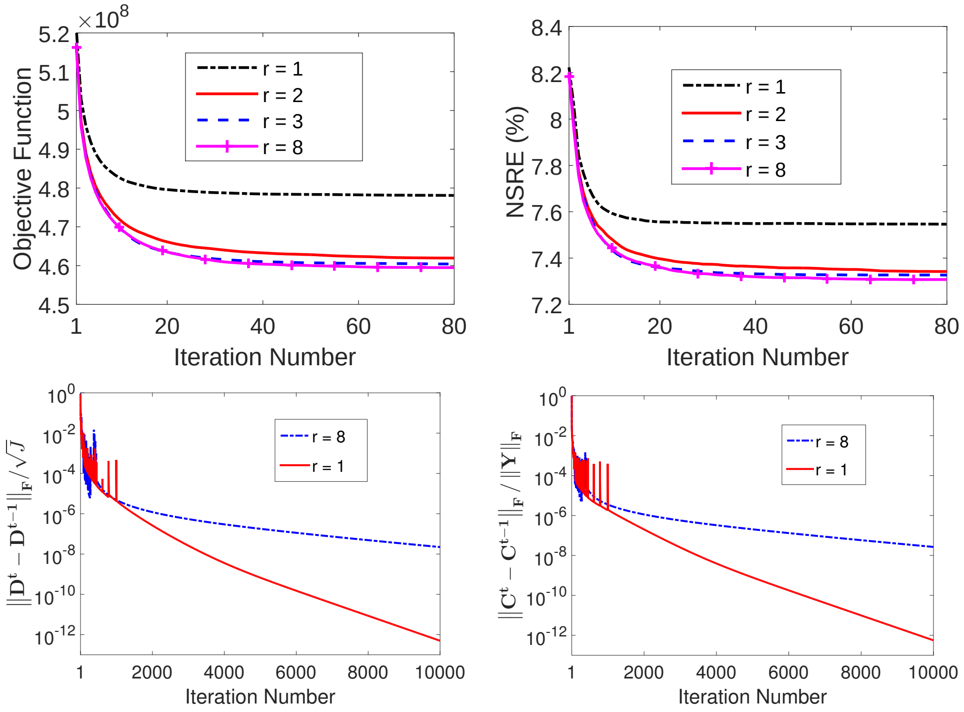


Figure 6.2: Algorithm behavior: Objective function (top left); NSRE (top right); normalized changes between successive $D$ iterates $\|D^t - D^{t-1}\|_F/\sqrt{J}$ (bottom left); and normalized changes between successive $C$ iterates $\|C^t - C^{t-1}\|_F/\|Y\|_F$ (bottom right).

### 6.3.2 Inverse Problem: Blind Compressed Sensing

In compressed sensing (CS) [69], the goal is to recover an image $x \in \mathbb{C}^p$ from its measurements $y = Ax + h$, where $A \in \mathbb{C}^{m\times p}$ with $m \ll p$ is a known sensing matrix, and $h$ denotes noise. CS methods reconstruct the image (or video) by modeling it (or its patches) as sparse in a known transform or dictionary. Here, we consider blind compressed sensing (BCS) [120], where the sparse model is assumed unknown a priori. The image and the model are jointly estimated in BCS. We propose the following BCS problem based on (P1):

$$\text{(P2)} \quad \min_{x,D,B} \; \nu\|Ax - y\|_2^2 + \sum_{k=1}^{N} \|P_k x - D b_k\|_2^2 \; + \; \lambda^2 \|B\|_0$$
$$\text{s.t.} \quad \|b_i\|_\infty \leq L, \;\; \textbf{rank}(\mathcal{R}(d_j)) \leq r, \;\; \|d_j\|_2 = 1, \; \forall i, j.$$

Here, $B$ is a matrix with sparse codes $b_i$ as its columns. We propose an algorithm for (P2) that alternates between updating $(D, B)$ and $x$. In the first step, $x$ is fixed,

| Case | Initial | Cubic | Fixed $D$ | $r = 8$ | $r = 3$ | $r = 2$ | $r = 1$ |
|---|---|---|---|---|---|---|---|
| 50% | 11.1 | 36.9 | 34.8 | **37.9** | **37.9** | **37.9** | **37.9** |
| 30% | 9.7 | 34.9 | 31.9 | 35.6 | 35.9 | **36.0** | 35.9 |
| 20% | 9.1 | 33.4 | 30.2 | 34.6 | **34.9** | 34.8 | 34.8 |
| 10% | 8.6 | 31.0 | 27.8 | 32.3 | 32.3 | 32.4 | **32.6** |

Table 6.1: Inpainting PSNR values in decibels (dB) at various percentages of measured pixels for the initial image, the result with cubic interpolation, the results using (P2) with $r = 1$, $r = 2$, $r = 3$, and $r = 8$, and for the reconstructions obtained with fixed dictionary in our algorithm. Results are for the Microscopy image. The best PSNRs are marked in bold.

and the problem reduces to DL using (P1). The second step involves a simple least squares problem in $x$ that can be solved either directly or using iterative solvers such as the proximal gradient method.

Here, we study the usefulness of (P2) for dynamic MRI (dMRI) and for compressive scanning electron microscopy (SEM) [125].

#### 6.3.2.1 Compressive SEM

We consider the SEM image [126] in Figure 6.1 and simulate CS (inpainting) by sampling a subset of image pixels. We used (P2) with a $64 \times 20$ $D$ learned on $8 \times 8$ overlapping image patches using 100 alternations between $(D, B)$ and $x$ with $\nu = 10^7$ and $\lambda = 0.05$. (We use larger $\lambda$ values during initial alternations, which accelerates convergence.) We update $(D, B)$ using 1 iteration of the algorithm for (P1). We set the initial $x = A^{\dagger}y$, the initial $B = 0$, and the initial $D$ was a $64 \times 20$ DCT (generated as in [127]).

Table 6.1 shows the PSNR values at various undersampling factors for reconstructions obtained using our method, and with cubic interpolation (using Matlab's *griddata* function), and using the proposed method with fixed $D$ (fixed to initialization). The proposed BCS scheme clearly achieves better reconstructions compared to cubic interpolation or conventional CS (fixed $D$). Importantly, in cases involving very limited data, enforcing the low-rank constraint ($r = 1, 2, 3$) on reshaped ($8 \times 8$) dictionary atoms leads to considerably better PSNRs compared to the unstructured ($r = 8$) case.

| Acceleration | 4x | 8x | 12x | 16x | 20x | 24x |
|---|---|---|---|---|---|---|
| NRMSE (L+S) % | 10.93 | 14.00 | 15.80 | 18.87 | 21.33 | 23.36 |
| NRMSE (Fixed D) % | 11.29 | 13.76 | 15.33 | 18.31 | 20.77 | 22.82 |
| NRMSE (r = 5) % | 10.85 | 13.08 | 14.37 | 17.01 | 19.19 | 21.35 |
| NRMSE (r = 1) % | **10.57** | **12.90** | **14.20** | **16.77** | **18.74** | **20.91** |
| Gain over L + S (dB) | 0.29 | 0.71 | 0.92 | 1.03 | 1.13 | 0.96 |
| Gain over r = 5 (dB) | 0.23 | 0.12 | 0.10 | 0.13 | 0.21 | 0.18 |

Table 6.2: NRMSE values at several undersampling factors for the L+S method and for the algorithm for (P2) with $r = 5$ (full rank), $r = 1$ (DINO-KAT MRI) and fixed dictionary cases. The best NRMSE values for each undersampling are marked in bold, and the improvements by DINO-KAT MRI are indicated in decibels (dB).

#### 6.3.2.2 CS Dynamic MRI

We perform simulations with the multi-coil Cartesian-sampled cardiac perfusion data used in prior work [1]. Fully-sampled data with an image matrix size of 128 × 128 and 40 temporal frames were retrospectively undersampled (in k-t space) using a different variable-density random Cartesian undersampling pattern for each time frame. We use normalized root mean square error (NRMSE), defined as

$$\mathbf{NRMSE}(x_{\text{recon}}) = \frac{\|x_{\text{recon}} - x_{\text{ref}}\|_2}{\|x_{\text{ref}}\|_2}, \tag{6.12}$$

where $x_{\text{ref}}$ is a reference reconstruction computed from the fully-sampled data and $x_{\text{recon}}$ the reconstruction from the undersampled data, as our performance metric. We compare the performance of the proposed method to that of the recent L+S method [1, 4], where the dynamic data is modeled as a sum of a low-rank (L) and a sparse (S) (with respect to a temporal Fourier transform) component. For the L+S method, the parameters $\lambda_L$ and $\lambda_S$ were tuned to obtain good NRMSE in our experiments. For the proposed method for (P2), we use spatiotemporal patches of size $8 \times 8 \times 5$ with spatial and temporal patch overlap strides of 2 pixels, $\nu = 66.67$, $\lambda = 0.025$, and we initialize the algorithm by setting $x$ to be the output of the L+S method, $D$ to be the $320 \times 320$ DCT matrix, and $B = 0$.

Table 6.2 lists the NRMSE values for conventional L+S [1] and the proposed DINO-KAT MRI ($r = 1$ and $64 \times 5$ (space-time) reshaped atoms) method at various undersampling factors. The NRMSEs achieved by the algorithm for (P2) with fixed $D$ (DCT) and for the adaptive $r = 5$ (full rank) case are also shown. DINO-KAT MRI with rank-1 atoms provides the best reconstruction errors for each undersampling factor tested. In particular, it provides improvements up to 1.13 dB over the L+S

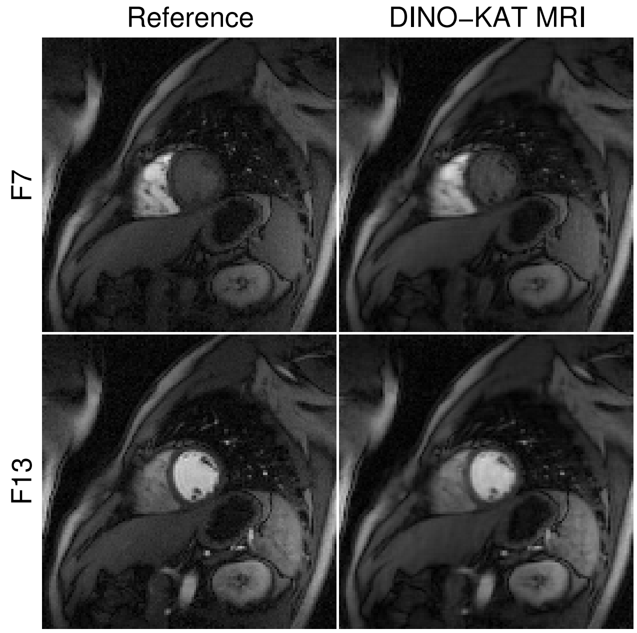


Figure 6.3: 8x undersampling: Frames 7 and 13 of the proposed DINO-KAT MRI ($r = 1$) reconstruction along with the reference frames.

method and up to 0.23 dB over the full rank $r = 5$ case. Figure 6.3 shows two representative frames of the DINO-KAT MRI reconstruction from 8x undersampled data. The reconstructed frames are visually very close to the reference frames.

## 6.4 Conclusions

In this chapter, we investigated a novel framework for structured synthesis dictionary learning. In particular, we considered the learning of dictionaries whose atoms or columns, after reshaping, have low rank, and we used the $\ell_0$ norm to measure sparsity in our formulations. We adopted a highly efficient block coordinate descent approach for dictionary learning in our algorithm, and we presented theoretical convergence results for the highly non-convex problem. Importantly, the proposed structured dictionary learning method converges faster in practice than unstructured dictionary learning methods. Our experiments showed the promise and efficiency of the proposed schemes for applications such as image denoising and inpainting.

# CHAPTER VII

# Online Data-Driven Image Reconstruction Using Efficiently Learned Dictionaries

## 7.1 Introduction

Signal models involving sparsity, low-rank, and other properties have been widely used in image and video processing. Such models are especially important in inverse problem settings such as in denoising, deblurring, inpainting, etc., where often they are used to construct regularizers that reflect known or assumed properties of data. While sparsity in wavelet or discrete cosine transform (DCT) domains has been extensively used in image and video restoration [128,129], more recently the data-driven adaptation of synthesis dictionary [104, 111] or sparsifying transform [130, 130, 131] models has shown promise in numerous applications [101, 103].

There has also been growing interest in dictionary learning-based dynamic image reconstruction or restoration methods [12,132,133]. For example, in blind compressed sensing [103, 133], the dictionary for the underlying image or video is assumed unknown, and it is estimated together with the image from undersampled measurements. Recently, the learning of a structured DIctioNary with lOw-ranK AToms (DINOKAT) was explored [12], where the dictionary atoms are constrained to be low-rank upon reshaping the atoms into appropriate-sized matrices (e.g., space-time). Such structured dictionaries are particularly useful in applications involving limited or corrupted data, because they are less prone to over-fitting problems and provide better reconstructions in scenarios such as blind compressed sensing compared to methods such as L+S [12].

For inverse problems involving large-scale or streaming data—e.g., in interventional imaging or inpainting large or streaming videos—it is often critical to obtain reconstructions in an online or time-sequential manner to limit latency, because batch methods that process all the data at once are typically prohibitively expensive in terms of time and memory usage. Methods for online or sequential learning of dictionary or transform models from streaming measurements have been recently proposed [109, 134, 135].

One important class of dynamic image reconstruction problems is dynamic magnetic resonance imaging (MRI). In dynamic MRI, measurements are collected in the form of samples in k-space or Fourier space of the object, and the samples are acquired sequentially over time of the dynamic object. MRI is a relatively slow imaging modality due to the sequential nature of the measurements, and, as a result, there has been much interest in accelerating MRI acquisition by sampling fewer k-space locations. Methods to reconstruct MR images from limited measurements typically assume that the image is sparse in some transform domain or dictionary [136] and optimize problems with sparsity-based regularizers like the $\ell_0$ and $\ell_1$ norms.

Dynamic MRI data are inherently or naturally undersampled because the object is changing as the data is collected. Various techniques have been proposed for reconstructing dynamic MR image sequences from limited (randomly sampled) k-t space measurements [1, 137, 138]. Such methods may achieve improved spatial or temporal resolution by using more explicit signal models rather than conventional k-space data sharing approaches, where data is pooled in time to make sets of k-space data such as in the form of a Casorati matrix [139], but they typically achieve increased accuracy at the price of increased computation.

While sparse signal models have been popular [137], alternative models have also been studied for dynamic MRI reconstruction in recent years including low-rank models [139–143]. The popular L+S method [1, 31] models the image sequence as the sum of a low-rank component (L) and a sparse component (S) and jointly estimates the components from k-t space data. The S component may be directly sparse or sparse in a known transform or dictionary. There has also been interest in dictionary learning-based approaches for dynamic MRI reconstruction [16, 133, 144], which tend to often involve expensive computation or memory use.

### 7.1.1 Contributions

In this chapter, we investigate a framework for online data-driven reconstruction of dynamic image sequences from linear (typically undersampled) measurements. In particular, we model the spatiotemporal patches of the underlying dynamic image sequence as sparse in an (unknown) adaptive dictionary model, and we develop a method to jointly estimate the dictionary, sparse codes, and images from streaming measurements. The proposed online data-driven reconstruction algorithm involves simple and efficient updates and requires a small, fixed amount of data to be stored in memory at a time, which greatly reduces the computation and memory demands during processing compared to conventional batch methods that process all of the data together.

We perform extensive numerical experiments that demonstrate the effectiveness of the proposed online method for performing video inpainting from limted and noisy pixels, as well as dynamic MRI reconstruction from highly undersampled measurements. The experiments show that our proposed method is able to learn dictionaries adaptively from corrupted measurements with important representational features that improve the accuracy of the reconstructions produced. Importantly, the proposed online method leads to lower reconstruction errors compared to an online method with a fixed DCT dictionary.

We also present a variation on our proposed method where we impose a unitary constraint on the learned dictionary, and we show that this modified problem can be efficiently solved with no inner block coordinate descent updates. We evaluate the low-rank atoms and unitary dictionary variations of our proposed method and show that both are useful in practice. Finally, we investigate the properties of our proposed online methods, including quantifying their ability to learn dictionaries from corrupted data by comparing to an online scheme with oracle dictionary learned from the patches of the true image sequence.

### 7.1.2 Organization

The chapter is organized as follows. In Section 7.2, we review the dictionary learning problem and introduce our framework for efficiently learning dictionaries with low-rank atoms. We formulate our proposed online dictionary learning method in Section 7.3 and present our algorithm for solving it. In Section 7.4 we present extensive numerical experiments that demonstrate the state-of-the-art performance of our proposed method on inverse problems such as video reconstruction and dynamic MRI. Finally, we conclude in Section 7.5 and discuss opportunities for future work.

## 7.2 Efficient Dictionary Learning

Given a set of signals (or vectorized image patches) that are represented as columns of a matrix $P \in \mathbb{C}^{n\times M}$, the goal of dictionary learning (DL) is to learn a dictionary $D \in \mathbb{C}^{n\times m}$ and a matrix $Z \in \mathbb{C}^{m\times M}$ of sparse codes such that $P \approx DZ$. Traditionally, the DL problem is often formulated [104] as

$$\begin{aligned} \min_{D,Z} \quad & \|P - DZ\|_F^2 \\ \text{s.t.} \quad & \|z_i\|_0 \leq s, \ \|d_j\|_2 = 1, \ \forall i, j, \end{aligned} \tag{7.1}$$

where $z_i$ and $d_j$ denote the $i$-th column of $Z$ and the $j$-th column of $D$, respectively, and $s$ denotes a target sparsity level for each signal. Here, the familiar $\ell_0$ "norm" counts the number of non-zero entries in a vector, and the columns of $D$ are set to unit norm to avoid scaling ambiguity between $D$ and $Z$ [105]. Various alternative versions of (7.1) exist that replace the $\ell_0$ "norm" with other sparsity-promoting functions, or enforce additional properties on the dictionary [106–108], or enable dictionary learning in an online manner [109].

Dictionary learning algorithms [104, 109–113] typically attempt to solve (7.1) or its variants in an alternating manner by performing a sparse coding step (updating $Z$) followed by a dictionary update step (updating $D$). Methods such as K-SVD [104] also partially update the coefficients in $Z$ in the dictionary update step, while a few recent methods attempt to solve for the variables jointly and iteratively [114]. However, (7.1) is non-convex and NP-hard, and most popular algorithms lack proven convergence guarantees and tend to be computationally expensive.

The DINO-KAT learning problem [12] is an alternative dictionary learning framework that imposes a low-rank constraint on the dictionary atoms. The problem formulation is

$$\begin{aligned} \min_{D,Z} \quad & \|P - DZ\|_F^2 + \lambda^2 \|Z\|_0 \\ \text{s.t.} \quad & \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \ \|d_i\|_2 = 1, \ \|z_l\|_\infty \leq L, \ \forall i, l, \end{aligned} \tag{7.2}$$

where the $\ell_0$ "norm" is applied elementwise to matrix arguments. The operator $\mathcal{R}(\cdot)$ reshapes dictionary atoms $d_i \in \mathbb{C}^n$ into matrices of size $n_1 \times n_2$ for some $n_1$ and $n_2$ such that $n = n_1 n_2$, and $r > 0$ is the maximum allowed rank for each reshaped atom. The dimensions of the reshaped atoms can be chosen on an application-specific basis. For example, in the case where spatiotemporal (3D) patches are extracted from dynamic data, the atoms could be reshaped into space-time (2D) matrices. Spatiotemporal patches of videos typically have high temporal correlation, so they may be well represented by a dictionary with low-rank space-time (reshaped) atoms [12].

The parameter $\lambda > 0$ in (7.2) controls the overall sparsity of the matrix $Z$, and enables variable sparsity levels across signals. The $\ell_\infty$ constraints for $L > 0$ prevent pathologies that could theoretically arise (e.g., unbounded algorithm iterates) due to the objective being non-coercive [119]. In practice, we set $L$ very large, and the constraint is typically inactive.

Unlike the fixed-sparsity-per-patch constraints in (7.1), the adaptive dictionary learning formulation (7.2) penalizes the number of non-zeros in the entire coefficient matrix, which allows for variable sparsity levels across the patches. Variable sparsity is a useful model for patch data in practice. For example, in imaging applications, the dictionary is usually learned on (possibly-overlapping) image patches. Patches from different regions of an image typically contain different amounts of information, and thus enforcing a fixed or common sparsity level across patches is too restrictive and does not reflect typical image properties, and thus it can lead to poor performance in practice.

## 7.3 Problem Formulation and Algorithms

We now present our proposed problem formulation for online dictionary learning-driven dynamic image reconstruction and our algorithm for solving it.

### 7.3.1 Problem Formulation

We propose an online image reconstruction framework based on an adaptive dictionary regularizer as in (7.2). Let $\{g^t \in \mathbb{C}^{N_x \times N_y}\}$ denote the sequence of dynamic image frames to be reconstructed. We assume that noisy, undersampled linear measurements of these frames are observed, and we process the streaming measurements in minibatches of $\tilde{M} \leq 1$ consecutive frames. Let $x^t$ denote the vectorized version of the 3D tensor obtained by (temporally) concatenating the $\tilde{M}$ consecutive frames of dynamic images. In practice, we construct $\{x^t\}$ using a sliding window (over time) strategy, which may involve overlapping or non-overlapping minibatches. We model the spatiotemporal (3D) patches of each $x^t$ as sparse with respect to a latent dictionary $D$. Under this model, we propose to solve the following online dictionary learning-driven image reconstruction problem for each time $t = 1, 2, 3, \ldots$

$$
\begin{aligned}
\text{(P1)} \quad \left\{\hat{x}^t, \hat{D}^t, \hat{Z}^t\right\} = \underset{x^t, D, Z^t}{\arg\min} \; & \frac{1}{K_t} \sum_{j=1}^{t} \rho^{t-j} \|y^j - A^j x^j\|_2^2 \\
& + \frac{\lambda_S}{K_t} \sum_{j=1}^{t} \rho^{t-j} \left( \sum_{l=1}^{M} \|P_l x^j - D z_l^j\|_2^2 + \lambda_Z^2 \|Z^j\|_0 \right) \\
& \text{s.t. } \|z_l^t\|_\infty \leq L, \; \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \; \|d_i\|_2 = 1, \; \forall i, l.
\end{aligned}
$$

In (P1), $j$ indexes time, and $y^t$ denotes the (typically undersampled) measurements that are related to the underlying frames $x^t$ that we would like to reconstruct through the linear sensing operator $A^t$. For example, in video inpainting, $A^t$ samples a subset of pixels in $x^t$, or in dynamic MRI, it corresponds to an undersampled Fourier encoding. The operator $P_l$ is a patch extraction matrix that extracts an $n_x \times n_y \times n_t$ spatiotemporal patch from $x^t$ as a vector. A total of $M$ (possibly) overlapping 3D patches are assumed. Matrix $D \in \mathbb{C}^{n \times m}$ with $n = n_x n_y n_t$ is the synthesis dictionary to be learned and $z_l^t \in \mathbb{C}^m$ is the unknown sparse code for the $l$-th patch of $x^t$, with $P_l x^t \approx D z_l^t$. Matrix $Z^t$ has $z_l^t$ as its columns. The weights $\lambda_S, \lambda_Z \geq 0$ are regularization parameters that control the relative adaptive dictionary regularization and sparsity of $Z$, respectively, in the model.

Problem (P1) jointly estimates the adaptive dictionary model for the patches of $x^t$ together with the underlying image frames. Note that, for each time index $t$, we only solve (P1) for the latest group of frames $x^t$ and the latest sparse coefficients $Z^t$, while the previous images and sparse coefficients are set to their estimates from previous minibatches (i.e., $x^j = \hat{x}^j$ and $Z^j = \hat{Z}^j$ for $j < t$). However, the dictionary $D$ is adapted to all spatiotemporal patches observed up to time $t$.[1] An exponential forgetting factor $\rho^{t-j}$ with $0 < \rho < 1$ is used for the terms in (P1), and $K_t = \sum_{j=1}^{t} \rho^{t-j}$ is a normalization constant for the objective. The forgetting factor $\rho$ diminishes the influence of "old" data on the dictionary adaptation process. When the dynamic object or scene changes slowly over time, a large $\rho$ (close to 1) is preferable so that past information can be used effectively. As written in (P1), the dictionary $D$ is updated based on patches from all previous times; however, we do not store this information. Indeed, our proposed algorithm in Section 7.3.2 computes only a few constant-sized matrices that contain the necessary cumulative (over time) information to solve (P1).

---

[1] We emphasize the global dependence of $D$ on all previous data by using the optimization variable $D$ rather than a time-indexed variable $D^t$ as for the variables $x^t$ and $Z^t$.

When minibatches $x^t$ and $x^{t+1}$ do not overlap (i.e., no common frames), each frame $g^t$ is reconstructed exactly once in its corresponding window in (P1). However, it is often beneficial to construct the $x^t$'s using an overlapping sliding window strategy [135], in which case a frame $g^t$ may be reconstructed in multiple windows (minibatches of frames). In this case, we independently produce estimates $\hat{x}^t$ for each time index as indicated in (P1), and then we produce a final estimate of the underlying frame $g^t$ by computing a weighted average of the reconstructions of that frame from each window in which it appeared. We empirically found that an exponentially $\rho$-weighted average performed better than alternatives such as an unweighted average of the estimates from each windows or using the most recent reconstruction from the latest window.

We propose a simple alternating minimization scheme for solving (P1). At each time index $t$, we alternate a few times between updating $(D, Z^t)$ while holding $x^t$ fixed (the *dictionary learning step*) and then updating $x^t$ with $(D, Z^t)$ held fixed (the *image update step*). For each $t$, we initialize the dictionary $D$ with the most recent dictionary ($\hat{D}^{t-1}$). Frames of $x^t$ that were estimated in the previous (temporal) windows are initialized with the most recent $\rho$-weighted reconstructions, and new frames are initialized using simple approaches (e.g., interpolation in the case of inpainting). Initializing the sparse coefficients $Z^t$ with the codes estimated in the preceding window ($\hat{Z}^{t-1}$) worked well. All updates are performed efficiently and with modest memory usage. Figure 7.1 provides a graphical flowchart depicting our proposed alternating minimization scheme at a given time index. We derive the solutions to the subproblems in the following sections.

### 7.3.2 Dictionary Learning Step

Let $C^t := (Z^t)^H$. Minimizing (P1) with respect to $(D, C^t)$ yields the optimization problem

$$\begin{aligned} \min_{D,C^t} \quad & \sum_{j=1}^{t} \rho^{t-j} \|P^j - D(C^j)^H\|_F^2 + \lambda_Z^2 \|C^t\|_0 \\ \text{s.t.} \quad & \|c_i^t\|_\infty \leq L,\ \mathbf{rank}(\mathcal{R}(d_i)) \leq r,\ \|d_i\|_2 = 1\ \forall i, \end{aligned} \tag{7.3}$$

where $P^j \in \mathbb{C}^{n \times M}$ is the matrix whose columns contain the patches $P_l x^j$ for $1 \leq l \leq M$, and $c_i^t$ is the $i$-th column of $C^t$. We use a block coordinate descent approach to update the sparse coefficients $c_i^t$ and atoms $d_i$ (columns of $D$) sequentially. For each $1 \leq i \leq m$, we first minimize (7.3) with respect to $c_i^t$ keeping the other variables fixed (the *sparse coding step*), and then we update $d_i$ keeping the other variables fixed (the *dictionary atom update step*).

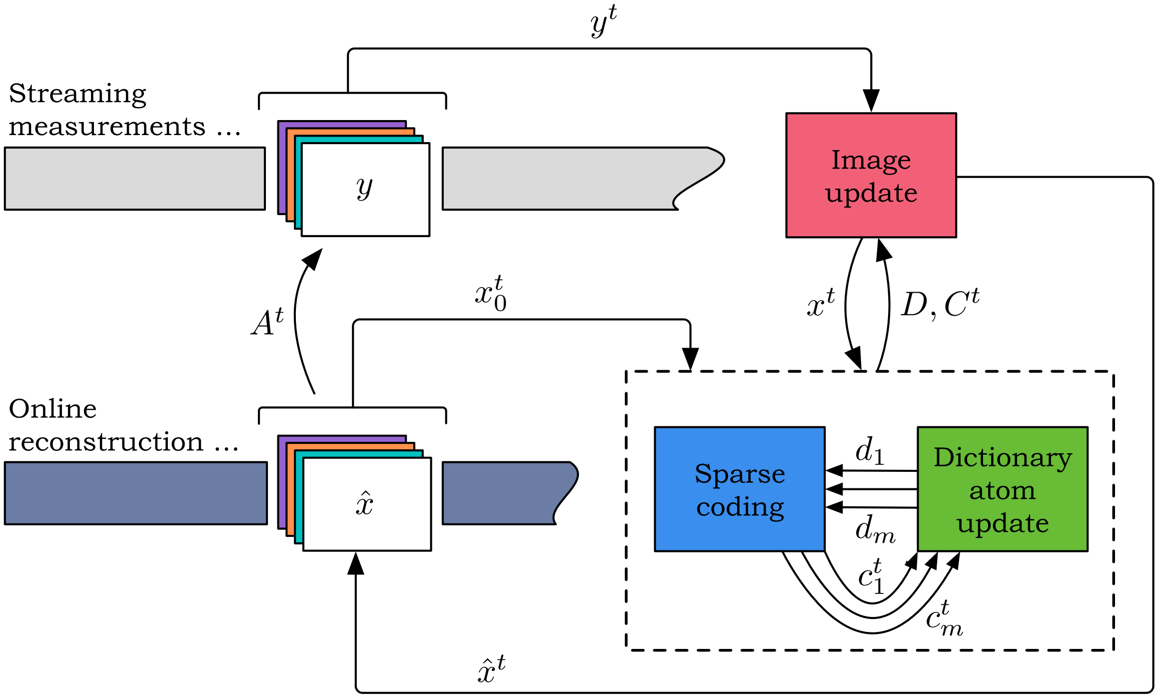


Figure 7.1: Flowchart of the proposed online adaptive dictionary learning alternating update scheme at time $t$. The input is a vector $y^t$ containing the streaming measurements for the current minibatch, and $x^t$ denotes the corresponding reconstructed frames. In the *dictionary learning step*, $(D, C^t)$ are updated with $x^t$ held fixed by performing block coordinate descent over the columns of $C^t$ (the *dictioanry atom update*) and the columns of $D$ (the *sparse coding step*). Then, the frames $x^t$ are updated (the *image update step*) with $(D, C^t)$ held fixed. This process is repeated a few times, and the final frame estimates $\hat{x}^t$ are integrated into the streaming reconstruction, $\hat{x}$.

#### 7.3.2.1 Sparse Coding Step

Minimizing (7.3) with respect to $c_i^t$ leads to the problem:

$$\begin{aligned} \min_{c_i^t \in \mathbb{C}^M} \quad & \|E_i^t - d_i (c_i^t)^H\|_F^2 + \lambda_Z^2 \|c_i^t\|_0 \\ \text{s.t.} \quad & \|c_i^t\|_\infty \le L, \end{aligned} \tag{7.4}$$

where the matrix

$$E_i^t := P^t - \sum_{k \neq i} d_k (c_k^t)^H \tag{7.5}$$

is defined based on the most recent estimates of the other atoms and sparse coefficients. The solution to (7.4), assuming $L > \lambda_Z$, is given by [119]

$$\hat{c}_i^t = \min\left(|H_{\lambda_Z}((E_i^t)^H d_i)|,\ L 1_M\right) \odot e^{j\angle (E_i^t)^H d_i}, \tag{7.6}$$

where $H_{\lambda_Z}(\cdot)$ is the elementwise hard thresholding operator that sets entries with (complex) magnitude less than $\lambda_Z$ to zero and leaves other entries unaffected, $1_M$ is a vector of ones of length $M$, $\odot$ denotes elementwise multiplication, $\min(\cdot,\cdot)$ denotes elementwise minimum, and $e^{j\angle\cdot}$ is computed elementwise, with $\angle$ denoting the phase. We do not construct $E_i^t$ explicitly; rather we efficiently compute the matrix-vector product $(E_i^t)^H d_i = (P^t)^H d_i - C^t D^H d_i + c_i^t$ based on the most recent estimates of each quantity using sparse matrix-vector operations [119].

#### 7.3.2.2 Dictionary Atom Update Step

Here, we minimize (7.3) with respect to $d_i$. This update makes use of past information via the forgetting factor $\rho$. Let $\tilde{P}^j := \sqrt{\rho^{t-j}}P^j$ and $\tilde{C}^j := \sqrt{\rho^{t-j}}C^j$ denote the $\rho$-weighted patches and sparse coefficients, respectively, and let $\tilde{P}^{1:t}$ and $\tilde{C}^{1:t}$ denote the matrices formed by stacking the $\tilde{P}^j$'s horizontally and $\tilde{C}^j$'s vertically, respectively, for times 1 to $t$. Finally, define $\tilde{E}_i^{1:t} := \tilde{P}^{1:t} - \sum_{k\neq i} d_k(\tilde{c}_k^{1:t})^H$ using the most recent estimates of all variables, with $\tilde{c}_k^{1:t}$ denoting the $k$-th column of $\tilde{C}^{1:t}$. Using this notation, we can write the minimization of (7.3) with respect to $d_i$ as

$$\begin{aligned} \min_{d_i\in\mathbb{C}^n} \quad & \|\tilde{E}_i^{1:t} - d_i(\tilde{c}_i^{1:t})^H\|_F^2 \\ \text{s.t.} \quad & \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \ \|d_i\|_2 = 1. \end{aligned} \tag{7.7}$$

Let $U_r\Sigma_r V_r^H$ be the rank-$r$ truncated singular value decomposition (SVD) of the matrix $\mathcal{R}(\tilde{E}_i^{1:t}\tilde{c}_i^{1:t})$ that is obtained by computing the $r$ leading singular vectors and singular values of the full SVD $\mathcal{R}(\tilde{E}_i^{1:t}\tilde{c}_i^{1:t}) := U\Sigma V^H$. Then a solution to (7.7) is given by [12]

$$\mathcal{R}(\hat{d}_i) = \begin{cases} \dfrac{U_r\Sigma_r V_r^H}{\|\Sigma_r\|_F}, & \text{if } \tilde{c}_i^{1:t} \neq 0 \\ W, & \text{if } \tilde{c}_i^{1:t} = 0, \end{cases} \tag{7.8}$$

where $W$ is any matrix of appropriate dimension with rank at most $r$ such that $\|W\|_F = 1$.[2]

The main computation in (7.8) is computing $\tilde{E}_i^{1:t}\tilde{c}_i^{1:t}$, since the SVD of the small $n_y n_x \times n_t$ matrix $\mathcal{R}(\tilde{E}_i^{1:t}\tilde{c}_i^{1:t})$ has negligible computational cost. In principal, the matrix-vector multiplication $\tilde{E}_i^{1:t}\tilde{c}_i^{1:t}$ depends on all past information processed by the streaming algorithm; however, it can be recursively computed using constant time

---

[2] We set $W$ to be the reshaped first column of the $n \times n$ identity matrix.

and memory. Indeed, observe that

$$\begin{aligned}
\tilde{E}_i^{1:t}\tilde{c}_i^{1:t} &= \sum_{j=1}^{t} \tilde{E}_i^j \tilde{c}_i^j = \sum_{j=1}^{t} \rho^{t-j} E_i^j c_i^j \\
&= \sum_{j=1}^{t} \rho^{t-j} \left(P^j - D(C^j)^H + d_i (c_i^j)^H\right) c_i^j \\
&= \underbrace{\left[\sum_{j=1}^{t} \rho^{t-j} P^j c_i^j\right]}_{=: f_i^t} - D \underbrace{\left[\sum_{j=1}^{t} \rho^{t-j} (C^j)^H c_i^j\right]}_{=: g_i^t} + d_i \underbrace{\left[\sum_{j=1}^{t} \rho^{t-j} \|c_i^j\|^2\right]}_{=[g_i^t]_i},
\end{aligned} \tag{7.9}$$

where $[z]_i$ denotes the $i$-th element of vector $z$. The vectors $f_i^t$ and $g_i^t$ depend on all previous data, but they can be recursively computed as

$$\begin{aligned}
f_i^t &= \rho f_i^{t-1} + P^t c_i^t \\
g_i^t &= \rho g_i^{t-1} + (C^t)^H c_i^t,
\end{aligned} \tag{7.10}$$

where, for each column index $i$, the matrix $C^t$ is understood to contain the latest versions of the sparse codes already updated during the dictionary learning step. Using these recursive formulas, the product $\tilde{E}_i^{1:t}\tilde{c}_i^{1:t}$ can be readily computed in our algorithm. Thus, the update in (7.8) can be performed in a fully online manner.

In practice, we collect the vectors $f_i^t$ and $g_i^t$ as columns of matrices $F^t \in \mathbb{C}^{n\times m}$ and $G^t \in \mathbb{C}^{m\times m}$, and we perform the following recursive update at the end of each outer iteration of our algorithm:

$$\begin{aligned}
F^t &= \rho F^{t-1} + P^t C^t \\
G^t &= \rho G^{t-1} + (C^t)^H C^t.
\end{aligned} \tag{7.11}$$

Here, $C^t$ denotes the final sparse codes after updating each column for the current outer iteration. The matrices $F^t \in \mathbb{C}^{n\times m}$ and $G^t \in \mathbb{C}^{m\times m}$ are small, constant-sized matrices whose dimensions are independent of the time index $t$ and the dimensions of frame sequence, so they can be efficiently stored for use in the next minibatch. Moreover, the matrix $C^t$ is sparse, so the matrix-matrix multiplications in (7.11) can be efficiently computed using sparse matrix operations.

### 7.3.3 Image Update Step

Minimizing (P1) with respect to $x^t$ yields the sub-problem:

$$\min_{x^t} \; \|A^t x^t - y^t\|_2^2 + \lambda_S \sum_{l=1}^{M} \|P_l x^t - D z_l^t\|_2^2. \tag{7.12}$$

Problem (7.12) is a least squares problem with normal equation

$$\left( (A^t)^H A^t + \lambda_S \sum_{l=1}^{M} P_l^H P_l \right) x^t = (A^t)^H y^t + \lambda_S \sum_{l=1}^{M} P_l^H D z_l^t. \tag{7.13}$$

In applications such as video denoising or inpainting, the matrix pre-multiplying $x^t$ in (7.13) is a fixed diagonal matrix that can be efficiently inverted (and pre-computed). More generally, in inverse problems where the matrix pre-multiplying $x^t$ in (7.13) is not diagonal or readily diagonalizable (e.g., in dynamic MRI with multiple coils), we instead solve (7.12) by applying a few iterations (indexed by $k$) of the proximal gradient method [25, 145], which prescribes updates of the form

$$x^{t,k+1} = \mathbf{prox}_{\tau_k h} \left( x^{t,k} - \tau_k (A^t)^H (A^t x^{t,k} - y^t) \right), \tag{7.14}$$

where $h(x) := \lambda_S \sum_{l=1}^{M} \|P_l x - D z_l^t\|_2^2$ and the proximal operator of a function $f$ is defined as

$$\mathbf{prox}_f(x) := \arg\min_z \; \tfrac{1}{2}\|x - z\|_2^2 + f(z). \tag{7.15}$$

The proximal operator in (7.14) is a simple least squares problem with diagonal normal equation, so it can be solved exactly and efficiently by inverting a fixed diagonal matrix, which can be pre-computed. A constant step-size $\tau_k = \tau < 2/\|A^t\|_2^2$ suffices for the proximal gradient method to converge [145]. In fact, the iterations (7.14) will monotonically decrease the objective (7.12) when a constant step size $\tau \leq 1/\|A^t\|_2^2$ is used [25].

### 7.3.4 Unitary Dictionary Variation

In (P1) we imposed a low-rank constraint on the dictionary atoms. In this section, we instead constrain the dictionary to be a unitary matrix. The resulting problem thus becomes

$$
\begin{aligned}
\text{(P2)}\quad \left\{\hat{x}^t, \hat{D}^t, \hat{Z}^t\right\} = \underset{x^t, D, Z^t}{\arg\min}\ & \frac{1}{K_t} \sum_{j=1}^{t} \rho^{t-j} \|y^j - A^j x^j\|_2^2 \\
& + \frac{\lambda_S}{K_t} \sum_{j=1}^{t} \rho^{t-j} \left( \sum_{l=1}^{M} \|P_l x^j - D z_l^j\|_2^2 + \lambda_Z^2 \|Z^j\|_0 \right) \\
& \text{s.t. } D^H D = I,
\end{aligned}
$$

where all terms are defined as in (P1). Note that we do not require the $\ell_\infty$-norm constraints on the sparse coefficients $Z^t$ in this formulation because the unitary constraint on the dictionary precludes the possibility of repeated dictionary atoms, which was the original motivation for including these constraints in (P1). As in Section 7.3.1, we propose to solve (P2) by an alternating minimization scheme where we sequentially minimize (P2) with respect to $Z^t$, $D$, and $x^t$ with all other variables held fixed. The image update step is identical to Section 7.3.3. However, unlike (P1), we do not perform block coordinate descent over the columns of $D$ and $(Z^t)^H$. Rather we derive simple closed-form matrix-valued updates. The following subsections explicitly describe the solutions to the $Z^t$ and $D$ subproblems.

#### 7.3.4.1 Sparse Coding Step

Minimizing (P2) with respect to $Z^t$ yields the subproblem

$$
\min_{Z^t} \ \|D^H P^t - Z^t\|_F^2 + \lambda_Z^2 \|Z\|_0, \tag{7.16}
$$

where $P^t$ is the matrix whose $l$-th column contains the image patch $P_l x^t$ and we have used the fact that $D$ is a unitary matrix and the unitary invariance of the Frobenius norm to isolate $Z^t$. The solution to (7.16) is the simple elementwise hard thresholding operation

$$
\hat{Z}^t = H_{\lambda_Z}(D^H P^t). \tag{7.17}
$$

#### 7.3.4.2 Dictionary Update Step

Minimizing (P2) with respect to $D$ yields the subproblem

$$\begin{aligned} \min_{D} \quad & \sum_{j=1}^{t} \rho^{t-j} \| P^j - D (C^j)^H \|_F^2 \\ \text{s.t.} \quad & D^H D = I, \end{aligned} \tag{7.18}$$

where we have reintroduced the matrix $C^j := (Z^j)^H$ for notational convenience. Using the definitions of the matrices $\tilde{P}^{1:t}$ and $\tilde{C}^{1:t}$ from the dictionary atom updates in Section 7.3.2, we can equivalently write (7.18) as

$$\begin{aligned} \min_{D} \quad & \| \tilde{P}^{1:t} - D (\tilde{C}^{1:t})^H \|_F^2 \\ \text{s.t.} \quad & D^H D = I. \end{aligned} \tag{7.19}$$

Problem (7.19) is a well-known Procrustes analysis problem [146]. The solution is given by $\hat{D} = UV^T$, where $U\Sigma V^T$ is the SVD of $\tilde{P}^{1:t}\tilde{C}^{1:t} = F^t$ and $F^t$ is defined as in (7.11). The matrix $F^t \in \mathbb{C}^{n \times m}$ can be recursively updated according to (7.11), so the dictionary update step can be performed efficiently and fully online.

### 7.3.5 Computational Cost and Convergence

The computational cost for each time index $t$ of the proposed algorithm for solving the online image reconstruction problem (P1) scales as $O(n^2 M)$, where $D \in \mathbb{C}^{n \times m}$, we assume $m \propto n$, and $M$ is the number of (possibly overlapping) patches in each temporal window. The cost is dominated by various matrix-vector multiplications. Assuming the window length $\tilde{M} \ll n$, the memory (storage) requirement for the proposed algorithm scales as $O(nM)$, which is the space required to store the image patches of $x^t$ when performing the updates for (P1). Since the minibatch size $\tilde{M}$ is typically small, the number of 3D patches in each window is also small, which ensures modest memory usage for our proposed online method.

The computational cost and memory requirements for each time index $t$ of the proposed algorithm for solving the unitary variation in (P2) are identical to (P1). However, the simple matrix-valued forms of the alternating updates result in a several-fold decrease in runtimes in practice due to optimizations inherent to matrix-valued computations in modern linear algebra libraries.

Each variable update in the proposed algorithms are either exact block coordinate

| % Missing Pixels | Coastguard | | | | | Bus | | | | | Flower Garden | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 50% | 60% | 70% | 80% | 90% | 50% | 60% | 70% | 80% | 90% | 50% | 60% | 70% | 80% | 90% |
| Online DINO-KAT | 33.1 | **31.4** | **29.6** | **27.3** | 22.5 | 28.7 | 27.1 | **25.5** | **23.7** | **21.5** | **24.4** | **22.8** | **21.0** | **18.8** | 15.8 |
| Online (unitary) | **33.8** | 31.3 | 28.1 | 24.8 | 21.9 | **29.7** | **27.6** | **25.5** | 23.4 | 21.1 | **24.4** | 22.1 | 19.6 | 17.1 | 15.6 |
| Online (DCT) | 32.7 | 30.3 | 27.8 | 25.3 | 22.6 | 28.4 | 26.7 | 25.0 | 23.1 | 20.8 | 23.3 | 21.6 | 19.9 | 18.1 | 16.3 |
| Batch DINO-KAT | 33.1 | 31.2 | 29.1 | 26.3 | 22.8 | 27.8 | 26.3 | 24.7 | 22.9 | 20.9 | 23.5 | 21.8 | 20.1 | 18.2 | 16.1 |
| Interpolation (3D) | 29.8 | 28.5 | 27.3 | 25.9 | **24.1** | 27.3 | 25.7 | 24.0 | 22.1 | 20.0 | 20.6 | 19.6 | 18.5 | 17.5 | **16.4** |
| Interpolation (2D) | 28.2 | 26.5 | 24.9 | 23.1 | 21.1 | 26.0 | 24.8 | 23.7 | 22.5 | 21.1 | 20.1 | 18.8 | 17.5 | 16.2 | 14.8 |

Table 7.1: PSNR values in decibels (dB) for video inpainting on three videos from the BM4D dataset at various percentages of missing pixels. The methods considered are the proposed online DINO-KAT learning method with $r = 5$, the proposed online method with unitary dictionary, online inpainting with a fixed DCT dictionary, the batch DINO-KAT learning method with $r = 5$, 2D (frame-by-frame cubic) interpolation, and 3D interpolation The best PSNR for each undersampling on each video is in bold.

descent updates or, in the image update step with non-diagonalizable $A^t$, proximal gradient iterations that can be guaranteed to monotonically decrease the objectives in (P1) or (P2), respectively, with an appropriate choice of step size. Thus the overall proposed algorithms are guaranteed to monotonically decrease their objectives. The objectives are bounded below by zero, so the objectives must converge. Whether or not the iterate sequences produced by each algorithm also converge is an interesting open problem that we leave for future work.

## 7.4 Numerical Experiments

In this section we perform extensive numerical experiments to illustrate the usefulness of the proposed online adaptive dictionary learning methods. We consider two inverse problem applications in this section: video reconstruction (inpainting) from noisy and subsampled pixels, and dynamic MRI reconstruction from highly undersampled data.

### 7.4.1 Video Inpainting

#### 7.4.1.1 Framework

First, we consider video inpainting from noisy and subsampled pixels. We work with the publicly available videos[3] provided by the authors of the BM4D method [147]. We process the first 150 frames of each video at native resolution. We measure a uniformly random subset of the pixels in each frame of the video, and in some experiments

---

[3] The data is available at `http://www.cs.tut.fi/~foi/GCF-BM3D/`.

| % Missing Pixels | Coastguard (25 dB PSNR) | | | | |
|---|---|---|---|---|---|
| | 50% | 60% | 70% | 80% | 90% |
| Online DINO-KAT | 28.6 | 27.9 | **27.2** | **26.1** | **23.9** |
| Online (unitary) | **29.4** | **28.6** | 26.6 | 24.9 | 22.0 |
| Online (DCT) | 28.6 | 27.8 | 26.7 | 25.1 | 22.9 |
| Batch DINO-KAT | 28.6 | 27.8 | 26.6 | 25.1 | 22.5 |
| Interpolation (3D) | 26.2 | 25.9 | 25.4 | 24.6 | 23.4 |
| Interpolation (2D) | 24.7 | 24.0 | 23.0 | 21.9 | 20.2 |

Table 7.2: PSNR values in decibels (dB) for video inpainting on the Coastguard video corrupted by Gaussian noise with 25dB PSNR at various percentages of missing pixels. The methods considered are the proposed online DINO-KAT learning method with $r = 5$, the proposed online method with unitary dictionary, the batch DINO-KAT learning method with $r = 5$, online inpainting with a fixed DCT dictionary, 2D (frame-by-frame cubic) interpolation, and 3D interpolation The best PSNR for each undersampling is in bold.

we also add Gaussian noise to the measured pixels. The proposed online method is then used to reconstruct the video from the corrupted (noisy and/or subsampled) measurements.

For the proposed method, we used sliding (temporal) windows of length $\tilde{M} = 5$ to construct minibatches of frames with a temporal stride of 1 frame. In each window, we extracted $8 \times 8 \times 5$ overlapping spatiotemporal patches with a spatial stride of 2 pixels. We learned a square $320 \times 320$ dictionary, and the operator $\mathcal{R}(\cdot)$ reshaped dictionary atoms into $64 \times 5$ space-time matrices. We ran the algorithm for (P1) for 7 iterations in each temporal window, with 1 inner iteration of block coordinate descent for updating $(D, Z^t)$, and we chose forgetting factor $\rho = 0.9$. We ran the algorithm for more (50) iterations for the first temporal window to warm start the algorithm. We initialized $D$ to the discrete cosine transform (DCT) matrix, set the initial sparse codes to zero, and initialized the frames with 2D (per-frame) cubic interpolation. We simulated various levels of subsampling of the video (with and without noise), and we chose $r = 5$ (full-rank) atoms. The weights $\lambda_S$ and $\lambda_Z$ were tuned at an intermediate undersampling factor (70% missing pixels) for each video.

Note that we still refer to the proposed method here as the online DINO-KAT method even when full-rank atoms are used, because the variable patch sparsity and other properties of the DINO-KAT learning model (7.2) are still important characteristics in (P1). The videos considered in this section have substantial temporal motion, so allowing full-rank atoms enabled the algorithm to learn useful dynamic temporal features of the data. We demonstrate the performance of low-rank atoms in Section 7.4.2.

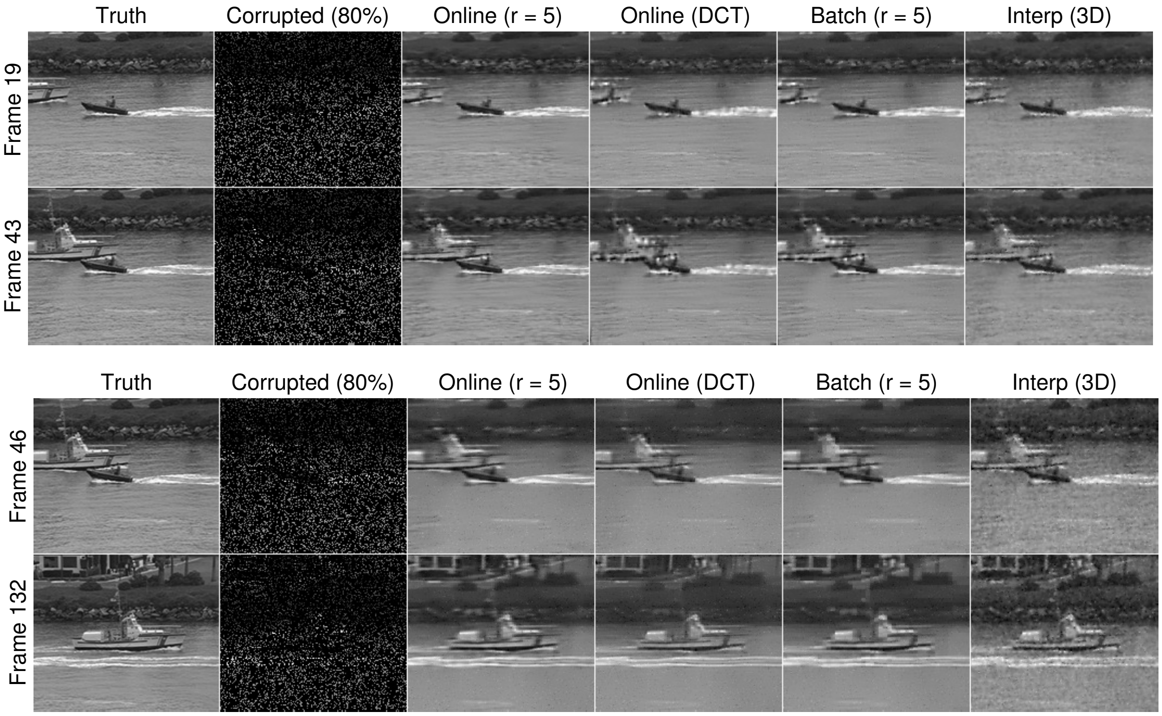


Figure 7.2: Two representative frames from the reconstructions produced by each method on the Coastguard video with 80% missing pixels (top) and 80% pixels with 25 dB Gaussian noise added (bottom). The methods considered are the proposed online DINO-KAT learning method ($r = 5$), the online method with fixed DCT dictionary, the batch DINO-KAT learning method, and 3D interpolation. Top: the proposed method method achieves PSNR improvements of 2.0 dB, 1.0 dB, and 1.4 dB, respectively, compared to the other methods. Bottom: the proposed method achieves PSNR improvements of 1.0 dB, 1.0 dB, and 1.5 dB, respectively, compared to the other methods.

We compare the performance of the proposed online DINO-KAT method, the proposed method with a (learned) unitary dictionary, and the proposed online method with a fixed DCT dictionary. We also produce reconstructions using the batch DINO-KAT method with $r = 5$ [12] that processes all frames jointly.[4] For each method, we used the same patch dimensions, initializations, etc. (if applicable for the batch scheme), and we tuned the parameters of each method individually. The one exception is that we used a spatial stride of 4 pixels for the batch method rather than the 2 pixel strides used for the online methods. The batch method is memory intensive as it requires extracting image patches from the entire video concurrently, so it was

---

[4] The batch DINO-KAT method is equivalent to the proposed online method with $\tilde{M}$ set to the number of frames in the video.

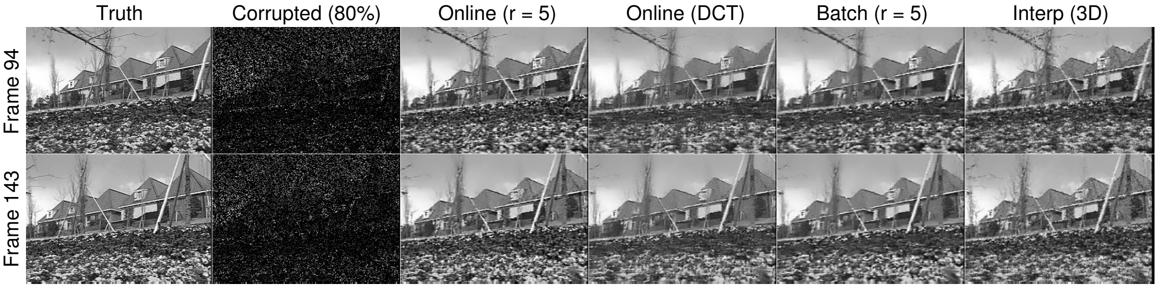


Figure 7.3: Two representative frames from the reconstructions produced by each method on the Flower Garden video with 80% missing pixels. The methods considered are the proposed online DINO-KAT learning method ($r = 5$), the online method with fixed DCT dictionary, the batch DINO-KAT learning method, and 3D interpolation. The proposed method achieves PSNR improvements of 0.7 dB, 0.6 dB, and 1.3 dB, respectively, compared to the other methods.

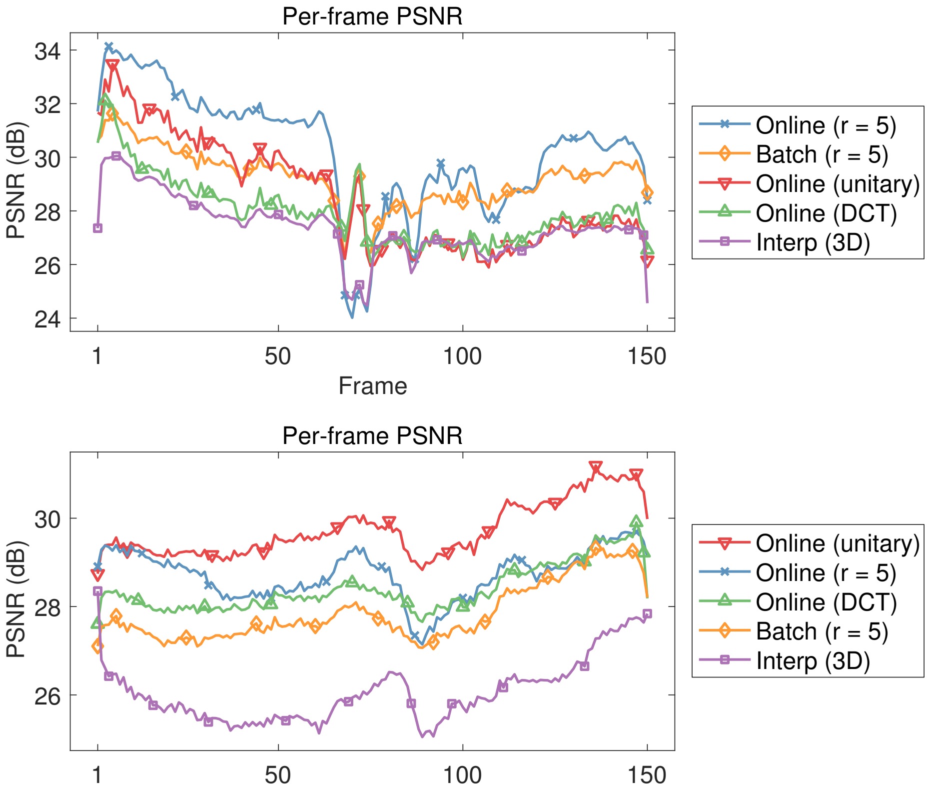


Figure 7.4: Per-frame PSNR for the reconstructions produced by each method on the Coastguard video with 70% missing pixels (left) and the Bus video with 50% missing pixels (right). The methods considered are the proposed online DINO-KAT learning method ($r = 5$), the proposed online method with unitary dictionary, the online method with fixed DCT dictionary, the batch DINO-KAT learning method, and 3D interpolation.

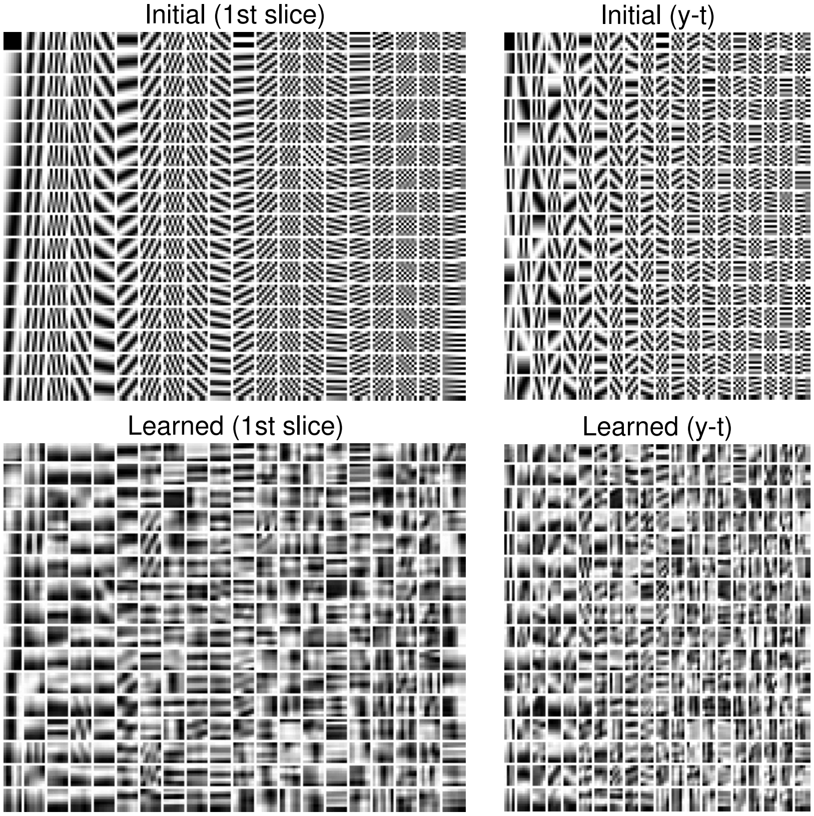


Figure 7.5: Dictionaries for the Bus video with 50% missing pixels. Top: the initial DCT dictoinary; bottom: the learned dictionary produced by the proposed online DINO-KAT learning method with $r = 5$. Left: the first $8 \times 8$ slice of each atom; right: the $y - t$ profiles of a vertical cross-section through each $8 \times 8 \times 5$ tensor.

necessary to process fewer patches to make the computation feasible. We ran the batch method for 20 iterations. Finally, we compute baseline reconstructions produced by 2D interpolation (frame-by-frame cubic) and 3D interpolation (using the natural neighbor method in MATLAB).

### 7.4.1.2 Reconstructions

Table 7.1 lists the peak signal to noise ratio (PSNR) values in decibels (dB) for the various reconstruction methods at different levels of subsampling (from 50% to 90% missing pixels) on three videos from the BM4D dataset. Table 7.2 shows the analogoous results on the Coastguard video when Gaussian noise with 25 dB PSNR was added before sampling. The proposed online DINO-KAT method typically provides the best PSNRs at higher undersampling rates, and the proposed unitary method typically performs better at lower undersampling rates. Both variations consistently outperform the online method with fixed DCT dictionary, suggesting that the dictionaries learned by the proposed methods are successfully adapting to underlying features of the videos.

Figures 7.2 and 7.3 show the original and reconstructed frames for a few representative frames for each method on the Coastguard and Flower Garden videos. Figure 7.2 shows that the proposed online DINO-KAT method produces visually more accurate reconstructions of the texture in the waves and produces fewer artifacts near the boats in the water and the rocks on the shore. From Figure 7.3 we see that the proposed method produces a sharper reconstruction with less smoothing artifacts than the online method with fixed DCT dictionary and the batch DINO-KAT method, and it is less noisy than the interpolation-based reconstruction.

#### 7.4.1.3 Properties

Figure 7.4 shows the frame-by-frame PSNRs for the inpainted Coastguard and Bus videos using the various methods. Clearly the proposed online method achieves consistently higher PSNRs across frames on the Coastguard video, and the proposed online method with unitary dictionary achieves consistently higher PSNRs on the Bus video. The overall trends in PSNR are similar across each method and are due to motion in the original videos, with more motion generally resulting in lower PSNRs.

Finally, Figure 7.5 shows a representative example of a learned dictionary produced by the proposed online DINO-KAT method on the Bus video compared to the initial DCT dictionary. The learned dictionary contains 320 atoms, each of which are $8 \times 8 \times 5$ space-time tensors. We visualize each atom by plotting the first $8 \times 8$ slice of each atom and the plotting the $y - t$ profiles of the atoms for a vertical slice through the middle of each atom. The first slice images show that the learned dictionary has adapted to both smooth and sharp gradients in the image, and the dynamic nature of the $y - t$ profiles shows that the dictionary atoms have adapted to temporal trends in the data.

### 7.4.2 Dynamic MRI

#### 7.4.2.1 Framework

We next demonstrate the usefulness of the proposed online DINO-KAT method for reconstructing dynamic MRI data from highly undersampled measurements. We work with the multi-coil (12-element coil array) cardiac perfusion data [1] and the PINCAT data [133, 148] from prior works. For the cardiac perfusion data, we retrospectively undersampled the k-t space using variable-density random Cartesian undersampling with a different undersampling pattern for each time frame, and for the PINCAT data we used pseudo-radial sampling with a random rotation of radial lines between frames. For each dataset, we obtained reconstructions using the proposed online DINO-KAT method, the online method with fixed DCT dictionary, and the batch DINO-KAT method [12]. We also ran the recent L+S [1] and k-t SLR [2] methods, two sophisticated batch methods for dynamic MRI that process all frames jointly. Finally, we also computed a baseline reconstruction for each dataset by performing zeroth order interpolation across time at non-sampled k-t space locations and then backpropagating the filled k-t space to image space by pre-multiplying with the $A^H$ corresponding to fully sampled data.

For the online schemes, we used $8 \times 8 \times 5$ spatiotemporal patches with $\tilde{M} = 5$ frames per temporal window and a temporal stride of 1. We extracted overlapping patches using a spatial stride of 2 along each dimension. We chose forgetting factor $\rho = 0.9$, and we learned a square $320 \times 320$ dictionary with rank $r = 1$ atoms when reshaped into $64 \times 5$ space-time matrices. We ran the online scheme at each time index $t$ for 10 outer iterations, with 1 block coordinate descent pass in the dictionary update step and 10 proximal gradient steps in the image update step, respectively. We used 50 outer iterations for the first temporal window to warm start the algorithm. The dictionary and image frames were not updated during the first 3 outer iterations to allow the sparse coefficients to adapt to new patches. The dictionary was initialized to the DCT matrix, and the sparse codes were set to zero. New frames—those not appearing in any preceding windows—were initialized by filling the non-sampled k-t space locations with the synthesized k-space data from the latest reconstructed frame and then backpropagating the filled k-t space to image space. After one complete pass over the frames, we performed another pass over the frames, using the reconstructed frames and learned dictionary from the first pass to initialize the second pass.

For the batch DINO-KAT method, we used the same patch dimensions, strides, and initializations (where applicable) as for the online methods. We ran the batch method for 50 iterations. For the L+S methods, we used the publicly available MATLAB implementations from [4] and [5], respectively, and we ran each method to convergence. The regularization parameters for all methods were obtained by sweeping over a range of values and selecting values that achieved good reconstruction quality at intermediate undersampling factors. We measured reconstruction quality using the

| Acceleration | Cardiac Perfusion | | | | | | PINCAT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4x | 8x | 12x | 16x | 20x | 24x | 5x | 6x | 7x | 9x | 14x | 27x |
| Online DINO-KAT | **10.1%** | **12.8%** | **14.8%** | **16.7%** | **18.1%** | **18.0%** | **8.9%** | **9.7%** | **11.0%** | **12.4%** | **15.5%** | 21.8% |
| Online (DCT) | 10.8% | 13.7% | 15.8% | 18.2% | 20.7% | 20.8% | 9.5% | 10.2% | 11.5% | 13.2% | 16.4% | 22.5% |
| Batch DINO-KAT | 10.7% | 13.7% | 15.9% | 18.2% | 22.0% | 23.9% | 10.0% | 10.7% | 11.8% | 13.2% | 15.9% | **20.9%** |
| L+S | 11.0% | 13.8% | 16.1% | 18.4% | 21.5% | 22.5% | 11.8% | 12.9% | 14.4% | 16.6% | 20.0% | 25.9% |
| k-t SLR | 11.2% | 15.7% | 18.4% | 21.3% | 24.3% | 26.5% | 9.8% | 10.9% | 12.4% | 14.7% | 18.2% | 24.2% |
| Baseline | 12.8% | 15.9% | 18.9% | 21.1% | 24.5% | 28.1% | 22.3% | 24.7% | 27.5% | 31.3% | 36.6% | 44.5% |

Table 7.3: Left: NRMSE values as percentages for the cardiac perfusion data at several undersampling factors with Cartesian sampling. Right: NRMSE values as percentages for the PINCAT data at several undersampling factors with pseudo-radial sampling. The methods considered are the proposed online DINO-KAT learning method with $r = 1$, the online scheme with fixed DCT dictionary, the batch DINO-KAT learning method with $r = 1$, the L+S method, the k-t SLR method, and a baseline reconstruction. The best NRMSE for each undersampling on each dataset is in bold.

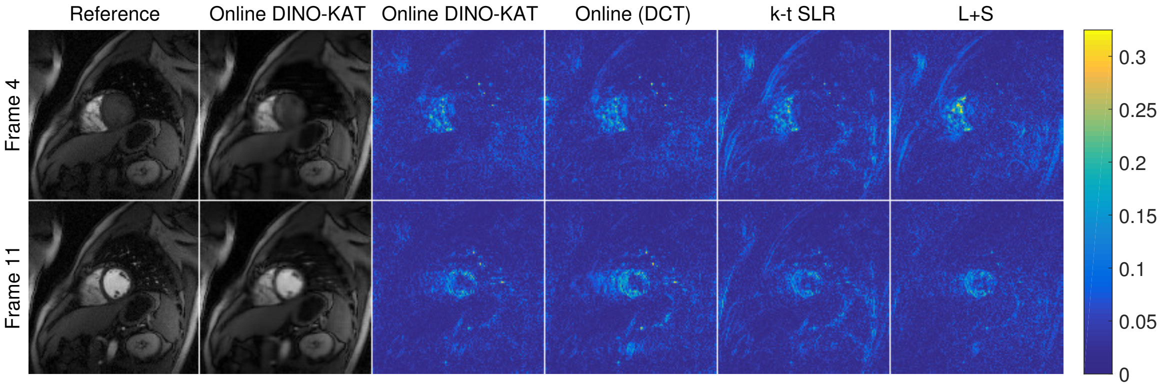


Figure 7.6: Two representative frames from a reference (fully sampled) reconstruction along with the corresponding frames from the proposed online DINO-KAT learning-based reconstruction on the cardiac perfusion data with 12x undersampling (Cartesian sampling). The right four columns depict the corresponding reconstruction error maps (w.r.t. reference) for the proposed online DINO-KAT learning method, the online method with fixed DCT dictionary, the k-t SLR method, and the L+S method, respectively. The proposed online method achieves NRMSE improvements of 0.6 dB, 1.9 dB, and 0.7 dB, respectively, compared to the other methods.

normalized root mean square error (NRMSE) metric that is computed as

$$\mathbf{NRMSE}(\hat{x}) = \frac{\|\hat{x} - x_{\text{ref}}\|_2}{\|x_{\text{ref}}\|_2} \times 100\%, \tag{7.20}$$

where $\hat{x}$ is a candidate reconstruction and $x_{\text{ref}}$ is a reference reconstruction computed from "fully" sampled data.

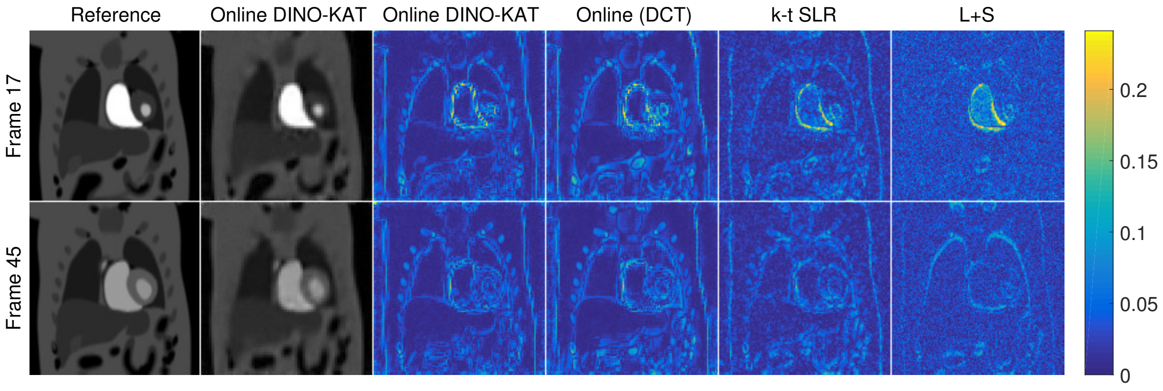


Figure 7.7: Two representative frames from a reference (fully sampled) reconstruction along with the corresponding frames from the proposed online DINO-KAT learning-based reconstruction on the PINCAT data with 7x undersampling (pseudo-radial sampling). The right four columns depict the corresponding reconstruction error maps (w.r.t. reference) for the proposed online DINO-KAT learning method, the online method with fixed DCT dictionary, the k-t SLR method, and the L+S method, respectively. The proposed online method achieves NRMSE improvements of 0.4 dB, 1.0 dB, and 2.3 dB, respectively, compared to the other methods.

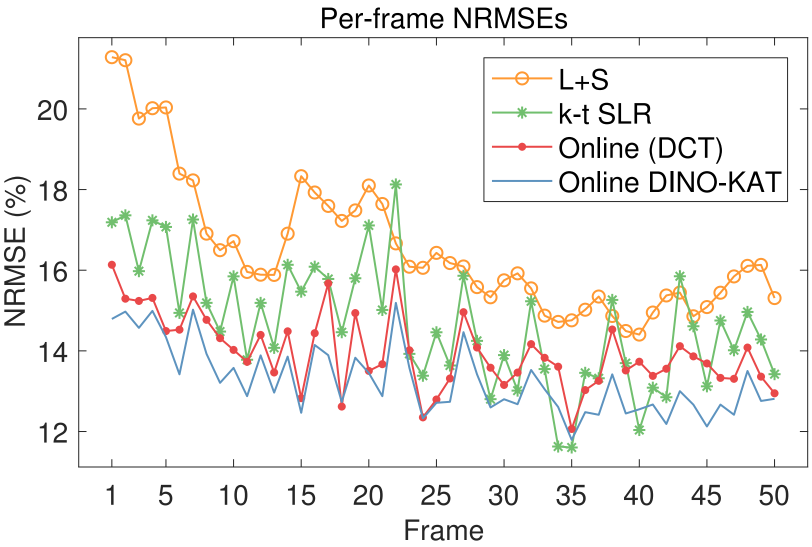


Figure 7.8: Per-frame PSNR for the reconstructions produced by each method on the PINCAT data with 9x undersampling (pseudo-radial sampling). The methods shown are the proposed online DINO-KAT learning method, the online method with fixed DCT dictionary, the k-t SLR method, and the L+S method.

### 7.4.2.2 Reconstructions

Table 7.3 shows the reconstruction NRMSE values obtained using each method as a function of undersampling factor on the cardiac perfusion and PINCAT datasets.

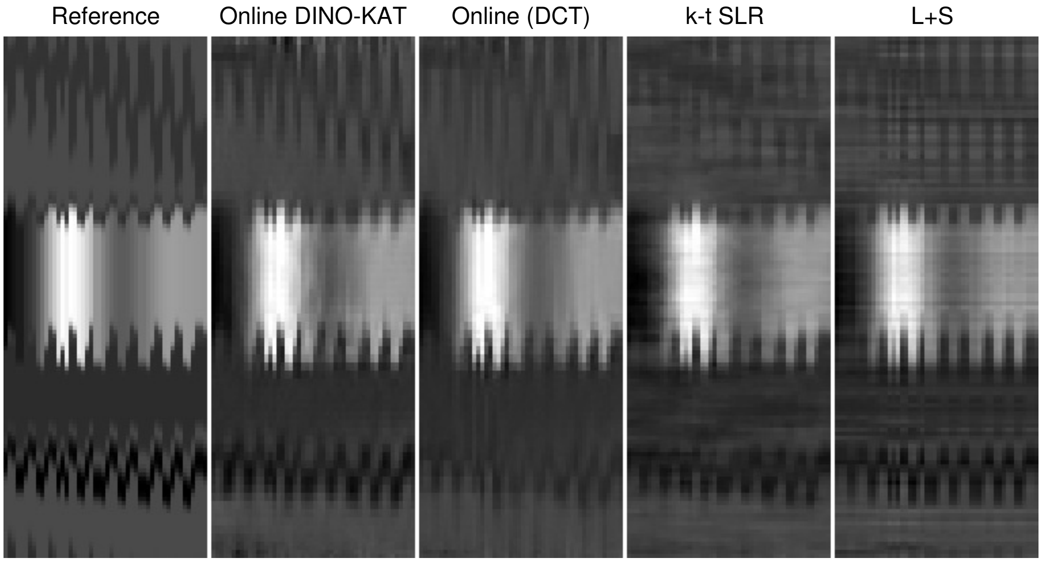


Figure 7.9: Temporal ($y - t$) profiles of a spatial vertical line cross section for the reference PINCAT reconstruction, the proposed online DINO-KAT learning method, the online method with fixed DCT dictionary, the k-t SLR method, and the L+S method for 14x undersampling (pseudo-radial sampling).

| Acceleration | 4x | 8x | 12x | 16x | 20x | 24x |
|---|---|---|---|---|---|---|
| Online (oracle) | **10.2%** | **12.4%** | **14.4%** | **16.4%** | **17.8%** | **17.9%** |
| Online DINO-KAT (2 passes) | **10.2%** | 12.8% | 14.8% | 16.7% | 18.1% | 18.0% |
| Online DINO-KAT (1 pass) | **10.2%** | 12.9% | 14.8% | 16.6% | 18.3% | 18.1% |
| Online (unitary) | 10.5% | 13.6% | 15.7% | 17.8% | 20.4% | 20.1% |
| Online (DCT) | 10.8% | 13.7% | 15.8% | 18.2% | 20.7% | 20.8% |

Table 7.4: NRMSE values as percentages for the cardiac perfusion data at seversal undersampling factors with Cartesian sampling. The methods considered are the online scheme with a fixed dictionary learned from patches of a reference reconstruction, the proposed online DINO-KAT learning method with two passes over the frames, the proposed online DINO-KAT learning method with a single pass over the frames, the proposed online method with unitary dictionary, and the online scheme with fixed DCT dictionary. The best NRMSE for each undersampling is in bold.

The proposed online DINO-KAT method achieves lower NRMSE values in almost every case, despite the fact that the online scheme only processes and stores data corresponding to 5 frames (in $x^t$) at any time while the L+S, k-t SLR, and batch DINO-KAT methods process all data jointly. These results show that the proposed online method is well-suited for processing streaming data.

Figures 7.6 and 7.7 show reconstructions and reconstruction error maps (magnitudes displayed) for some representative frames from the cardiac perfusion and PINCAT datasets, respectively. The error maps indicate that the proposed online

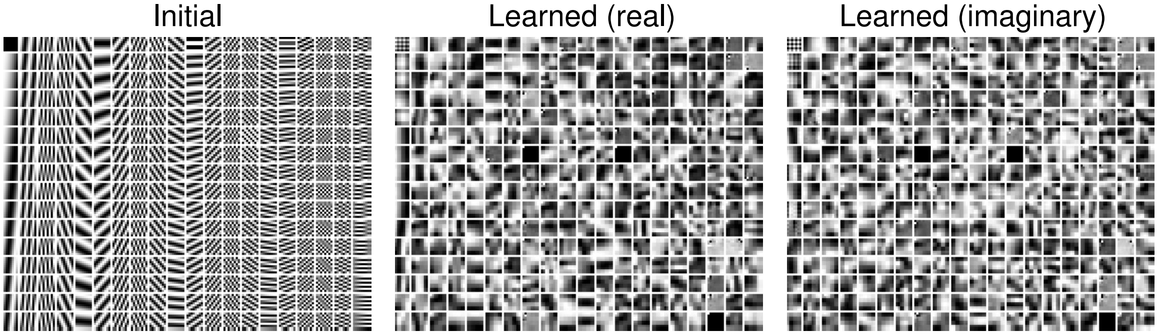


Figure 7.10: Dictionaries for the PINCAT data with 9x undersampling. Left: the atoms of the initial DCT dictionary. Right: the real and imaginary parts of the learned dictionary produced by the proposed online DINO-KAT learning method with $r = 1$. The dictionary atoms are $8 \times 8 \times 5$ tensors, so only the first $8 \times 8$ slice of each atom is displayed.

method produces fewer artifacts compared to the existing methods.

#### 7.4.2.3 Properties

Figure 7.8 shows that the online DINO-KAT scheme typically provides better frame-by-frame NRMSE compared to the other methods. Finally, Figure 7.9 shows $y - t$ profiles for each method obtained by extracting the same vertical line segment from each reconstructed frame for the PINCAT data. The online DCT-based and batch methods show line-like or additional smoothing artifacts that are not produced by the proposed online DINO-KAT method.

Table 7.4 investigates the properties of the proposed online method in more detail on the cardiac perfusion data. Specifically, it compares the NRMSE values produced by the proposed online DINO-KAT method with one and two passes over the data, the proposed online method with (learned) unitary dictionary, and the online method with a fixed DCT dictionary. In addition, we ran the online method with an "oracle" dictionary learned from patches of the reference reconstructions by solving the DINO-KAT learning problem (7.2). The oracle dictionary was computed based on the fully-sampled data, so it can be viewed as the "best" dictionary that one could learn from the undersampled data. From Table 7.4 we see that the NRMSE values achieved by the online method with two passes are within 0.0% - 0.5% of the oracle NRMSE values, which suggests that the proposed scheme is able to learn dictionaries with good representational qualities from highly undersampled data.

Figure 7.10 shows an example of a learned dictionary produced by the proposed online DINO-KAT method on the PINCAT dataset compared to the initial DCT dictionary. The learned dictionary contains 320 atoms, each of which are $8 \times 8 \times 5$ complex-valued space-time tensors with rank $r = 1$ when reshaped into $64 \times 5$ space-time matrices. Therefore, we display the real and imaginary parts of the first $8 \times 8$ slice of each atom. The figures show that the learned dictionaries have significantly evolved from the initial atoms and have adapted to certain smooth and sharp textures at various orientations. Recall that we chose full-rank ($r = 5$) atoms in the video inpainting experiments, while here we chose low-rank ($r = 1$) atoms. Intuitively, low-rank atoms are a better model for the dynamic MRI data because the videos have high temporal correlation and rank-1 atoms are necessarily constant across time. Conversely, the videos from Section 7.4.1 contained significant camera motion and thus dictionary atoms with more temporal variation (i.e., higher rank) enabled more accurate reconstructions.

## 7.5 Conclusion

In this chapter, we presented a novel framework for online estimation of dynamic image sequences by learning dictionaries with low-rank (upon reshaping) atoms. The proposed algorithm sequentially and efficiently updates the images, dictionary, and sparse coefficients for each image patch from streaming measurements. We also proposed a variation of our method that enforces a unitary constraint on the learned dictionary. Importantly, our algorithms are fully-online and thus can process arbitrarily long video sequences without scaling memory usage over time. Our numerical experiments demonstrate that the proposed methods produce accurate reconstructions on both video inpainting and dynamic MRI reconstructions tasks. These results suggest that our methods may also be suitable for other inverse problems, including medical imaging applications such as interventional imaging and other video-processing tasks from computer vision. We hope to investigate these application domains in future work.

# CHAPTER VIII

# Low-Rank and Adaptive Sparse Signal Models for Highly Accelerated Dynamic Imaging

## 8.1 Introduction

Sparsity-based techniques are popular in many applications in image processing and imaging. Sparsity in either a fixed or data-adaptive dictionary or transform is fundamental to the success of popular techniques such as compressed sensing that aim to reconstruct images from limited sensor measurements. In this chapter, we focus on low-rank and adaptive dictionary-sparse models for dynamic imaging data and exploit such models to perform image reconstruction from limited (compressive) measurements. In the following, we briefly review compressed sensing (CS), CS-based magnetic resonance imaging (MRI), and dynamic data modeling. We then outline the contributions of this chapter.

### 8.1.1 Background

CS [69, 70, 149, 150] is a popular technique that enables recovery of signals or images from far fewer measurements (or at a lower rate) than the number of unknowns or than required by Nyquist sampling conditions. CS assumes that the underlying signal is sparse in some transform domain or dictionary and that the measurement acquisition procedure is incoherent in an appropriate sense with the dictionary. CS has been shown to be very useful for MRI [136,137]. MRI is a relatively slow modality because the data, which are samples in the Fourier space (or k-space) of the object, are acquired sequentially in time. In spite of advances in scanner hardware and pulse sequences, the rate at which MR data are acquired is limited by MR physics and physiological constraints [136].

CS has been applied to a variety of MR techniques such as static MRI [136, 151, 152], dynamic MRI (dMRI) [137,138,153,154], parallel imaging (pMRI) [155–158], and perfusion imaging and diffusion tensor imaging (DTI) [159]. For static MR imaging, CS-based MRI (CSMRI) involves undersampling the k-space data (e.g., collecting fewer phase encodes) using random sampling techniques to accelerate data acquisition. However, in dynamic MRI the data is inherently undersampled because the object is changing as the data is being collected, so in a sense *all* dynamic MRI scans (of k-t space) involve some form of CS because one must reconstruct the dynamic images from under-sampled data. The traditional approach to this problem in MRI is to use "data sharing" where data is pooled in time to make sets of k-space data (e.g., in the form of a Casorati matrix [139]) that appear to have sufficient samples, but these methods do not fully model the temporal changes in the object. CS-based dMRI can achieve improved temporal (or spatial) resolution by using more explicit signal models rather than only implicit k-space data sharing, albeit at the price of increased computation.

CSMRI reconstructions with fixed, non-adaptive signal models (e.g., wavelets or total variation sparsity) typically suffer from artifacts at high undersampling factors [103]. Thus, there has been growing interest in image reconstruction methods where the dictionary is adapted to provide highly sparse representations of data. Recent research has shown benefits for such data-driven adaptation of dictionaries [104, 109, 110, 160] in many applications [100, 101, 103, 161]. For example, the DLMRI method [103] jointly estimates the image and a synthesis dictionary for the image patches from undersampled k-space measurements. The model there is that the unknown (vectorized) image patches can be well approximated by a sparse linear combination of the columns or atoms of a learned (a priori unknown) dictionary $D$. This idea of joint dictionary learning and signal reconstruction from undersampled measurements [103], known as (dictionary) blind compressed sensing (BCS) [162], has been the focus of several recent works (including for dMRI reconstruction) [103, 133, 144, 163–170]. The BCS problem is harder than conventional (non-adaptive) compressed sensing. However, the dictionaries learned in BCS typically reflect the underlying image properties better than pre-determined models, thus improving image reconstructions.

While CS methods use sparse signal models, various alternative models have been explored for dynamic data in recent years. Several works have demonstrated the efficacy of low-rank models (e.g., by constraining the Casorati data matrix to have low-rank) for dynamic MRI reconstruction [139–142]. A recent work [143] also considered a low-rank property for local space-time image patches. For data such as videos (or collections of related images [171]), there has been growing interest in decomposing the data into the sum of a low-rank (L) and a sparse (S) component [31, 32, 63]. In this L+S (or equivalently Robust Principal Component Analysis (RPCA) [31]) model, the L component may capture the background of the video, while the S component captures the sparse (dynamic) foreground. The L+S model has been recently shown to be promising for CS-based dynamic MRI [1, 172]. The S component of the L+S decomposition could either be sparse by itself or sparse in some known dictionary or transform domain. Some works alternatively consider modeling the dynamic image sequence as both low-rank and sparse (L & S) [2, 173], with a recent work [174] learning dictionaries for the S part of L & S. In practice, which model provides better image reconstructions may depend on the specific properties of the underlying data.

When employing the L+S model, the CS reconstruction problem can be formulated as follows:

$$\text{(P0)} \quad \min_{x_L, x_S} \ \frac{1}{2}\|A(x_L + x_S) - d\|_2^2 + \lambda_L \|\mathcal{R}_1(x_L)\|_* + \lambda_S \|T x_S\|_1.$$

In (P0), the underlying unknown dynamic object is $x = x_L + x_S \in \mathbb{C}^{N_x N_y N_t}$, where $x_L$ and $x_S$ are vectorized versions of space-time (3D) tensors corresponding to $N_t$ temporal frames, each an image[1] of size $N_x \times N_y$. The operator $A$ is the sensing or encoding operator and $d$ denotes the (undersampled) measurements. For parallel imaging with $N_c$ receiver coils, applying the operator $A$ involves frame-by-frame multiplication by coil sensitivities followed by applying an undersampled Fourier encoding (i.e., the SENSE method) [37]. The operation $\mathcal{R}_1(x_L)$ reshapes $x_L$ into an $N_x N_y \times N_t$ matrix, and $\|\cdot\|_*$ denotes the nuclear norm that sums the singular values of a matrix. The nuclear norm serves as a convex surrogate for matrix rank in (P0). Traditionally, the operator $T$ in (P0) is a *known* sparsifying transform for $x_S$, and $\lambda_L$ and $\lambda_S$ are non-negative weights.

### 8.1.2 Contributions

This chapter investigates in detail the extension of the L+S model for dynamic data to a Low-rank + Adaptive Sparse SIgnal (LASSI) model. In particular, we decompose the underlying temporal image sequence into a low-rank component and a component whose overlapping spatiotemporal (3D) patches are assumed sparse in

---

[1] We focus on 2D + time for simplicity but the concepts generalize readily to 3D + time.

some *adaptive* dictionary domain.[2] We propose a framework to jointly estimate the underlying signal components and the spatiotemporal dictionary from limited measurements. We compare using $\ell_0$ and $\ell_1$ penalties for sparsity in our formulations, and also investigate adapting structured dictionaries, where the atoms of the dictionary, after being reshaped into space-time matrices are low-rank. The proposed iterative LASSI reconstruction algorithms involve efficient block coordinate descent-type updates of the dictionary and sparse coefficients of patches, and an efficient proximal gradient-based update of the signal components. We also obtain novel sparsity penalized dictionary-blind compressed sensing methods as special cases of our LASSI approaches.

Our experiments demonstrate the promising performance of the proposed data-driven schemes for dMRI reconstruction from limited k-t space data. In particular, we show that the LASSI methods give much improved reconstructions compared to the recent L+S method and methods involving joint L & S modeling [2]. We also show improvements with LASSI compared to the proposed spatiotemporal dictionary-BCS methods (that are special cases of LASSI). Moreover, learning structured dictionaries and using the $\ell_0$ sparsity "norm" in LASSI are shown to be advantageous in practice. Finally, in our experiments, we compare the use of conventional singular value thresholding (SVT) for updating the low-rank signal component in the LASSI algorithms to alternative approaches including the recent OptShrink method [6, 7, 26].

### 8.1.3 Organization

The rest of this chapter is organized as follows. Section 8.2 describes our models and problem formulations for dynamic image reconstruction. Section 8.3 presents efficient algorithms for the proposed problems and discusses the algorithms' properties. Section 8.4 presents experimental results demonstrating the convergence behavior and performance of the proposed schemes for the dynamic MRI application. Section 8.5 concludes with proposals for future work.

---

[2] The LASSI method differs from the scheme in [175] that is not (overlapping) patch-based and involves only a 2D (spatial) dictionary. The model in [175] is that $\mathcal{R}_1(x_S) = DZ$ with sparse $Z$ and the atoms of $D$ have size $N_x N_y$ (typically very large). Since often $N_t < N_x N_y$, one can easily construct trivial (degenerate) sparsifying dictionaries (e.g., $D = \mathcal{R}_1(x_S)$) in this case. On the other hand, in our framework, the dictionaries are for small spatiotemporal patches, and there are many such overlapping patches for a dynamic image sequence to enable the learning of rich models that capture local spatiotemporal properties.

## 8.2 Models and Problem Formulations

### 8.2.1 LASSI Formulations

We model the dynamic image data as $x = x_L + x_S$, where $x_L$ is low-rank when reshaped into a (space-time) matrix, and we assume that the spatiotemporal (3D) patches in the vectorized tensor $x_S$ are sparse in some adaptive dictionary domain. We replace the regularizer $\zeta(x_s) = \|Tx_S\|_1$ with weight $\lambda_S$ in (P0) with the following patch-based dictionary learning regularizer

$$\begin{aligned} \zeta(x_s) \;=\; & \min_{D,Z} \; \sum_{j=1}^{M} \|P_j x_S - D z_j\|_2^2 + \lambda_Z^2 \|Z\|_0 \\ & \text{s.t. } \; \|Z\|_\infty \leq a, \; \mathbf{rank}(\mathcal{R}_2(d_i)) \leq r, \; \|d_i\|_2 = 1, \; \forall i \end{aligned} \tag{8.1}$$

to arrive at the following problem for joint image sequence reconstruction and dictionary estimation:

$$\begin{aligned} \text{(P1)} \quad \min_{D,Z,x_L,x_S} \; & \frac{1}{2}\|A(x_L + x_S) - d\|_2^2 + \lambda_L \|\mathcal{R}_1(x_L)\|_* \\ + \; & \lambda_S \left( \sum_{j=1}^{M} \|P_j x_S - D z_j\|_2^2 + \lambda_Z^2 \|Z\|_0 \right) \\ \text{s.t. } \; & \|Z\|_\infty \leq a, \; \mathbf{rank}(\mathcal{R}_2(d_i)) \leq r, \; \|d_i\|_2 = 1, \; \forall i. \end{aligned}$$

Here, $P_j$ is a patch extraction matrix that extracts an $m_x \times m_y \times m_t$ spatiotemporal patch from $x_S$ as a vector. A total of $M$ (spatially and temporally) overlapping 3D patches are assumed. Matrix $D \in \mathbb{C}^{m \times K}$ with $m = m_x m_y m_t$ is the synthesis dictionary to be learned and $z_j \in \mathbb{C}^K$ is the unknown sparse code for the $j$th patch, with $P_j x_S \approx D z_j$.

We use $Z \in \mathbb{C}^{K \times M}$ to denote the matrix that has the sparse codes $z_j$ as its columns, $\|Z\|_0$ (based on the $\ell_0$ "norm") counts the number of nonzeros in the matrix $Z$, and $\lambda_Z \geq 0$. Problem (P1) penalizes the number of nonzeros in the (entire) coefficient matrix $Z$, allowing variable sparsity levels across patches. This is a general and flexible model for image patches (e.g., patches from different regions in the dynamic image sequence may contain different amounts of information and therefore all patches may not be well represented at the same sparsity) and leads to promising performance in our experiments. The constraint $\|Z\|_\infty \triangleq \max_j \|z_j\|_\infty \leq a$ with $a > 0$ is used in (P1) because the objective (specifically the regularizer (8.1)) is non-coercive with respect to $Z$ [119].[3] The $\ell_\infty$ constraint prevents pathologies that could theoretically arise (e.g., unbounded algorithm iterates) due to the non-coercive objective. In practice, we set $a$ very large, and the constraint is typically inactive.

The atoms or columns of $D$, denoted by $d_i$, are constrained to have unit norm in (P1) to avoid scaling ambiguity between $D$ and $Z$ [105, 119]. We also model the reshaped dictionary atoms $\mathcal{R}_2(d_i)$ as having rank at most $r > 0$, where the operator $\mathcal{R}_2(\cdot)$ reshapes $d_i$ into a $m_x m_y \times m_t$ space-time matrix. Imposing low-rank (small $r$) structure on reshaped dictionary atoms is motivated by our empirical observation that the dictionaries learned on image patches (without such a constraint) tend to have reshaped atoms with only a few dominant singular values. Our numerical results in Section 8.4 show that dictionaries learned on dynamic image patches with low-rank atom constraints tend to represent such data as well as learned dictionaries with full-rank atoms. Importantly, such structured dictionary learning may be less prone to over-fitting in scenarios involving limited or corrupted data. We illustrate this for the dynamic MRI application in Section 8.4.

When $z_j$ is highly sparse (with $\|z_j\|_0 \ll \min(m_t, m_x m_y)$) and $\mathcal{R}_2(d_i)$ has low rank (say rank-1), the model $P_j x_S \approx D z_j$ corresponds to approximating the space-time patch matrix as a sum of a few reshaped low-rank (rank-1) atoms. This special (extreme) case would correspond to approximating the patch itself as low-rank. However, in general the decomposition $Dz_j$ could involve numerous ($> \min(m_t, m_x m_y)$) active atoms, corresponding to a rich, not necessarily low-rank, patch model. Experimental results in Section 8.4 illustrate the benefits of such rich models.

Problem (P1) jointly learns a decomposition $x = x_L + x_S$ and a dictionary $D$ along with the sparse coefficients $Z$ (of spatiotemporal patches) from the measurements $d$. Unlike (P0), the fully-adaptive Problem (P1) is nonconvex. An alternative to (P1) involves replacing the $\ell_0$ "norm" with the convex $\ell_1$ norm (with $\|Z\|_1 = \sum_{j=1}^{M} \|z_j\|_1$) as follows:

$$\begin{aligned}
\text{(P2)} \quad \min_{D, Z, x_L, x_S} \quad & \frac{1}{2}\|A(x_L + x_S) - d\|_2^2 + \lambda_L \|\mathcal{R}_1(x_L)\|_* \\
+ \quad & \lambda_S \left( \sum_{j=1}^{M} \|P_j x_S - D z_j\|_2^2 + \lambda_Z \|Z\|_1 \right) \\
\text{s.t.} \quad & \|Z\|_\infty \le a, \ \mathbf{rank}(\mathcal{R}_2(d_i)) \le r, \ \|d_i\|_2 = 1, \forall i.
\end{aligned}$$

---

[3] Such a non-coercive function remains finite even in cases when $\|Z\| \to \infty$. For example, consider a dictionary $D$ that has a column $d_i$ that repeats. Then, in this case, the patch coefficient vector $z_j$ in (P1) could have entries $\alpha$ and $-\alpha$ respectively, corresponding to the two repeated atoms in $D$, and the objective would be invariant to arbitrarily large scaling of $|\alpha|$ (i.e., non-coercive).

Problem (P2) is also nonconvex due to the product $Dz_j$ (and the nonconvex constraints), so the question of choosing (P2) or (P1) is one of image quality, not convexity.

Finally, the convex nuclear norm penalty $\|\mathcal{R}_1(x_L)\|_*$ in (P1) or (P2) could be alternatively replaced with a nonconvex penalty on the rank of $\mathcal{R}_1(x_L)$, or the function $\|\cdot\|_p^p$ for $p < 1$ (based on the Schatten $p$-norm) that is applied to the vector of singular values of $\mathcal{R}_1(x_L)$ [2]. While we focus mainly on the popular nuclear norm penalty in our investigations, we also briefly study some of the alternatives in Section 8.3 and Section 8.4.5.

### 8.2.2 Special Case of LASSI Formulations: Dictionary-Blind Image Reconstruction

When $\lambda_L \to \infty$ in (P1) or (P2), the optimal low-rank component of the dynamic image sequence becomes inactive (zero). The problems then become pure spatiotemporal dictionary-blind image reconstruction problems (with $x_L = 0$ and $x = x_S$) involving $\ell_0$ or $\ell_1$ overall sparsity [119] penalties. For example, Problem (P1) reduces to

$$\begin{aligned} \min_{D,Z,x} \quad & \frac{1}{2}\|Ax - d\|_2^2 + \lambda_S \left( \sum_{j=1}^{M} \|P_j x - D z_j\|_2^2 + \lambda_Z^2 \|Z\|_0 \right) \\ \text{s.t.} \quad & \|Z\|_\infty \leq a, \ \mathbf{rank}(\mathcal{R}_2(d_i)) \leq r, \ \|d_i\|_2 = 1, \forall i, \end{aligned} \tag{8.2}$$

which is exactly the DINO-KAT (DIctioNary with lOw-ranK AToms) blind image reconstruction problem from Chapter VI. A similar formulation is obtained from (P2) but with an $\ell_1$ penalty. These formulations differ from the ones proposed for dynamic image reconstruction in prior works such as [133, 144], [166]. In [144], dynamic image reconstruction is performed by learning a common real-valued dictionary for the spatiotemporal patches of the real and imaginary parts of the dynamic image sequence. The algorithm therein involves dictionary learning using K-SVD [104], where sparse coding is performed using the approximate and expensive orthogonal matching pursuit method [176]. In contrast, the algorithms in this chapter (cf. Section 8.3) for the overall sparsity penalized DINO-KAT blind image reconstruction problems involve simple and efficient updating of the *complex-valued* spatiotemporal dictionary (for complex-valued 3D patches) and sparse coefficients (by simple thresholding) in the formulations. The advantages of employing sparsity penalized dictionary learning over conventional approaches like K-SVD are discussed in more detail elsewhere [119]. In [166], a spatiotemporal dictionary is learned for the complex-valued 3D patches of the dynamic image sequence (a total variation penalty is also used), but the method again involves dictionary learning using K-SVD. In the blind compressed sensing method of [133], the time-profiles of individual image pixels were modeled as sparse in a learned dictionary. The 1D voxel time-profiles are a special case of general overlapping 3D (spatiotemporal) patches. Spatiotemporal dictionaries as used here may help capture redundancies in both spatial and temporal dimensions in the data. Finally, unlike the prior works, the DINO-KAT schemes in this chapter involve structured dictionary learning with low-rank reshaped atoms.

## 8.3 Algorithms and Properties

### 8.3.1 Algorithms

We propose efficient block coordinate descent-type algorithms for (P1) and (P2), where, in one step, we update $(D, Z)$ keeping $(x_L, x_S)$ fixed (the *dictionary learning step*), and then we update $(x_L, x_S)$ keeping $(D, Z)$ fixed (the *image reconstruction step*). We repeat these alternating steps in an iterative manner. The algorithm for the DINO-KAT blind image reconstruction problem (8.2) (or its $\ell_1$ version) is similar, except that $x_L = 0$ during the update steps. Therefore, we focus on the algorithms for (P1) and (P2) in the following.

#### 8.3.1.1 Dictionary Learning Step

Here, we optimize (P1) or (P2) with respect to $(D, Z)$. We first describe the update procedure for (P1). Denoting by $P$ the matrix that has the patches $P_j x_S$ for $1 \leq j \leq M$ as its columns, and with $C \triangleq Z^H$, the optimization problem with respect to $(D, Z)$ in the case of (P1) can be rewritten as follows:

$$\begin{aligned} \text{(P3)} \quad & \min_{D,C} \; \|P - DC^H\|_F^2 + \lambda_Z^2 \|C\|_0 \\ & \text{s.t.} \;\; \|C\|_\infty \leq a, \; \mathbf{rank}(\mathcal{R}_2(d_i)) \leq r, \; \|d_i\|_2 = 1, \; \forall i. \end{aligned}$$

Here, we express the matrix $DC^H$ as a Sum of OUter Products (SOUP) $\sum_{i=1}^{K} d_i c_i^H$. We then employ an iterative block coordinate descent method for (P3), where the columns $c_i$ of $C$ and atoms $d_i$ of $D$ are updated sequentially by cycling over all $i$ values [119]. Specifically, for each $1 \leq i \leq K$, we solve (P3) first with respect to $c_i$ (*sparse coding*) and then with respect to $d_i$ (*dictionary atom update*).

For the minimization with respect to $c_i$, we have the following subproblem, where

$E_i \triangleq P - \sum_{k \neq i} d_k c_k^H$ is computed using the most recent estimates of the other variables:

$$\begin{aligned} \min_{c_i \in \mathbb{C}^M} \quad & \|E_i - d_i c_i^H\|_F^2 + \lambda_Z^2 \|c_i\|_0 \\ \text{s.t.} \quad & \|c_i\|_\infty \leq a. \end{aligned} \tag{8.3}$$

The minimizer $\hat{c}_i$ of (8.3) is given by [119]

$$\hat{c}_i = \min(|H_{\lambda_Z}(E_i^H d_i)|,\ a 1_M) \ \odot \ e^{j \angle E_i^H d_i}, \tag{8.4}$$

where the hard thresholding operator $H_{\lambda_Z}(\cdot)$ zeros out vector entries with magnitude less than $\lambda_Z$ and leaves the other entries (with magnitude $\geq \lambda_Z$) unaffected. Here, $|\cdot|$ computes the magnitude of vector entries, $1_M$ denotes a vector of ones of length $M$, $\odot$ denotes element-wise multiplication, $\min(\cdot, \cdot)$ denotes element-wise minimum, and we choose $a$ such that $a > \lambda_Z$. For a vector $c \in \mathbb{C}^M$, $e^{j\angle c} \in \mathbb{C}^M$ is computed element-wise, with $\angle$ denoting the phase.

Optimizing (P3) with respect to the atom $d_i$ while holding all other variables fixed yields the following subproblem:

$$\begin{aligned} \min_{d_i \in \mathbb{C}^m} \quad & \|E_i - d_i c_i^H\|_F^2 \\ \text{s.t.} \quad & \textbf{rank}(\mathcal{R}_2(d_i)) \leq r,\ \|d_i\|_2 = 1. \end{aligned} \tag{8.5}$$

Let $U_r \Sigma_r V_r^H$ denote an optimal rank-$r$ approximation to $\mathcal{R}_2(E_i c_i) \in \mathbb{C}^{m_x m_y \times m_t}$ that is obtained using the $r$ leading singular vectors and singular values of the full singular value decomposition (SVD) $\mathcal{R}_2(E_i c_i) \triangleq U \Sigma V^H$. Then a global minimizer of (8.5), upon reshaping, is

$$\mathcal{R}_2(\hat{d}_i) = \begin{cases} \dfrac{U_r \Sigma_r V_r^H}{\|\Sigma_r\|_F} & \text{if } c_i \neq 0 \\ W & \text{if } c_i = 0, \end{cases} \tag{8.6}$$

where $W$ is any normalized matrix with rank at most $r$, of appropriate dimensions (e.g., we use the reshaped first column of the $m \times m$ identity matrix). See Chapter VI for the proof of (8.6). If $r = \min(m_x m_y, m_t)$, then no SVD is needed and the solution is [119]

$$\hat{d}_i = \begin{cases} \dfrac{E_i c_i}{\|E_i c_i\|_2} & \text{if } c_i \neq 0 \\ w & \text{if } c_i = 0, \end{cases} \tag{8.7}$$

where $w$ is any vector on the $m$-dimensional unit sphere (e.g., we use the first column of the $m \times m$ identity).

In the case of (P2), when minimizing with respect to $(D, Z)$, we again set $C = Z^H$, which yields an $\ell_1$ penalized dictionary learning problem (a simple variant of (P3)). The dictionary and sparse coefficients are then updated using a similar block coordinate descent method as for (P3). In particular, the coefficients $c_i$ are updated using soft thresholding:

$$\hat{c}_i = \max\left(|E_i^H d_i| - \frac{\lambda_Z}{2} 1_M,\ 0\right) \odot e^{j\angle E_i^H d_i}. \tag{8.8}$$

#### 8.3.1.2 Image Reconstruction Step

Minimizing (P1) or (P2) with respect to $x_L$ and $x_S$ yields the following subproblem:

$$\text{(P4)} \quad \min_{x_L, x_S} \ \frac{1}{2}\|A(x_L + x_S) - d\|_2^2 + \lambda_L \|\mathcal{R}_1(x_L)\|_* + \lambda_S \sum_{j=1}^{M} \|P_j x_S - D z_j\|_2^2.$$

Problem (P4) is convex but nonsmooth, and its objective has the form $f(x_L, x_S) + g_1(x_L) + g_2(x_S)$, with $f(x_L, x_S) \triangleq 0.5\|A(x_L + x_S) - d\|_2^2$, $g_1(x_L) \triangleq \lambda_L \|\mathcal{R}_1(x_L)\|_*$, and $g_2(x_S) \triangleq \lambda_S \sum_{j=1}^{M} \|P_j x_S - D z_j\|_2^2$. We employ the proximal gradient method [1] for (P4), whose iterates, denoted by superscript $k$, take the following form:

$$x_L^k = \text{prox}_{t_k g_1}(x_L^{k-1} - t_k \nabla_{x_L} f(x_L^{k-1},\ x_S^{k-1})), \tag{8.9}$$

$$x_S^k = \text{prox}_{t_k g_2}(x_S^{k-1} - t_k \nabla_{x_S} f(x_L^{k-1},\ x_S^{k-1})), \tag{8.10}$$

where the proximity function is defined as

$$\text{prox}_{t_k g}(y) \triangleq \arg\min_z \ \frac{1}{2}\|y - z\|_2^2 + t_k\, g(z), \tag{8.11}$$

and the gradients of $f$ are given by

$$\nabla_{x_L} f(x_L, x_S) = \nabla_{x_S} f(x_L, x_S) = A^H A(x_L + x_S) - A^H d.$$

The update in (8.9) corresponds to the singular value thresholding (SVT) operation [24]. Indeed, defining $\tilde{x}_L^{k-1} \triangleq x_L^{k-1} - t_k \nabla_{x_L} f(x_L^{k-1}, x_S^{k-1})$, it follows from (8.9) and (8.11) [24] that

$$\mathcal{R}_1(x_L^k) = \mathbf{SVT}_{t_k \lambda_L}(\mathcal{R}_1(\tilde{x}_L^{k-1})). \tag{8.12}$$

Here, the SVT operator for a given threshold $\tau > 0$ is

$$\mathbf{SVT}_{\tau}(Y) = \sum_{i} (\sigma_i - \tau)^{+} u_i v_i^H, \tag{8.13}$$

where $U \Sigma V^H$ is the SVD of $Y$ with $\sigma_i$ denoting the $i$th largest singular value and $u_i$ and $v_i$ denoting the $i$th columns of $U$ and $V$, and $(\cdot)^{+} = \max(\cdot, 0)$ sets negative values to zero.

Let $\tilde{x}_S^{k-1} \triangleq x_S^{k-1} - t_k \nabla_{x_S} f(x_L^{k-1}, x_S^{k-1})$. Then (8.10) and (8.11) imply that $x_S^k$ satisfies the following normal equation:

$$\left( I + 2t_k \lambda_S \sum_{j=1}^{M} P_j^T P_j \right) x_S^k = \tilde{x}_S^{k-1} + 2t_k \lambda_S \sum_{j=1}^{M} P_j^T D z_j. \tag{8.14}$$

Solving (8.14) for $x_S^k$ is straightforward because the matrix pre-multiplying $x_S^k$ is diagonal, and thus its inverse can be computed cheaply. The term $2t_k \lambda_S \sum_{j=1}^{M} P_j^T D z_j$ in (8.14) can also be computed cheaply using patch-based operations.

The proximal gradient method for (P4) converges [145] for a constant step-size $t_k = t < 2/\ell$, where $\ell$ is the Lipschitz constant of $\nabla f(x_L, x_S)$. For (P4), $\ell = 2\|A\|_2^2$. In practice, $\ell$ can be precomputed using standard techniques such as the power iteration method. In our dMRI experiments in Section 8.4, we normalize the encoding operator $A$ so that $\|A\|_2 = 1$ for fully-sampled measurements (cf. [1,4]) to ensure that $\|A\|_2^2 \leq 1$ in undersampled (k-t space) scenarios.

When the nuclear norm penalty in (P4) is replaced with a rank penalty, i.e., $g_1(x_L) \triangleq \lambda_L \, \mathbf{rank}(\mathcal{R}_1(x_L))$, the proximity function is a modified form of the SVT operation in (8.12) (or (8.13)), where the singular values smaller than $\sqrt{2t_k \lambda_L}$ are set to zero and the other singular values are left unaffected (i.e., hard-thresholding the singular values). Alternatively, when the nuclear norm penalty is replaced with $\| \cdot \|_p^p$ (for $p < 1$) applied to the vector of singular values of $\mathcal{R}_1(x_L)$ [2], the proximity function can still be computed cheaply when $p = 1/2$ or $p = 2/3$, for which the soft thresholding of singular values in (8.13) is replaced with the solution of an appropriate polynomial equation (see [177]). For general $p$, the $x_L$ update could be performed using strategies such as in [2].

The nuclear norm-based low-rank regularizer $\|\mathcal{R}_1(x_L)\|_\star$ is popular because it is the tightest convex relaxation of the (nonconvex) matrix rank penalty. However, this does not guarantee that the nuclear norm (or its alternatives) is the optimal (in any sense) low-rank regularizer in practice. Indeed, the argument $\mathcal{R}_1(\tilde{x}_L^{k-1})$ of the SVT operator in (8.12) can be interpreted as an estimate of the underlying (true) low-rank matrix $\mathcal{R}_1(x_L)$ plus a residual (noise) matrix. In [26], the low-rank denoising problem was studied from a random-matrix-theoretic perspective and an algorithm – OptShrink – was derived that asymptotically achieves minimum squared error among all estimators that shrink the singular values of their argument. We leverage this result for dMRI by proposing the following modification of (8.12):

$$\mathcal{R}_1(x_L^k) = \mathbf{OptShrink}_{r_L}(\mathcal{R}_1(\tilde{x}_L^{k-1})). \tag{8.15}$$

Here, $\mathbf{OptShrink}_{r_L}(.)$ is the data-driven OptShrink estimator from Algorithm 1 of [26].[4] In this variation, the regularization parameter $\lambda_L$ is replaced by a parameter $r_L \in \mathbb{N}$ that directly specifies the rank of $\mathcal{R}_1(x_L^k)$, and the (optimal) shrinkage for each of the leading $r_L$ singular values is implicitly estimated based on the distribution of the remaining singular values. Intuitively, we expect this variation of the aforementioned (SVT-based) proximal gradient scheme to yield better estimates of the underlying low-rank component of the reconstruction because, at each iteration $k$ (in (8.9)), the OptShrink-based update (8.15) should produce an estimate of the underlying low-rank matrix $\mathcal{R}_1(x_L)$ with smaller squared error than the corresponding SVT-based update (8.12). Similar OptShrink-based schemes have shown promise in practice [6,7]. In particular, in [7] it is shown that replacing the SVT-based low-rank updates in the algorithm [1] for (P0) with OptShrink updates can improve dMRI reconstruction quality. In practice, small $r_L$ values perform well due to the high spatiotemporal correlation of the background in dMRI.

Figure 8.1 shows the LASSI reconstruction algorithms for Problems (P1) and (P2), respectively. As discussed, we can obtain variants of these proposed LASSI algorithms by replacing the SVT-based $x_L$ update (8.12) in the image reconstruction step with an OptShrink-based update (8.15), or with the update arising from the rank penalty or from the Schatten $p$-norm ($p < 1$) penalty. The proposed LASSI algorithms start with an initial $(x_L^0, x_S^0, D^0, Z^0)$. For example, $D^0$ can be set to an analytical dictionary, $Z^0 = 0$, and $x_L^0$ and $x_S^0$ could be (for example) set based on some iterations of the recent L+S method [1]. In the case of Problem (8.2), the proposed algorithm is an efficient SOUP-based image reconstruction algorithm. We refer to it as the DINO-KAT image reconstruction algorithm in this case.

---

[4] See Chapter II for more details and discussion of OptShrink.

```
Algorithms for (P1) and (P2)
Inputs: Measurements $d$, weights $\lambda_L$, $\lambda_S$, and $\lambda_Z$, rank $r$, upper bound $a$, number of dictionary learning iterations $J$, number of proximal gradient iterations $\tilde{J}$, and number of outer iterations $\hat{J}$.
Outputs: Reconstructed dynamic image sequence components $x_L^{\hat{J}}$ and $x_S^{\hat{J}}$, learned dictionary $D^{\hat{J}}$, and learned coefficients of patches $Z^{\hat{J}}$.
Initial Estimates: $(x_L^0, x_S^0, D^0, Z^0)$, with $C^0 = (Z^0)^H$.
For $t = 1 : \hat{J}$ repeat
    1. Form $P^{t-1} = [P_1 x_S^{t-1} \mid P_2 x_S^{t-1} \mid ... \mid P_M x_S^{t-1}]$.
    2. Dictionary Learning: With training data $P^{t-1}$ and initialization $(D^{t-1}, C^{t-1})$, update $(c_i, d_i)$ sequentially for $1 \leq i \leq K$ using (8.4) (or (8.8)) and (8.6). Set $(D^t, C^t)$ to be the output after $J$ cycles of such updates, and $Z^t = (C^t)^H$.
    3. Image Reconstruction: Update $x_L^t$ and $x_S^t$ using $\tilde{J}$ iterations of the proximal gradient scheme using (8.9) and (8.10), and with initialization $(x_L^{t-1}, x_S^{t-1})$.
End
```

Figure 8.1: The LASSI reconstruction algorithms for Problems (P1) and (P2), respectively. Superscript $t$ denotes the iterates in the algorithm.

### 8.3.2 Convergence and Computational Cost

The proposed LASSI algorithms for (P1) and (P2) alternate between updating $(D, Z)$ and $(x_L, x_S)$. Since we update the dictionary atoms and sparse coefficients using an exact block coordinate descent approach, the objectives in our formulations only decrease in this step. When the $(x_L, x_S)$ update is performed using proximal gradients (which is guaranteed to converge to the global minimizer of (P4)), by appropriate choice of the constant-step size [25], the objective functions can be ensured to be monotone (non-increasing) in this step. Thus, the costs in our algorithms are monotone decreasing, and because they are lower-bounded (by 0), they must converge. Whether the iterates in the LASSI algorithms converge to the critical points [124] in (P1) or (P2) [119] is an interesting question that we leave for future work.

In practice, the computational cost per outer iteration of the proposed algorithms is dominated by the cost of the dictionary learning step, which scales (assuming $K \propto m$ and $M \gg K, m$) as $O(m^2 M J)$, where $J$ is the number of times the matrix $D$ is updated in the dictionary learning step. The SOUP dictionary learning cost is itself dominated by various matrix-vector products, whereas the costs of the truncated hard-thresholding (8.4) and low-rank approximation (8.6) steps are negligible. On the other hand, when dictionary learning is performed using methods like K-SVD [104] (e.g., in [103, 165]), the associated cost (assuming per-patch sparsity $\propto m$) may scale worse[5] as $O(m^3 MJ)$. Section 8.4 illustrates that our algorithms converge quickly in practice.

## 8.4 Numerical Experiments

### 8.4.1 Framework

The proposed LASSI framework can be used for inverse problems involving dynamic data, such as in dMRI, interventional imaging, video processing, etc. Here, we illustrate the convergence behavior and performance of our methods for dMRI reconstruction from limited k-t space data. Section 8.4.2 focuses on empirical convergence and learning behavior of the methods. Section 8.4.3 compares the image reconstruction quality obtained with LASSI to that obtained with recent techniques. Section 8.4.5 investigates and compares the various LASSI models and methods in detail. We compare using the $\ell_0$ "norm" (i.e., (P1)) to the $\ell_1$ norm (i.e., (P2)), structured (with low-rank atoms) dictionary learning to the learning of unstructured (with full-rank atoms) dictionaries, and singular value thresholding-based $x_L$ update to OptShrink-based or other alternative $x_L$ updates in LASSI. We also investigate the effects of the sparsity level (i.e., number of nonzeros) of the learned $Z$ and the overcompleteness of $D$ in LASSI, and demonstrate the advantages of adapting the patch-based LASSI dictionary compared to using fixed dictionary models in the LASSI algorithms. The LASSI methods are also shown to perform well for various initializations of $x_L$ and $x_S$.

We work with several dMRI datasets from prior works [1, 2]: 1) the Cartesian cardiac perfusion data [1, 4], 2) a 2D cross section of the physiologically improved nonuniform cardiac torso (PINCAT) [178] phantom data (see [2, 5]), and 3) the *in vivo* myocardial perfusion MRI data in [2,5]. The cardiac perfusion data were acquired with a modified TurboFLASH sequence on a 3T scanner using a 12-element coil array. The fully sampled data with an image matrix size of $128 \times 128$ (128 phase encode lines) and 40 temporal frames was acquired with FOV = $320 \times 320$ mm$^2$, slice thickness = 8 mm, spatial resolution = 3.2 mm$^2$, and temporal resolution of 307 ms [1]. The coil sensitivity maps are provided in [4]. The (single coil) PINCAT data (as in [5]) had image matrix size of $128 \times 128$ and 50 temporal frames. The single coil *in vivo*

---

[5] In [119], we have shown that efficient SOUP learning-based image reconstruction methods outperform methods based on K-SVD in practice.

myocardial perfusion data was acquired on a 3T scanner using a saturation recovery FLASH sequence with Cartesian sampling (TR/TE = 2.5/1 ms, saturation recovery time = 100 ms), and had a image matrix size of 90 × 190 (phase encodes × frequency encodes) and 70 temporal frames [2].

Fully sampled data (PINCAT and *in vivo* data were normalized to unit peak image intensity, and the cardiac perfusion data [1] had a peak image intensity of 1.27) were retrospectively undersampled in our experiments. We used Cartesian and pseudo-radial undersampling patterns. In the case of Cartesian sampling, we used a different variable-density random Cartesian undersampling pattern for each time frame. The pseudo-radial (sampling radially at uniformly spaced angles for each time frame and with a small random rotation of the radial lines between frames) sampling patterns were obtained by subsampling on a Cartesian grid for each time frame. We simulate several undersampling factors of k-t space in our experiments. We measure the quality of the dMRI reconstructions using the normalized root mean square error (NRMSE) metric defined as

$$\mathbf{NRMSE}(x_{\text{recon}}) = \frac{\|x_{\text{recon}} - x_{\text{ref}}\|_2}{\|x_{\text{ref}}\|_2}, \tag{8.16}$$

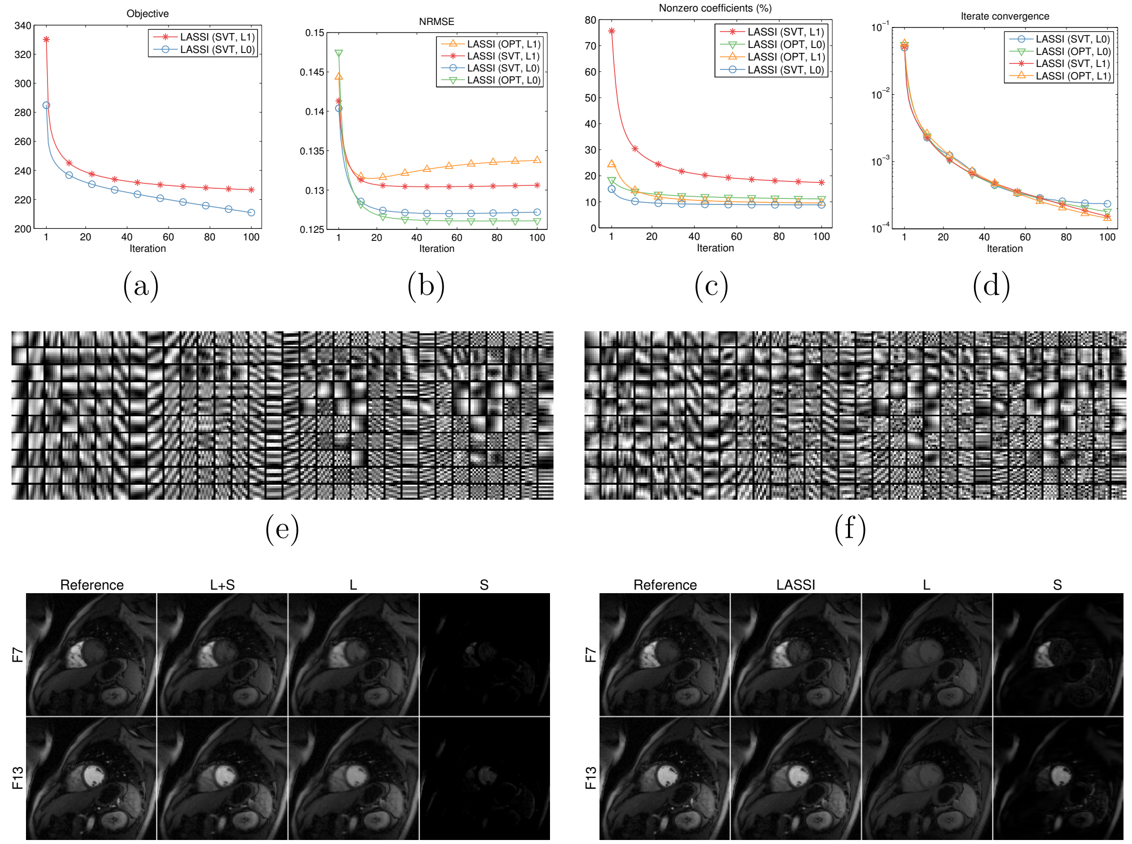
where $x_{\text{ref}}$ is a reference reconstruction from fully sampled data, and $x_{\text{recon}}$ is the reconstruction from undersampled data.

We compare the quality of reconstructions obtained with the proposed LASSI methods to those obtained with the recent L+S method [1] and the k-t SLR method involving joint L & S modeling [2]. For the L+S and k-t SLR methods, we used the publicly available MATLAB implementations [4, 5]. We chose the parameters for both methods (e.g., $\lambda_L$ and $\lambda_S$ for L+S in (P0) or $\lambda_1$, $\lambda_2$, etc. for k-t SLR [2, 5]) by sweeping over a range of values and choosing the settings that achieved good NRMSE in our experiments. We optimized parameters separately for each dataset to achieve the lowest NRMSE at some intermediate undersampling factors, and observed that these settings also worked well at other undersampling factors. The L+S method was simulated for 250 iterations and k-t SLR was also simulated for sufficient iterations to ensure convergence. The operator $T$ (in (P0)) for L+S was set to a temporal Fourier transform, and a total variation sparsifying penalty (together with a nuclear norm penalty for enforcing low-rankness) was used in k-t SLR. The dynamic image sequence in both methods was initialized with a baseline reconstruction (for the L+S method, L was initialized with this baseline and S with zero) that was obtained by first performing zeroth order interpolation at the non-sampled k-t space locations (by filling in with the nearest non-zero entry along time) and then backpropagating the filled k-t space to image space (i.e., pre-multiplying by the $A^H$ corresponding to fully sampled data).

For the LASSI method, we extracted spatiotemporal patches of size $8 \times 8 \times 5$ from $x_S$ in (P1) with spatial and temporal patch overlap strides of 2 pixels.[6] The dictionary atoms were reshaped into $64 \times 5$ space-time matrices, and we set the rank parameter $r = 1$, except for the invivo dataset [2, 5], where we set $r = 5$. We ran LASSI for 50 outer iterations with 1 and 5 inner iterations in the $(D, Z)$ and $(x_L, x_S)$ updates, respectively. Since Problem (P1) is nonconvex, the proposed algorithm needs to be initialized appropriately. We set the initial $Z = 0$, and the initial $x_L$ and $x_S$ were typically set based on the outputs of either the L+S or k-t SLR methods. When learning a square dictionary, we initialized $D$ with a $320 \times 320$ DCT, and, in the overcomplete $(K > m)$ case, we concatenated the square DCT initialization with normalized and vectorized patches that were selected from random locations of the initial reconstruction. We empirically show in Section 8.4.5 that the proposed LASSI algorithms typically improve image reconstruction quality compared to that achieved by their initializations. We selected the weights $\lambda_L$, $\lambda_S$, and $\lambda_Z$ for the LASSI methods separately for each dataset by sweeping over a range (3D grid) of values and picking the settings that achieved the lowest NRMSE at intermediate undersampling factors (as for L+S and k-t SLR) in our experiments. These tuned parameters also worked well at other undersampling factors (e.g., see Figure 8.9(h)).

We also evaluate the proposed variant of LASSI involving only spatiotemporal dictionary learning (i.e., dictionary blind compressed sensing). We refer to this method as DINO-KAT dMRI, with $r = 1$. We use an $\ell_0$ sparsity penalty for DINO-KAT dMRI (i.e., we solve Problem (8.2)) in our experiments, and the other parameters are set or optimized similarly as described above for LASSI.

The LASSI and DINO-KAT dMRI implementations were coded in Matlab R2016a. Our current Matlab implementations are not optimized for efficiency. Hence, here we perform our comparisons to recent methods based on reconstruction quality (NRMSE) rather than runtimes, since the latter are highly implementation dependant.



(a) (b) (c) (d)

(e) (f)

(g) (h)

Figure 8.2: Behavior of the LASSI algorithms with Cartesian sampling and 8x undersampling. The algorithms are labeled according to the method used for $x_L$ update, i.e., SVT or OptShrink (OPT), and according to the type of sparsity penalty employed for the patch coefficients ($\ell_0$ or $\ell_1$ corresponding to (P1) or (P2)). (a) Objectives, shown only for the algorithms for (P1) and (P2) with SVT-based updates, since the OptShrink-based updates do not correspond to minimizing a formal cost function); (b) NRMSE; (c) Sparsity fraction of $Z$ (i.e., $\|Z\|_0/mM$) expressed as a percentage; (d) normalized changes between successive dMRI reconstructions $\|x_L^t + x_S^t - x_L^{t-1} - x_S^{t-1}\|_2/\|x_{\text{ref}}\|_2$; (e) real and (f) imaginary parts of the atoms of the learned dictionaries in LASSI (using $\ell_0$ sparsity penalty and OptShrink-based $x_L$ update) shown as patches – only the $8 \times 8$ patches corresponding to the first time-point (column) of the rank-1 reshaped $(64 \times 5)$ atoms are shown; and frames 7 and 13 of the (g) conventional L+S reconstruction [1] and (h) the proposed LASSI (with $\ell_0$ penalty and OptShrink-based $x_L$ update) reconstruction shown along with the corresponding reference frames. The low-rank (L) and (transform or dictionary) sparse (S) components of each reconstructed frame are also individually shown. Only image magnitudes are displayed in (g) and (h).

### 8.4.2 LASSI Convergence and Learning Behavior

Here, we consider the fully sampled cardiac perfusion data in [1, 4] and perform eight fold Cartesian undersampling of k-t space. We study the behavior of the proposed LASSI algorithms for reconstructing the dMRI data from (multi-coil) undersampled measurements. We consider four different LASSI algorithms in our study here: the algorithms for (P1) (with $\ell_0$ "norm") and (P2) (with $\ell_1$ norm) with SVT-based $x_L$ update; and the variants of these two algorithms where the SVT update step is replaced with an OptShrink (OPT)-type update. The other variants of the SVT update including hard thresholding of singular values or updating based on the Schatten $p$-norm are studied later in Section 8.4.5. We learned $320 \times 320$ dictionaries (with atoms reshaped by the operator $\mathcal{R}_2(\cdot)$ into $64 \times 5$ space-time matrices) for the patches of $x_S$ with $r = 1$, and $x_L$ and $x_S$ were initialized using the corresponding components of the L+S method with $\lambda_L = 1.2$ and $\lambda_S = 0.01$ in (P0) [1]. Here, we jointly tuned $\lambda_L$, $\lambda_S$, and $\lambda_Z$ for each LASSI variation, to achieve the best NRMSE.

Figure 8.2 shows the behavior of the proposed LASSI reconstruction methods. The objective function values (Figure 8.2(a)) in (P1) and (P2) decreased monotonically and quickly for the algorithms with SVT-based $x_L$ update. The OptShrink-based $x_L$ update does not correspond to minimizing a formal cost function, so the OPT-based algorithms are omitted in Figure 8.2(a). All four LASSI methods improved the NRMSE over iterations compared to the initialization. The NRMSE converged (Figure 8.2(b)) in all four cases, with the $\ell_0$ "norm"-based methods outperforming the $\ell_1$ penalty methods. Moreover, when employing the $\ell_0$ sparsity penalty, the OPT-based method ($r_L = 1$) outperformed the SVT-based one for the dataset. The sparsity fraction ($\|Z\|_0/mM$) for the learned coefficients matrix (Figure 8.2(c)) converged to small values (about 10-20 %) in all cases indicating that highly sparse representations are obtained in the LASSI models. Lastly, the difference between successive dMRI reconstructions (Figure 8.2(d)) quickly decreased to small values, suggesting iterate convergence.

Figures 8.2(g) and (h) show the reconstructions[7] and $x_L$ and $x_S$ components of two representative frames produced by the L+S [1] (with parameters optimized to achieve best NRMSE) and LASSI (OPT update and $\ell_0$ sparsity) methods, respectively. The LASSI reconstructions are sharper and a better approximation of the ref-

---

[6] While we used a stride of 2 pixels, a spatial and temporal patch overlap stride of 1 pixel would further enhance the reconstruction performance of LASSI in our experiments, but at the cost of substantially more computation.

[7] Gamma correction was used to better display the images.

| Undersampling | 4x | 8x | 12x | 16x | 20x | 24x |
|---|---|---|---|---|---|---|
| NRMSE (k-t SLR) % | 11.1 | 15.4 | 18.8 | 21.7 | 24.3 | 27.0 |
| NRMSE (L+S) % | 10.9 | 13.9 | 15.8 | 17.8 | 20.1 | 23.0 |
| NRMSE (DINO-KAT) % | 10.4 | 12.6 | 14.5 | 16.7 | 18.8 | 22.1 |
| NRMSE (LASSI) % | **10.0** | **12.6** | **14.3** | **16.1** | **17.6** | **20.2** |
| Gain over k-t SLR (dB) | 0.9 | 1.7 | 2.4 | 2.6 | 2.8 | 2.5 |
| Gain over L+S (dB) | 0.7 | 0.8 | 0.9 | 0.9 | 1.2 | 1.2 |
| Gain over DINO-KAT (dB) | 0.3 | 0.0 | 0.1 | 0.3 | 0.6 | 0.8 |

Table 8.1: NRMSE values expressed as percentages for the L+S [1], k-t SLR [2], and the proposed DINO-KAT dMRI and LASSI methods at several undersampling factors for the cardiac perfusion data [1, 4] with Cartesian sampling. The NRMSE gain (in decibels (dB)) achieved by LASSI over the other methods is also shown. The best NRMSE for each undersampling factor is in bold.

erence frames (fully sampled reconstructions) shown. In particular, the $x_L$ component of the LASSI reconstruction is clearly low-rank, and the $x_S$ component captures the changes in contrast and other dynamic features in the data. On the other hand, the $x_L$ component of the conventional L+S reconstruction varies more over time (i.e., it has higher rank), and the $x_S$ component contains relatively little information. The richer $(x_L, x_S)$ decomposition produced by LASSI suggests that both the low-rank and adaptive dictionary-sparse components of the model are well-suited for dMRI.

Figs. 8.2(e) and (f) show the real and imaginary parts of the atoms of the learned $D$ in LASSI with OptShrink-based $x_L$ updating and $\ell_0$ sparsity. Only the first columns (time-point) of the (rank-1) reshaped $64 \times 5$ atoms are shown as $8 \times 8$ patches. The learned atoms contain rich geometric and frequency-like structures that were jointly learned with the dynamic signal components from limited k-t space measurements.

### 8.4.3 Dynamic MRI Results and Comparisons

Here, we consider the fully sampled cardiac perfusion data [1,4], PINCAT data [2, 5], and *in vivo* myocardial perfusion data [2,5], and simulate k-t space undersampling at various acceleration factors. Cartesian sampling was used for the first dataset, and pseudo-radial sampling was employed for the other two. The performance of LASSI and DINO-KAT dMRI is compared to that of L+S [1] and k-t SLR [2]. The LASSI and DINO-KAT dMRI algorithms were simulated with an $\ell_0$ sparsity penalty and a $320 \times 320$ dictionary. OptShrink-based $x_L$ updates were employed in LASSI for the cardiac perfusion data, and SVT-based updates were used in the other cases. For the cardiac perfusion data, the initial $x_L$ and $x_S$ in LASSI were from the L+S framework

| Undersampling | 5x | 6x | 7x | 9x | 14x | 27x |
|---|---|---|---|---|---|---|
| NRMSE (k-t SLR) % | 9.7 | 10.7 | 12.2 | 14.5 | 18.0 | 23.7 |
| NRMSE (L+S) % | 11.7 | 12.8 | 14.2 | 16.3 | 19.6 | 25.4 |
| NRMSE (DINO-KAT) % | 8.6 | 9.5 | 10.7 | 12.6 | 15.9 | 21.8 |
| NRMSE (LASSI) % | **8.4** | **9.1** | **10.1** | **11.4** | **13.6** | **18.3** |
| Gain over k-t SLR (dB) | 1.2 | 1.4 | 1.7 | 2.1 | 2.4 | 2.2 |
| Gain over L+S (dB) | 2.8 | 2.9 | 3.0 | 3.1 | 3.2 | 2.8 |
| Gain over DINO-KAT (dB) | 0.2 | 0.3 | 0.6 | 0.9 | 1.4 | 1.5 |

Table 8.2: NRMSE values expressed as percentages for the L+S [1], k-t SLR [2], and the proposed DINO-KAT dMRI and LASSI methods at several undersampling factors for the PINCAT data [2,5] with pseudo-radial sampling. The best NRMSE values for each undersampling factor are marked in bold.

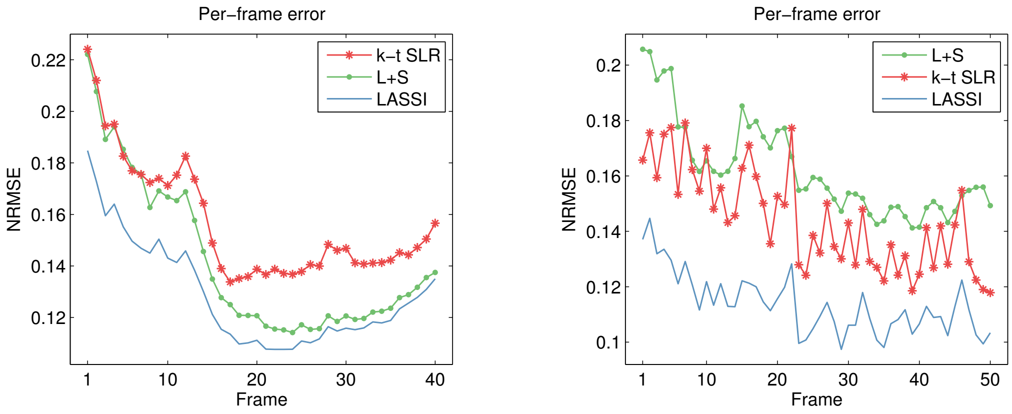
[1] (and the initial $x$ in DINO-KAT dMRI was an L+S dMRI reconstruction). For the PINCAT and *in vivo* myocardial perfusion data, the initial $x_S$ in LASSI (or $x$ in DINO-KAT dMRI) was the (better) k-t SLR reconstruction and the initial $x_L$ was zero. All other settings are as discussed in Section 8.4.1.

Tables 8.1, 8.2 and 8.3 list the reconstruction NRMSE values for LASSI, DINO-KAT dMRI, L+S [1] and k-t SLR [2] for the cardiac perfusion, PINCAT, and *in vivo* datasets, respectively. The LASSI method provides the best NRMSE values, and the proposed DINO-KAT dMRI method also outperforms the prior L+S and k-t SLR methods. The NRMSE gains achieved by LASSI over the other methods are indicated in the tables for each dataset and undersampling factor. The LASSI framework provides an average improvement of 1.9 dB, 1.5 dB, and 0.5 dB respectively, over the L+S, k-t SLR, and (proposed) DINO-KAT dMRI methods. This suggests the suitability of the richer LASSI model for dynamic image sequences compared to the jointly low-rank and sparse (k-t SLR), low-rank plus non-adaptive sparse (L+S), and purely adaptive dictionary-sparse (DINO-KAT dMRI) signal models.

Figure 8.3 shows the NRMSE values computed between each reconstructed and reference frame for the LASSI, L+S, and k-t SLR outputs for two datasets. The proposed LASSI scheme clearly outperforms the previous L+S and k-t SLR methods across frames (time). Figure 8.4 shows the LASSI reconstructions of some representative frames for each dataset in Tables 8.1-8.3. The reconstructed frames are visually similar to the reference frames (fully sampled reconstructions) shown. Figure 8.4 also shows the reconstruction error maps (i.e., the magnitude of the difference between the magnitudes of the reconstructed and reference frames) for LASSI, L+S, and k-t SLR for the representative frames of each dataset. The error maps for LASSI show

| Undersampling | 4x | 5x | 6x | 8x | 12x | 23x |
|---|---|---|---|---|---|---|
| NRMSE (k-t SLR) % | 10.7 | 11.6 | 12.7 | 14.0 | 16.7 | 22.1 |
| NRMSE (L+S) % | 12.5 | 13.4 | 14.6 | 16.1 | 18.8 | 24.2 |
| NRMSE (DINO-KAT) % | 10.2 | 11.0 | 12.1 | 13.5 | 16.4 | 21.9 |
| NRMSE (LASSI) % | **9.9** | **10.7** | **11.8** | **13.2** | **16.2** | **21.9** |
| Gain over k-t SLR (dB) | 0.7 | 0.7 | 0.6 | 0.5 | 0.3 | 0.1 |
| Gain over L+S (dB) | 2.1 | 2.0 | 1.8 | 1.7 | 1.3 | 0.9 |
| Gain over DINO-KAT (dB) | 0.3 | 0.3 | 0.2 | 0.2 | 0.1 | 0.0 |

Table 8.3: NRMSE values expressed as percentages for the L+S [1], k-t SLR [2], and the proposed DINO-KAT dMRI and LASSI methods at several undersampling factors for the myocardial perfusion MRI data in [2, 5], using pseudo-radial sampling. The best NRMSE values for each undersampling factor are marked in bold.



(a) (b)

Figure 8.3: NRMSE values computed between each reconstructed and reference frame for LASSI, L+S, and k-t SLR for (a) the cardiac perfusion data [1, 4] at 8x undersampling, and (b) the PINCAT data at 9x undersampling.

fewer artifacts and smaller distortions than the other methods.

Figure 8.5 shows reconstruction results for the PINCAT data [2, 5] with pseudo-radial sampling and nine fold undersampling. The time series ($x - t$) plots, which correspond to the line marked in green on a reference PINCAT frame (Figure 8.5), are shown for the reference, LASSI, DINO-KAT dMRI, L+S [1], and k-t SLR [2] reconstructions. The NRMSE values computed between the reconstructed and reference $x-t$ slices are also shown. The reconstruction for LASSI has lower NRMSE and clearly shows fewer artifacts and distortions (with respect to the reference) compared to the L+S and k-t SLR results. The LASSI result is also better than the DINO-KAT dMRI reconstruction that shows more smoothing (blur) effects (particularly in the

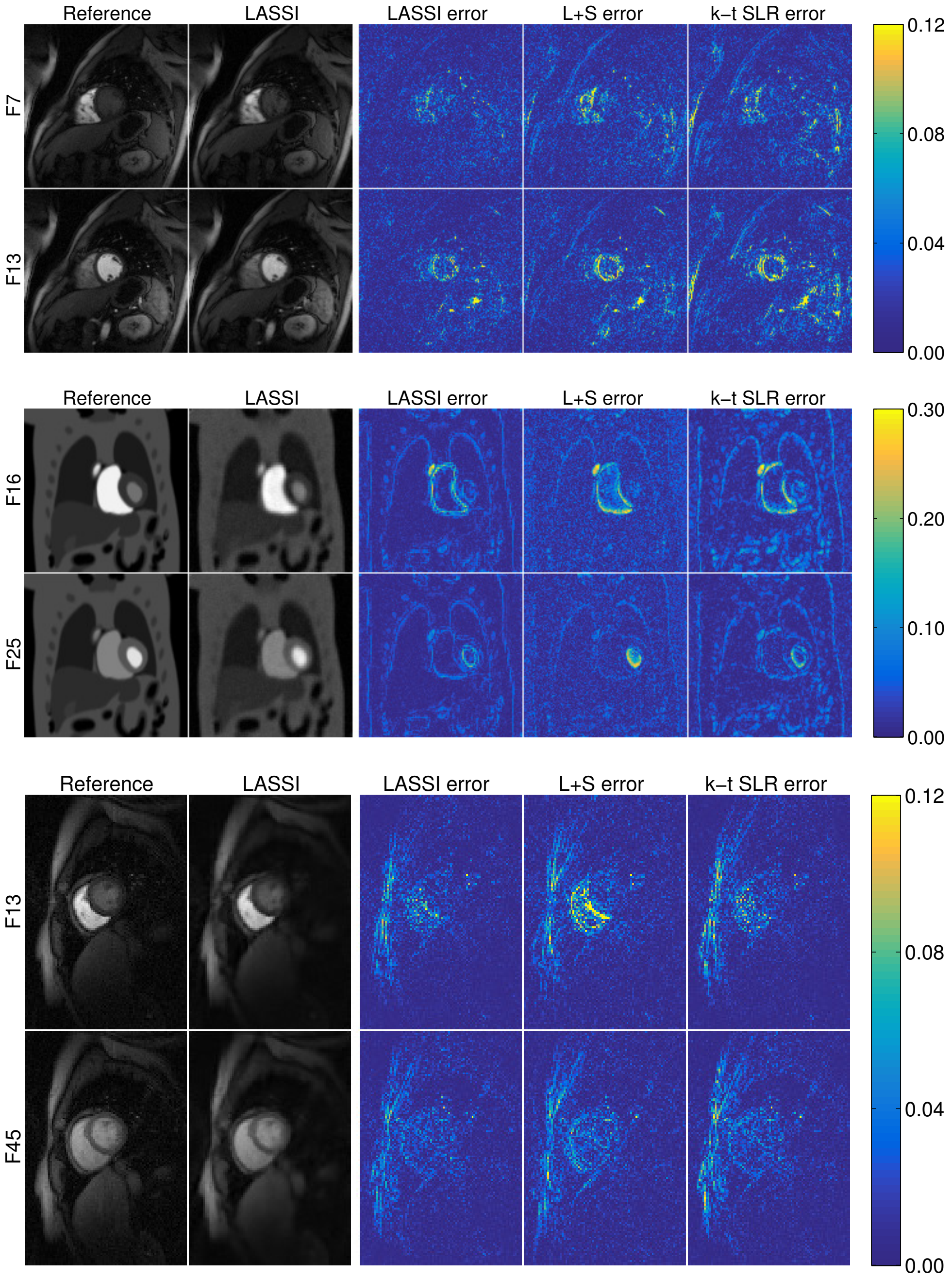


Figure 8.4: LASSI reconstructions and the error maps (clipped for viewing) for LASSI, L+S, and k-t SLR for frames of the cardiac perfusion data [1,4] (first row), PINCAT data [2,5] (second row), and *in vivo* myocardial perfusion data [2,5] (third row), shown along with the reference reconstruction frames. Undersampling factors (top to bottom): 8x, 9x, and 8x. The frame numbers and method names are indicated on the images.

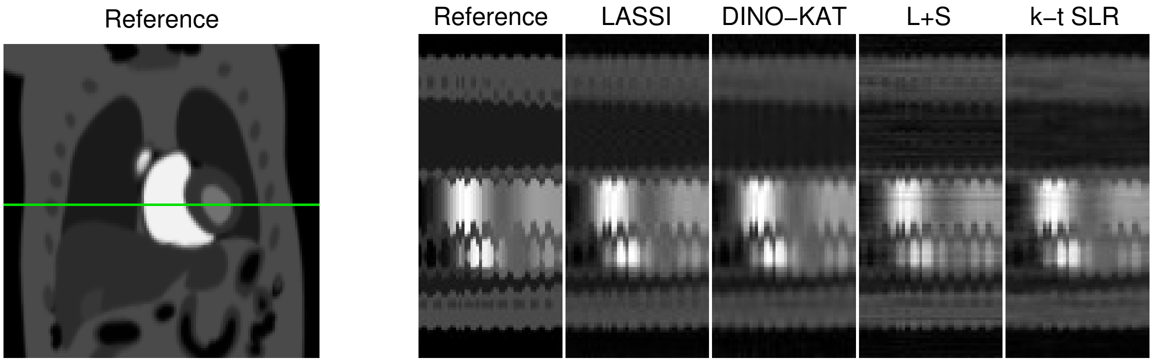


Figure 8.5: A frame of the reference PINCAT [2, 5] reconstruction is shown (left) with a spatial line cross section marked in green. The temporal ($x - t$) profiles of that line are shown for the reference, LASSI, DINO-KAT dMRI, L+S [1], and k-t SLR [2] reconstructions for pseudo-radial sampling and nine fold undersampling. The NRMSE values computed between the reconstructed and reference $x - t$ profiles are 0.107, 0.116 , 0.153, and 0.131 respectively, for LASSI, DINO-KAT dMRI, L+S, and k-t SLR.

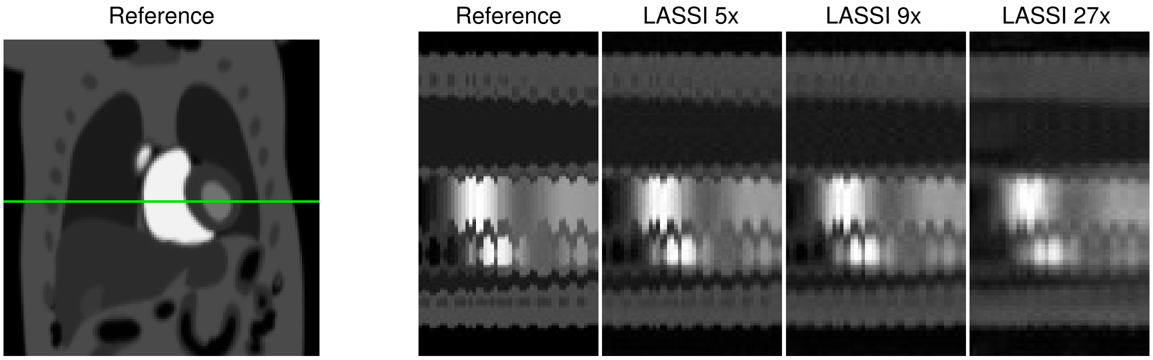


Figure 8.6: A frame of the reference PINCAT [2,5] reconstruction is shown (left) with a spatial line cross section marked in green. The temporal ($x - t$) profiles of that line are shown for the reference, and the LASSI reconstructions at 5x, 9x, and 27x undersampling and pseudo-radial sampling.

top and bottom portions of the $x - t$ map).

Figure 8.6 shows time series ($x - t$) plots for the LASSI reconstructions of the PINCAT data at several undersampling factors. At an undersampling factor of 27x, the LASSI result shows temporal smoothing. Nevertheless, LASSI still reconstructs many features well, despite the high undersampling. Figure 8.7 shows the LASSI reconstructions and reconstruction error maps for some representative frames of the cardiac perfusion data [1, 4], at several undersampling factors. Notably, even at high

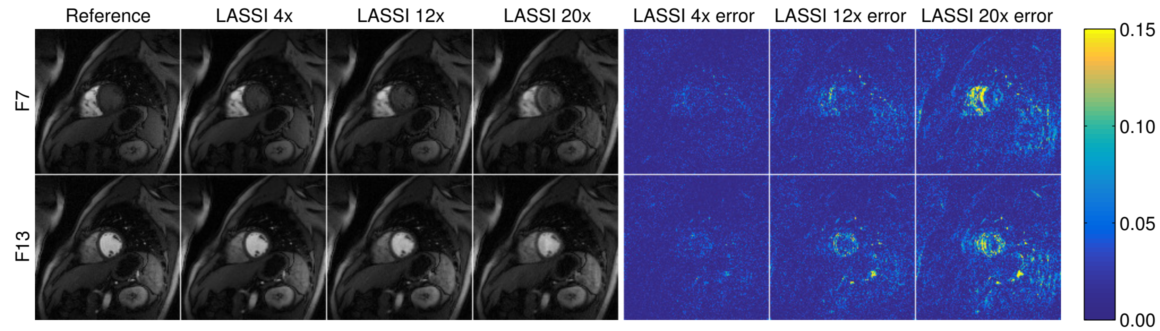


Figure 8.7: LASSI reconstructions and error maps (clipped for viewing) for frames of the cardiac perfusion data [1, 4] at 4x, 12x, and 20x undersampling (Cartesian sampling), shown along with the reference reconstruction frames. The images are labeled with the frame numbers and undersampling factors.

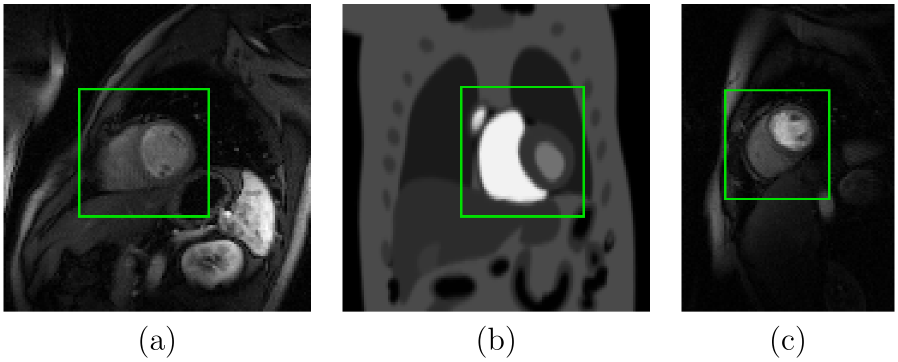


Figure 8.8: Regions of interest containing the heart shown using green bounding boxes for a frame of (a) the cardiac perfusion data [1], (b) PINCAT data [2, 5], and (c) *in vivo* myocardial perfusion MRI data [2, 5], respectively.

undersampling factors, LASSI still accurately reconstructs many image features.

### 8.4.4 Dynamic MRI Results over Heart ROIs

We also report the NRMSE of the dynamic MRI reconstructions from Section 8.4.3 computed over specific regions of interest (ROIs) containing the heart. Figure 8.8 shows the ROIs (as a rectangular box in a frame) for the cardiac perfusion data [1,4], PINCAT data [2,5], and *in vivo* myocardial perfusion MRI data [2,5]. Tables 8.4, 8.5, and 8.6 list the NRMSE values computed over these ROIs for the LASSI, DINO-KAT dMRI, L+S [1], and k-t SLR [2] reconstructions at several undersampling factors. The various methods tend to provide even better reconstruction quality (i.e., NRMSE) within the specific ROIs than over the entire images (cf. Section 8.4.3). Tables 8.4-

| Undersampling | 4x | 8x | 12x | 16x | 20x | 24x |
|---|---|---|---|---|---|---|
| NRMSE (k-t SLR) % | 10.4 | 14.2 | 17.2 | 19.5 | 22.4 | 24.2 |
| NRMSE (L+S) % | 10.7 | 14.0 | 16.3 | 18.8 | 22.2 | 24.1 |
| NRMSE (DINO-KAT) % | 9.8 | **12.5** | **14.2** | **16.4** | 19.1 | 21.2 |
| NRMSE (LASSI) % | **9.7** | 12.7 | 14.4 | 16.7 | **18.3** | **20.1** |
| Gain over k-t SLR (dB) | 0.6 | 0.9 | 1.5 | 1.4 | 1.8 | 1.6 |
| Gain over L+S (dB) | 0.8 | 0.9 | 1.1 | 1.1 | 1.6 | 1.6 |
| Gain over DINO-KAT (dB) | 0.1 | -0.1 | -0.1 | -0.1 | 0.4 | 0.5 |

Table 8.4: NRMSE values for an ROI (Figure 8.8(a)) in the cardiac perfusion data [1] expressed as percentages for the L+S [1], k-t SLR [2], and the proposed DINO-KAT dMRI and LASSI methods at several undersampling factors and Cartesian sampling. The best NRMSE value at each undersampling factor is indicated in bold.

| Undersampling | 5x | 6x | 7x | 9x | 14x | 27x |
|---|---|---|---|---|---|---|
| NRMSE (k-t SLR) % | 8.7 | 9.6 | 11.1 | 13.2 | 16.7 | 22.8 |
| NRMSE (L+S) % | 11.1 | 12.0 | 13.2 | 15.0 | 18.1 | 23.9 |
| NRMSE (DINO-KAT) % | 8.2 | 8.9 | 10.1 | 11.6 | 14.6 | 20.6 |
| NRMSE (LASSI) % | **8.0** | **8.7** | **9.6** | **10.9** | **13.2** | **18.3** |
| Gain over k-t SLR (dB) | 0.7 | 0.9 | 1.2 | 1.6 | 2.1 | 1.9 |
| Gain over L+S (dB) | 2.9 | 2.8 | 2.8 | 2.8 | 2.8 | 2.3 |
| Gain over DINO-KAT (dB) | 0.2 | 0.3 | 0.4 | 0.5 | 0.9 | 1.0 |

Table 8.5: NRMSE values for an ROI (Figure 8.8(b)) in the PINCAT data [2, 5] expressed as percentages for the L+S [1], k-t SLR [2], and the proposed DINO-KAT dMRI and LASSI methods at several undersampling factors and pseudo-radial sampling. The best NRMSE value at each undersampling factor is indicated in bold.

| Undersampling | 4x | 5x | 6x | 8x | 12x | 23x |
|---|---|---|---|---|---|---|
| NRMSE (k-t SLR) % | 7.6 | 8.3 | 9.2 | 10.4 | 12.4 | 17.1 |
| NRMSE (L+S) % | 9.2 | 10.0 | 11.0 | 12.3 | 14.5 | 18.9 |
| NRMSE (DINO-KAT) % | 7.1 | 7.8 | 8.7 | 10.0 | 12.0 | 16.8 |
| NRMSE (LASSI) % | **6.8** | **7.5** | **8.4** | **9.7** | **11.8** | **16.8** |
| Gain over k-t SLR (dB) | 0.9 | 0.9 | 0.8 | 0.6 | 0.4 | 0.2 |
| Gain over L+S (dB) | 2.6 | 2.5 | 2.3 | 2.1 | 1.8 | 1.0 |
| Gain over DINO-KAT (dB) | 0.4 | 0.4 | 0.3 | 0.2 | 0.1 | 0.0 |

Table 8.6: NRMSE values for an ROI (Figure 8.8(c)) in the myocardial perfusion MRI data [2, 5] expressed as percentages for the L+S [1], k-t SLR [2], and the proposed DINO-KAT dMRI and LASSI methods at several undersampling factors and pseudo-radial sampling. The best NRMSE value at each undersampling factor is indicated in bold.

8.6 also indicate the NRMSE gains achieved by LASSI over the other methods for each dataset and undersampling factor. The proposed LASSI and DINO-KAT dMRI methods provide much lower NRMSE in the heart ROIs compared to the previous L+S and k-t SLR methods. The LASSI scheme also outperforms DINO-KAT dMRI in most cases, and provides an average improvement within the ROIs of 2.0 dB, 1.1 dB, and 0.3 dB respectively, over the L+S, k-t SLR, and the proposed DINO-KAT dMRI methods.

### 8.4.5 A Study of Various LASSI Models and Methods

Here, we investigate the various LASSI models and methods in detail. We work with the cardiac perfusion data [1] and simulate the reconstruction performance of LASSI for Cartesian sampling at various undersampling factors. Unless otherwise stated, we simulate LASSI here with the $\ell_0$ sparsity penalty, the SVT-based $x_L$ update, $r = 1$, an initial $320 \times 320$ (1D) DCT dictionary, and $x_S$ initialized with the dMRI reconstruction from the L+S method [1] and $x_L$ initialized to zero. In the following, we first compare SVT-based updating of $x_L$ to alternatives in the algorithms and the use of $\ell_0$ versus $\ell_1$ sparsity penalties. The weights $\lambda_L$, $\lambda_S$, and $\lambda_Z$ were tuned for each LASSI variation. Second, we study the behavior of LASSI for different initializations of the underlying signal components or dictionary. Third, we study the effect of the number of atoms of $D$ on LASSI performance. Fourth, we study the effect of the sparsity level of the learned $Z$ on the reconstruction quality in LASSI. Lastly, we study the effect of the atom rank parameter $r$ in LASSI.

#### 8.4.5.1 SVT vs. Alternatives and $\ell_0$ vs. $\ell_1$ patch sparsity

Figs. 8.9(a) and (b) show the behavior of the LASSI algorithms using $\ell_0$ and $\ell_1$ sparsity penalties, respectively. In each case, the results obtained with $x_L$ updates based on SVT, OptShrink (OPT), or based on the Schatten $p$-norm ($p = 0.5$), and rank penalty are shown. The OptShrink-based singular value shrinkage (with $r_L = 1$) and Schatten $p$-norm-based shrinkage typically outperform the conventional SVT (based on nuclear norm penalty) as well as the hard thresholding of singular values (for rank penalty) for the cardiac perfusion data. The OptShrink and Schatten $p$-norm-based $x_L$ updates also perform quite similarly at lower undersampling factors, but OptShrink outperforms the latter approach at higher undersampling factors. Moreover, the $\ell_0$ "norm"-based methods outperformed the corresponding $\ell_1$ norm methods in many cases (with SVT or alternative approaches). These results

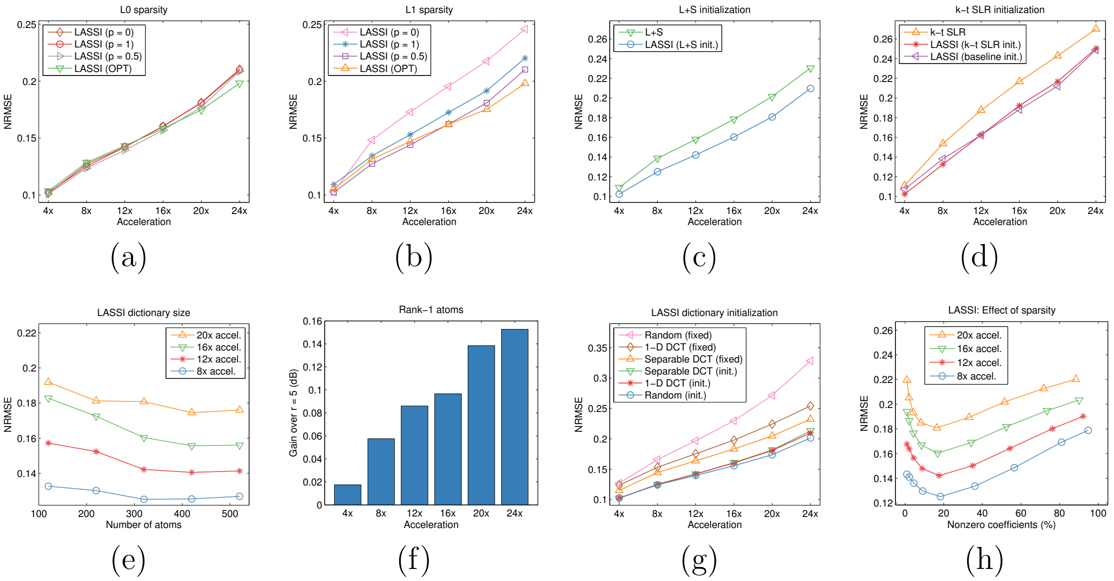


Figure 8.9: Study of LASSI models, methods, and initializations at various undersampling factors for the cardiac perfusion data in [1, 4] with Cartesian sampling: (a) NRMSE for LASSI with $\ell_0$ "norm" for sparsity and with $x_L$ updates based on SVT ($p = 1$), OptShrink (OPT), or based on the Schatten $p$-norm ($p = 0.5$) or rank penalty ($p = 0$); (b) NRMSE for LASSI with $\ell_1$ sparsity and with $x_L$ updates based on SVT ($p = 1$), OptShrink (OPT), or based on the Schatten $p$-norm ($p = 0.5$) or rank penalty ($p = 0$); (c) NRMSE for LASSI when initialized with the output of the L+S method [1] (used to initialize $x_S$ with $x_L^0 = 0$) together with the NRMSE for the L+S method; (d) NRMSE for LASSI when initialized with the output of the k-t SLR method [2] or with the baseline reconstruction (performing zeroth order interpolation at the nonsampled k-t space locations and then backpropagating to image space) mentioned in Section 8.4.1 (these are used to initialize $x_S$ with $x_L^0 = 0$), together with the NRMSE values for k-t SLR; (e) NRMSE versus dictionary size at different acceleration factors; (f) NRMSE improvement (in dB) achieved with $r = 1$ compared to the $r = 5$ case in LASSI; (g) NRMSE for LASSI with different dictionary initializations (a random dictionary, a $320 \times 320$ 1D DCT and a separable 3D DCT of the same size) together with the NRMSEs achieved in LASSI when the dictionary is fixed to its initial value; and (h) NRMSE versus the fraction of nonzero coefficients (expressed as percentage) in the learned $Z$ at different acceleration factors.

demonstrate the benefits of appropriate nonconvex regularizers in practice.

#### 8.4.5.2 Effect of Initializations

Here, we explore the behavior of LASSI for different initializations of the dictionary and the dynamic signal components. First, we consider the LASSI algorithm initialized by the L+S and k-t SLR methods as well as with the baseline reconstruction (obtained by performing zeroth order interpolation at the nonsampled k-t space locations and then backpropagating to image space) mentioned in Section 8.4.1 (all other parameters fixed). The reconstructions from the prior methods are used to initialize $x_S$ in LASSI with $x_L^0 = 0$[8]. Figs. 8.9(c) and (d) show that LASSI significantly improves the dMRI reconstruction quality compared to the initializations at all undersampling factors tested. The baseline reconstructions had high NRMSE values (not shown in Figure 8.9) of about 0.5. Importantly, the reconstruction NRMSE for LASSI with the simple baseline initialization (Figure 8.9(d)) is comparable to the NRMSE obtained with the more sophisticated k-t SLR initialization. In general, better initializations (for $x_L$, $x_S$) in LASSI may lead to a better final NRMSE in practice.

Next, we consider initializing the LASSI method with the following types of dictionaries (all other parameters fixed): a random i.i.d. gaussian matrix with normalized columns, the $320 \times 320$ 1D DCT, and the separable 3D DCT of size $320 \times 320$. Figure 8.9(g) shows that LASSI performs well for each choice of initialization. We also simulated the LASSI algorithm by keeping the dictionary $D$ fixed (but still updating $Z$) to each of the aforementioned initializations. Importantly, the NRMSE values achieved by the adaptive-dictionary LASSI variations are substantially better than the values achieved by the fixed-dictionary schemes.

#### 8.4.5.3 Effect of Overcompleteness of $D$

Figure 8.9(e) shows the performance (NRMSE) of LASSI for various choices of the number of atoms ($K$) in $D$ at several acceleration factors. The weights in (P1) were tuned for each $K$. As $K$ is increased, the NRMSE initially shows significant improvements (decrease) of more than 1 dB. This is because LASSI learns richer models that provide sparser representations of patches and, hence, better reconstructions. However, for very large $K$ values, the NRMSE saturates or begins to degrade, since it is harder to learn very rich models using limited imaging measurements (without overfitting artifacts).

#### 8.4.5.4 Effect of the Sparsity Level in LASSI

While Section 8.4.5.1 compared the various ways of updating the low-rank signal component in LASSI, here we study the effect of the sparsity level of the learned

---

[8] We have also observed that LASSI improves the reconstruction quality over other alternative initializations such as initializing $x_L$ and $x_S$ using corresponding outputs of the L+S framework.

$Z$ on LASSI performance. In particular, we simulate LASSI at various values of the parameter $\lambda_Z$ that controls sparsity (all other parameters fixed). Figure 8.9(h) shows the NRMSE of LASSI at various sparsity levels of the learned $Z$ and at several acceleration factors. The weight $\lambda_Z$ decreases from left to right in the plot and the same set of $\lambda_Z$ values were selected (for the simulation) at the various acceleration factors. Clearly, the best NRMSE values occur around 10-20% sparsity (when 32-64 dictionary atoms are used on the average to represent the reshaped $64 \times 5$ space-time patches of $x_S$), and the NRMSE degrades when the number of nonzeros in $Z$ is either too high (non-sparse) or too low (when the dictionary model reduces to a low-rank approximation of space-time patches in $x_S$). This illustrates the effectiveness of the rich sparsity-driven modeling in LASSI[9].

#### 8.4.5.5 Effect of Rank of Reshaped Atoms

Here, we simulate LASSI with (reshaped) atom ranks $r = 1$ (low-rank) and $r = 5$ (full-rank). Figure 8.9(f) shows that LASSI with $r = 1$ provides somewhat improved NRMSE values over the $r = 5$ case at several undersampling factors, with larger improvements at higher accelerations. This result suggests that structured (fewer degrees of freedom) dictionary adaptation may be useful in scenarios involving very limited measurements. In practice, the effectiveness of the low-rank model for reshaped dictionary atoms also depends on the properties of the underlying data.

### 8.4.6 Dictionary Learning for Representing Dynamic Image Patches

Here, we present results on the effectiveness of learned (SOUP) dictionaries for representing dynamic image data. In particular, we compare dictionary learning with low-rank atom constraints to learning without such constraints. We extract the $8 \times 8 \times 5$ overlapping spatiotemporal patches of the fully sampled cardiac perfusion data [1], with a spatial and temporal patch overlap stride of 2 pixels. The vectorized 3D patches are then stacked as columns of the training matrix $P$, and we solve Problem (P3) to learn the approximation $DC^H$ for $P$. Dictionaries of size $320 \times 320$ (with atoms reshaped into $64 \times 5$ matrices) were learned for various values of the $\ell_0$ sparsity penalty parameter $\lambda_Z$ and for $r = 1, 2, 3, 4,$ and 5. The block coordinate

---

[9] Figure 8.9(h) shows that the same $\lambda_Z$ value is optimal at various accelerations. An intuitive explanation for this is that as the undersampling factor increases, the weighting of the (first) data-fidelity term in (P1) or (P2) decreases (fewer k-t space samples, or rows of the sensing matrix are selected). Thus, even with fixed $\lambda_Z$, the relative weighting of the sparsity penalty would increase, creating a stronger sparsity regularization at higher undersampling factors.

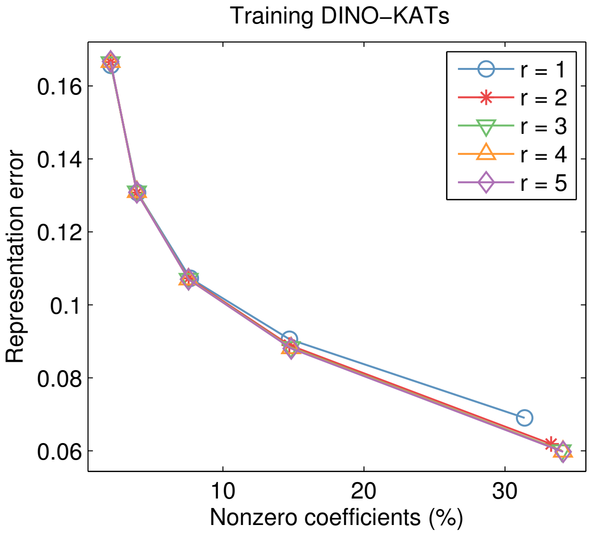


Figure 8.10: The normalized sparse representation error (NSRE) $\|Y - DC^H\|_F / \|Y\|_F$ for the $320 \times 320$ dictionaries learned on the $8 \times 8 \times 5$ overlapping spatiotemporal patches of the fully sampled cardiac perfusion data [1]. The results are shown for various choices of the $\ell_0$ sparsity penalty parameter $\lambda_Z$ corresponding to different fractions of nonzero coefficients in the learned $C$ and for various choices of the atom rank parameter $r$.

descent learning method ran for 50 iterations and was initialized with $C = 0$ and a $320 \times 320$ DCT.

The quality of the learned data approximations was measured using the normalized sparse representation error (NSRE) given as $\|Y - DC^H\|_F / \|Y\|_F$. Figure 8.10 shows the NSRE for various choices of $\lambda_Z$ corresponding to different fractions of nonzero coefficients in the learned $C$ and for various choices of the reshaped atom rank $r$. The learned dictionaries achieved small NSRE values together with sparse coefficients $C$. Importantly, the learned dictionaries with low-rank ($r < 5$) reshaped atoms represented the spatiotemporal patches about as well as the learned dictionaries with full-rank ($r = 5$) atoms. Thus, the low-rank model on reshaped dictionary atoms, although a constrained model, effectively captures the properties of dynamic image patches.

## 8.5 Conclusions

In this chapter, we investigated a novel framework for reconstructing spatiotemporal data from limited measurements. The proposed LASSI framework jointly learns a low-rank and dictionary-sparse decomposition of the underlying dynamic image sequence together with a spatiotemporal dictionary. The proposed algorithms involve simple updates. Our experimental results showed the superior performance of LASSI methods for dynamic MR image reconstruction from limited k-t space data compared to recent works such as L+S and k-t SLR. The LASSI framework also outperformed the proposed efficient dictionary-blind compressed sensing framework (a special case of LASSI) called DINO-KAT dMRI. We also studied and compared various LASSI methods and formulations such as with $\ell_0$ or $\ell_1$ sparsity penalties, or with low-rank or full-rank reshaped dictionary atoms, or involving singular value thresholding-based optimization versus some alternatives including OptShrink-based optimization. The usefulness of LASSI-based schemes in other inverse problems and image processing applications merits further study. The LASSI schemes involve parameters (like in most regularization-based methods) that need to be set (or tuned) in practice. We also leave the study of automating the parameter selection process to future work.

# CHAPTER IX

# Robust Photometric Stereo via Dictionary Learning

## 9.1 Introduction

Photometric stereo [179] is a method that seeks to reconstruct the normal vectors of an object from a set of images of the object illuminated under different light sources. Concretely, we have images $I_1, \ldots, I_d$ of the three-dimensional object and, in each image, the object is illuminated by a (distant) light source with light incident on the object in directions $\ell_1, \ldots, \ell_d \in \mathbb{R}^3$. Given $I_1, \ldots, I_d$ and $\ell_1, \ldots, \ell_d$, the goal is to estimate the normal vector map of the object, which can be numerically integrated to obtain a three-dimensional representation of the object. The appeal of photometric stereo is its simplicity: it requires only a camera and a movable light source to generate a three-dimensional representation of an object.

### 9.1.1 Background

Since its introduction by Woodham [179], significant work has been performed to increase the generality and robustness of photometric stereo [180–193]. This body of work typically seeks to weaken one of two underlying assumptions in Woodham's original model: that the position of the object relative to the position of the light source is known or that the object follows the Lambertian reflectance model. Works addressing the first assumption are solving the so-called *uncalibrated photometric stereo* problem, and they generally attempt to estimate the normal vectors of the object without any knowledge of the lighting directions [194–203]. Works addressing the latter assumption generally attempt to either reconstruct the normal vectors of objects whose reflectance properties deviate from the Lambertian model, or they try to develop methods that are robust to corruptions in the observed images. In this chapter, our focus is primarily on the design of robust photometric stereo algorithms, but we also incorporate a non-Lambertian model for increased generality.

The Lambertian reflectance model holds that the intensity of light reflected by a point on a surface is linearly proportional to the inner product of the direction of illumination and the normal vector of the surface at that point [179]. Given a set of images of a Lambertian object illuminated under several (known) lighting directions, a simple system of equations can be solved to determine the normal vector at each point on the surface. In practice, while this is a reasonable model for some objects, the reflectance properties of many real-world objects differ significantly from the Lambertian model. Furthermore, shadows, specularities, and other non-idealities can cause additional deviations from linear reflectance. Performing classical photometric stereo on such non-Lambertian data typically yields large errors in the estimated normal vectors. As such, developing photometric stereo methods for objects that are inherently non-Lambertian and improving robustness to other imperfections in the data are essential to extending the applicability and accuracy of photometric stereo.

Two primary approaches have found success addressing these problems. Several works assume the Lambertian model is fundamentally correct and seek to account for deviations from the model through explicit outlier removal [186, 189]—often assuming that non-idealities are sparse. While achieving some level of success, these approaches can place overly restrictive assumptions on the data, which may result in falsely rejecting useful data as outliers, and they make no attempt to model the true reflectance properties of objects. In turn, other works propose more complex reflectance models that enable non-Lambertian photometric stereo [191–193]. Approaches in this class are able to accurately model a wider range of objects, but they still break down when their modeling assumptions fail. Furthermore, they often fail when the data contains corruptions not accounted for by their reflectance models. In addition to the aforementioned difficulties, state-of-the-art methods in both categories typically rely on a large number of images to accurately estimate the normal vectors, which may be infeasible to gather in practice.

### 9.1.2 Contributions

In this chapter, we propose a novel approach to photometric stereo that relies on dictionary learning [100, 104] to robustly handle a wide range of non-idealities in the data. Dictionary learning seeks to represent local patches of the data as sparse with respect to a learned collection of atoms. Such models effectively act as dynamic regularization that adapts to the underlying structure of the data and removes spurious corruptions. Inspired by recent successes applying dictionary learning to a variety of imaging problems [12, 16, 17, 103], we adopt this methodology to improve the robustness of photometric stereo. Our approach is data-driven and adapts to the underlying structure of the data without imposing additional explicit constraints. Furthermore, we incorporate an existing non-Lambertian reflectance model into our method to better handle non-Lambertian surfaces. In total we present three dictionary learning-based formulations of robust photometric stereo. We investigate the performance of each method in a variety of different scenarios. In particular, we evaluate their performance on the benchmark DiLiGenT dataset [204] and their ability to handle general, non-sparse corruptions.

### 9.1.3 Organization

The remainder of this chapter is organized as follows. In Section 9.2, we provide a brief overview of related works on photometric stereo. In Section 9.3, we carefully define the photometric stereo problem and the non-Lambertian reflectance model we will incorporate into our method. We present our proposed dictionary learning-based methods in Section 9.4, and in Section 9.5 we present the associated algorithms for solving them. Finally, Section 9.6 provides an extensive numerical study of the performance of our proposed methods compared to state-of-the-art methods.

## 9.2 Related Work

Lambertian photometric stereo was originally proposed by Woodham [179] in 1980. Since then, much work has been done extending it to more general settings where the Lambertian model does not hold exactly. This body of work has generally proceeded by either treating non-Lambertian effects as outliers or directly accounting for non-Lambertian effects in the reflectance model.

A variety of approaches have been proposed that perform robust photometric stereo via outlier rejection. In general, these methods assume that the data is inherently Lambertian, seek to isolate non-Lambertian effects as outliers, and then reject the outliers to increase the accuracy of the computed normal vectors. Early works in robust photometric stereo—typically referred to as *four source photometric stereo*—utilized four images to identify and reject specularities [180, 205, 206]. Recently, more complex methods have been developed that rely on maximum likelihood estimation [207], expectation maximization [184], and a maximum feasible subsystem framework [208]. Other approaches include a graph cuts-based algorithm to identify shadows [209], a method that seeks to map color images into a two-dimensional subspace invariant to specularities [210], and several methods that utilize RANSAC-based algorithms [211, 212] to detect and reject outliers.

The most recent works on robust photometric stereo via outlier rejection—and the current state-of-the-art in this area—are those by Wu et al. [186] and Ikehata et al. [189]. These works rely on the observation that images of a Lambertian object lie in a three-dimensional subspace. This observation, together with modeling non-Lambertian effects as sparse corruptions to the underlying Lambertian data, motivates the authors to propose rank minimization-based approaches, which are shown to effectively separate the Lambertian portion of the data from non-Lambertian effects.

Regardless of their robustness to outliers, approaches that rely on the Lambertian reflection model as the underlying model of the data are inherently limited in scope due to the wide variety of non-Lambertian surfaces that exist in the real world. As a result, another body of work has been incorporating more general reflectance models into photometric stereo [213]. For example, uncalibrated photometric stereo based on the Torrance and Sparrow reflectance model has been proposed [197] as well as calibrated photometric stereo based on the Ward reflectance model [183, 187, 214].

A large amount of work has also been done developing photometric stereo algorithms that incorporate reflectance models based on general reflectance properties exhibited by materials. In particular, the property of isotropy has been successfully utilized in a variety of works [188, 190, 215–217]. The current state-of-the-art in this category are the works of Shi et al. [193] and Ikehata et al. [192]. Ikehata et al. models the reflectance function using a sum-of-lobes representation [185], utilizing Bernstein polynomials as a basis for the inverse reflectance function and performing bivariate regression to determine the normal vectors. Shi et al. instead models the low-frequency reflectance component using polynomials of up to order three while discarding the high-frequency reflectance components. Of particular interest in this chapter is another work by Ikehata et al. [191] that models the reflectance function as piecewise-linear. We explore this method in more detail in the following section.

In addition to the aforementioned approaches, a variety of other robust photometric stereo methods have been proposed [181, 182, 218, 219]. The recent work of Shi et al. [204] seeks to standardize future work in photometric stereo by introducing an extensive dataset to facilitate future testing and evaluation. Furthermore, they compare a variety of existing approaches on this dataset, providing a benchmark for future work.

## 9.3 Problem Formulation

### 9.3.1 Basis of Photometric Stereo

The Lambertian reflectance model states that, to an observer, the brightness of a point on a Lambertian surface is independent of the observer's viewing angle. Surfaces that follow this model are matte in appearance. Indeed, consider an image taken of a Lambertian object. The light intensity measured at pixel $(x, y)$ of the image satisfies the relationship

$$I_{xy} = \rho_{xy} \ell^T n_{xy}, \tag{9.1}$$

where $I_{xy}$ is the image intensity at pixel $(x, y)$, $\ell \in \mathbb{R}^3$ is the direction of the light source incident on the surface of the object, $\|\ell\|_2$ is the light source intensity, $n_{xy} \in \mathbb{R}^3$ is the (unit) normal vector of the surface at $(x, y)$, and $\rho_{xy} \in \mathbb{R}$ is the surface albedo at $(x, y)$—a measure of the reflectivity of the surface.

Suppose we fix the position of a camera facing the surface and vary the position of the light source over $d$ unique locations. Then we can write $d$ equations of the form (9.1) and stack them into the matrix equation

$$\begin{bmatrix} I_{xy}^1 \\ \vdots \\ I_{xy}^d \end{bmatrix} = \begin{bmatrix} \ell_1^T \\ \vdots \\ \ell_d^T \end{bmatrix} \rho_{xy} n_{xy}, \tag{9.2}$$

where $I_{xy}^k$ denotes the image intensity at $(x, y)$ in the $k$th image. Assuming each of our $d$ images has dimension $m_1 \times m_2$, (9.2) can be solved $m_1 m_2$ times to obtain the normal vector of the object at each point on the surface. We may also combine these $m_1 m_2$ equations into a single matrix equation. Indeed, define the observation matrix

$$Y \triangleq \left[\mathbf{vec}(I^1) \ \ldots \ \mathbf{vec}(I^d)\right] \in \mathbb{R}^{m_1 m_2 \times d}, \tag{9.3}$$

where $\mathbf{vec}(I^k)$ is the vector formed by stacking the columns of $I^k$. Then, assuming the light sources are ideal (i.e., the incident light rays are parallel and of equal intensity at each point on the surface), we can collect (9.2) into the single equation

$$Y = NL, \tag{9.4}$$

where $N = [\rho_{11} n_{11} \ \ldots \ \rho_{m_1 m_2} n_{m_1 m_2}]^T \in \mathbb{R}^{m_1 m_2 \times 3}$ and $L = [\ell_1 \ \ldots \ \ell_d] \in \mathbb{R}^{3 \times d}$. To avoid scaling ambiguity, we assume all light sources have intensity $\|\ell_k\|_2 = 1$.

Each normal vector $n_{xy}$ contains three unknown components. Thus, given $d \geq 3$ images and the corresponding light directions, we can solve (9.4) to obtain the normal vector at each point on the object. Once computed, we can integrate the normal vectors to produce a full three-dimensional model of our surface [220].

### 9.3.2 Deviations From the Lambertian Model

While the Lambertian reflectance model is a good approximation of the reflectance properties of some surfaces, it is a poor approximation for many real-world objects. Lambertian objects are matte in appearance and thus any non-matte objects necessarily deviate from the Lambertian reflectance model. The latter class includes any object that exhibits specularities—bright points observed when light reflects off a shiny surface. Furthermore, even if an object is Lambertian, shadows (both self-cast and those produced by other objects) can cause the Lambertian model to break down.

One approach to modeling these effects is to modify (9.4) to

$$Y = NL + E, \tag{9.5}$$

where $E$ is an additive error matrix accounting for non-Lambertian effects. Under this model, a simple, naive approach for estimating $N$ is to solve the least squares problem

$$\min_{N} \ \|Y - NL\|_F^2 \,, \tag{9.6}$$

which has solution $\hat{N} = YL^{\dagger}$, where $\dagger$ denotes the Moore-Penrose pseudoinverse. In this setting, one typically gathers $d > 3$ images so that the problem is overdetermined and thus provides some robustness to the non-Lambertian effects.

Several works apply further constraints to (9.5)—such as constraining $E$ to be *sparse*—allowing them to (in cases where their assumptions hold) derive more accurate estimates of $N$ than those obtained by (9.6) [186,189]. In Section 9.4, we propose two novel approaches that apply dictionary learning to the model (9.5).

### 9.3.3 Piecewise Linear Reflectance Model

Regardless of the constraints imposed on the additive error $E$, the model (9.5) fundamentally relies on the Lambertian reflectance model, thus limiting its generality. Recent works have sought to move beyond the Lambertian assumption and utilize more general reflectance models that can accurately model the normal vectors of a wider range of objects [191–193]. In this chapter, we are particularly interested in the model presented in [191], which we briefly summarize here.

A simple extension to the Lambertian model is to assume that the image intensity is related to the inner product of $\ell$ and $n$ through a nonlinear function. In other words, we modify (9.1) to read

$$I_{xy} = f_{xy}(\ell^T n_{xy}) \tag{9.7}$$

for some nonlinear function $f_{xy}$. Assuming the reflectance function at each pixel, $f_{xy}$, is monotonically increasing, a unique inverse is guaranteed to exist. We can thus invert (9.7) and write

$$f_{xy}^{-1}(I_{xy}) = g_{xy}(I_{xy}) = \ell^T n_{xy}. \tag{9.8}$$

Given a set of lighting vectors and corresponding images, our task is then to jointly estimate $g_{xy}(.)$ and $n_{xy}$ for each pixel. This is a highly underdetermined problem and so, to solve it in practice, further constraints must be imposed. A natural possibility is to assume $g_{xy}$ is piecewise linear. That is, we let

$$g_{xy}(t) = \sum_{k=1}^{p} a_{xy}^k g_{xy}^k(t), \tag{9.9}$$

where

$$g_{xy}^k(t) = \begin{cases} 0 & \text{if } t < b_{xy}^{k-1}, \\ t - b_{xy}^{k-1} & \text{if } b_{xy}^{k-1} \leq t \leq b_{xy}^k, \\ b_{xy}^k - b_{xy}^{k-1} & \text{if } t > b_{xy}^k. \end{cases} \tag{9.10}$$

Here, $p$ is a design parameter that determines the number of piecewise segments in $g_{xy}$, $b_{xy}^k$ are the inflection points of $g_{xy}$ (a strictly increasing sequence), and $a_{xy}^k > 0$ is the slope of each segment. For simplicity, we set $b_{xy}^0 = 0$ and choose the remaining values of $b_{xy}^k$ to be equally spaced along the range of intensity values among the $d$ images at pixel $(x, y)$. Under these assumptions, model (9.8) reduces to the problem of estimating the slopes $a_{xy}^1, \ldots, a_{xy}^p$ and the normal vector $n_{xy}$ at each pixel. Note that the case $a_{xy}^1 = \ldots = a_{xy}^p$ reduces to the Lambertian model (9.1).

To simplify notation, let

$$a_{xy} = [a_{xy}^1 \ \ldots \ a_{xy}^p]^T \in \mathbb{R}^p \tag{9.11}$$

and

$$\bar{g}_{xy}(t) = [g_{xy}^1(t) \ \ldots \ g_{xy}^p(t)]^T \in \mathbb{R}^p \tag{9.12}$$

and rewrite (9.9) as the vector product

$$g_{xy}(t) = \bar{g}_{xy}(t)^T a_{xy}. \tag{9.13}$$

Similarly, (9.8) can be written as

$$\bar{g}_{xy}(I_{xy})^T a_{xy} = \ell^T n_{xy}. \tag{9.14}$$

Given $d$ images, let $C_{xy} \in \mathbb{R}^{d \times p}$ be the matrix whose $j$th row is $\bar{g}_{xy}(I_{xy}^j)^T$. Then we can collect the data from the $d$ images at pixel $(x, y)$ into the single equation

$$C_{xy} a_{xy} = L^T n_{xy}, \tag{9.15}$$

which is the analogue of (9.2) for the Lambertian model. Equation (9.15) can be solved for $n_{xy}$ and $a_{xy}$ to determine the normal vector and the corresponding nonlinear reflectance function at $(x, y)$. To avoid scaling ambiguity, one can constrain $1^T a_{xy} = 1$ and then normalize $n_{xy}$ to unit norm after solving (9.15).

As in the Lambertian case, the model (9.8) only accounts for the reflectance properties of the object. While significantly more general than the Lambertian model, nonidealities present in the images that do not conform to these reflectance properties—or, more explicitly, that do not follow a piecewise linear relationship between $I_{xy}$ and $\ell^T n_{xy}$—will prevent (9.15) from holding exactly. Thus, analogous to (9.5), we modify (9.15) to yield the model

$$C_{xy} a_{xy} = L^T n_{xy} + e, \tag{9.16}$$

where $e \in \mathbb{R}^d$ accounts for any corruptions in the data not captured by the piecewise reflectance model. A simple approach to fiting model (9.16) to data is to solve the constrained least squares problem[1]

$$\begin{aligned} \min_{n_{xy},\ a_{xy}} \quad & \|C_{xy} a_{xy} - L^T n_{xy}\|_2^2 \\ \text{s.t.} \quad & 1^T a_{xy} = 1. \end{aligned} \tag{9.17}$$

In practice, one can improve robustness by applying some regularization to the modeling error $e$ from (9.16). In particular, [191] utilizes this reflectance model and assumes the corruptions to the data are *sparse*. In the next section we propose an alternative model based on dictionary learning to robustly solve (9.17).

---

[1] Note that we do not explicitly constrain values of $a_{xy}$ to be positive, although this is strictly required to interpret $g_{xy}(.)$ as the inverse of a reflectance model $f_{xy}(.)$.

## 9.4 Dictionary Learning Approaches

Dictionary learning refers to a class of algorithms that seek to sparsely represent some data of interest with respect to a learned "dictionary"—a collection of basis or atom elements. Intuitively, dictionary learning methods allow one to uncover structure present in data without a priori knowledge of the form of the structure. In this section, we propose three adaptive dictionary learning algorithms for photometric stereo.

### 9.4.1 Preprocessing of Images through Dictionary Learning (DLPI)

We first propose applying dictionary learning to our data in a preprocessing step performed on the images before reconstructing the normal vectors. Our formulation utilizes a dictionary learning regularization term to represent local patches of the data matrix $Y$ from (9.3) as sparse in an adaptive (learned) dictionary, thereby removing certain non-idealities from the data that are not represented by the dictionary. This approach can be thought of as applying a denoising step to the raw images. Specifically, we propose to solve the optimization problem:

$$\begin{aligned} \min_{v,B,D} \quad & \|y-v\|_2^2 + \lambda\Big(\sum_{j=1}^{c} \|P_j v - D b_j\|_2^2 + \mu^2 \|B\|_0\Big) \\ \text{s.t.} \quad & \|B\|_\infty \le q,\ \|d_i\|_2 = 1,\ \forall i. \end{aligned} \tag{9.18}$$

Here, $y = \mathbf{vec}(Y) \in \mathbb{R}^{m_1 m_2 d}$ and $P_j \in \mathbb{R}^{c_x c_y c_z \times m_1 m_2 d}$ is a diagonal $\{0,1\}$ matrix that extracts vectorized patches of dimensions $c_x \times c_y \times c_z$ from $v$, where $c_x$ and $c_y$ correspond to the dimensions of the patches extracted from each image and $c_z$ corresponds to the number of consecutive images these patches are extracted from. In practice, we extract patches from $v$ using a simple sliding window strategy. $D \in \mathbb{R}^{c_x c_y c_z \times K}$ is the learned dictionary with atoms (columns) $d_i$. Note that, while each atom is stored as a vector, it can be interpreted upon reshaping as an $c_x \times c_y \times c_z$ tensor. $B \in \mathbb{R}^{K \times c}$ is a sparse coding matrix whose columns $b_j$ define the (usually sparse) linear combinations of dictionary atoms used to represent each patch. Also, $\|.\|_0$ is the familiar $\ell_0$ (pseudo-)norm, and $\lambda \ge 0$ and $\mu \ge 0$ are regularization parameters. The parameter $K$ specifies the number of atoms in our dictionary $D$, and $c$ is the number of patches extracted from $v$—the denoised images.

We impose the constraint $||B||_\infty \triangleq \max_j ||b_j||_\infty \le q$, where $q$ is typically very large, to prevent any instability that could theoretically arise due to (9.18) being non-coercive with respect to $B$, but the constraint is inactive in practice [16, 120]. Without loss of generality, we also constrain the dictionary atoms $d_i$ to unit-norm to avoid scaling ambiguity between $D$ and $B$ [105].

After obtaining an (approximate) solution $\hat{v}$ to (9.18), we reshape $\hat{v}$ into an $m_1 m_2 \times d$ matrix $\hat{V}$ and estimate the associated normal vectors using the standard least-squares model (9.6) with the denoised images $\hat{V}$ in place of the original images $Y$.[2] Henceforth, we refer to this approach as the Dictionary Learning with Preprocessed Images (DLPI) method, and we present our algorithm for solving (9.18) in Section 9.5.

### 9.4.2 Normal Vectors through Dictionary Learning

We next propose modifying (9.6) by adding an adaptive dictionary learning regularization term applied to the normal vectors. Under this approach, we seek a normal map that agrees with the Lambertian model (9.6) while also having a locally sparse representation with respect to a learned dictionary—resulting in a smoother normal map that is robust to non-idealities in the data. Specifically, we propose to solve the optimization problem:

$$\begin{aligned} \min_{n,B,D} \quad & \|y - An\|_2^2 + \lambda\Big(\sum_{j=1}^{w} \|P_j n - Db_j\|_2^2 + \mu^2 \|B\|_0\Big) \\ \text{s.t.} \quad & \|B\|_\infty \le q, \quad \|d_i\|_2 = 1 \ \forall i. \end{aligned} \tag{9.19}$$

Here, $y = \mathbf{vec}(Y) \in \mathbb{R}^{m_1 m_2 d}$ and $A = L^T \otimes I \in \mathbb{R}^{m_1 m_2 d \times 3 m_1 m_2}$, where $\otimes$ denotes the Kronecker product and $I$ is the $m_1 m_2 \times m_1 m_2$ identity matrix. Furthermore, $n = \mathbf{vec}(N) \in \mathbb{R}^{3 m_1 m_3}$ are the vectorized normal vectors. As in the DLPI formulation, $P_j$ denotes a patch extraction matrix that extracts vectorized patches from $n$ of dimensions $w_x \times w_y \times w_z$, where $n$ is treated as an $m_1 \times m_2 \times 3$ tensor during extraction. In practice, we extract patches from $N$ using a simple sliding window strategy. Also, $D \in \mathbb{R}^{w_x w_y w_z \times K}$ denotes the learned dictionary whose columns (atoms) $d_i$ can be thought of as vectorized $w_x \times w_y \times w_z$ tensors, and $B \in \mathbb{R}^{K \times w}$ are the sparse coefficients needed to represent each patch of normal vectors as a linear combination of dictionary atoms. We impose the same constraints on $B$ and $D$ from the DLPI formulation (9.18).

Intuitively, the adaptive dictionary learning regularization in (9.19) is able to uncover underlying local structure in $N$ that the least squares formulation (9.6) alone

---

[2] Although we do not investigate this here, one could use a more sophisticated method in place of the least-squares model (9.6), e.g. one that incorporates a non-Lambertain model.

cannot deduce from the images. This results in normal vectors that are "smooth" and free from noise and other non-idealities that may otherwise corrupt them. Henceforth, we refer to this approach as the Dictionary Learning on Normal Vectors (DLNV) method, and we present our algorithm for solving (9.19) in Section 9.5.

### 9.4.3 Non-Lambertian Normal Vectors through Dictionary Learning

Finally, we present a method that is based on the non-Lambertian reflectance model from Section 9.3.3. Using (9.17) as the baseline, our approach is to incorporate a dictionary learning term to increase robustness to corruptions. In particular, we again apply dictionary learning regularization to the normal vectors, thus constraining them to agree with the non-Lambertian model (9.17) while also admitting a sparse representation in the learned dictionary. Specifically, we propose to solve the optimization problem:

$$\begin{aligned}\min_{n,B,D,a} \;& \sum_{x,y} \Big( \|C_{xy}a_{xy} - L^T n_{xy}\|_2^2 + \gamma \|1^T a_{xy} - 1\|_2^2 \Big) \\ + \;& \lambda \Big( \sum_{j=1}^{w} \|P_j n - D b_j\|_2^2 + \mu^2 \|B\|_0 \Big) \\ \text{s.t.} \;& \|B\|_\infty \leq q, \; \|d_i\|_2 = 1, \; \forall i. \end{aligned} \tag{9.20}$$

Here, all terms here are defined analogously as in Sections 9.3.3 and 9.4.2. Note that we include the constraint $1^T a_{xy} = 1$ from (9.17) in penalty form, where we typically set parameter $\gamma \geq 0$ to be very large.

The problem (9.20) can be thought of as a generalization of DLNV. Indeed, if we set $p = 1$ and $\gamma = \infty$, then (9.20) reduces to (9.19). However, we present both models as distinct methods in this chapter to highlight the differences between models that rely on the Lambertian assumption versus models that incorporate more complex reflectance models. We investigate the performance of both approaches in detail in Section 9.6. Henceforth, we refer to this approach as the Dictionary Learning on Normal Vectors with Piecewise-Linear Reflectance (PDLNV) method, and we present our algorithm for solving (9.20) in Section 9.5.

## 9.5 Algorithms and Properties

We propose solving (9.18), (9.19), and (9.20) via block coordinate descent-type algorithms. Specifically, for (9.18) and (9.19) we alternate between updating $v$ and $n$, respectively, with $(D, B)$ fixed and updating $(D, B)$ with $v$ or $n$ fixed. For (9.20) we use a similar strategy where we alternate between updating $n$, $(D, B)$, and $a$ with all other variables held fixed. For each subproblem, we now derive simple and efficient schemes for minimizing the associated cost.

### 9.5.1 Updating $(D, B)$

The $(D, B)$ update is identical for all three methods. Here we present the update using the notation from (9.18). For notational convenience, we define $G \triangleq B^T$ and denote by $P$ the matrix whose $j$th column is $P_j v$. With $v$ fixed, the optimization with respect to $(D, B)$ can be written as

$$\begin{aligned} \min_{G,D} \quad & \|P - DG^T\|_F^2 + \mu^2 \|G\|_0 \\ \text{s.t.} \quad & \|G\|_\infty \leq a, \quad \|d_i\|_2 = 1 \ \forall i. \end{aligned} \tag{9.21}$$

We (approximately) solve (9.21) by applying a few iterations of block coordinate descent, where we iterate over the columns $g_i$ of $G$ and columns $d_i$ of $D$ sequentially. For each $1 \leq i \leq K$, we minimize (9.21) first with respect to $g_i$ and then with respect to $d_i$, holding all other variables fixed.

We first consider the minimization of (9.21) with respect to $g_i$. Define $E_i \triangleq P - \sum_{k \neq i} d_k g_k^T$, where $E_i$ is computed using the most recent values of the dictionary atoms and coefficients. Then we can write the $g_i$ subproblem as

$$\begin{aligned} \min_{g_i} \quad & \left\|E_i - d_i g_i^T\right\|_F^2 + \mu^2 \left\|g_i\right\|_0 \\ \text{s.t.} \quad & ||g_i||_\infty \leq q. \end{aligned} \tag{9.22}$$

The solution to (9.22) is given by [119]

$$\hat{g}_i = \min\left(|H_\mu(E_i^T d_i)|, q 1_w\right) \odot \mathbf{sign}\left(H_\mu(E_i^T d_i)\right), \tag{9.23}$$

where $1_w \in \mathbb{R}^w$ is a vector of ones, $\min(.,.)$ is applied element-wise to vector arguments, and $\odot$ denotes element-wise multiplication. Furthermore, $H_\mu(.)$ denotes the element-wise hard thresholding function, defined as

$$H_\mu(y) = \begin{cases} 0 & \text{if } |y| < \mu \\ y & \text{if } |y| \geq \mu. \end{cases} \tag{9.24}$$

Minimizing (9.21) with respect to $d_i$ can be written as

$$\begin{aligned} \min_{d_i} \quad & \|E_i - d_i g_i^T\|_F^2 \\ \text{s.t.} \quad & \|d_i\|_2 = 1. \end{aligned} \tag{9.25}$$

The solution to (9.25) is given by [119]

$$\hat{d}_i = \begin{cases} \dfrac{E_i g_i}{\|E_i g_i\|_2}, & \text{if } g_i \neq 0 \\ u, & \text{if } g_i = 0, \end{cases} \tag{9.26}$$

where $u \in \mathbb{R}^{c_x c_y c_z}$ is an arbitrary unit-norm vector (e.g., the first column of the $c_x c_y c_z \times c_x c_y c_z$ identity matrix).

### 9.5.2 Updating $v$

Minimizing (9.18) with respect to $v$ yields the problem

$$\min_v \|y - v\|_2^2 + \lambda \sum_{j=1}^{c} \|P_j v - D b_j\|_2^2, \tag{9.27}$$

which is a least-squares problem with normal equation

$$\Big(I + 2\lambda \sum_{j=1}^{c} P_j^T P_j\Big) v = y + 2\lambda \sum_{j=1}^{c} P_j^T D b_j. \tag{9.28}$$

The matrix pre-multiplying $v$ in (9.28) is diagonal, so its inverse can be cheaply computed and hence $v$ can be updated efficiently.

### 9.5.3 Updating $n$

Minimizing (9.19) with respect to $n$ yields the problem

$$\min_n \|y - An\|_2^2 + \lambda \sum_{j=1}^{w} \|P_j n - D b_j\|_2^2. \tag{9.29}$$

Although (9.29) is a least-squares problem, its normal equation cannot be easily inverted due to the presence of the $A$ matrix. Instead, we perform a few iterations of proximal gradient [25] to (approximately) solve (9.29).[3] The cost function can be written in the form $f(n) + g(n)$ where $f(n) = \|y - An\|_2^2$ and $g(n) = \lambda \sum_{j=1}^{w} \|P_j n - Db_j\|_2^2$, so we perform the proximal steps

$$n^{k+1} = \mathbf{prox}_{\tau g}(n^k - \tau \nabla f(n^k)), \tag{9.30}$$

where

$$\mathbf{prox}_{\tau g}(y) := \arg\min_{x} \ \frac{1}{2}\|y - x\|_2^2 + \tau g(x) \tag{9.31}$$

is the proximal operator of $g$ and $\tau > 0$ is a chosen step size. The updates (9.30) are guaranteed to converge to a solution of (9.29) when $\tau < 1/\|A\|^2 = 1/\|L\|^2$, and in fact the cost will monotonically decrease when $\tau \leq 1/2\|L\|^2$ is used [25].

Define $\tilde{n}^k \triangleq n^k - \tau \nabla f(n^k) = n^k - 2\tau A^T(An^k - y)$. Then, after substituting (9.31) into (9.30) and simplifying, one can show that $n^{k+1}$ satisfies the normal equation

$$\Big(I + 2\tau\lambda \sum_{j=1}^{w} P_j^T P_j\Big) n^{k+1} = \tilde{n}^k + 2\tau\lambda \sum_{j=1}^{w} P_j^T D b_j. \tag{9.32}$$

As in (9.28), the matrix multiplying $n^{k+1}$ in (9.32) is diagonal and thus can be efficiently inverted to compute $n^{k+1}$.

In the case of PDLNV, the $n$ update for (9.20) can be solved in an identical manner, where the analogous data matrix $y$ is constructed as

$$y = \mathbf{vec}([C_{11}a_{11} \ \ldots \ C_{m_1 m_2} a_{m_1 m_2}]) \tag{9.33}$$

from the most recent values of $a_{xy}$.

### 9.5.4 Updating $a$

Minimizing (9.20) with respect to $a_{xy}$ yields $m_1 m_2$ problems of the form

$$\min_{a_{xy}} \ \sum_{x,y} \left\|C_{xy} a_{xy} - L^T n_{xy}\right\|_2^2 + \gamma \left\|1^T a_{xy} - 1\right\|_2^2. \tag{9.34}$$

---

[3] Proximal gradient is one of many possible iterative schemes for minimizing the quadratic objective (9.29); one could also employ a different algorithm, such as preconditioned conjugate gradient.

These are simple least squares problems with $d+1$ equations and $p$ unknowns that can be solved exactly and in parallel. Indeed, the solution to (9.34) is

$$\hat{a}_{xy} = \begin{bmatrix} C_{xy} \\ \gamma 1^T \end{bmatrix}^{\dagger} \begin{bmatrix} L^T n_{xy} \\ \gamma \end{bmatrix}. \tag{9.35}$$

The pseudoinverse in (9.35) is a constant that can be pre-computed from the raw images, so $a_{xy}$ can be updated efficiently.

### 9.5.5 Convergence

The proposed algorithms for solving (9.18), (9.19), and (9.20) alternate between updating $(D, B)$, $v$ or $n$, and $a$ (PDLNV only) with the other variables held fixed. Except for the $n$ updates of DLNV and PDLNV, all update schemes are either exact block coordinate descent updates or composed of inner iterations of exact block coordinate descent updates, so the objectives in our formulations must be monotonically decreasing (non-increasing) during these updates. Moreover, the proximal gradient step size for the $n$ update can be chosen to guarantee that these iterations also monotonically decrease their objectives. Thus, the cost functions for all three proposed algorithms are monotonically decreasing and bounded below by zero, so they must converge. Whether the algorithm iterates themselves converge to critical points of the (non-convex) costs is an interesting theoretical question for future work.

## 9.6 Numerical Experiments

We now investigate the performance of our proposed dictionary learning-based methods experimentally. To obtain quantitative results, we rely primarily on the recent DiLiGenT benchmark dataset [204]. This dataset contains images of a variety of surfaces of different materials and provides the true normal vectors of each object, allowing us to measure the accuracy of the normal vectors produced by each method. We quantify the error in each estimated normal vector by measuring the angular difference between it and the correponding true normal vector.

We evaluate our methods in a variety of settings. For each experiment, we compare the results of our methods to the robust PCA (RPCA) approach of Wu et al [186], the sparse regression (SR) method of Ikehata et al [189], and the constrained bivariate regression (CBR) approach of Ikehata et al [192]. In addition, we compare with the baseline least squares (LS) model defined by (9.6).

| Dataset | Mean Angular Error (degrees) | | | | | | |
|---|---|---|---|---|---|---|---|
| | PDNLV | DLPI | DLNV | CBR | SR | RPCA | LS |
| Ball | 3.60 | 3.99 | 3.82 | 6.78 | **2.08** | 3.20 | 4.10 |
| Cat | **6.40** | 8.39 | 8.10 | 8.05 | 6.73 | 7.96 | 8.41 |
| Pot1 | **6.99** | 8.88 | 8.67 | 8.57 | 7.24 | 8.81 | 8.89 |
| Bear | 8.51 | 8.31 | 8.32 | 9.77 | **6.01** | 7.89 | 8.39 |
| Pot2 | **10.37** | 14.57 | 13.88 | 10.56 | 11.98 | 11.94 | 14.65 |
| Buddha | 13.56 | 14.91 | 14.72 | 14.90 | **11.11** | 13.88 | 14.92 |
| Goblet | 15.49 | 18.43 | 17.69 | **15.10** | 15.53 | 15.14 | 18.50 |
| Reading | 20.28 | 19.66 | 19.58 | 19.39 | **12.56** | 17.42 | 19.80 |
| Cow | 21.80 | 25.48 | 17.58 | 15.68 | 22.42 | **11.96** | 25.60 |
| Harvest | **20.89** | 30.55 | 27.07 | 26.93 | 26.80 | 25.50 | 30.62 |

Table 9.1: Mean angular errors of the estimated normal vectors for the full, uncorrupted DiLiGenT datasets.

With the exception of LS, each method contains one or more regularization parameters. For each method, we sweep the parameters across a wide range of values and select the optimal parameters for each trial. For existing methods, we include any recommended parameters from the respective papers in our sweep. We run all iterative algorithms to convergence, and we repeat each experiment with additive noise over multiple realizations and average the results.

The majority of the photometric stereo literature has focused primarily on the problem of reconstructing normal vectors from uncorrupted and generally large datasets (many images of each object), such as the DiLiGenT dataset. In cases where additional corruptions were added, the corruptions were typically sparse to better align with the modeling assumptions of each method. In our experiments, we endeavor to fully investigate the robustness of our proposed methods and existing methods to general non-sparse corruptions. Specifically, we corrupt the raw images with Poisson noise, which is a realistic model for noise in real images [221]. This model is applicable, for example, when performing photometric stereo in low-light conditions, where noise levels can be significant.

Many existing photometric stereo algorithms apply a pixel-wise mask as a preprocessing step to remove shadows from the images. Such masks are typically computed by performing a simple thresholding operation on the data and excluding any pixels below a chosen threshold from subsequent computations. While this strategy can improve results in some cases, it does not capture the complexity of shadows present in the image and often results in useful data being rejected. This is of particular importance when working with small or heavily corrupted datasets, where it is important that the reconstruction method has access to as much data as possible to uncover the relevant information. Such robust methods should have the capacity to adapt to shadows in images without the use of a shadow mask. As such, all of the experiments we present here are performed without the use of shadow masks.

### 9.6.1 Evaluation on Uncorrupted DiLiGenT Dataset

We first investigate the performance of our proposed methods on the DiLiGenT dataset [204]. For each object, we use all 96 images present in the dataset and do not add any additional corruptions to the images. For PDLNV, we set $p = 2$ for each dataset except Harvest, where we set $p = 3$.

Table 9.1 presents the mean angular errors of the reconstructed normal vectors for each method on each dataset. PDLNV outperforms all existing approaches on 4 of the 10 objects. In cases where our methods do not outperform existing approaches, with the exception of the Reading and Cow datasets, we achieve comparable angular errors to the best performer. As we will demonstrate, the primary strength of our proposed methods is constructing normal vectors from images that are less pristine than the DiLiGenT datasets. However, Table 9.1 shows that PDLNV is still able to perform better or comparable to methods specifically designed to operate on large, clean datasets.

In practice, it may be infeasible to collect 96 images of an object under varying lighting conditions. As such, it is important to develop methods that can accurately estimate normal vectors from smaller datasets. Figures 9.1 and 9.2 illustrate the angular errors of the normal vectors estimated by each method on the uncorrupted DiLiGenT Cat and Harvest datasets as a function of the number of images used. In these experiments, we randomly selected images from among the original 96 images and averaged the results across 10 trials. From both figures, it is clear that PDLNV significantly outperforms all other methods for most dataset sizes.

### 9.6.2 Evaluation on Corrupted DiLiGenT Dataset

We next compare the performance of our proposed methods to existing methods on images corrupted with Poisson noise. Specifically, we subsample the DiLiGenT Pot2 dataset to 20 images and then corrupt these images with Poisson noise of a given signal-to-noise-ratio (SNR).

Figure 9.3 plots the angular errors of the estimated normal vectors for each algorithm as a function of SNR. It is clear that our proposed dictionary learning-based

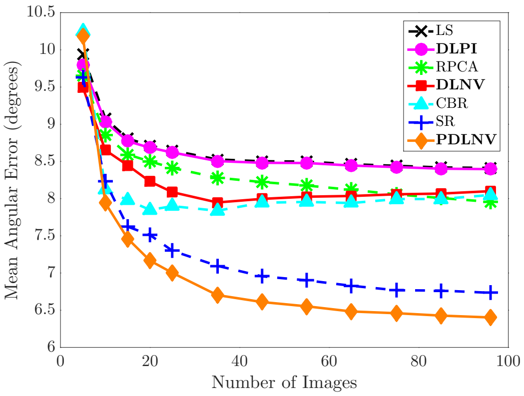


Figure 9.1: Mean angular errors (in degrees) of the estimated normal vectors for the DiLiGenT Cat dataset as a function of number of images used during reconstruction.

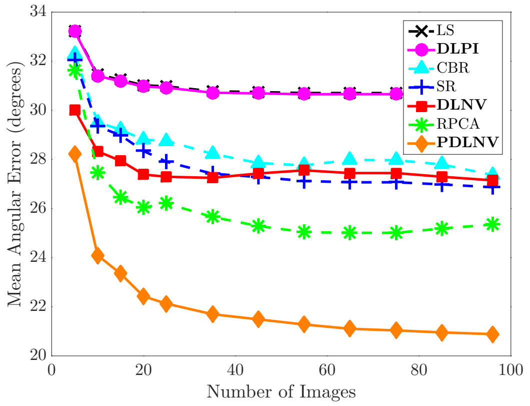


Figure 9.2: Mean angular errors (in degrees) of the estimated normal vectors for the DiLiGenT Harvest dataset as a function of number of images used during reconstruction.

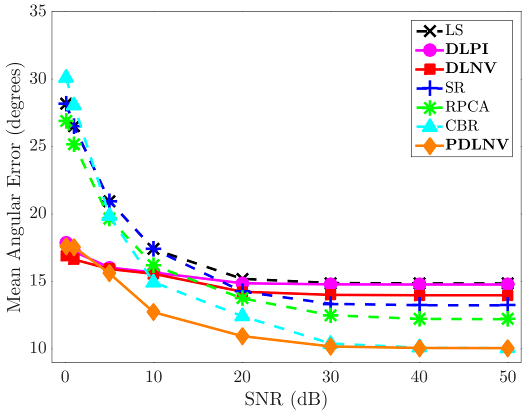
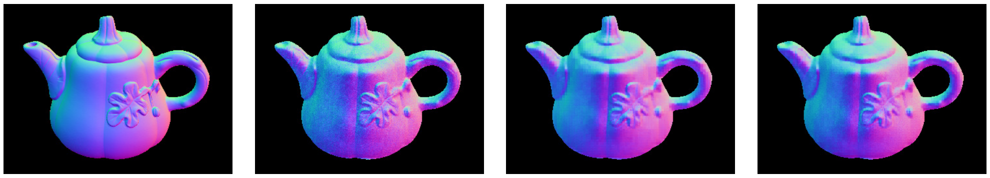


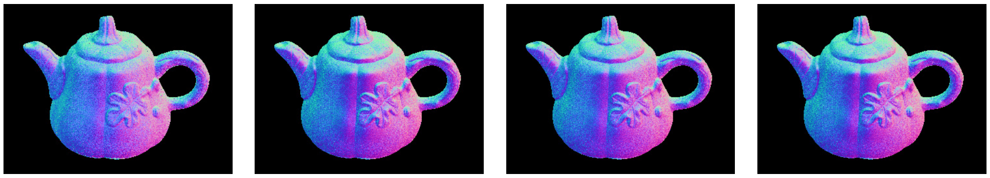
Figure 9.3: Mean angular errors (in degrees) of the estimated normal vectors for the DiLiGenT Pot2 dataset with 20 images versus SNR.

approaches are significantly more robust to high levels of non-sparse corruptions than existing methods. In particular, for SNR values below 10 dB, our methods outperform the existing methods by up to 10 degrees. Furthermore, the angular errors produced by our dictionary learning-based methods vary significantly less than existing approaches, indicating that the normal vector reconstructions are much more stable and robust to these corruptions.

Figures 9.4 and 9.5 show the normal vector reconstructions and the corresponding error maps produced by each method on the Pot2 dataset with 20 images at a noise level of 5 dB. As these figures illustrate, the dictionary learning-based reconstructions are significantly more accurate and robust to noise than the existing methods. Figure 9.6 shows the error maps for the normal vectors produced by PDLNV on

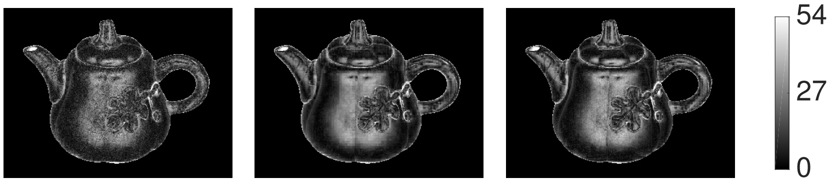


(a) Truth (b) **PDLNV** (c) **DLNV** (d) **DLPI**

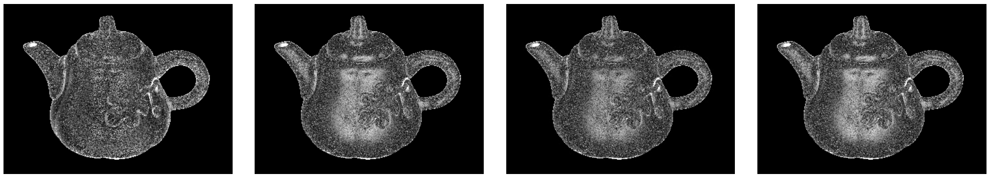


(e) CBR (f) SR (g) RPCA (h) LS

Figure 9.4: Normal vector reconstructions for the DiLiGenT Pot2 dataset with 20 images and 5 dB Poisson noise.



(a) **PDLNV** (b) **DLNV** (c) **DLPI**



(d) CBR (e) SR (f) RPCA (g) LS

Figure 9.5: Normal vector error maps (in degrees) for the DiLiGenT Pot2 dataset with 20 images and 5 dB Poisson noise.

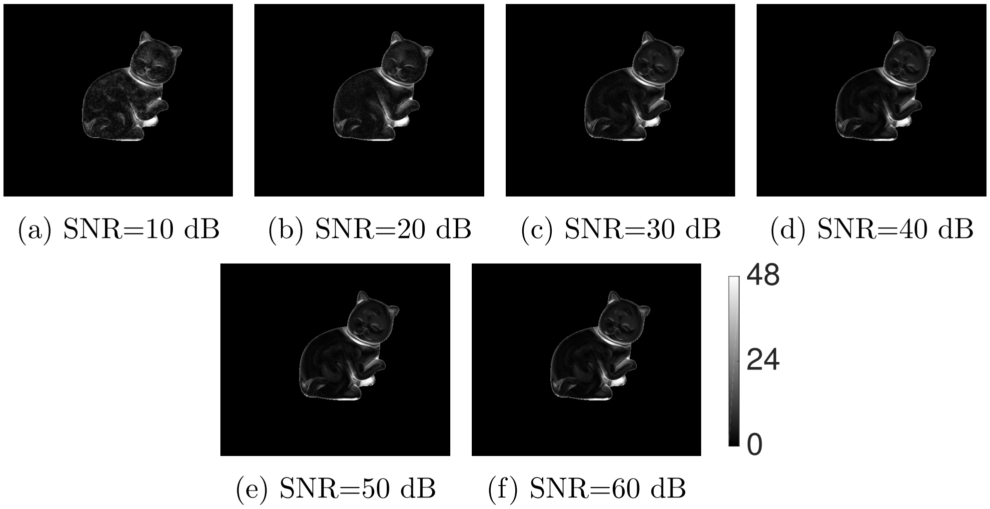
the DiLiGenT Cat dataset for different SNR levels. As this figure shows, the error maps are quite uniform across the noise levels, indicating that PDLNV is surprisingly insensitive to noise strength.

### 9.6.3 Evaluation on non-DiLiGenT Datasets

In addition to the DiLiGenT dataset, we also consider the dataset[4] from [222]. This dataset contains images of several real objects with no corresponding normal vectors. To obtain reference (ground truth) normal vectors for this dataset, we assume the objects are Lambertian. While this assumption does not hold exactly, the objects

---

[4] The data can be found at `http://vision.seas.harvard.edu/qsfs/Data.html`.


(a) SNR=10 dB (b) SNR=20 dB (c) SNR=30 dB (d) SNR=40 dB

(e) SNR=50 dB (f) SNR=60 dB

Figure 9.6: Normal vector error maps (in degrees) computed with PDLNV for the DiLiGenT Cat dataset with 20 images and varying SNR.

are matte in appearance and thus nearly Lambertian. We compute the reference normal vectors by applying the standard least squares method (9.6) to the raw images.

Our motivation for considering this (approximately) Lambertian dataset is as follows. Even when additional noise is added, the primary challenge with the DiLiGenT datasets is dealing with the fundamentally non-Lambertian properites of the data (specularities, shadows, etc.) As such, our experiments thus far do not necessarily evaluate the ability of each method to estimate a Lambertian surface in the presence of noise, despite the fact that the majority of the methods we are investigating are based on a Lambertian model. Therefore, in this section we assume our data is Lambertian, add corruptions, and then evaluate the ability of each method to reject the corruptions and produce normal vectors that agree with the underlying Lambertian model.

Figure 9.7 plots the mean angular errors of the estimated normal vectors for the Hippo dataset as a function of SNR. For high SNR, the errors approach zero, as expected since the uncorrupted data is Lambertian. However, in the high SNR regime, the proposed dictionary learning-based approaches are significantly more robust to imperfections compared to existing approaches. Unlike in the DiLiGenT experiments, the DLPI method now outperforms both DLNV and PDLNV. This suggests that the dictionary learning-based preprocessing step is able to robustly remove corruptions from Lambertian data.

We also evaluate the qualitative performance of each method. Figure 9.8 shows the reference normal vectors for the Cat dataset (computed from the uncorrupted

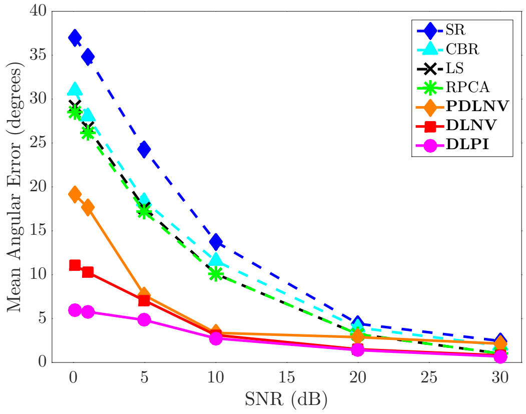


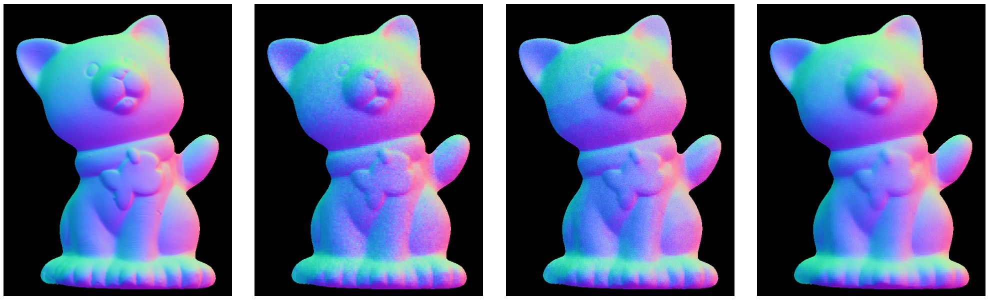
Figure 9.7: Mean angular errors (in degrees) of estimated normal vectors for the Hippo dataset with 20 images versus SNR.

data using the least squares method) together with the normal vectors estimated by each method from data corrupted by Poisson noise with 5 dB SNR. Figure 9.9 shows the error maps for the estimated normal vectors with respect to the reference normal vectors. Clearly the proposed dictionary learning-based methods produce much more accurate normal vectors compared to the existing methods. Note that the DLPI method achieves particularly small errors on the smooth portions of the surface, where the normal vectors are slowly varying.

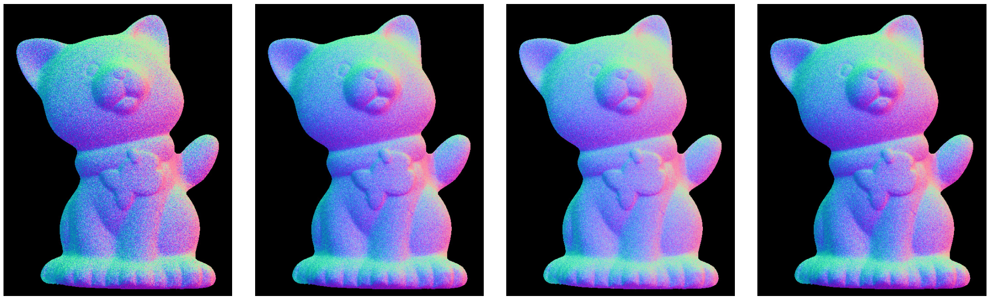
Figure 9.10 plots the surfaces computed for the normal vectors from Figure 9.8 using the method outlined in [220]. Qualitatively, we see that the surfaces computed from the dictionary learning-based methods are much smoother and more accurate representations of the actual surface. In contrast, the surfaces obtained from the existing methods, though they preserve the general shape of the surface, are quite rough and/or contain significant spike artifacts. Note that the DLNV method does exhibit some flattening artifacts on the side of the head, and the DLPI method produces an extremely smooth surface at the cost of some loss of definition near sharp edges. PDLNV, though slightly less smooth than DLNV, retains many of the sharp edges of the reference surface.

### 9.6.4 Algorithm Properties

Finally, we investigate the properties of our proposed dictionary learning-based methods and how the various model parameters affect the results. In addition to regularization parameters, which were directly optimized for each method in our experiments, there are multiple model parameters that can be tuned. The dimensions of the dictionary atoms—which correspond to the patch sizes that are extracted from

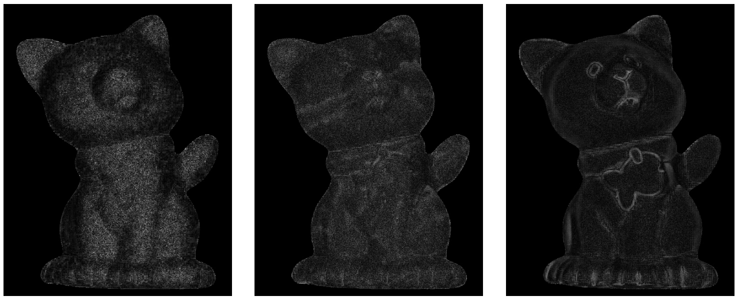


(a) Reference (b) **PDLNV** (c) **DLNV** (d) **DLPI**

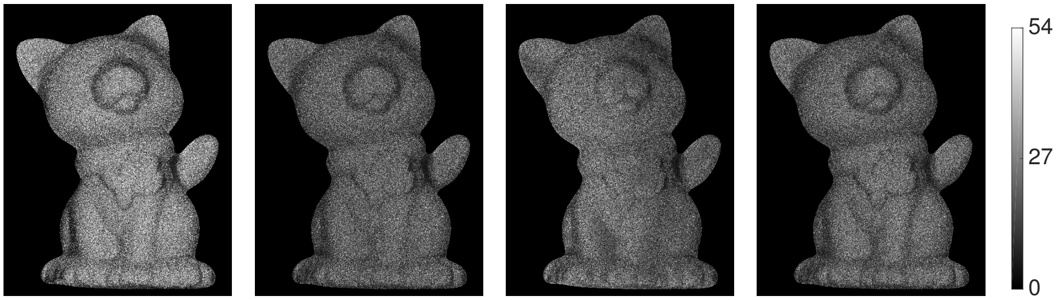


(e) SR (f) RPCA (g) CBR (h) LS

Figure 9.8: Normal vector reconstructions for the Cat dataset with 20 images and 5 dB Poisson noise.

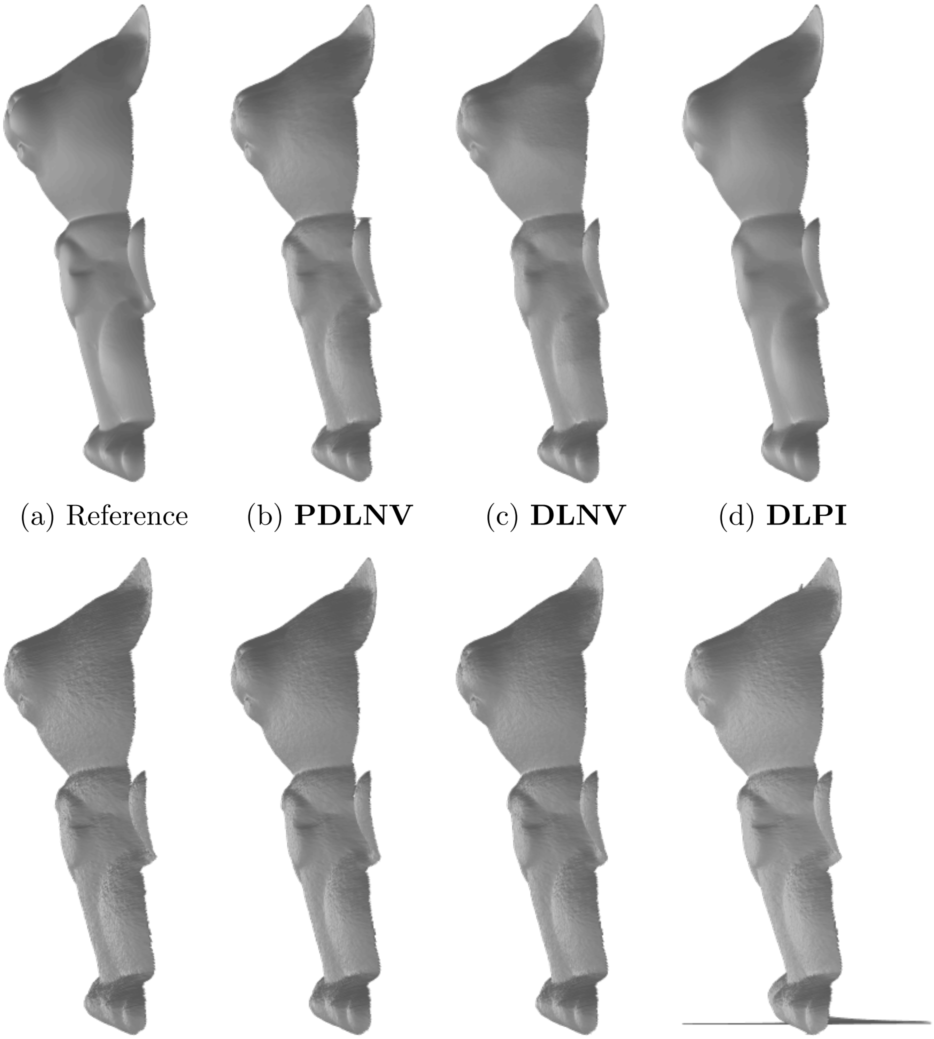


(a) **PDLNV** (b) **DLNV** (c) **DLPI**



(d) SR (e) RPCA (f) CBR (g) LS

Figure 9.9: Normal vector error maps (in degrees) for the Cat dataset with 20 images and 5 dB Poisson noise.



(a) Reference (b) **PDLNV** (c) **DLNV** (d) **DLPI**

(e) SR (f) RPCA (g) LS (h) CBR

Figure 9.10: Surfaces computed from the estimated normal vectors of the Cat dataset with 20 images and 5 dB Poisson noise.

the images or normal vectors—can be changed, the patch extraction strategy—e.g., non-overlapping patches or overlapping patches with a given spatial stride—can be changed, and one must choose how to initialize the dictionary. Another interesting parameter is the number of atoms (columns) in the dictionary, where we are free to choose between tall (undercomplete), square, or wide (overcomplete) dictionaries. In the case of PDNLV, we can also choose the parameter $p$ that controls the number of piecewise segments in the non-Lambertian model; the optimal value may depend on the properties of a given surface.

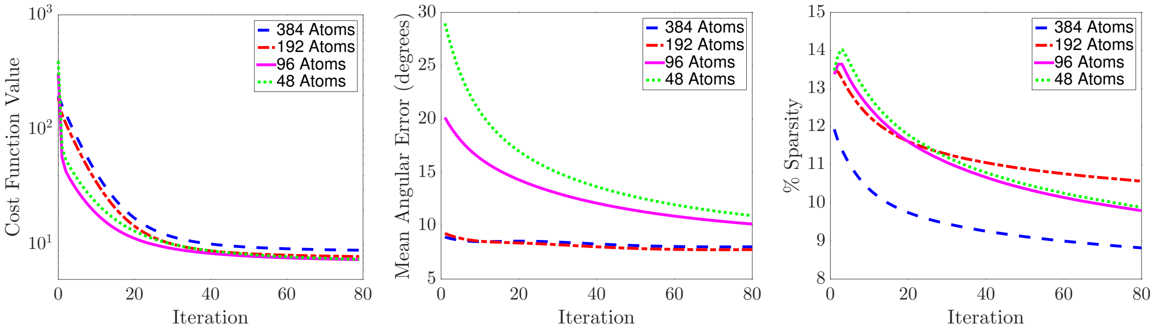
In the preceding experiments, we used $8 \times 8$ atoms for DLPI, which is a standard choice for dictionary learning methods. For PDNLV, we choose $8 \times 8 \times 3$ atoms, where the third dimension corresponds to the $(x, y, z)$ coordinates of each normal vector. In each case, we used a sliding window strategy with a spatial stride of 4 pixels in each direction to extract overlapping patches from the images or normal vectors. We used square dictionaries for our experiments (containing 64 atoms for DLPI and 192 atoms for DLNV and PDLNV), and we initialized each dictionary to the discrete cosine transform (DCT) matrix of appropriate size. For the $n$ updates of DLNV and PDLNV, we used the step size $\tau = 1/2\|L\|^2$ to guarantee that the updates will monotonically decrease their objectives.

Figure 9.11 shows the per-iteration properties of the PDLNV method with $p = 2$ from a representative trial on the DiLiGenT Cat dataset. Figure 9.11a plots the cost function at each iteration, and Figure 9.11b shows the corresponding mean angular error of the normal vector estimates. While the cost is guaranteed to decrease at each iteration, angular error can increase. Empirically we have found, however, that angular error typically decreases with iteration. Figure 9.11c plots the sparsity (percentage of nonzero elements) of the sparse coding matrix $B$ at each iteration. The sparsity of $B$ can be changed by varying the regularization parameter $\mu$. Empirically, we have found that sparsity values around 10% often yield good results. Each plot in Figure 9.11 includes multiple curves for several different dictionary sizes (number of columns). Of particular interest is how the size of the dictionary affects mean angular error. As Figure 9.11b illustrates, larger dictionaries typically perform better than smaller, undercomplete dictionaries. However, we do not observe a significant boost in performance when the dictionary becomes overcomplete. Note that $8 \times 8 \times 3$ dictionary atoms were used for PDLNV, so a size of 192 corresponds to a square dictionary.

In practice, for the trials included in this chapter, we terminated PDNLV after 50 iterations and terminated both DLNV and DLPI after 20 iterations. At every iteration, we would iterate once over the columns of $(D, B)$ and take 25 proximal gradient steps updating $n$ or $v$.

We next illustrate the effect of varying parameter $p$ on the performance of PDLNV. Figure 9.12 shows the angular error of the estimated normal vectors when varying the number of images in the DiLiGenT Pot1 dataset for several values of $p$. As illustrated, for this dataset $p = 2$ is the optimal choice. In general, we have found that $p = 2$ typically produces good results, but in some cases (e.g., the DiLiGenT Harvest dataset) $p = 3$ does perform better.

Figure 9.13 illustrates the initial (DCT) and final (learned) dictionaries produced by PDLNV on the full DiLiGenT Pot1 dataset. Each dictionary atom is an $8 \times 8 \times 3$



(a) PDLNV cost function. (b) Angular errors. (c) Sparsity of $B$.

Figure 9.11: Cost function, normal vector angular errors (in degrees), and sparsity of the sparse coding matrix $B$ for the PDLNV method with $p = 2$ applied to the DiLiGenT Cat dataset with 20 images and 20 dB Poisson noise for several different dictionary sizes.

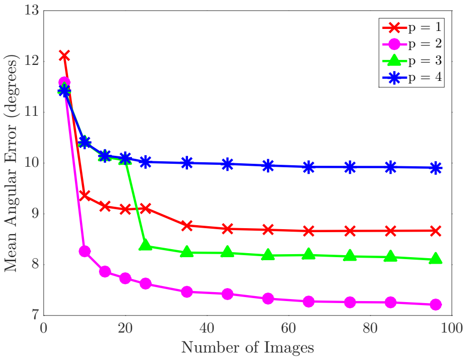
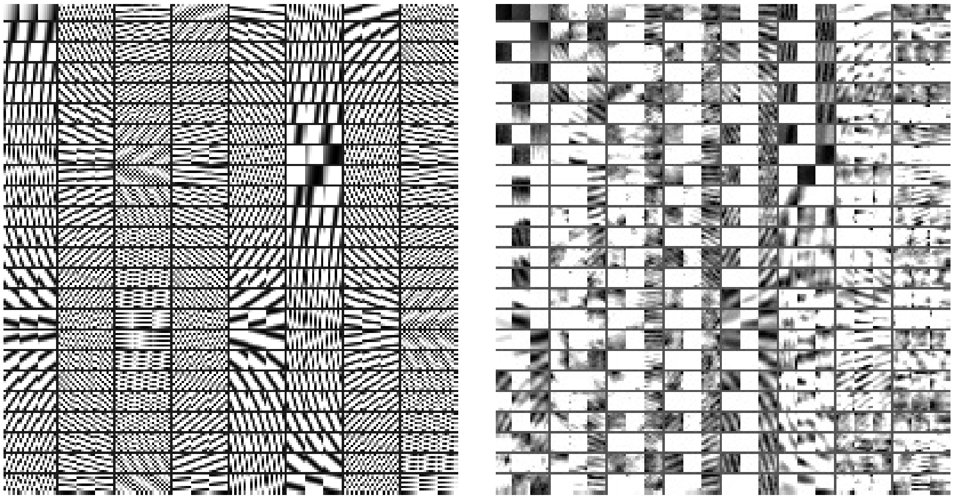


Figure 9.12: Mean angular error (in degrees) of the estimated normal vectors for PDLNV with multiple values of $p$ on the DiLiGenT Pot1 dataset as a function of number of images used.

tensor, so we visualize each atom as three $8 \times 8$ images arranged horizontally. The learned dictionary exhibits interesting structure. Some atoms have learned structure across all three normal vector dimensions, while other atoms have learned structure in one or two dimensions and are trivial (constant) in the other dimension(s). Note that this behavior has emerged organically—the dictionary learning methods automatically adapt to the underlying structure on a per-dataset basis.

## 9.7 Conclusion

In this chapter, we proposed three methods for applying dictionary learning to photometric stereo. Each method seeks to represent some form of the data—either the original images or the reconstructed normal vectors—as sparse with respect to



(a) Initial DCT dictionary.  (b) Final learned dictionary.

Figure 9.13: Initial and final learned dictionaries for the PDLNV method with $p = 2$ applied to the full DiLiGenT Pot1 dataset.

an adaptive dictionary. We showed through extensive numerical studies that our proposed methods are significantly more robust than existing methods in the high-noise regime while preserving accuracy in the low-noise regime. Dictionary learning is a general purpose adaptive regularization framework, and, as such, it could be coupled with more complex reflectance models from the photometric stereo literature to further improve reconstruction quality. We plan to investigate this line of inquiry in future work.

# CHAPTER X

# Robust Surface Reconstruction via Dictionary Learning

## 10.1 Introduction

Imaging techniques such as photometric stereo [179] allow one to efficiently estimate the normal vector map of an object. The primary goal of such methods is to ultimately derive a three-dimensional representation of the object, a task which requires some flavor of numerical integration of the gradient fields defined by the normal vector map. Robust photometric stereo—the problem of accurately determining the normal map of a non-ideal surface or from noisy data—has attracted considerable attention in recent years [186, 189, 192]. In this chapter, we seek to develop a robust approach to the problem of reconstructing surfaces from gradient fields that can accurately estimate the depth map of an object in the presence of noise.

The problem of reconstructing a surface from estimates of its photometric stereo gradient fields has been investigated since the late 1980s. The seminal works of Simchony *et al.* [220] and Frankot and Chellappa [223] seek to solve the problem through a least squares approach, utilizing efficient Discrete Fourier Transform or Discrete Cosine Transform based solvers—essentially attempting to project the surface onto Fourier basis functions or the DCT basis. Harker and O'Leary [224] propose a modified "global" least squares problem and extend this method to incorporate regularization [225], solving a Sylvester equation to obtain the solution. Recently, Quéau and Durou [226] introduced a weighted-least squares formulation as well as formulations minimizing total-variation and incorporating the $\ell_1$ norm to promote sparsity. Further attempts at applying a regularization term while integrating the gradients have also been proposed at the expense of computation time [227, 228]. Additional approaches include line-integral based methods [229, 230] and reconstructions based on the calculus of variations [231–233]. A range of other possible methods have also been proposed with mixed results [234–239].

Our work builds on these previous works, specifically those that utilize a least squares-type formulation to relate the underlying surface and its gradient fields. In particular, we propose a novel adaptive dictionary learning based regularizer that enables the robust estimation of surfaces from noisy gradients. Dictionary learning [100, 104, 240] has, in recent years, been successfully applied to many imaging applications, e.g., [16, 17, 103]. In dictionary learning models, one typically seeks to learn sparse representations of the local patches of the data. These models often induce a type of smoothness constraint on the underlying data that, in the case of surface reconstruction, we show leads to robust reconstructions with desirable noise rejection properties. Our framework is general and can be easily combined with any existing method that utilizes a least squares-type objective to estimate the underlying surface.

## 10.2 Surface Reconstruction from Gradient Fields

Let $n(x,y) \in \mathbb{R}^3$ denote the normal vector of a differentiable surface $z(x,y)$ at position $(x,y)$, and let $n_1(x,y)$, $n_2(x,y)$, and $n_3(x,y)$ denote the $x$, $y$, and $z$ components of this vector, respectively. Under this ideal model, one can relate the $x$ and $y$ derivatives of the surface $z$ to its normal vectors via the relation

$$\frac{\partial z(x,y)}{\partial x} = -p(x,y), \quad \frac{\partial z(x,y)}{\partial y} = q(x,y), \tag{10.1}$$

where we have defined $p(x,y) := n_1(x,y)/n_3(x,y)$ and $q(x,y) := n_2(x,y)/n_3(x,y)$. In practice, the estimated (e.g., by photometric stereo) normal vectors of a surface and its gradient fields will not exactly satisfy (10.1), so one must instead find a function $z(x,y)$ with derivatives *close* to $p(x,y)$ and $q(x,y)$ in an appropriate sense, often by minimizing a variational problem of the form

$$\iint_\Omega \left(\frac{\partial z(x,y)}{\partial x} - p(x,y)\right)^2 + \left(\frac{\partial z(x,y)}{\partial y} - q(x,y)\right)^2 dx\; dy. \tag{10.2}$$

When our data is instead sampled on a discrete grid, we will not have access to a continuous normal map $n(x,y)$ but will instead have some matrix $N \in \mathbb{R}^{m \times n \times 3}$ containing the normal vectors of the object on the grid. Following (10.1), we can compute matrices $P \in \mathbb{R}^{m \times n}$ and $Q \in \mathbb{R}^{m \times n}$ containing the measured gradients, and our goal then becomes to estimate the matrix $Z \in \mathbb{R}^{m \times n}$ containing the values of the surface $z(x, y)$ sampled on the grid. The discrete analogue of (10.2) is commonly expressed [220, 223, 224] as a standard least squares problem of the form

$$z^* = \arg\min_z \; \|Az - v\|_2^2, \tag{10.3}$$

where $z = \mathbf{vec}(Z) \in \mathbb{R}^{mn}$ is the vectorized version of $Z$, $A$ is a numerical differentiation operator, and the vector $v$ is an appropriate function of the measured gradients, $P$ and $Q$. Solving this problem yields a representation of our surface optimal in the least squares sense.

Note that the specific form of $A$ and $v$ can vary. One possible formulation is

$$A = \begin{bmatrix} D_n \otimes I_m \\ I_n \otimes D_m \end{bmatrix}, \qquad v = \begin{bmatrix} \mathbf{vec}(P) \\ \mathbf{vec}(Q) \end{bmatrix}, \tag{10.4}$$

where $D_n$ is the discrete first differences matrix, and $\otimes$ denotes the Kronecker product. However, multiple models are possible, each yielding reconstructed surfaces with different properties. Importantly, the dictionary learning based approach that we will introduce in Section 10.3 can be coupled with any least squares model of the form (10.3), so our proposed approach is quite flexible.

## 10.3 Adaptive Dictionary Learning Regularization

Given normal vectors corrupted by noise or other non-idealities, solving (10.3) directly generally produces a rough, bumpy surface, even when the underlying true surface is smooth. Thus, in this chapter, we propose an adaptive dictionary regularizer that can be combined with the least squares model (10.3) to more accurately estimate the underlying surface. In particular, we propose to solve the following dictionary learning problem

$$\begin{gathered}\{z^*, B^*, D^*\} = \arg\min_{z,B,D} \frac{1}{2}\|Az - v\|_2^2 + \lambda\left(\sum_{j=1}^{c} \|P_j z - Db_j\|_2^2 + \mu^2\|B\|_0\right) \\ \text{s.t. } \|B\|_\infty \le a, \; \|d_i\|_2 = 1, \; \forall i. \end{gathered} \tag{10.5}$$

In (10.5), $P_j$ is a patch extraction operator that extracts a vectorized $c_x \times c_y$ spatial patch from $z$, $D \in \mathbb{R}^{c_x c_y \times K}$ is a dictionary matrix whose columns $d_i$ are vectorized $c_x \times c_y$ atoms, and $B \in \mathbb{R}^{K \times c}$ is a matrix of sparse codes, where the columns $b_j$ of $B$

define the (usually sparse) linear combination of atoms used to represent the patch $P_j z$ of $z$. Also, $\|\cdot\|_0$ is the familiar $\ell_0$ (pseudo-)norm and $\lambda \geq 0$ and $\mu \geq 0$ are regularization parameters.

We include the constraint $\|B\|_\infty \triangleq \max_j \|b_j\|_\infty \leq a$, where $a$ is typically very large, since (10.5) is non-coercive with respect to $B$, although the constraint is inactive in practice [120]. In addition, we constrain the columns $D$ to be unit-norm to avoid scaling ambiguity between $D$ and $B$ [105]. Note that (10.5) is a non-convex problem.

By solving (10.5), we are attempting to estimate our surface $z$ by numerically integrating it through a least squares functional while simultaneously enforcing that local patches of the reconstructed surface should have sparse representations with respect to the dictionary $D$. As $D$ itself is learned, our proposed algorithm can automatically adapt to the underlying properties of the surface and its gradients.

### 10.3.1 Dictionary Learning on Surfaces (DLS) Algorithm

We propose to solve (10.5) via a block coordinate descent-type algorithm where we alternate between updating $z$ with $(D, B)$ fixed and updating $(D, B)$ with $z$ fixed. Henceforward, we refer to this algorithm as the Dictionary Learning on Surfaces (DLS) method.

#### 10.3.1.1 (D, B) updates

Let $P$ be the matrix with columns $P_j z$. With $z$ fixed, the minimization of (10.5) with respect to $(D, B)$ is

$$\begin{aligned} \min_{B,D} \quad & \|P - DB\|_F^2 + \mu^2 \|B\|_0 \\ \text{s.t.} \quad & \|B\|_\infty \leq a, \ \|d_i\|_2 = 1, \forall i. \end{aligned} \tag{10.6}$$

We solve (10.6) via a block coordinate descent method where we iteratively minimize the cost with respect to the $i$th column, $g_i$, of $G := B^T$ and the $i$th column, $d_i$, of $D$ for every $1 \leq i \leq K$ with all other variables held fixed.

For a given $i$, define $E_i \triangleq P - \sum_{k \neq i} d_k g_k^T$ computed using the most recent values of the other dictionary atoms and coefficients. Then the minimizer of (10.6) with respect to $g_i$ is given by [120]

$$\hat{g}_i = \min\left(|H_\mu(E_i^T d_i)|,\ a 1_C\right) \ \odot\ \mathbf{sign}(H_\mu(E_i^T d_i)), \tag{10.7}$$

where $1_C$ is a vector of ones of length $C$, $\min(\cdot,\cdot)$ is the element-wise minimum operator, $\odot$ denotes element-wise multiplication, and $H_\mu(\cdot)$ is the element-wise hard thresholding operator

$$H_\mu(y) = \begin{cases} y & \text{if } |y| \geq \mu \\ 0 & \text{if } |y| < \mu. \end{cases} \tag{10.8}$$

On the other hand, the minimizer of (10.6) with respect to $d_i$ with all other variables held fixed is given by [120]

$$\hat{d}_i = \begin{cases} \dfrac{E_i g_i}{\|E_i g_i\|_2} & \text{if } g_i \neq 0 \\ u & \text{if } g_i = 0, \end{cases} \tag{10.9}$$

where $u$ is any unit norm vector, e.g., the first column of the identity matrix.

#### 10.3.1.2 $z$ update

With $D$ and $B$ fixed, our problem becomes

$$\min_z \; \frac{1}{2}\|Az - v\|_2^2 + \lambda \sum_{j=1}^{c} \|P_j z - D b_j\|_2^2. \tag{10.10}$$

The cost function in (10.10) can be written in the form $f(z) + g(z)$ where $f(z) = \frac{1}{2}\|Az - v\|_2^2$, and $g(z) = \lambda \sum_{j=1}^{c} \|P_j z - D b_j\|_2^2$. We utilize a proximal gradient strategy to solve (10.10) [25], iteratively updating $z$ according to

$$z^{k+1} = \mathbf{prox}_{\tau g}(z^k - \tau \nabla f(z^k)), \tag{10.11}$$

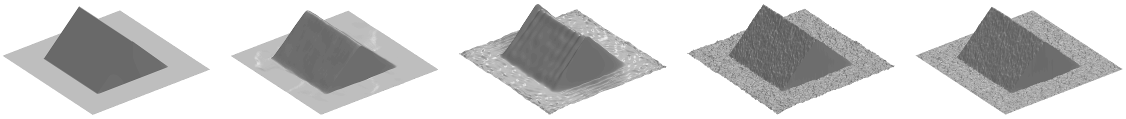
where

$$\mathbf{prox}_{\tau g}(y) := \arg\min_x \; \frac{1}{2}\|y - x\|_2^2 + \tau g(x) \tag{10.12}$$

is the proximal operator of $g(x)$. If we define $\tilde{z}^k \triangleq z^k - \tau \nabla f(z^k)$, we see that (10.11) and (10.12) imply that $z^{k+1}$ satisfies the normal equation

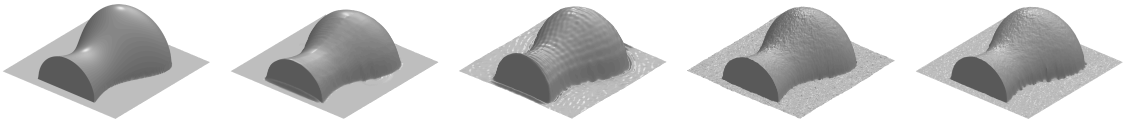
$$\Big(I + 2\tau\lambda \sum_{j=1}^{c} P_j^T P_j\Big) z^{k+1} = \tilde{z}^k + 2\tau\lambda \sum_{j=1}^{c} P_j^T D b_j. \tag{10.13}$$

The matrix on the left hand side of (10.13) is diagonal, so its inverse can be cheaply computed to solve for $z^{k+1}$. Note that (10.10) is a simple least squares problem and, as such, could be minimized by a variety of iterative schemes.



(a) Truth (b) **DLS** (c) SR (d) TV (e) DCTLS

Figure 10.1: Reconstructions of the Tent surface with SNR = 20 dB.



(a) Truth (b) **DLS** (c) SR (d) TV (e) DCTLS

Figure 10.2: Reconstructions of the Vase surface with SNR = 30 dB.

## 10.4 Results

In this section, we numerically evaluate our proposed DLS method on several example datasets. In each case we compare our method against the spectral regularization method (SR) [225], the isotropic total variation (TV) approach [226], and DCT based least squares (DCTLS) [220]. For methods that include tunable weight parameters, we sweep over a wide range of values, reporting the best results obtained over this parameter sweep. As we have noted, the dictionary learning approach can incorporate any least squares based solver by simply defining $A$ and $v$ in (10.5) accordingly. For all results given here, we have used the least squares cost found in [220]. In order to evaluate the robustness of our approach, we add Gaussian noise to the data.

For our proposed DLS method, we used dictionary atoms of size $8 \times 8$ pixels and a square $64 \times 64$ dictionary $D$, initialized with a DCT matrix. We extracted patches from $z$ using a spatial stride of two pixels in each direction, allowing adjacent patches to overlap. Finally, we initialized $z$ as the vectorized surface produced by solving the stand-alone least squares problem in [220], and initialized $B = 0$.

### 10.4.1 Synthetic Surface Reconstructions

To quantitatively evaluate our method, we first considered two synthetic datasets, which we call Tent and Vase, for which we have analytic expressions for the surface $z = f(x, y)$. Given $f(x, y)$, we can differentiate to obtain the gradients, $\partial f(x, y)/\partial x$ and $\partial f(x, y)/\partial y$, and sample on a discrete grid. By reconstructing the surface from these gradients subject to additive noise, the integrity of the reconstructions can then

| SNR | Tent | | | | Vase | | | |
|---|---|---|---|---|---|---|---|---|
| | **DLS** | SR | TV | DCTLS | **DLS** | SR | TV | DCTLS |
| 1 dB | **0.969** | 0.944 | 0.918 | 0.924 | **0.958** | 0.930 | 0.889 | 0.894 |
| 5 dB | **0.971** | 0.950 | 0.938 | 0.944 | **0.966** | 0.934 | 0.911 | 0.915 |
| 10 dB | **0.976** | 0.956 | 0.957 | 0.962 | **0.971** | 0.942 | 0.933 | 0.936 |
| 20 dB | **0.988** | 0.969 | 0.979 | 0.983 | **0.977** | 0.961 | 0.965 | 0.966 |
| 30 dB | **0.995** | 0.978 | 0.989 | 0.992 | **0.982** | 0.975 | 0.981 | 0.981 |
| 40 dB | **0.997** | 0.985 | 0.994 | 0.996 | **0.990** | 0.982 | 0.990 | 0.989 |
| 50 dB | **0.998** | 0.988 | 0.996 | **0.998** | **0.993** | 0.984 | **0.993** | 0.992 |
| 60 dB | **0.999** | 0.989 | 0.997 | 0.998 | **0.995** | 0.985 | **0.995** | 0.993 |

Table 10.1: Quality of Tent (left) and Vase (right) surface reconstructions in SSIM as a function of SNR.

be evaluated against a ground truth, $f(x, y)$. We utilize the SSIM error metric [241] to evaluate the quality of the computed surfaces against this ground truth. For these experiments, noise is added directly to the gradient fields.

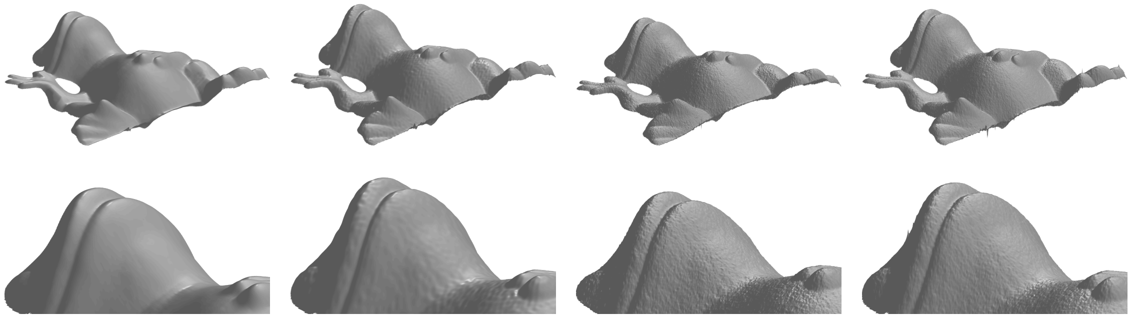
Figures 10.1 and 10.2 show images of the reconstructed surfaces produced by each algorithm. As these images illustrate, the proposed DLS method produces much smoother surfaces from noisy data compared to the existing methods. Intuitively, the locally sparse model imposed by the dictionary regularization denoises the surfaces, while the adaptive nature of the dictionary allows DLS to represent and reconstruct both sharp edges and smooth regions in a data-dependent basis.

The surfaces obtained by SR, TV, and DCTLS are much more sensitive to the noisy gradients. Indeed, while they retain the general shape of the surface, they exhibit significantly more corruption. In particular, the spectral regularization method, which attempts to represent the surface in a low-dimensional subspace, seems to introduce a systematic error into the reconstructions.

Table 10.1 numerically corroborates the qualitative results illustrated by Figures 10.1 and 10.2. The low signal-to-noise-ratio (SNR) regime, DLS significantly outperforms other approaches. As SNR increases, this gap decreases. When the data is essentially noiseless, DLS, TV, and DCTLS are all able to reconstruct the surfaces with nearly zero error, yielding comparable reconstruction quality.

### 10.4.2 Photometric Stereo

We now return to the problem of reconstructing a depth map of an object from normal vectors obtained through photometric stereo. We consider a dataset contain-



(a) **DLS** (b) SR (c) TV (d) DCTLS

Figure 10.4: Surface reconstructions for the Frog dataset with SNR = 17 dB.

ing 10 images, each taken under a unique, known lighting direction, and we corrupt the images with Gaussian noise. The normal vectors are computed from the noisy images using the standard least squares approach [186]. Once we have obtained the normal vectors, we utilize them to determine the gradient fields, as discussed in Section 2, and reconstruct a depth map of the object from these gradient fields. The results of this procedure performed on the Frog dataset [1] [222] are illustrated in Figure 10.4.

The reconstructions displayed in Figure 10.4 again illustrate the ability of the proposed DLS approach to effectively denoise the gradients and produce a smooth surface. Indeed, the surface produced by DLS is considerably smoother and less corrupted than those produced by the existing methods. The denoising capability of DLS may prove valuable when running photometric stereo on real-world data, where noise and other non-idealities are inevitable.

## 10.5 Conclusion

In this chapter, we explored the use of adaptive dictionary learning based regularization for the estimation of surfaces from their gradient fields. We showed that our proposed dictionary learning approach is able to effectively reject the addition of noise to gradient fields/images and produce more accurate and smooth representations of the underlying surfaces compared to existing methods. Our dictionary learning framework is very general and would be straightforward to combine with many existing algorithms. In future work, we hope to investigate these combinations and also perform a more thorough study of the influence of the various parameters of our dictionary learning model on the computed surfaces.

---

[1] This dataset can be found at `http://vision.seas.harvard.edu/qsfs/Data.html`

# CHAPTER XI

# Conclusions and Future Work

In the first half of this thesis, we studied robust algorithms for decomposing data into the sum of low-rank and sparse matrices. Such robust PCA models are useful in practice because the low-rank component captures static or highly correlated features of the data—e.g., the background of a video whose vectorized frames are arranged as columns of a data matrix—and the sparse component captures dynamic or ephemeral features of the data—e.g., the foreground of a video—while disentangling these features from dense corruptions.

We began in Chapter III by arguing that conventional convex optimization-based robust PCA algorithms can lead to suboptimal low-rank components in practice, and we proposed a new robust PCA algorithm based on an optimal low-rank matrix estimator (OptShrink) to overcome these shortcomings. Our background subtraction and dynamic MRI reconstruction experiments showed that the proposed method produces more accurate and physically meaningful low-rank components compared to conventional robust PCA methods. Our proposed alternating minimization algorithm is data-driven in the sense that the low-rank update does not correspond to minimizing a particular fixed cost function. Empirically, our proposed method behaves well, but it would be interesting to study the algorithm from a theoretical perspective and establish some guarantees on the convergence of the iterates. It may also be fruitful to apply our proposed method in other inverse problem settings where conventional robust PCA methods have shown promise. We hope that this work motivates further study of non-convex low-rank matrix regularizers for applications with noisy data.

In Chapter IV, we studied the problem of robust foreground-background separation in more detail. In particular, we modified our robust PCA method from Chapter III to handle videos with freely moving cameras and arbitrary dense and sparse corruptions. The key contributions of our method are a pre-processing step that registers the frames of a video to a common (reference) perspective and a two-component sparsity model that relies on a weighted total-variation framework to capture the foreground of the scene and a typical $\ell_1$-based term to isolate sparse corruptions from the foreground. Our proposed method produces impressive foreground-background decompositions on videos with substantial camera motion and corruption—cases where state-of-the-art methods fail. In future work, we plan to improve our model by integrating the frame registration procedure as an additional alternating update in the model, which should improve the accuracy of the registration process on severely corrupted videos. We hope that this work will enable foreground-background separation in challenging real-world scenarios such as handheld camera video with significant motion and low-light applications such as nighttime surveillance, where poor lighting leads to significant noise in the video.

A common theme of our work is designing robust PCA algorithms that can produce accurate reconstructions from noisy data. In the noiseless setting, it is known that convex optimization-based robust PCA algorithms can provably decompose a low-rank plus sparse matrix into its constituent components. Much less is known, however, about the noisy setting, except the unsurprising fact that one cannot expect perfect recovery. We partially bridge this gap in Chapter V by providing a new analysis of the singular vectors of a thresholded low-rank plus sparse plus noise matrix. Our results show that one can reliably detect and remove outliers from a data matrix by applying hard thresholding and then estimating the underlying low-rank subspace to a certain (known) accuracy by computing the leading singular vectors of the outlier-rejected matrix. Our results can be viewed as a first-principles analysis of the first iteration of an alternating minimization algorithm for robust PCA, and, as such, they may be useful in establishing some theoretical convergence results for robust PCA in the noisy setting. We do not have a concrete recommendation in this direction, but perhaps our asymptotic results can be used to bound the distance from the low-rank and sparse iterates from their respective ground truth values after a single iteration. Such initialization bounds are often required to establish recovery results for optimization problems.

In the second half of this thesis, we turned our attention to sparse signal models based on adaptive dictionary learning. The overarching motivation of our work was that dictionary learning models possess important data representational capabilities that allow one to reconstruct signals from highly undersampled measurements. In particular, our focus was on proposing adaptive dictionary learning models with structured (e.g., low-rank) atoms and deriving efficient updates for solving them.

In Chapter VI, we proposed an adaptive dictionary learning framework (DINO-

KAT) for learning synthesis dictionaries with atoms that are low-rank upon appropriate reshaping into matrices. We proposed an efficient method for learning models of this form, and we demonstrated their suitability for inverse problems such as video inpainting and highly accelerated dynamic MRI reconstruction. We extended our (batch) method to the online setting in Chapter VII, which enables the processing of streaming measurements with efficient memory usage and computation. The proposed problems are highly non-convex, but we showed that the cost function values of our batch update scheme converge. In future work, it would be interesting to prove a similar convergence result for the online version of the algorithm; additionally, it would be valuable to strengthen these results to show that the iterate sequences themselves converge. We hope that this work will spur further research into dictionary learning methods with structured atoms, which may uncover local structure from highly contaminated data more effectively than unstructured dictionary models.

In Chapter VIII, we combined our work on improved low-rank updates for robust PCA from Chapter III and our adaptive dictionary learning-based methods from Chapter VI to form the LASSI algorithm. We focused on highly accelerated dynamic MRI reconstruction in this chapter, although it would be an interesting line of inquiry to apply LASSI in other inverse problem settings where robust PCA models are popular. We performed an extensive numerical study comparing LASSI variations such as conventional SVT-based (convex) low-rank updates versus OptShrink-based updates, $\ell_0$ versus $\ell_1$-based sparsity regularization, and low-rank versus unstructured dictionary atoms. Our analysis suggests that our proposed OptShrink-based updates and low-rank dictionary atoms are useful models for highly corrupted or undersampled data that may lead to more accurate reconstructions in practice. We hope that this work will impact clinical medical imaging systems and find fruitful use in other inverse problem settings.

Finally, in Chapters IX and X, we applied our adaptive dictionary learning models to photometric stereo. Specifically, we showed that dictionary learning-based regularization can be incorporated in various models and efficiently solved to generate accurate estimates of the normal vectors and surfaces of objects from a small number of corrupted images. Our dictionary learning strategy is quite general and could be combined with other state-of-the-art methods for robust photometric stereo to yield further improvements in reconstruction accuracy. Investigating these adaptations is an interesting future research direction. We hope that this work can be applied in practice to generate accurate 3D models of real objects with shiny surfaces, sharp edges, and other complex reflectance properties that current methods cannot handle.

# APPENDICES

# APPENDIX A

# Incorporating Missing Data into Existing Foreground-Background Separation Algorithms

In this appendix, we describe how we adapt the RPCA [1,31], DECOLOR [65], and TVRPCA [59] algorithms for our inpainting experiments in Chapter IV. Throughout, we use $Y \in \mathbb{R}^{mn \times p}$ to denote the matrix whose columns contain the vectorized frames of the input video with missing data.

## A.1 RPCA

The standard robust PCA [1, 31] method minimizes the cost

$$\min_{L,S} \; \tfrac{1}{2}\|Y - L - S\|_F^2 + \lambda_L \|L\|_\star + \lambda_S \|S\|_1, \tag{A.1}$$

where $L$ is the low-rank background component and $S$ is the sparse foreground component. We incorporate a missing data mask into (A.1) analogously to our approach in our proposed PRPCA method; that is, we solve the modified RPCA problem

$$\min_{L,S} \; \tfrac{1}{2}\|\mathcal{P}_M(Y - L - S)\|_F^2 + \lambda_L \|L\|_\star + \lambda_S \|S\|_1, \tag{A.2}$$

where the missing data mask $M \in \{0,1\}^{mn \times d}$ with entries

$$M_{ij} = \begin{cases} 0 & Y_{ij} \text{ is missing} \\ 1 & Y_{ij} \text{ is observed} \end{cases} \tag{A.3}$$

omits unobserved pixels from the data fidelity term in (A.1). Applying the same proximal gradient strategy to (A.2) as for the standard RPCA problem (A.1) leads to the updates

$$\begin{aligned} Z^{k+1} &= \mathcal{P}_M(L^k + S^k - Y) \\ L^{k+1} &= \mathbf{SVT}_{\tau\lambda_L}(L^k - \tau Z^{k+1}) \\ S^{k+1} &= \mathbf{soft}_{\tau\lambda_S}(S^k - \tau Z^{k+1}), \end{aligned} \tag{A.4}$$

with constant step size $\tau^k = \tau < 1$ sufficing to guarantee convergence [25]. One can view the updates (A.4) as a special case of our proposed PRPCA updates when the camera is static (so that no frame registration is performed) and the TV regularization parameter tends to infinity.

## A.2 DECOLOR

The DECOLOR method minimizes the cost from Equation (20) of [65], which, in our notation, is

$$\min_{\tau,L,S} \tfrac{1}{2}\|\mathcal{P}_{S^\perp}(Y \circ \tau - L)\|_F^2 + \alpha\|L\|_\star + \beta\|S\|_1 + \gamma\mathbf{TV}(S), \tag{A.5}$$

where $L$ is the low-rank (registered) background, $S_{ij} \in \{0,1\}$ is the (registered) foreground mask, $S^\perp$ is the orthogonal complement of $S$, $\tau$ are the 2D parameteric transforms that register the input frames $Y$, and $\mathbf{TV}(\cdot)$ denotes unweighted anisotropic total variation.

The DECOLOR algorithm proceeds by alternating between updating $\tau$, $L$, and $S$ sequentially with all other variables held fixed. The $\tau$ subproblem is approximately solved using an iterative strategy where one linearizes (A.5) with respect to $\tau$, solves the resulting weight least-squares problem, and then repeats the process to refine $\tau$. The $L$ subproblem for (A.5) is a missing data version of the proximal operator for the nuclear norm and can be approximately solved by performing a few iterations of the SOFT-IMPUTE algorithm [242]. Finally, the $S$ subproblem is a Markov random field problem that is solved exactly via graph cuts [65].

At any given step of the DECOLOR algorithm, the matrix $Y \circ \tau$ denotes the current estimate of the registered frames, so the appropriate missing data mask to consider is

$$M_{ij} = \begin{cases} 0 & \text{if } [Y \circ \tau]_{ij} \text{ is missing} \\ 1 & \text{if } [Y \circ \tau]_{ij} \text{ is observed,} \end{cases} \tag{A.6}$$

which implicitly depends on the current value of the parameteric transformations $\tau$.

Thus, to incorporate this mask into (A.5), we solve the modified problem

$$\min_{\tau,L,S} \tfrac{1}{2}\|\mathcal{P}_{S^\perp \odot M}(Y \circ \tau - L))\|_F^2 + \alpha\|L\|_\star + \beta\|S\|_1 + \gamma\mathbf{TV}(S), \tag{A.7}$$

where $\odot$ denotes elementwise multiplication. Our modified problem (A.7) omits unobserved data in the registered perspective defined by $\tau$ from the data fidelity term. Note that we have the relation $\mathcal{P}_{S^\perp \odot M}(\cdot) = \mathcal{P}_{S^\perp}(\mathcal{P}_M(\cdot)) = \mathcal{P}_M(\mathcal{P}_{S^\perp}(\cdot))$, which can be used to appropriately isolate $S$ in the projection operators when minimizing (A.7) with respect to $S$.

The same alternating minimization algorithm proposed in Algorithm 1 of [65] can be extended to solve (A.7). Indeed, after linearizing (A.7) around $\tau$, the inner iterations for updating $\tau$ can be written as

$$\tau^{k+1} = \tau^k + \arg\min_{\Delta\tau} \ \|\mathcal{P}_{S^\perp \odot M}(Y \circ \tau - L + J_{\tau^k}\Delta\tau)\|_F^2, \tag{A.8}$$

where $J_\tau$ denotes the Jacobian matrix of (A.7) with respect to $\tau$. The iteration (A.8) is still a weighted least squares problem that can be solved in closed-form. The $L$ subproblem can be approximately solved by performing a few inner iterations of the SOFT-IMPUTE updates

$$L^{k+1} = \mathbf{SVT}_\alpha\left(\mathcal{P}_{S^\perp \odot M}(Y \circ \tau) + \mathcal{P}_{(S^\perp \odot M)^\perp}(L^k)\right). \tag{A.9}$$

Finally, the $S$ subproblem can be written as

$$\min_S \sum_{ij} \left(\beta - \tfrac{1}{2}[\mathcal{P}_M(Y \circ \tau - L)]_{ij}^2\right) S_{ij} + \gamma\mathbf{TV}(S), \tag{A.10}$$

which can be solved using the same graph cuts algorithm from [65] with residual matrix $\mathcal{P}_M(Y \circ \tau - L)$ in place of $Y \circ \tau - L$.

Aside from the modified subproblem updates in (A.8)-(A.10), we retain all other features of the DECOLOR method as outlined in Algorithm 1 of [65]. Note that the above updates reduce to the original DECOLOR algorithm when $M$ is the all-ones matrix (no missing data).

## A.3 TVRPCA

The TVRPCA method minimizes the cost from Equation (7) of [59], which, in our notation, is

$$\begin{aligned}\min_{L,G,E,S} \quad & \|L\|_\star + \lambda_1\|G\|_1 + \lambda_2\|E\|_1 + \lambda_3\mathbf{TV}(S) \\ \text{s.t.} \quad & Y = L + G,\ G = E + S.\end{aligned} \tag{A.11}$$

In (A.11), $L$ is the low-rank background component and $G$ is a residual component, which is further decomposed into a smooth foreground component $S$ and a sparse error term $E$. The authors propose to solve (A.11) by applying an alternating minimization scheme to the augmented Lagrangian of (A.11):

$$\begin{aligned}&\mathcal{L}_\mu(L,G,E,S,X,Z) = \\ &\quad \|L\|_\star + \lambda_1\|G\|_1 + \lambda_2\|E\|_1 + \lambda_3\mathbf{TV}(S) \\ &\qquad +\tfrac{\mu}{2}\|Y - L - G\|_F^2 + \langle X,\ Y - L - G\rangle \\ &\qquad\quad +\tfrac{\mu}{2}\|G - E - S\|_F^2 + \langle Z,\ G - E - S\rangle.\end{aligned} \tag{A.12}$$

In particular, in [59] one sequentially updates each component $\{L,G,E,S,X,Z\}$ by minimizing (A.12) with all other components held fixed.

We incorporate a missing data mask into (A.11) by solving the related problem

$$\begin{aligned}\min_{L,G,E,S} \quad & \|L\|_\star + \lambda_1\|G\|_1 + \lambda_2\|E\|_1 + \lambda_3\mathbf{TV}(S) \\ \text{s.t.} \quad & \mathcal{P}_M(Y) = \mathcal{P}_M(L + G),\ G = E + S,\end{aligned} \tag{A.13}$$

which omits equality constraints involving unobserved pixels from (A.13). The augmented Lagrangian for (A.13) is

$$\begin{aligned}&\mathcal{L}_\mu(L,G,E,S,X,Z) = \\ &\quad \|L\|_\star + \lambda_1\|G\|_1 + \lambda_2\|E\|_1 + \lambda_3\mathbf{TV}(S) \\ &\qquad +\tfrac{\mu}{2}\|\mathcal{P}_M(Y - L - G)\|_F^2 + \langle X,\ \mathcal{P}_M(Y - L - G)\rangle \\ &\qquad\quad +\tfrac{\mu}{2}\|G - E - S\|_F^2 + \langle Z,\ G - E - S\rangle,\end{aligned} \tag{A.14}$$

and we solve (A.13) by applying the same alternating minimization strategy to (A.14) from the TVRPCA method. The subproblem updates for minimizing (A.14) are the same as those derived in Section III-C of [59] for the original cost (A.11), with the following modifications.[1] Fist, the $L$ subproblem for (A.14) can be written in the form of a SOFT-IMPUTE problem [242], so it can be approximately solved using a

---

[1] In the modified $L$ and $G$ updates, we assume that the initial $X^0$ satisfies $\mathcal{P}_{M^\perp}(X^0) = 0$, which is true when one chooses $X^0 = 0$.

few inner iterations of the updates

$$L^{k+1} = \mathbf{SVT}_{\frac{1}{\mu}}(\mathcal{P}_M(Y - G + \tfrac{1}{\mu}X) + \mathcal{P}_{M^\perp}(L^k)). \tag{A.15}$$

After suitable manipulation, the $G$ subproblem for (A.14) can be written as two disjoint soft-thresholding problems with different shrinkage parameters. Indeed, the minimizer $\hat{G}$ of (A.14) with respect to $G$ can be written as

$$\begin{aligned} \mathcal{P}_M(\hat{G}) &= \mathcal{P}_M\left[\mathbf{soft}_{\frac{\lambda_1}{2\mu}}\left(\tfrac{1}{2}(Y - L + E + S) + \tfrac{1}{2\mu}(X - Z))\right)\right] \\ \mathcal{P}_{M^\perp}(\hat{G}) &= \mathcal{P}_{M^\perp}\left[\mathbf{soft}_{\frac{\lambda_1}{\mu}}\left(E + S - \tfrac{1}{\mu}Z\right)\right]. \end{aligned} \tag{A.16}$$

Finally, the $X$ subproblem for (A.14) can be solved exactly using the simple update

$$X \leftarrow X + \mu\mathcal{P}_M(Y - L - G). \tag{A.17}$$

All other subproblems for (A.13) are identical to the method outlined in Section III-C of [59] for the original cost (A.11). Note that the above updates reduce to the original TVRPCA algorithm when $M$ is the all-ones matrix (no missing data).

# APPENDIX B

# Useful Results for Appendices C, D, and E

This appendix collects two useful results that we use in the proofs of Appendices C, D, and E. We use $C$, $\gamma > 0$ to denote arbitrary absolute constants whose values may change from line to line.

**Proposition B.1** (Latala). *Suppose $X \in \mathbb{R}^{m\times n}$ is a random matrix with $m = O(n)$ and independent zero-mean entries such that $\mathbb{E}X_{ij}^2 \le A^2$ and $\mathbb{E}X_{ij}^4 \le B^4$. Then*

$$\mathbb{E}\sigma_1(X) \le C\sqrt{n}\max(A, B). \tag{B.1}$$

*Proof.* Applying Theorem 2 of [243], we have

$$\begin{aligned}\mathbb{E}\sigma_1(X) &\le C\left[\max_i \sqrt{\textstyle\sum_j \mathbb{E}X_{ij}^2} + \max_j \sqrt{\textstyle\sum_i \mathbb{E}X_{ij}^2} + \sqrt[4]{\textstyle\sum_{ij} \mathbb{E}X_{ij}^4}\right] \\ &\le C\left[\sqrt{nA^2} + \sqrt{mA^2} + \sqrt[4]{mnB^4}\right] \\ &\le C\sqrt{n}\max(A, B).\end{aligned} \tag{B.2}$$

□

**Proposition B.2** (Talagrand). *Fix $K > 0$ and suppose that $X_{ij}$ are independent random variables such that $|X_{ij}| \le K$ for all $1 \le i \le m$ and $1 \le j \le n$. Then, for any $\epsilon > 0$, one has*

$$\mathbb{P}(|\sigma_1(X) - \mathbb{E}\sigma_1(X)| \ge \epsilon) \le C\exp(-\gamma\epsilon^2/K^2). \tag{B.3}$$

*Proof.* Follows from [244] and the observation that $\sigma_1 : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a convex 1-Lipschitz function. □

# APPENDIX C

# Proof of Theorem V.7

Throughout this appendix, we use $C$, $\gamma > 0$ to denote arbitrary absolute constants whose values may change from line to line. By construction of $\widetilde{X}^\star$ from (5.15), the non-zero elements of $S$ are deterministically canceled, so we can write

$$\begin{aligned}\widetilde{X}^\star &\overset{\mathrm{d}}{=} (L+G)\odot M \\ &= L\odot M + G\odot M \\ &= \mathbb{E}[L\odot M] + \underbrace{G\odot M}_{=:G^\star} + \underbrace{(L\odot M - \mathbb{E}[L\odot M])}_{=:\Delta_L^\star},\end{aligned} \tag{C.1}$$

where

$$M_{ij} = \begin{cases} 0 & \text{with probability } p_s, \\ 1 & \text{with probability } 1-p_s. \end{cases} \tag{C.2}$$

From the definition of $M$,

$$\mathbb{E}[L\odot M] = (1-p_s)L, \tag{C.3}$$

and the elements of $G^\star$ are independent zero-mean random variables with variance $(1-p_s)\sigma^2/n$. Therefore, provided that $\sigma_1(\Delta_L^\star) \xrightarrow{\text{a.s.}} 0$, we can apply Theorem 2.9 and Section 3.1 of [27] to conclude that the asymptotic accuracies of the left and right singular vectors of $\widetilde{X}^\star$ by given by (5.11) and (5.12), respectively, with effective SNR

$$\overline{\theta}_i^\star = \lim_{n\to\infty} \frac{(1-p_s)\theta_i}{\sqrt{(1-p_s)\sigma^2}} = \lim_{n\to\infty} \frac{\sqrt{1-p_s}\;\theta_i}{\sigma}. \tag{C.4}$$

To establish $\sigma_1(\Delta_L^\star) \xrightarrow{\text{a.s.}} 0$, observe that

$$(\Delta_L^\star)_{ij} = \begin{cases} p_s L_{ij} & \text{with probability } 1 - p_s \\ -(1-p_s)L_{ij} & \text{with probability } p_s, \end{cases} \tag{C.5}$$

and therefore that $\mathbb{E}(\Delta_L^\star)_{ij} = 0$. Now, let us define

$$\ell = \ell_n := \max_{ij} |L_{ij}|, \tag{C.6}$$

so that, by definition of $\Delta_L^\star$, we have

$$\mathbb{E}|(\Delta_L^\star)_{ij}|^\alpha \leq \ell^\alpha \tag{C.7}$$

for any $\alpha > 0$. Moreover, by Assumption V.1, we have

$$\ell \leq \frac{\log n \text{ factors}}{n}, \tag{C.8}$$

so we can apply Proposition B.1 to conclude

$$\begin{aligned} \mathbb{E}\sigma_1(\Delta_L^\star) &\leq C\sqrt{n}\ell \\ &\leq C\ \frac{\log n \text{ factors}}{\sqrt{n}}. \end{aligned} \tag{C.9}$$

Also, since $|(\Delta_L^\star)_{ij}| \leq \ell$, we can apply Proposition B.2 to conclude

$$\begin{aligned} \mathbb{P}(|\sigma_1(\Delta_L^\star) - \mathbb{E}\sigma_1(\Delta_L^\star)| \geq \epsilon) &\leq C\exp(-\gamma\epsilon^2/\ell^2) \\ &\leq C\exp\left(\frac{-\gamma\epsilon^2 n^2}{\log n \text{ factors}}\right). \end{aligned} \tag{C.10}$$

The result follows by the Borel-Cantelli lemma. □

# APPENDIX D

# Proof of Theorem V.8 for Hard Thresholding

In this appendix, we prove Theorem V.8 for the hard thresholding estimator $\widetilde{X}_{\tau}^{\mathrm{HT}}$ defined by (5.20). That is, we prove the first row of Table 5.1. The proof is organized as follows. In Section D, we introduce common notation that will be used throughout this appendix. In Sections D - D, we derive some preliminary results, and in Section D we combine the preliminary results to complete the proof.

## D.1 Notation

We use $C$, $\gamma$, $N > 0$ to denote arbitrary absolute constants whose values may change from line to line. Also, let us define

$$\ell = \ell_n := \max_{ij} |L_{ij}|, \tag{D.1}$$

for which, by Assumption V.1, we have the element-wise incoherence condition

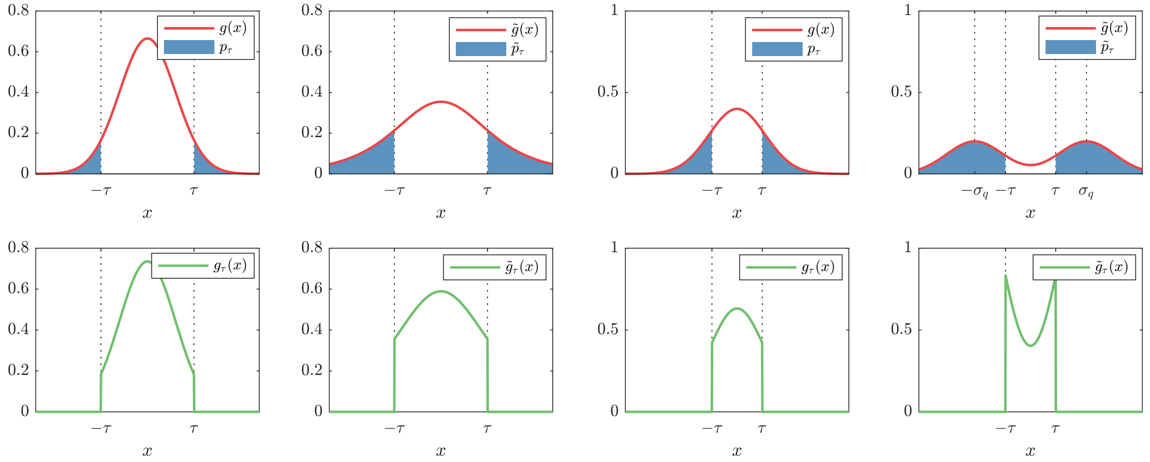
$$\ell \leq \frac{\log n \text{ factors}}{n}. \tag{D.2}$$

Define the random variable

$$\widetilde{g} = g + q \tag{D.3}$$

that we henceforth refer to as the *outlier-noise distribution*. Also, define the tail probabilities

$$\begin{aligned} p_{\tau} &:= \mathbb{P}(|g| \geq \tau), \\ \widetilde{p}_{\tau} &:= \mathbb{P}(|\widetilde{g}| \geq \tau). \end{aligned} \tag{D.4}$$



(a) Gaussian noise and Laplacian outliers. (b) Gaussian noise and Rademacher outliers.

Figure D.1: Illustration of the notation defined in Section D for two possible outlier distributions.

Let $g_\tau$ and $\widetilde{g}_\tau$ be the random variables formed by restricting $g$ and $\widetilde{g}$, respectively, to the interval $[-\tau, \tau]$. By symmetry, we have $\mathbb{E}g_\tau = \mathbb{E}\widetilde{g}_\tau = 0$, and we define

$$\begin{aligned} \sigma_\tau^2 &:= \mathbb{E}g_\tau^2, \\ \widetilde{\sigma}_\tau^2 &:= \mathbb{E}\widetilde{g}_\tau^2. \end{aligned} \tag{D.5}$$

Finally, we define the tail mixture probability

$$p := (1 - p_s)p_\tau + p_s\widetilde{p}_\tau. \tag{D.6}$$

Figure D.1 illustrates the notation introduced in this section for two possible outlier distributions.

## D.2 Hard Thresholding Estimator

Note that we can explicitly write the entries of $\widetilde{X}_\tau^{\mathrm{HT}}$ as

$$(\widetilde{X}_\tau^{\mathrm{HT}})_{ij} = \begin{cases} L_{ij} + G_{ij} & \text{if } S_{ij} = 0 \text{ and } |L_{ij} + G_{ij}| \le \tau, \\ L_{ij} + G_{ij} + S_{ij} & \text{if } S_{ij} \ne 0 \text{ and } |L_{ij} + G_{ij} + S_{ij}| \le \tau, \\ 0 & \text{if } |L_{ij} + G_{ij} + S_{ij}| > \tau. \end{cases} \tag{D.7}$$

The statistics of the matrix $\widetilde{X}_\tau^{\mathrm{HT}}$ are complicated by the presence of $L_{ij}$ in the condition statements of (D.7), which make the probability of each case occurring *entry-dependent*. Removing $L_{ij}$ from the condition statements of (D.7) yields a closely-related random matrix $\overline{X}_{\tau}^{\mathrm{HT}}$ with elements

$$(\overline{X}_{\tau}^{\mathrm{HT}})_{ij} := \begin{cases} L_{ij} + G_{ij} & \text{if } S_{ij} = 0 \text{ and } |G_{ij}| \leq \tau, \\ L_{ij} + G_{ij} + S_{ij} & \text{if } S_{ij} \neq 0 \text{ and } |G_{ij} + S_{ij}| \leq \tau, \\ 0 & \text{if } |G_{ij} + S_{ij}| > \tau \end{cases} \tag{D.8}$$

that is easier to analyze because the case statements in (D.8) occur with fixed probabilities independent of $i$ and $j$. In Section D, we establish that the modeling error[1]

$$\Delta_X^{\mathrm{HT}} := \widetilde{X}_{\tau}^{\mathrm{HT}} - \overline{X}_{\tau}^{\mathrm{HT}} \tag{D.9}$$

is small in the sense that $\sigma_1(\Delta_X^{\mathrm{HT}}) \xrightarrow{\text{a.s.}} 0$. Therefore, we can determine the asymptotic performance of the estimator $\widetilde{X}_{\tau}^{\mathrm{HT}}$ by analyzing the asymptotics of the random matrix $\overline{X}_{\tau}^{\mathrm{HT}}$. We analyze $\overline{X}_{\tau}^{\mathrm{HT}}$ in Section D, and then we revisit the modeling error $\Delta_X^{\mathrm{HT}}$ in Section D.

## D.3 Equivalent Random Matrix Model

In this section, we analyze the random matrix model $\overline{X}_{\tau}^{\mathrm{HT}}$ defined by (D.8). Using the notation from Section D, we have

$$\begin{aligned} \mathbb{P}(S_{ij} = 0 \text{ and } |G_{ij}| \leq \tau) &= (1 - p_s)(1 - p_\tau), \\ \mathbb{P}(S_{ij} \neq 0 \text{ and } |G_{ij} + S_{ij}| \leq \tau) &= p_s(1 - \widetilde{p}_\tau), \\ \mathbb{P}(|G_{ij} + S_{ij}| > \tau) &= p, \end{aligned} \tag{D.10}$$

and so the distributions of the elements of $\overline{X}_{\tau}^{\mathrm{HT}}$ from (D.8) are

$$(\overline{X}_{\tau}^{\mathrm{HT}})_{ij} \overset{\mathrm{d}}{=} \begin{cases} L_{ij} + (G_\tau)_{ij} & \text{with probability } (1 - p_s)(1 - p_\tau), \\ L_{ij} + (\widetilde{G}_\tau)_{ij} & \text{with probability } p_s(1 - \widetilde{p}_\tau), \\ 0 & \text{with probability } p, \end{cases} \tag{D.11}$$

---

[1] The quantity $\Delta_X^{\mathrm{HT}}$ is a random matrix defined implicitly by the random matrices in Section 5.2 and the thresholding schemes defined by $\widetilde{X}_{\tau}^{\mathrm{HT}}$ and $\overline{X}_{\tau}^{\mathrm{HT}}$.

where $(G_\tau)_{ij}$ are drawn i.i.d. from $g_\tau$ and $(\widetilde{G}_\tau)_{ij}$ are drawn i.i.d. from $\widetilde{g}_\tau$. From (D.11), we have

$$\mathbb{E}\overline{X}_\tau^{\mathrm{HT}} = (1-p)L, \tag{D.12}$$

and so

$$\begin{aligned}\overline{X}_\tau^{\mathrm{HT}} &= \mathbb{E}\overline{X}_\tau^{\mathrm{HT}} + (\overline{X}_\tau^{\mathrm{HT}} - \mathbb{E}\overline{X}_\tau^{\mathrm{HT}}) \\ &= (1-p)L + G_\tau^{\mathrm{HT}} + \Delta_L,\end{aligned} \tag{D.13}$$

where

$$(G_\tau^{\mathrm{HT}})_{ij} = \begin{cases} (G_\tau)_{ij} & \text{with probability } (1-p_s)(1-p_\tau), \\ (\widetilde{G}_\tau)_{ij} & \text{with probability } p_s(1-\widetilde{p}_\tau), \\ 0 & \text{with probability } p, \end{cases} \tag{D.14}$$

and

$$(\Delta_L)_{ij} = \begin{cases} pL_{ij} & \text{with probability } 1-p, \\ -(1-p)L_{ij} & \text{with probability } p. \end{cases} \tag{D.15}$$

Theorem D.1 establishes almost sure convergence of the spectral norm of the error matrix $\Delta_L$ to zero.

**Theorem D.1.** *We have*

$$\mathbb{E}\sigma_1(\Delta_L) \to 0, \tag{D.16}$$

*and, consequently,* $\sigma_1(\Delta_L) \xrightarrow{a.s.} 0$.

*Proof.* From (D.15), we have $\mathbb{E}(\Delta_L)_{ij} = 0$, and, by Assumption V.1, we have

$$\mathbb{E}|(\Delta_L)_{ij}|^\alpha \le |L_{ij}|^\alpha \le \ell^\alpha \tag{D.17}$$

for any $\alpha > 0$. Thus, applying Proposition B.1 gives

$$\begin{aligned}\mathbb{E}\sigma_1(\Delta_L) &\le C\sqrt{n}\ell \\ &\le C\frac{\log n \text{ factors}}{\sqrt{n}},\end{aligned} \tag{D.18}$$

where in the second inequality we have applied the incoherence bound (D.2). This establishes (D.16). Moreover, since $|(\Delta_L)_{ij}| \le |L_{ij}| \le \ell$, we can apply Proposition B.2 to conclude that

$$\begin{aligned}\mathbb{P}(|\sigma_1(\Delta_L) - \mathbb{E}\sigma_1(\Delta_L)| \ge \epsilon) &\le C\exp(-\gamma\epsilon^2/\ell^2) \\ &\le C\exp\left(\frac{-\gamma\epsilon^2 n^2}{\log n \text{ factors}}\right)\end{aligned} \tag{D.19}$$

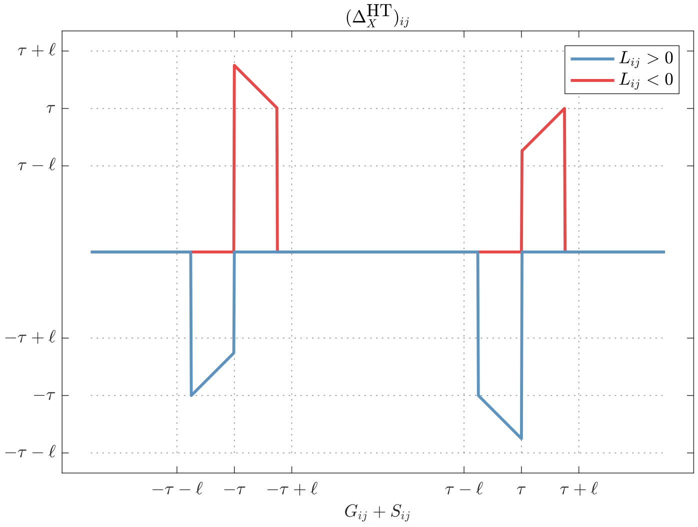


Figure D.2: Graphical depiction of $(\Delta_X^{\mathrm{HT}})_{ij}$ as a function of $G_{ij} + S_{ij}$ for two fixed values of $L_{ij}$.

for all $\epsilon > 0$. The result follows by the Borel-Cantelli lemma. $\square$

## D.4 Modeling Error Bounds

In this section, we analyze the modeling error $\Delta_X^{\mathrm{HT}}$ defined by (D.9). Towards this end, observe that one can explicitly describe the entries of $\Delta_X^{\mathrm{HT}}$ as

$$\left(\Delta_X^{\mathrm{HT}}\right)_{ij} = \begin{cases} -(L_{ij}+G_{ij}) & \text{if } S_{ij}=0,\ |G_{ij}| \le \tau,\ \text{and } |L_{ij}+G_{ij}| > \tau, \\ L_{ij}+G_{ij} & \text{if } S_{ij}=0,\ |G_{ij}| > \tau,\ \text{and } |L_{ij}+G_{ij}| \le \tau, \\ -(L_{ij}+G_{ij}+S_{ij}) & \text{if } S_{ij}\neq 0,\ |G_{ij}+S_{ij}| \le \tau,\ \text{and } |L_{ij}+G_{ij}+S_{ij}| > \tau, \\ L_{ij}+G_{ij}+S_{ij} & \text{if } S_{ij}\neq 0,\ |G_{ij}+S_{ij}| > \tau,\ \text{and } |L_{ij}+G_{ij}+S_{ij}| \le \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{D.20}$$

Intuitively, $\Delta_X^{\mathrm{HT}}$ captures the cases when an entry of $\widetilde{X}_\tau^{\mathrm{HT}}$ triggers the hard thresholding function but the corresponding entry of $\overline{X}_\tau^{\mathrm{HT}}$ does not, and vice versa. Figure D.2 plots $(\Delta_X^{\mathrm{HT}})_{ij}$ as a function of the underlying random variable $G_{ij} + S_{ij}$.

The following lemma establishes a bound on the expected magnitude of the elements of $\Delta_X^{\mathrm{HT}}$, using the following notation:

$$\begin{aligned} \mathcal{I} &= [\tau - \ell,\ \tau + \ell]\,, \\ a &= \mathbb{P}(|g| \in \mathcal{I}), \\ b &= \mathbb{P}(|\widetilde{g}| \in \mathcal{I}). \end{aligned} \tag{D.21}$$

**Lemma D.2.** *We have*

$$\mathbb{E}|(\Delta_X^{HT})_{ij}|^\alpha \le [(1-p_s)a + p_s b](\tau + \ell)^\alpha \tag{D.22}$$

*for any $\alpha > 0$ and $n \ge N$.*

*Proof.* Choose $N$ large enough that $\tau_n \ge \ell_n$ for all $n \ge N$, which, comparing (5.23) and (D.2), must be possible. Note that this implies that the interval $\mathcal{I}$ is nonnegative. Upon inspection of (D.20) and Figure D.2, we see that $(\Delta_X^{\mathrm{HT}})_{ij} \neq 0$ only when $|L_{ij} + G_{ij} + S_{ij}|$ and $|G_{ij} + S_{ij}|$ lie on opposite sides of $\tau$. Moreover, since $|L_{ij}| \le \ell$, in such cases we have either $|\widetilde{X}_\tau^{\mathrm{HT}}|_{ij} \in \mathcal{I}$ and $|\overline{X}_\tau^{\mathrm{HT}}|_{ij} = 0$ or vice versa. Therefore, $|(\Delta_X^{\mathrm{HT}})_{ij}| \le \tau + \ell$, and a sufficient condition for $(\Delta_X^{\mathrm{HT}})_{ij} \neq 0$ is $|G_{ij} + S_{ij}| \in \mathcal{I}$. In other words, we have that

$$|\Delta_X^{\mathrm{HT}}|_{ij} \le \begin{cases} \tau + \ell & \text{with probability } (1-p_s)a, \\ \tau + \ell & \text{with probability } p_s b, \\ 0 & \text{otherwise.} \end{cases} \tag{D.23}$$

for $n \ge N$. Computing the $\alpha$-moments of (D.23) yields the desired result. □

The following corollary refines the bound from Lemma D.2 using the properties of our model parameters.

**Corollary D.3.** *We have*

$$\mathbb{E}|(\Delta_X^{HT})_{ij}|^\alpha \le C^\alpha \tau^\alpha \left(\ell + \exp(-\gamma n \tau^2)\right) \tag{D.24}$$

*for any $\alpha > 0$ and $n \ge N$.*

*Proof.* Choose $N$ large enough that $\tau_n \ge 2\ell_n$ for all $n \ge N$, which, comparing (5.23) and (D.2), must be possible. Then, continuing from Lemma D.2, we have

$$\begin{aligned}\mathbb{E}|(\Delta_X^{\mathrm{HT}})_{ij}|^\alpha &\leq [(1-p_s)a + p_s b](\tau+\ell)^\alpha \\ &\leq (a+b)(\tau+\ell)^\alpha \\ &\leq (3/2)^\alpha (a+b)\tau^\alpha, \ \forall n \geq N.\end{aligned} \tag{D.25}$$

We also have

$$\begin{aligned} a &= \mathbb{P}(|g| \in \mathcal{I}) \\ &\leq \mathbb{P}(|g| \geq \tau - \ell) \\ \text{since } \tau \geq 2\ell \rightarrow &\leq \mathbb{P}(|g| \geq \tau/2) \quad , \ \forall n \geq N \\ \text{Assumption } V.2 \rightarrow &\leq 2\exp(-\gamma n \tau^2) \ \ , \ \forall n \geq N.\end{aligned} \tag{D.26}$$

Also, by the symmetry of $g$ and $q$, we have

$$\begin{aligned} b/2 &= \mathbb{P}(q+g \in \mathcal{I}) \\ &= \mathbb{P}(q+g \in \mathcal{I} \mid |g| \leq \tau)\mathbb{P}(|g| \leq \tau) + \mathbb{P}(q+g \in \mathcal{I} \mid |g| > \tau)\mathbb{P}(|g| > \tau) \\ &\leq \mathbb{P}(q+g \in \mathcal{I} \mid |g| \leq \tau) + \mathbb{P}(|g| \geq \tau),\end{aligned} \tag{D.27}$$

where, by Assumption V.2, we have

$$\mathbb{P}(|g| \geq \tau) \leq 2\exp(-\gamma n \tau^2). \tag{D.28}$$

By the independence of $g$ and $q$, we have

$$\begin{aligned}\mathbb{P}(q+g \in \mathcal{I} \mid |g| \leq \tau) &= \mathbb{P}(q + g_\tau \in \mathcal{I}) \\ &= \mathbb{E}\left[\mathbb{1}_{\{q+g_\tau \in \mathcal{I}\}}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{q \in \mathcal{I} - g_\tau\}} \mid g_\tau\right]\right] \\ &= \mathbb{E}\left[\mathbb{P}(q \in \mathcal{I} - g_\tau)\right],\end{aligned} \tag{D.29}$$

where $\mathcal{I} - x$ denotes shifting the interval $\mathcal{I}$ by $x \in \mathbb{R}$. Since $\tau \rightarrow 0$, the shifted interval $\mathcal{I} - g_\tau$ will lie in the neighborhood of zero for all sufficiently large $n$ uniformly over all realizations of $g_\tau$. Thus, since $\mathcal{I} - g_\tau$ has width $2\ell$, we can apply Lemma V.3 to conclude that

$$\mathbb{P}(q \in \mathcal{I} - g_\tau) \leq C\ell, \ \forall n \geq N, \tag{D.30}$$

and so

$$\mathbb{P}(q+g \in \mathcal{I} \mid |g| \leq \tau) = \mathbb{E}\left[\mathbb{P}(q \in \mathcal{I} - g_\tau)\right] \leq C\ell, \ \forall n \geq N. \tag{D.31}$$

Combining (D.25)-(D.31) gives the desired result. □

The following proposition applies Corollary D.3 to bound the spectral norm of $\mathbb{E}\Delta_X^{\mathrm{HT}}$.

**Proposition D.4.** *We have*

$$\sigma_1\left(\mathbb{E}\Delta_X^{HT}\right) \leq Cn\tau\left(\ell + \exp(-\gamma n\tau^2)\right) \tag{D.32}$$

*for $n \geq N$.*

*Proof.* Applying Corollary D.3 with $\alpha = 1$ yields

$$\begin{aligned} |\mathbb{E}(\Delta_X^{\mathrm{HT}})_{ij}| &\leq \mathbb{E}|(\Delta_X^{\mathrm{HT}})_{ij}| \\ &\leq D_n := C\tau\left(\ell + \exp(-\gamma n\tau^2)\right), \ \forall n \geq N. \end{aligned} \tag{D.33}$$

Therefore we have the bound

$$\begin{aligned} \sigma_1\left(\mathbb{E}\Delta_X^{\mathrm{HT}}\right) &\leq \sqrt{\|\mathbb{E}\Delta_X^{\mathrm{HT}}\|_1\|\mathbb{E}\Delta_X^{\mathrm{HT}}\|_\infty} \\ &\leq \sqrt{(D_n m)(D_n n)} \qquad , \ \forall n \geq N \\ &\leq CD_n n \qquad , \ \forall n \geq N, \end{aligned} \tag{D.34}$$

where the last inequality follows from the fact that $m/n \to c \in (0,1]$. □

The following proposition bounds the deviation of $\Delta_X^{\mathrm{HT}}$ from its mean.

**Proposition D.5.** *We have*

*(i)* $$\mathbb{E}\sigma_1(\Delta_X^{HT} - \mathbb{E}\Delta_X^{HT}) \leq C\sqrt{n}\tau\left(\sqrt{\ell} + \exp(-\gamma n\tau^2)\right), \ \forall n \geq N, \tag{D.35}$$

*and for all $\epsilon > 0$:*

*(ii)* $$\mathbb{P}(|\sigma_1(\Delta_X^{HT} - \mathbb{E}\Delta_X^{HT}) - \mathbb{E}\sigma_1(\Delta_X^{HT} - \mathbb{E}\Delta_X^{HT})| \geq \epsilon) \leq C\exp(-\gamma\epsilon^2\tau^{-2}), \ \forall n \geq N. \tag{D.36}$$

*Proof.* For any $\beta \in \mathbb{N}$, we have

$$\begin{aligned}
\mathbb{E}\left[\left((\Delta_X^{\mathrm{HT}})_{ij} - \mathbb{E}(\Delta_X^{\mathrm{HT}})_{ij}\right)^{\beta}\right] &\leq \mathbb{E}\left[\sum_{k=0}^{\beta}\binom{\beta}{k}|(\Delta_X^{\mathrm{HT}})_{ij}|^{k}|\mathbb{E}(\Delta_X^{\mathrm{HT}})_{ij}|^{\beta-k}\right] \\
&\leq \sum_{k=0}^{\beta}\binom{\beta}{k}\mathbb{E}|(\Delta_X^{\mathrm{HT}})_{ij}|^{k}\mathbb{E}|(\Delta_X^{\mathrm{HT}})_{ij}|^{\beta-k} \\
\text{Corollary D.3} \rightarrow &\leq \left[\sum_{k=0}^{\beta}\binom{\beta}{k}\right] C^{\beta}\tau^{\beta}(\ell + \exp(-\gamma n\tau^2))^2 \quad , \forall n \geq N. \\
&= C^{\beta}\tau^{\beta}(\ell + \exp(-\gamma n\tau^2))^2 \quad , \forall n \geq N.
\end{aligned} \tag{D.37}$$

Now, applying Proposition B.1 with the parameters $A := C\tau(\ell + \exp(-\gamma n\tau^2))$ and $B := C\tau\sqrt{\ell + \exp(-\gamma n\tau^2)}$ as suggested by (D.37) yields

$$\begin{aligned}
\mathbb{E}\sigma_1(\Delta_X^{\mathrm{HT}} - \mathbb{E}\Delta_X^{\mathrm{HT}}) &\leq C\sqrt{n}\tau \max\left(\ell + \exp(-\gamma n\tau^2), \sqrt{\ell + \exp(-\gamma n\tau^2)}\right) \quad , \forall n \geq N, \\
&\leq C\sqrt{n}\tau\sqrt{\ell + \exp(-\gamma n\tau^2)} \quad , \forall n \geq N, \\
&\leq C\sqrt{n}\tau\left(\sqrt{\ell} + \exp(-\gamma n\tau^2)\right) \quad , \forall n \geq N,
\end{aligned} \tag{D.38}$$

where the last two inequalities follow from the facts that $\ell \to 0$ and $n\tau^2 \to \infty$. This establishes (D.35). Equation (D.36) follows from Proposition B.2 and the observation that, by (D.23), we have

$$\begin{aligned}
|(\Delta_X^{\mathrm{HT}})_{ij} - \mathbb{E}(\Delta_X^{\mathrm{HT}})_{ij}| &\leq |(\Delta_X^{\mathrm{HT}})_{ij}| + |\mathbb{E}(\Delta_X^{\mathrm{HT}})_{ij}| \\
&\leq 2(\tau + \ell) \\
&\leq 4\tau, \ \forall n \geq N,
\end{aligned} \tag{D.39}$$

where we choose $N$ large enough that $\tau_n \geq \ell_n$ for all $n \geq N$. □

The following theorem combines Propositions D.4 and D.5 to establish the targeted result of this section: control of the modeling error $\Delta_X^{\mathrm{HT}}$.

**Theorem D.6.** *We have*

$$\mathbb{E}\sigma_1(\Delta_X^{HT}) \to 0, \tag{D.40}$$

*and, consequently,* $\sigma_1(\Delta_X^{HT}) \xrightarrow{a.s.} 0$.

*Proof.* We have

$$\begin{aligned}\mathbb{E}\sigma_1(\Delta_X^{\mathrm{HT}}) &\le \mathbb{E}\left[\sigma_1(\Delta_X^{\mathrm{HT}} - \mathbb{E}\Delta_X^{\mathrm{HT}}) + \sigma_1(\mathbb{E}\Delta_X^{\mathrm{HT}})\right] \\ &= \mathbb{E}\sigma_1(\Delta_X^{\mathrm{HT}} - \mathbb{E}\Delta_X^{\mathrm{HT}}) + \sigma_1(\mathbb{E}\Delta_X^{\mathrm{HT}}).\end{aligned} \tag{D.41}$$

Applying Propositions D.4 and D.5, we obtain the bounds

$$\begin{aligned}\mathbb{E}\sigma_1(\Delta_X^{\mathrm{HT}} - \mathbb{E}\Delta_X^{\mathrm{HT}}) &\le A_n := C\sqrt{n}\tau\left(\sqrt{\ell} + \exp(-\gamma n\tau^2)\right), \ \forall n \ge N, \\ \sigma_1\left(\mathbb{E}\Delta_X^{\mathrm{HT}}\right) &\le B_n := Cn\tau\left(\ell + \exp(-\gamma n\tau^2)\right) \qquad, \ \forall n \ge N.\end{aligned} \tag{D.42}$$

For the particular choice of $\tau$ in (5.23) and the incoherence bound (D.2) on $\ell$, simple computations show that

$$A_n, B_n \le \frac{\log n \text{ factors}}{\sqrt{n}} + \frac{\log n \text{ factors}}{n^{\log n \text{ factors}}} \to 0. \tag{D.43}$$

This establishes (D.40). We also have $\sum_n \exp(-\gamma\tau^{-2}) < \infty$ for all $\gamma > 0$, so the result follows by Equation (D.36) of Proposition D.5 and the Borel-Cantelli lemma. □

## D.5 Effective SNR

In this section, we combine the results of Sections D through D to complete the proof of Theorem V.8 for the case of hard thresholding. Combining (D.9) and (D.13), we have that

$$\widetilde{X}_\tau^{\mathrm{HT}} = (1-p)L + G_\tau^{\mathrm{HT}} + \Delta_L + \Delta_X^{\mathrm{HT}}, \tag{D.44}$$

where Theorems D.1 and D.6 establish that $\sigma_1(\Delta_X^{\mathrm{HT}}) \xrightarrow{\text{a.s.}} 0$ and $\sigma_1(\Delta_L) \xrightarrow{\text{a.s.}} 0$. Thus, asymptotically, $\widetilde{X}_\tau^{\mathrm{HT}}$ is a low-rank plus noise matrix, where the elements of the noise matrix $G_\tau^{\mathrm{HT}}$ are i.i.d. with variance $(\sigma_\tau^{\mathrm{HT}})^2/n$ with

$$(\sigma_\tau^{\mathrm{HT}})^2 := (1-p_s)\left[n(1-p_\tau)\sigma_\tau^2\right] + p_s\left[n(1-\widetilde{p}_\tau)\widetilde{\sigma}_\tau^2\right]. \tag{D.45}$$

We can therefore complete the proof of Theorem V.8 by applying Theorem 2.9 and Section 3.1 of [27] to conclude that the asymptotic accuracies of the left and right singular vectors of $\widetilde{X}_\tau^{\mathrm{HT}}$ are given by (5.11) and (5.12), respectively, with effective SNRs

$$\overline{\theta}_i^{\mathrm{HT}} = \lim_{n\to\infty} \frac{(1-p)\theta_i}{\sigma_\tau^{\mathrm{HT}}}, \tag{D.46}$$

where it remains only to show that

$$\lim_{n\to\infty} \frac{(1-p)\theta_i}{\sigma_\tau^{\mathrm{HT}}} = \lim_{n\to\infty} \frac{\sqrt{1-p_s}\ \theta_i}{\sigma} = \overline{\theta}_i^\star. \tag{D.47}$$

The following theorem establishes the limit (D.47) and thus completes the proof.

**Theorem D.7.** *We have*

$$\lim_{n\to\infty} \frac{1-p}{\sigma_\tau^{HT}} = \lim_{n\to\infty} \frac{\sqrt{1-p_s}}{\sigma}. \tag{D.48}$$

*Proof.* Applying Chebyshev's inequality yields

$$p_\tau = \mathbb{P}(|g| \geq \tau) \leq \frac{\sigma^2}{n\tau^2}, \tag{D.49}$$

and so $p_\tau \to 0$ since $n\tau^2 \to \infty$. We also have

$$\begin{aligned} \widetilde{p}_\tau &= \mathbb{P}(|\widetilde{g}| \geq \tau) \\ &= \mathbb{P}(|q+g| \geq \tau) \\ &\geq \mathbb{P}(|q| \geq |g| + \tau) \\ &\geq \mathbb{P}(|q| \geq |g| + \tau \mid |g| \leq \tau)\mathbb{P}(|g| \leq \tau) \\ &\geq \mathbb{P}(|q| \geq 2\tau)(1-p_\tau). \end{aligned} \tag{D.50}$$

We can assume without loss of generality (by redefining $p_s$ if necessary) that $q$ has no point mass at zero. Thus, since $\tau \to 0$, we have $\mathbb{P}(|q| \geq 2\tau) \to 1$ as $n \to \infty$. Therefore

$$\lim_{n\to\infty} \widetilde{p}_\tau = \lim_{n\to\infty} (1-p_\tau) = 1. \tag{D.51}$$

Combining these results, we have

$$\begin{aligned} \lim_{n\to\infty}(1-p) &= 1 - \underbrace{\left[\lim_{n\to\infty}(1-p_s)\right]}_{\leq 1} \underbrace{\left[\lim_{n\to\infty} p_\tau\right]}_{=0} - \left[\lim_{n\to\infty} p_s\right] \underbrace{\left[\lim_{n\to\infty} \widetilde{p}_\tau\right]}_{=1} \\ &= \lim_{n\to\infty}(1-p_s). \end{aligned} \tag{D.52}$$

Next, observe that

$$\begin{aligned} n(1-p_\tau)\sigma_\tau^2 &= n(1-p_\tau)\mathbb{E}\left[|g|^2 \mid |g| \le \tau\right] \\ &= n(1-p_\tau)\frac{\mathbb{E}\left[|g|^2 \mathbb{1}_{\{|g|\le\tau\}}\right]}{\mathbb{P}(|g|\le\tau)} \\ &= n\mathbb{E}\left[|g|^2 \mathbb{1}_{\{|g|\le\tau\}}\right] \\ &= \sigma^2\mathbb{E}\left[\left|\frac{\sqrt{n}g}{\sigma}\right|^2 \mathbb{1}_{\left\{\left|\sqrt{n}g/\sigma\right|\le\sqrt{n}\tau/\sigma\right\}}\right], \end{aligned} \tag{D.53}$$

where $\overline{g} := \sqrt{n}g/\sigma$ is a zero-mean unit variance random variable.[2] Therefore, since $n\tau^2 \to \infty$, we have

$$\begin{aligned} \lim_{n\to\infty} n(1-p_\tau)\sigma_\tau^2 &= \lim_{n\to\infty} \sigma^2\mathbb{E}\left[|\overline{g}|^2 \mathbb{1}_{\{|\overline{g}|\le\sqrt{n}\tau/\sigma\}}\right] \\ &= \sigma^2\mathbb{E}[|\overline{g}|^2] \\ &= \sigma^2. \end{aligned} \tag{D.54}$$

We also have

$$\widetilde{\sigma}_\tau^2 = \mathbb{E}[\widetilde{g}^2 \mid |\widetilde{g}| \le \tau] \le \tau^2, \tag{D.55}$$

and so, combining (D.50) and (D.55),

$$\begin{aligned} n(1-\widetilde{p}_\tau)\widetilde{\sigma}_\tau^2 &\le n\tau^2(1-\widetilde{p}_\tau) \\ &\le n\tau^2\left[1-\mathbb{P}(|q|\ge 2\tau)(1-p_\tau)\right] \\ &\le \underbrace{n\tau^2\mathbb{P}(|q|\le 2\tau)}_{=:T_1} + \underbrace{p_\tau n\tau^2}_{=:T_2}. \end{aligned} \tag{D.56}$$

Now, since $\tau \to 0$, $\tau$ will eventually fall in a neighborhood of zero, so we can apply Assumption V.3 to conclude that

$$\mathbb{P}(|q| \le 2\tau) \le C\tau, \ \forall n \ge N. \tag{D.57}$$

Thus, eventually $T_1 \le Cn\tau^3$, and therefore $T_1 \to 0$ for the particular choice of $\tau$ in (5.23). Also, by Assumption V.2, we have

$$p_\tau \le 2\exp(-\gamma n\tau^2), \tag{D.58}$$

---

[2] The random variable $\overline{g}$ does not depend on $n$ and $\sigma$, which appeared in $g$ only through its variance.

and so $T_2 \to 0$. Combining (D.54) and (D.56), we have

$$\begin{aligned}\lim_{n\to\infty}(\sigma_\tau^{\mathrm{HT}})^2 &= \left[\lim_{n\to\infty}(1-p_s)\right]\underbrace{\left[\lim_{n\to\infty} n(1-p_\tau)\sigma_\tau^2\right]}_{=\sigma^2} + \left[\lim_{n\to\infty} p_s\right]\underbrace{\left[\lim_{n\to\infty} n(1-\widetilde{p}_\tau)\widetilde{\sigma}_\tau^2\right]}_{=0} \\ &= \lim_{n\to\infty}(1-p_s)\sigma^2.\end{aligned} \tag{D.59}$$

Taking the ratio of (D.52) and (D.59) gives the desired result. □

# APPENDIX E

# Proof of Theorem V.8 for Soft Thresholding

In this appendix we prove Theorem V.8 for the soft thresholding estimator $\widetilde{X}_{\tau}^{\mathrm{ST}}$ defined by (5.21). That is, we prove the second row of Table 5.1. The proof is organized as follows. In Sections E through E, we derive some preliminary results, and in Section E we combine the preliminary results to complete the proof. Throughout the proof, we adopt the notation from Section D. The proof for the soft thresholding case is very similar to the hard thresholding proof from Appendix D, so we omit redundant proofs when possible.

## E.1 Soft Thresholding Estimator

Note that we can explicitly write the entries of $\widetilde{X}_{\tau}^{\mathrm{ST}}$ as

$$(\widetilde{X}_{\tau}^{\mathrm{ST}})_{ij} = \begin{cases} L_{ij} + G_{ij} & \text{if } S_{ij} = 0 \text{ and } |L_{ij} + G_{ij}| \leq \tau, \\ L_{ij} + S_{ij} + G_{ij} & \text{if } S_{ij} \neq 0 \text{ and } |L_{ij} + S_{ij} + G_{ij}| \leq \tau, \\ \tau & \text{if } L_{ij} + S_{ij} + G_{ij} > \tau, \\ -\tau & \text{if } L_{ij} + S_{ij} + G_{ij} < -\tau. \end{cases} \tag{E.1}$$

The statistics of the matrix $\widetilde{X}_{\tau}^{\mathrm{ST}}$ are complicated by the presence of $L_{ij}$ in the condition statements of (E.1), which make the probability of each case occurring *entry-dependent*. Removing $L_{ij}$ from the condition statements of (E.1) yields a closely-related matrix $\overline{X}_{\tau}^{\mathrm{ST}}$ with elements

$$(\overline{X}_{\tau}^{\mathrm{ST}})_{ij} = \begin{cases} L_{ij} + G_{ij} & \text{if } S_{ij} = 0 \text{ and } |G_{ij}| \leq \tau, \\ L_{ij} + S_{ij} + G_{ij} & \text{if } S_{ij} \neq 0 \text{ and } |S_{ij} + G_{ij}| \leq \tau, \\ \tau & \text{if } S_{ij} + G_{ij} > \tau, \\ -\tau & \text{if } S_{ij} + G_{ij} < -\tau \end{cases} \tag{E.2}$$

that is easier to analyze because the case statements in (E.2) occur with fixed probabilities independent of $i$ and $j$. In Section E, we establish that the modeling error[1]

$$\Delta_X^{\mathrm{ST}} := \widetilde{X}_{\tau}^{\mathrm{ST}} - \overline{X}_{\tau}^{\mathrm{ST}} \tag{E.3}$$

is small in the sense that $\sigma_1(\Delta_X^{\mathrm{ST}}) \xrightarrow{\text{a.s.}} 0$. Therefore, we can determine the asymptotic performance of the estimator $\widetilde{X}_{\tau}^{\mathrm{ST}}$ by analyzing the asymptotics of the random matrix $\overline{X}_{\tau}^{\mathrm{ST}}$. We analyze $\overline{X}_{\tau}^{\mathrm{ST}}$ in Section E, and then we revisit the modeling error $\Delta_X^{\mathrm{ST}}$ in Section E.

## E.2 Equivalent Random Matrix Model

In this section we analyze the random matrix model $\overline{X}_{\tau}^{\mathrm{ST}}$ defined by (D.8). By construction, we have

$$\begin{aligned} \mathbb{P}(S_{ij} = 0 \text{ and } |G_{ij}| \leq \tau) &= (1 - p_s)(1 - p_\tau), \\ \mathbb{P}(S_{ij} \neq 0 \text{ and } |S_{ij} + G_{ij}| \leq \tau) &= p_s(1 - \widetilde{p}_\tau), \\ \mathbb{P}(S_{ij} + G_{ij} > \tau) &= p/2, \\ \mathbb{P}(S_{ij} + G_{ij} < -\tau) &= p/2, \end{aligned} \tag{E.4}$$

and so the distributions of the elements of $\overline{X}_{\tau}^{\mathrm{ST}}$ from (E.2) are

$$(\overline{X}_{\tau}^{\mathrm{ST}})_{ij} \overset{\mathrm{d}}{=} \begin{cases} L_{ij} + (G_\tau)_{ij} & \text{with probability } (1 - p_s)(1 - p_\tau), \\ L_{ij} + (\widetilde{G}_\tau)_{ij} & \text{with probability } p_s(1 - \widetilde{p}_\tau), \\ \tau & \text{with probability } p/2, \\ -\tau & \text{with probability } p/2, \end{cases} \tag{E.5}$$

---

[1] The quantity $\Delta_X^{\mathrm{ST}}$ is a random matrix defined implicitly by the random matrices in Section 5.2 and the thresholding schemes defined by $\widetilde{X}_{\tau}^{\mathrm{ST}}$ and $\overline{X}_{\tau}^{\mathrm{ST}}$.

where $(G_\tau)_{ij}$ are drawn i.i.d. from $g_\tau$ and $(\widetilde{G}_\tau)_{ij}$ are drawn i.i.d. from $\widetilde{g}_\tau$. From (E.5), we have

$$\mathbb{E}\overline{X}_\tau^{\mathrm{ST}} = (1-p)L, \tag{E.6}$$

and so

$$\begin{aligned} \overline{X}_\tau^{\mathrm{ST}} &= \mathbb{E}\overline{X}_\tau^{\mathrm{ST}} + (\overline{X}_\tau^{\mathrm{ST}} - \mathbb{E}\overline{X}_\tau^{\mathrm{ST}}) \\ &= (1-p)L + G_\tau^{\mathrm{ST}} + \Delta_L, \end{aligned} \tag{E.7}$$

where

$$(G_\tau^{\mathrm{ST}})_{ij} = \begin{cases} (G_\tau)_{ij} & \text{with probability } (1-p_s)(1-p_\tau), \\ (\widetilde{G}_\tau)_{ij} & \text{with probability } p_s(1-\widetilde{p}_\tau), \\ \tau & \text{with probability } p/2, \\ -\tau & \text{with probability } p/2, \end{cases} \tag{E.8}$$

and $\Delta_L$ is the same matrix (D.15) that appeared in the analogous hard thresholding result (D.13).

## E.3 Modeling Error Bounds

In this section we analyze the modeling error $\Delta_X^{\mathrm{ST}}$ defined by (E.3). We reuse the definitions (D.21) from Section D. Comparing (D.7)-(D.8) and (E.1)-(E.2), we see that the modeling errors $\Delta_X^{\mathrm{HT}}$ and $\Delta_X^{\mathrm{ST}}$ have similar properties. In particular, assuming without loss of generality that $\tau \geq \ell$, we have $(\Delta_X^{\mathrm{ST}})_{ij} \neq 0$ if any only if $(\Delta_X^{\mathrm{HT}})_{ij} \neq 0$. Furthermore, when $(\Delta_X^{\mathrm{ST}})_{ij} \neq 0$, we have either $|\widetilde{X}_\tau^{\mathrm{ST}}|_{ij} \in \mathcal{I}$ and $|\overline{X}_\tau^{\mathrm{HT}}|_{ij} = \tau$ or vice versa. As a result, $|(\Delta_X^{\mathrm{ST}})_{ij}| \leq \ell$. Figure E.1 plots $(\Delta_X^{\mathrm{ST}})_{ij}$ as a function of the underlying random variable $G_{ij} + S_{ij}$.

The following results are the soft thresholding analogs of the hard thresholding results from Section D. The proofs are identical to those from Section D with the bound $|(\Delta_X^{\mathrm{ST}})_{ij}| \leq \ell$ in place of the bound $|(\Delta_X^{\mathrm{HT}})_{ij}| \leq \tau + \ell$, so we omit the details here.

**Lemma E.1.** *We have*

$$\mathbb{E}|(\Delta_X^{ST})_{ij}|^\alpha \leq [(1-p_s)a + p_s b]\ell^\alpha. \tag{E.9}$$

*for any $\alpha > 0$ and $n \geq N$.*

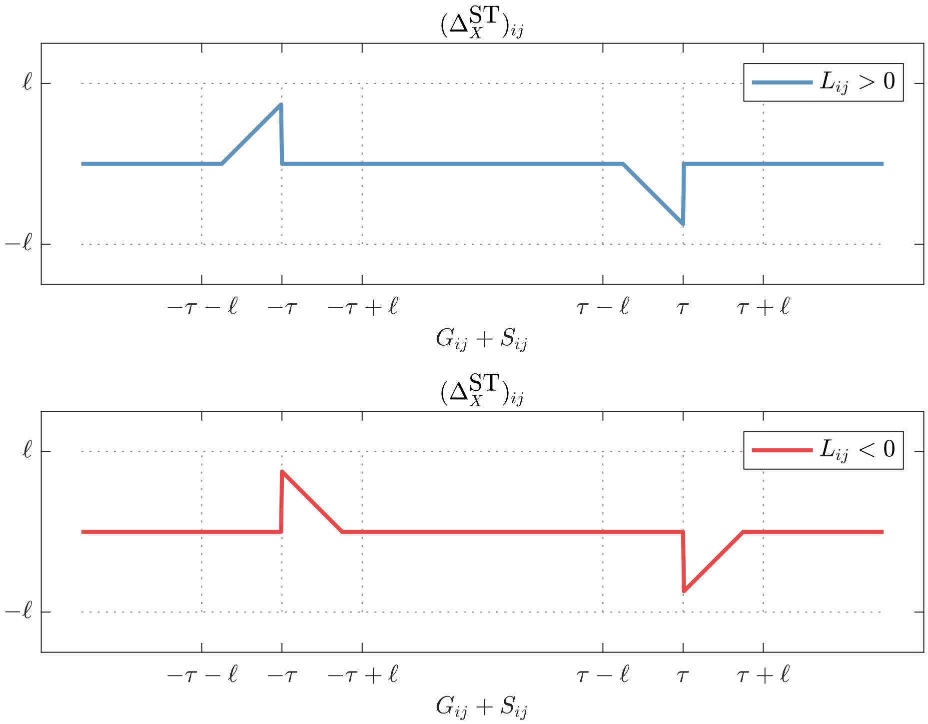


Figure E.1: Graphical depiction of $(\Delta_X^{\mathrm{ST}})_{ij}$ as a function of $G_{ij}+S_{ij}$ for two fixed values of $L_{ij}$.

**Corollary E.2.** *We have*

$$\mathbb{E}|(\Delta_X^{ST})_{ij}|^\alpha \leq C^\alpha \ell^\alpha \left(\ell + \exp(-\gamma n \tau^2)\right). \tag{E.10}$$

*for any $\alpha > 0$ and $n \geq N$.*

**Proposition E.3.** *We have*

$$\sigma_1\left(\mathbb{E}\Delta_X^{ST}\right) \leq Cn\ell\left(\ell + \exp(-\gamma n\tau^2)\right) \tag{E.11}$$

*for $n \geq N$.*

**Proposition E.4.** *We have*

$$\text{(i)} \quad \mathbb{E}\sigma_1(\Delta_X^{ST} - \mathbb{E}\Delta_X^{ST}) \leq C\sqrt{n}\ell\left(\sqrt{\ell} + \exp(-\gamma n\tau^2)\right), \ \forall n \geq N \tag{E.12}$$

*and for all $\epsilon > 0$:*

$$\text{(ii)} \quad \mathbb{P}(|\sigma_1(\Delta_X^{ST} - \mathbb{E}\Delta_X^{ST}) - \mathbb{E}\sigma_1(\Delta_X^{ST} - \mathbb{E}\Delta_X^{ST})| \geq \epsilon) \leq C\exp(-\gamma\epsilon^2\ell^{-2}), \ \forall n \geq N. \tag{E.13}$$

**Theorem E.5.** *We have*

$$\mathbb{E}\sigma_1(\Delta_X^{ST}) \to 0, \tag{E.14}$$

*and, consequently,* $\sigma_1(\Delta_X^{ST}) \xrightarrow{a.s.} 0$.

## E.4 Effective SNR

In this section, we combine the results of Sections E through E to complete the proof of Theorem V.8 for the case of soft thresholding. Combining (E.3) and (E.7), we have that

$$\widetilde{X}_\tau^{\mathrm{ST}} = (1-p)L + G_\tau^{\mathrm{ST}} + \Delta_L + \Delta_X^{\mathrm{ST}}, \tag{E.15}$$

where Theorems D.1 and E.5 establish that $\sigma_1(\Delta_L) \xrightarrow{\text{a.s.}} 0$ and $\sigma_1(\Delta_X^{\mathrm{ST}}) \xrightarrow{\text{a.s.}} 0$. Thus, asymptotically, $\widetilde{X}_\tau^{\mathrm{ST}}$ is a low-rank plus noise matrix, where the elements of the noise matrix $G_\tau^{\mathrm{ST}}$ are i.i.d. with variance $(\sigma_\tau^{\mathrm{ST}})^2/n$ with

$$(\sigma_\tau^{\mathrm{ST}})^2 := (\sigma_\tau^{\mathrm{HT}})^2 + pn\tau^2. \tag{E.16}$$

Here, $(\sigma_\tau^{\mathrm{HT}})^2$ is the effective variance of the hard thresholding model defined by (D.45). We can therefore complete the proof of Theorem V.8 by applying Theorem 2.9 and Section 3.1 of [27] to conclude that the asymptotic accuracies of the left and right singular vectors of $\widetilde{X}_\tau^{\mathrm{ST}}$ are given by (5.11) and (5.12), respectively, with effective SNRs

$$\overline{\theta}_i^{\mathrm{ST}} = \lim_{n\to\infty} \frac{(1-p)\theta_i}{\sigma_\tau^{\mathrm{ST}}}, \tag{E.17}$$

where it remains only to show that

$$\lim_{n\to\infty} \frac{(1-p)\theta_i}{\sigma_\tau^{\mathrm{ST}}} = \begin{cases} \lim_{n\to\infty} \dfrac{\sqrt{1-p_s}\;\theta_i}{\sigma} = \overline{\theta}_i^\star & \text{if } p_s \log^\eta n \to 0 \\ 0 & \text{if } p_s \log^\eta n \to \infty. \end{cases} \tag{E.18}$$

The following theorem establishes the limit (E.18) and thus completes the proof.

**Theorem E.6.** *We have*

$$\lim_{n\to\infty} \frac{1-p}{\sigma_\tau^{ST}} = \lim_{n\to\infty} \frac{1-p_s}{\sqrt{(1-p_s)\sigma^2 + p_s n\tau^2}}. \tag{E.19}$$

*Proof.* By (D.52) and (D.59) of Theorem D.7, it suffices to show that

$$\lim_{n\to\infty} pn\tau^2 = \lim_{n\to\infty} p_s n\tau^2, \tag{E.20}$$

because $p_s n\tau^2 = Cp_s \log^\eta n$. To establish (E.20), observe that, by (D.51) of Theorem D.7 and the definition of $p$ from (D.6), we have

$$\lim_{n\to\infty} pn\tau^2 = \left[\lim_{n\to\infty}(1-p_s)p_\tau n\tau^2\right] + \underbrace{\left[\lim_{n\to\infty}\widetilde{p}_\tau\right]}_{=1}\left[\lim_{n\to\infty}p_s n\tau^2\right]. \tag{E.21}$$

Furthermore, by Assumption V.2, we have

$$p_\tau \leq 2\exp(-\gamma n\tau^2), \tag{E.22}$$

and so

$$\begin{aligned}\lim_{n\to\infty}(1-p_s)p_\tau n\tau^2 &\leq \lim_{n\to\infty} p_\tau n\tau^2 \\ &\leq \lim_{n\to\infty} 2n\tau^2\exp(-\gamma n\tau^2) \qquad \text{(E.23)} \\ &= 0.\end{aligned}$$

This completes the proof. □

# BIBLIOGRAPHY

# BIBLIOGRAPHY

[1] R. Otazo, E. Candes, and D. K. Sodickson, “Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components,” *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[2] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, “Accelerated dynamic MRI exploiting sparsity and low-rank structure: k-t SLR,” *IEEE Trans. Med. Imaging*, vol. 30, no. 5, pp. 1042–1054, May 2011.

[3] S. G. Lingala, Y. Hu, E. Dibella, and M. Jacob, “Accelerated first pass cardiac perfusion MRI using improved k-t SLR,” in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 1280–1283.

[4] R. Otazo, “L+S reconstruction MATLAB code,” http://cai2r.net/resources/software/ls-reconstruction-matlab-code, 2014, [Online; accessed Mar. 2015].

[5] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, “k-t SLR: MATLAB package,” http://user.engineering.uiowa.edu/~jcb/software/ktslr_matlab/Software.html, 2011, [Online; accessed Mar. 2015].

[6] B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, “Improved robust PCA using low-rank denoising with optimal singular value shrinkage,” in *IEEE Workshop on Statistical Signal Processing (SSP)*, June 2014, pp. 13–16.

[7] B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, “Dynamic MRI reconstruction using low-rank plus sparse model with optimal rank regularized eigen-shrinkage,” in *Proceedings of the International Society for Magnetic Resonance in Medicine Annual Meeting (ISMRM)*, May 2014, p. 740.

[8] B. E. Moore, C. Gao, and R. R. Nadakuditi, “Panoramic robust PCA for foreground-background separation on noisy, free-motion camera video,” *IEEE Transactions on Computational Imaging*, December 2017, submitted.

[9] C. Gao, B. E. Moore, and R. R. Nadakuditi, “Augmented robust PCA for foreground-background separation on noisy, moving camera video,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, November 2017, to appear.

[10] B. E. Moore and R. R. Nadakuditi, “Robust PCA: Fundamental limits of thresholding-based outlier rejection,” *IEEE Transactions on Information Theory*, January 2018, submitted.

[11] B. E. Moore and R. R. Nadakuditi, “Robust PCA: Low rank matrix estimation with hard or soft thresholding-based outlier rejection,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, December 2016, pp. 80–84.

[12] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, “Efficient learning of dictionaries with low-rank atoms,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, December 2016, pp. 222–226.

[13] B. E. Moore, S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, “Online data-driven image reconstruction using efficiently learned dictionaries,” *IEEE Transactions on Computational Imaging*, January 2018, submitted.

[14] B. E. Moore and S. Ravishankar, “Online data-driven dynamic image restoration using DINO-KAT models,” in *IEEE International Conference on Image Processing (ICIP)*, September 2017, to appear.

[15] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, “Efficient online dictionary adaptation and image reconstruction for dynamic MRI,” in *Asilomar Conference on Signals, Systems, and Computers*, October 2017, to appear.

[16] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, “Low-rank and adaptive sparse signal (LASSI) models for highly accelerated dynamic imaging,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 5, pp. 1116–1128, May 2017.

[17] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, “LASSI: A low-rank and adaptive sparse signal model for highly accelerated dynamic imaging,” in *IEEE Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, July 2016, pp. 1–5.

[18] A. J. Wagenmaker, B. E. Moore, and R. R. Nadakuditi, “Robust photometric stereo via dictionary learning,” *IEEE Transactions on Computational Imaging*, December 2017, submitted.

[19] A. J. Wagenmaker, B. E. Moore, and R. R. Nadakuditi, “Robust photometric stereo using learned image and gradient dictionaries,” in *IEEE International Conference on Image Processing (ICIP)*, September 2017, to appear.

[20] A. J. Wagenmaker, B. E. Moore, and R. R. Nadakuditi, “Robust surface reconstruction from gradients via adaptive dictionary regularization,” in *IEEE International Conference on Image Processing (ICIP)*, September 2017, to appear.

[21] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[22] I. T. Jolliffe, "Principal component analysis," *Springer Series in Statistics*, vol. 1, 1986.

[23] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[24] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[25] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[26] R. R. Nadakuditi, "OptShrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 3002–3018, 2014.

[27] F. Benaych-Georges and R. R. Nadakuditi, "The singular values and vectors of low rank perturbations of large rectangular random matrices," *Journal of Multivariate Analysis*, vol. 111, pp. 120–135, 2012.

[28] T. Tao and V. Vu, "Random matrices: Universality of local eigenvalue statistics up to the edge," *Communications in Mathematical Physics*, vol. 298, no. 2, pp. 549–572, 2010.

[29] T. Tao and V. Vu, "Random matrices: Universality of local eigenvalue statistics," *Acta Mathematica*, vol. 206, no. 1, pp. 127–204, 2011.

[30] V. A. Marcenko and L. A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Mathematics of the USSR-Sbornik*, vol. 1, no. 4, p. 457, 1967.

[31] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.

[32] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[33] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.

[34] L. V. Tran, V. H. Vu, and K. Wang, "Sparse random graphs: Eigenvalues and eigenvectors," *Random Structures & Algorithms*, vol. 42, no. 1, pp. 110–134, 2013.

[35] A. Beck and M. Teboulle, "Gradient-based algorithms with applications to signal recovery," in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar, Eds. Cambridge University Press, 2010.

[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Img. Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[37] K. P. Pruessmann, M. Weiger, P. Börnert, and P. Boesiger, "Advances in sensitivity encoding with arbitrary k-space trajectories," *Magn Reson Med*, no. 4, pp. 638–651, 2001.

[38] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in neural information processing systems*, 2009, pp. 2080–2088.

[39] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7221–7234, 2011.

[40] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma, "TILT: Transform invariant low-rank textures," in *Asian Conference on Computer Vision.* Springer, 2010, pp. 314–328.

[41] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[42] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *2011 IEEE International Conference on Computer Vision.* IEEE, 2011, pp. 1615–1622.

[43] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[44] S.-C. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 1, pp. 1–14, 2011.

[45] Y. Tsaig and A. Averbuch, "Automatic segmentation of moving objects in video sequences: a region labeling approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 597–612, 2002.

[46] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014.

[47] A. Sobral and A. Vacavant, “A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos,” *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.

[48] X. Ye, J. Yang, X. Sun, K. Li, C. Hou, and Y. Wang, “Foreground–background separation from video clips via motion-assisted matrix restoration,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1721–1734, 2015.

[49] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” *Computer VisionECCV 2000*, pp. 751–767, 2000.

[50] M. Piccardi, “Background subtraction techniques: a review,” in *2004 IEEE international conference on systems, man and cybernetics*, vol. 4. IEEE, 2004, pp. 3099–3104.

[51] J. Rittscher, J. Kato, S. Joga, and A. Blake, “A probabilistic background model for tracking,” *Computer VisionECCV 2000*, pp. 336–350, 2000.

[52] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 246–252.

[53] A. Mittal and N. Paragios, “Motion-based background subtraction using adaptive kernel density estimation,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. Ieee, pp. II–II.

[54] V. Ramesh *et al.*, “Background modeling and subtraction of dynamic scenes,” in *IEEE International Conference on Computer Vision.* IEEE, 2003, pp. 1305–1312.

[55] J. Zhong *et al.*, “Segmenting foreground objects from a dynamic textured background via a robust kalman filter,” in *IEEE International Conference on Computer Vision.* IEEE, 2003, pp. 44–50.

[56] C. Guyon, T. Bouwmans, and E.-H. Zahzah, “Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint,” in *Asian Conference on Computer Vision.* Springer, 2012, pp. 315–320.

[57] T. Zhou and D. Tao, “Shifted subspaces tracking on sparse outlier for motion segmentation.” in *Artificial Intelligence Journal*, 2013.

[58] C. Guyon, T. Bouwmans, and E. Zahzah, “Foreground detection via robust low rank matrix factorization including spatial constraint with iterative reweighted regression,” in *International Conference on Pattern Recognition.* IEEE, 2012, pp. 2805–2808.

[59] X. Cao, L. Yang, and X. Guo, "Total variation regularized rpca for irregularly moving object detection under dynamic background," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 1014–1027, 2016.

[60] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Allerton Conference on Communication, Control, and Computing.* IEEE, 2010, pp. 704–711.

[61] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2012, pp. 1568–1575.

[62] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," in *Allerton Conference on Communication, Control, and Computing.* IEEE, 2010, pp. 591–598.

[63] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4284–4297, 2014.

[64] S. E. Ebadi, V. G. Ones, and E. Izquierdo, "Approximated robust principal component analysis for improved general scene background subtraction," *arXiv preprint arXiv:1603.05875*, 2016.

[65] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

[66] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice-Hall, 2002.

[67] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision*, 2006, pp. 404–417.

[68] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[69] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[70] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[71] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[72] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1, pp. 89–97, 2004.

[73] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented lagrangian method for total variation video restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097–3111, 2011.

[74] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[75] M. Tao and J. Yang, "Alternating direction algorithms for total variation deconvolution in image reconstruction," Dept. Math., Nanjing Univ., Nanjing, China, Tech. Rep. TR0918, 2009.

[76] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[77] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[78] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[79] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[80] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal component pursuit," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 1518–1522.

[81] A. Ganesh, Z. Lin, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast algorithms for recovering a corrupted low-rank matrix," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2009, pp. 213–216.

[82] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *preprint, arXiv:1009.5055*, 2010.

[83] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[84] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[85] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[86] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[87] D. L. Donoho and I. M. Johnstone, "Minimax risk over $\ell_p$-balls for $\ell_q$-error," *Probability Theory and Related Fields*, vol. 99, no. 2, pp. 277–303, 1994.

[88] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[89] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet shrinkage: asymptopia?" *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 301–369, 1995.

[90] D. L. Donoho and I. M. Johnstone, "Minimax estimation via wavelet shrinkage," *The Annals of Statistics*, vol. 26, no. 3, pp. 879–921, 1998.

[91] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[92] I. M. Johnstone, "Gaussian estimation: Sequence and wavelet models," *Manuscript, December*, 2011.

[93] B. Efron and C. Morris, "Stein's estimation rule and its competitorsan empirical bayes approach," *Journal of the American Statistical Association*, vol. 68, no. 341, pp. 117–130, 1973.

[94] B. Efron, "Biased versus unbiased estimation," *Advances in Mathematics*, vol. 16, no. 3, pp. 259–277, 1975.

[95] M. Elad, "Why simple shrinkage is still relevant for redundant representations?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5559–5569, 2006.

[96] Y. C. Eldar, "Uniformly improving the cramér–rao bound and maximum-likelihood estimation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 8, pp. 2943–2956, 2006.

[97] Y. C. Eldar, "Rethinking biased estimation: Improving maximum likelihood and the cramér–rao bound," *Foundations and Trends in Signal Processing*, vol. 1, no. 4, pp. 305–449, 2008.

[98] S. Kay and Y. C. Eldar, "Rethinking biased estimation," *IEEE Signal Processing Magazine*, vol. 25, no. 3, p. 133, 2008.

[99] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[100] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[101] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[102] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *IEEE International Conference on Computer Vision*, Sept 2009, pp. 2272–2279.

[103] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1028–1041, 2011.

[104] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[105] R. Gribonval and K. Schnass, "Dictionary identification–sparse matrix-factorization via $l_1$ -minimization," *IEEE Trans. Inform. Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.

[106] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2055–2065, 2013.

[107] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2010*, 2010, pp. 3501–3508.

[108] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.

[109] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[110] K. Engan, S. Aase, and J. Hakon-Husoy, "Method of optimal directions for frame design," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 2443–2446.

[111] M. Yaghoobi, T. Blumensath, and M. Davies, “Dictionary learning for sparse approximations with the majorization method,” *IEEE Transaction on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[112] K. Skretting and K. Engan, “Recursive least squares dictionary learning algorithm,” *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[113] L. N. Smith and M. Elad, “Improving dictionary learning: Multiple dictionary updates and coefficient reuse,” *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 79–82, Jan 2013.

[114] A. Rakotomamonjy, “Direct optimization of the dictionary learning problem,” *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5495–5506, 2013.

[115] D. A. Spielman, H. Wang, and J. Wright, “Exact recovery of sparsely-used dictionaries,” in *Proceedings of the 25th Annual Conference on Learning Theory*, 2012, pp. 37.1–37.18.

[116] S. Arora, R. Ge, and A. Moitra, “New algorithms for learning incoherent and overcomplete dictionaries,” in *Conference on Learning Theory*, 2014, pp. 779–806.

[117] C. Bao, H. Ji, Y. Quan, and Z. Shen, “L0 norm based dictionary learning by proximal methods with global convergence,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3858–3865.

[118] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon, “Learning sparsely used overcomplete dictionaries,” *Journal of Machine Learning Research*, vol. 35, pp. 1–15, 2014.

[119] S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, “Efficient sum of outer products dictionary learning (SOUP-DIL) and its application to inverse problems,” *IEEE Transactions on Computational Imaging*, 2017.

[120] S. Ravishankar and Y. Bresler, “Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2519–2557, 2015.

[121] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, “Learning overcomplete dictionaries based on atom-by-atom updating,” *IEEE Trans. on Signal Process.*, vol. 62, no. 4, pp. 883–891, 2014.

[122] L. Mirsky, “On the trace of matrix products,” *Mathematische Nachrichten*, vol. 20, no. 3-6, pp. 171–174, 1959.

[123] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit,” http://www.cs.technion.ac.il/~ronrubin/Publications/KSVD-OMP-v2.pdf, 2008, technion - Computer Science Department - Technical Report.

[124] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis.* Heidelberg, Germany: Springer-Verlag, 1998.

[125] H. S. Anderson, J. Ilic-Helms, B. Rohrer, J. Wheeler, and K. Larson, “Sparse imaging for fast electron microscopy,” in *Proc. SPIE*, vol. 8657, 2013, pp. 86 570C–86 570C–12.

[126] L. Howard, “Dartmouth college electron microscope facility - electron microscope images,” http://remf.dartmouth.edu/miscellaneous_SEM_P1/, 2011, [Online; accessed Dec. 2015].

[127] M. Elad, “Michael Elad personal page,” http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip, 2009, [Online; accessed Nov. 2015].

[128] N. Rajpoot, Z. Yao, and R. Wilson, “Adaptive wavelet restoration of noisy video sequences,” in *International Conference on Image Processing*, vol. 2, 2004, pp. 957–960 Vol.2.

[129] D. Rusanovskyy and K. Egiazarian, “Video denoising algorithm in sliding 3D DCT domain,” in *Advanced Concepts for Intelligent Vision Systems: 7th International Conference, ACIVS 2005*, 2005, pp. 618–625.

[130] S. Ravishankar and Y. Bresler, “Learning doubly sparse transforms for images,” *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4598–4612, 2013.

[131] B. Wen, S. Ravishankar, and Y. Bresler, “Learning flipping and rotation invariant sparsifying transforms,” in *2016 IEEE International Conference on Image Processing*, 2016, pp. 3857–3861.

[132] M. Protter and M. Elad, “Image sequence denoising via sparse and redundant representations,” *IEEE Trans. on Image Processing*, vol. 18, no. 1, pp. 27–36, 2009.

[133] S. G. Lingala and M. Jacob, “Blind compressive sensing dynamic MRI,” *IEEE Transactions on Medical Imaging*, vol. 32, no. 6, pp. 1132–1145, 2013.

[134] S. Ravishankar, B. Wen, and Y. Bresler, “Online sparsifying transform learning – Part I: Algorithms,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 625–636, 2015.

[135] B. Wen, S. Ravishankar, and Y. Bresler, “Video denoising by online 3D sparsifying transform learning,” in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 118–122.

[136] M. Lustig, D. L. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[137] M. Lustig, J. M. Santos, D. L. Donoho, and J. M. Pauly, "k-t SPARSE: High frame rate dynamic MRI exploiting spatio-temporal sparsity," in *Proc. ISMRM*, 2006, p. 2420.

[138] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye, "k-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI," *Magnetic Resonance in Medicine*, vol. 61, no. 1, pp. 103–116, 2009.

[139] Z. P. Liang, "Spatiotemporal imaging with partially separable functions," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, pp. 988–991.

[140] J. P. Haldar and Z. P. Liang, "Spatiotemporal imaging with partially separable functions: A matrix recovery approach," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010, pp. 716–719.

[141] B. Zhao, J. P. Haldar, C. Brinegar, and Z. P. Liang, "Low rank matrix recovery for real-time cardiac MRI," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010, pp. 996–999.

[142] H. Pedersen, S. Kozerke, S. Ringgaard, K. Nehrke, and W. Y. Kim, "k-t PCA: Temporally constrained k-t BLAST reconstruction using principal component analysis," *Magnetic Resonance in Medicine*, vol. 62, no. 3, pp. 706–716, 2009.

[143] J. Trzasko and A. Manduca, "Local versus global low-rank promotion in dynamic MRI series reconstruction," in *Proc. ISMRM*, 2011, p. 4371.

[144] J. Caballero, A. N. Price, D. Rueckert, and J. V. Hajnal, "Dictionary learning and time sparsity for dynamic mr data reconstruction," *IEEE Transactions on Medical Imaging*, vol. 33, no. 4, pp. 979–994, 2014.

[145] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[146] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[147] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE transactions on image processing*, vol. 22, no. 1, pp. 119–133, 2013.

[148] B. Sharif and Y. Bresler, "Adaptive real-time cardiac MRI using paradise: Validation by the physiologically improved NCAT phantom," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, pp. 1020–1023.

[149] P. Feng and Y. Bresler, "Spectrum-blind minimum-rate sampling and reconstruction of multiband signals," in *ICASSP*, vol. 3, 1996, pp. 1689–1692.

[150] Y. Bresler and P. Feng, "Spectrum-blind minimum-rate sampling and reconstruction of 2-D multiband signals," in *Proc. 3rd IEEE Int. Conf. on Image Processing, ICIP'96*, 1996, pp. 701–704.

[151] J. Trzasko and A. Manduca, "Highly undersampled magnetic resonance image reconstruction via homotopic $l_0$-minimization," *IEEE Trans. Med. Imaging*, vol. 28, no. 1, pp. 106–121, 2009.

[152] Y. Kim, M. S. Nadar, and A. Bilgin, "Wavelet-based compressed sensing using gaussian scale mixtures," in *Proc. ISMRM*, 2010, p. 4856.

[153] C. Qiu, W. Lu, and N. Vaswani, "Real-time dynamic MR image reconstruction using kalman filtered compressed sensing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 393–396.

[154] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," *Magnetic Resonance in Medicine*, vol. 59, no. 2, pp. 365–373, 2008.

[155] D. Liang, B. Liu, J. Wang, and L. Ying, "Accelerating SENSE using compressed sensing," *Magnetic Resonance in Medicine*, vol. 62, no. 6, pp. 1574–1584, 2009.

[156] R. Otazo, D. Kim, L. Axel, and D. K. Sodickson, "Combination of compressed sensing and parallel imaging for highly accelerated first-pass cardiac perfusion MRI," *Magnetic Resonance in Medicine*, vol. 64, no. 3, pp. 767–776, 2010.

[157] B. Wu, R. Watts, R. Millane, and P. Bones, "An improved approach in applying compressed sensing in parallel MR imaging," in *Proc. ISMRM*, 2009, p. 4595.

[158] B. Liu, F. M. Sebert, Y. Zou, and L. Ying, "SparseSENSE: Randomly-sampled parallel imaging using compressed sensing," in *Proc. ISMRM*, 2008, p. 3154.

[159] G. Adluru and E. V. R. DiBella, "Reordering for improved constrained reconstruction from undersampled k-space data," *Journal of Biomedical Imaging*, vol. 2008, pp. 1–12, 2008.

[160] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[161] X. Lu, Y. Yuan, and P. Yan, "Alternatively constrained dictionary learning for image superresolution," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 366–377, 2014.

[162] S. Gleichman and Y. C. Eldar, "Blind compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6958–6975, 2011.

[163] S. Ravishankar and Y. Bresler, “Multiscale dictionary learning for MRI,” in *Proc. ISMRM*, 2011, p. 2830.

[164] S. G. Lingala and M. Jacob, “Blind compressed sensing with sparse dictionaries for accelerated dynamic MRI,” in *2013 IEEE 10th International Symposium on Biomedical Imaging*, 2013, pp. 5–8.

[165] Y. Wang, Y. Zhou, and L. Ying, “Undersampled dynamic magnetic resonance imaging using patch-based spatiotemporal dictionaries,” in *2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI)*, April 2013, pp. 294–297.

[166] Y. Wang and L. Ying, “Compressed sensing dynamic cardiac cine mri using learned spatiotemporal dictionary,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1109–1120, 2014.

[167] J. Caballero, D. Rueckert, and J. V. Hajnal, “Dictionary learning and time sparsity in dynamic MRI,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2012*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7510, pp. 256–263.

[168] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X. P. Zhang, “Bayesian nonparametric dictionary learning for compressed sensing MRI,” *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5007–5019, 2014.

[169] S. P. Awate and E. V. R. DiBella, “Spatiotemporal dictionary learning for undersampled dynamic MRI reconstruction via joint frame-based and dictionary-based sparsity,” in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, 2012, pp. 318–321.

[170] S. Wang, X. Peng, P. Dong, L. Ying, D. D. Feng, and D. Liang, “Parallel imaging via sparse representation over a learned dictionary,” in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, pp. 687–690.

[171] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 763–770.

[172] B. Trémoulhéac, N. Dikaios, D. Atkinson, and S. R. Arridge, “Dynamic mr image reconstruction - separation from undersampled ( **k**,t )-space via low-rank plus sparse prior,” *IEEE Transactions on Medical Imaging*, vol. 33, no. 8, pp. 1689–1701, 2014.

[173] B. Zhao, J. P. Haldar, A. G. Christodoulou, and Z. P. Liang, “Image reconstruction from highly undersampled (**k**, t) -space data with joint partial separability and sparsity constraints,” *IEEE Transactions on Medical Imaging*, vol. 31, no. 9, pp. 1809–1820, 2012.

[174] A. Majumdar and R. Ward, “Learning space-time dictionaries for blind compressed sensing dynamic MRI reconstruction,” in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 4550–4554.

[175] A. Majumdar and R. K. Ward, “Learning the sparsity basis in low-rank plus sparse model for dynamic MRI reconstruction,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 778–782.

[176] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, “Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition,” in *Asilomar Conf. on Signals, Systems and Comput.*, 1993, pp. 40–44 vol.1.

[177] J. Woodworth and R. Chartrand, “Compressed sensing recovery via nonconvex shrinkage penalties,” *Inverse Problems*, vol. 32, no. 7, pp. 75 004–75 028, 2016.

[178] B. Sharif and Y. Bresler, “Adaptive real-time cardiac MRI using paradise: Validation by the physiologically improved NCAT phantom,” in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, pp. 1020–1023.

[179] R. J. Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical Engineering*, vol. 19, no. 1, pp. 191 139–191 139, 1980.

[180] S. Barsky and M. Petrou, “The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows,” *IEEE PAMI*, vol. 25, no. 10, pp. 1239–1252, Oct. 2003.

[181] I. Sato, T. Okabe, Q. Yu, and Y. Sato, “Shape reconstruction based on similarity in radiance changes under varying illumination,” in *ICCV*. IEEE, 2007, pp. 1–8.

[182] T. Okabe, I. Sato, and Y. Sato, “Attached shadow coding: Estimating surface normals from shadows under unknown reflectance and lighting conditions,” in *ICCV*. IEEE, 2009, pp. 1693–1700.

[183] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz, “Shape and spatially-varying brdfs from photometric stereo,” *IEEE PAMI*, vol. 32, no. 6, pp. 1060–1071, 2010.

[184] T.-P. Wu and C.-K. Tang, “Photometric stereo via expectation maximization,” *IEEE PAMI*, vol. 32, no. 3, pp. 546–560, Mar. 2010.

[185] M. Chandraker and R. Ramamoorthi, “What an image reveals about material reflectance,” in *ICCV*. IEEE, 2011, pp. 1076–1083.

[186] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, “Robust photometric stereo via low-rank matrix completion and recovery,” *ACCV*, pp. 703–717, 2011.

[187] J. Ackermann, F. Langguth, S. Fuhrmann, and M. Goesele, "Photometric stereo for outdoor webcams," in *CVPR*. IEEE, 2012, pp. 262–269.

[188] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi, "Elevation angle from reflectance monotonicity: Photometric stereo for general isotropic reflectances," *ECCV*, pp. 455–468, 2012.

[189] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa, "Robust photometric stereo using sparse regression," in *CVPR*, 2012, pp. 318–325.

[190] M. Chandraker, J. Bai, and R. Ramamoorthi, "On differential photometric reconstruction for unknown, isotropic BRDFs," *IEEE PAMI*, vol. 35, no. 12, pp. 2941–2955, 2013.

[191] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa, "Photometric stereo using sparse bayesian regression for general diffuse surfaces," *IEEE PAMI*, vol. 36, no. 9, pp. 1816–1831, 2014.

[192] S. Ikehata and K. Aizawa, "Photometric stereo using constrained bivariate regression for general isotropic surfaces," in *CVPR*, 2014, pp. 2179–2186.

[193] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi, "Bi-polynomial modeling of low-frequency reflectances," *IEEE PAMI*, vol. 36, no. 6, pp. 1078–1091, June 2014.

[194] H. Hayakawa, "Photometric stereo under a light source with arbitrary motion," *JOSA A*, vol. 11, no. 11, pp. 3079–3089, 1994.

[195] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, "The bas-relief ambiguity," *International Journal of Computer Vision*, vol. 35, no. 1, pp. 33–44, Nov. 1999.

[196] A. L. Yuille, D. Snow, R. Epstein, and P. N. Belhumeur, "Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 203–222, 1999.

[197] A. S. Georghiades, "Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo," in *ICCV*, vol. 2, 2003, pp. 816–823.

[198] M. K. Chandraker, F. Kahl, and D. J. Kriegman, "Reflections on the generalized bas-relief ambiguity," in *CVPR*, 2005, pp. 788–795.

[199] N. G. Alldrin, S. P. Mallick, and D. J. Kriegman, "Resolving the generalized bas-relief ambiguity by entropy minimization," in *CVPR*, June 2007, pp. 1–7.

[200] B. Shi, Y. Matsushita, Y. Wei, C. Xu, and P. Tan, "Self-calibrating photometric stereo," in *CVPR*, June 2010, pp. 1118–1125.

[201] P. Favaro, "A closed-form solution to uncalibrated photometric stereo via diffuse maxima," in *CVPR*, 2012, pp. 821–828.

[202] Z. Wu and P. Tan, "Calibrating photometric stereo by holistic reflectance symmetry analysis," in *CVPR*, 2013, pp. 1498–1505.

[203] Y. Quéau, F. Lauze, and J.-D. Durou, "Solving uncalibrated photometric stereo using total variation," *Journal of Mathematical Imaging and Vision*, vol. 52, no. 1, pp. 87–107, 2015.

[204] B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan, "A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo," in *CVPR*, 2016.

[205] E. N. Coleman and R. Jain, "Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry," *Computer graphics and image processing*, vol. 18, no. 4, pp. 309–328, 1982.

[206] F. Solomon and K. Ikeuchi, "Extracting the shape and roughness of specular lobe objects using four light photometric stereo," *IEEE PAMI*, vol. 18, no. 4, pp. 449–454, 1996.

[207] F. Verbiest and L. V. Gool, "Photometric stereo with coherent outlier handling and confidence estimation," in *CVPR*, June 2008, pp. 1–8.

[208] C. Yu, Y. Seo, and S. W. Lee, "Photometric stereo from maximum feasible lambertian reflections," in *ECCV*, 2010, pp. 115–126.

[209] M. Chandraker, S. Agarwal, and D. Kriegman, "Shadowcuts: Photometric stereo with shadows," in *CVPR*, June 2007, pp. 1–8.

[210] T. Zickler, S. P. Mallick, D. J. Kriegman, and P. N. Belhumeur, "Color subspaces as photometric invariants," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 13–30, 2008.

[211] K. Sunkavalli, T. Zickler, and H. Pfister, "Visibility subspaces: Uncalibrated photometric stereo with shadows," *ECCV*, pp. 251–264, 2010.

[212] Y. Mukaigawa, Y. Ishii, and T. Shakunaga, "Analysis of photometric factors based on photometric linearization," *JOSA A*, vol. 24, no. 10, pp. 3326–3334, 2007.

[213] M. Oren and S. K. Nayar, "Generalization of the lambertian model and implications for machine vision," *International Journal of Computer Vision*, vol. 14, no. 3, pp. 227–251, 1995.

[214] H.-S. Chung and J. Jia, "Efficient photometric stereo on glossy surfaces with wide specular lobes," in *CVPR*, June 2008, pp. 1–8.

[215] N. G. Alldrin and D. J. Kriegman, "Toward reconstructing surfaces with arbitrary isotropic reflectance: A stratified photometric stereo approach," in *ICCV*, 2007, pp. 1–8.

[216] N. Alldrin, T. Zickler, and D. Kriegman, "Photometric stereo with non-parametric and spatially-varying reflectance," in *CVPR*, June 2008, pp. 1–8.

[217] T. Higo, Y. Matsushita, and K. Ikeuchi, "Consensus photometric stereo," in *CVPR*, June 2010, pp. 1157–1164.

[218] A. Hertzmann and S. M. Seitz, "Example-based photometric stereo: Shape reconstruction with general, varying BRDFs," *IEEE PAMI*, vol. 27, no. 8, pp. 1254–1264, Aug. 2005.

[219] M. Holroyd, J. Lawrence, G. Humphreys, and T. Zickler, "A photometric approach for estimating normals and tangents," *ACM Trans. on Graphics*, vol. 27, no. 5, p. 133, 2008.

[220] T. Simchony, R. Chellappa, and M. Shao, "Direct analytical methods for solving poisson equations in computer vision problems," *IEEE PAMI*, vol. 12, no. 5, pp. 435–446, May 1990.

[221] S. W. Hasinoff, "Photon, poisson noise," in *Computer Vision.* Springer, 2014, pp. 608–610.

[222] Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. E. Zickler, "From shading to local shape," *IEEE PAMI*, vol. 37, no. 1, pp. 67–79, 2015.

[223] R. T. Frankot and R. Chellappa, "A method for enforcing integrability in shape from shading algorithms," *IEEE PAMI*, vol. 10, no. 4, pp. 439–451, 1988.

[224] M. Harker and P. O'Leary, "Least squares surface reconstruction from measured gradient fields," in *CVPR 2008.* IEEE, 2008, pp. 1–7.

[225] M. Harker and P. O'Leary, "Regularized reconstruction of a surface from its measured gradient field," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 46–70, 2015.

[226] Y. Quéau and J.-D. Durou, "Edge-preserving integration of a normal field: Weighted least-squares, tv and l1 approaches," in *International Conference on Scale Space and Variational Methods in Computer Vision.* Springer, 2015, pp. 576–588.

[227] A. Agrawal, R. Raskar, and R. Chellappa, "What is the range of surface reconstructions from a gradient field?" in *ECCV 2006.* Springer, 2006, pp. 578–591.

[228] H.-S. Ng, T.-P. Wu, and C.-K. Tang, "Surface-from-gradients without discrete integrability enforcement: A gaussian kernel approach," *IEEE PAMI*, vol. 32, no. 11, pp. 2085–2099, 2010.

[229] Z. Wu and L. Li, “A line-integration based method for depth recovery from surface normals,” *Computer Vision, Graphics, and Image Processing*, vol. 43, no. 1, pp. 53–66, 1988.

[230] A. Robles-Kelly and E. R. Hancock, “A graph-spectral method for surface height recovery,” *Pattern recognition*, vol. 38, no. 8, pp. 1167–1186, 2005.

[231] B. K. Horn and M. J. Brooks, “The variational approach to shape from shading,” *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 2, pp. 174–208, 1986.

[232] J. Balzer and T. Mörwald, “Isogeometric finite-elements methods and variational reconstruction tasks in vision—a perfect match,” in *CVPR 2012*. IEEE, 2012, pp. 1624–1631.

[233] J.-D. Durou and F. Courteille, “Integration of a normal field without boundary condition,” in *PACV 2007*. INRIA, 2007, pp. 8–p.

[234] I. Horovitz and N. Kiryati, “Depth from gradient fields and control points: Bias correction in photometric stereo,” *Image and Vision Computing*, vol. 22, no. 9, pp. 681–694, 2004.

[235] K. M. Lee and C.-C. J. Kuo, “Surface reconstruction from photometric stereo images,” *JOSA A*, vol. 10, no. 5, pp. 855–868, 1993.

[236] B. Karaçalı and W. Snyder, “Reconstructing discontinuous surfaces from a given gradient field using partial integrability,” *Computer Vision and Image Understanding*, vol. 92, no. 1, pp. 78–111, 2003.

[237] B. Karaçalı and W. Snyder, “Noise reduction in surface reconstruction from a given gradient field,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 25–44, 2004.

[238] P. Kovesi, “Shapelets correlated with surface normals produce surfaces,” in *ICCV 2005*, vol. 2. IEEE, 2005, pp. 994–1001.

[239] J. Balzer, “A gauss-newton method for the integration of spatial normal fields in shape space,” *Journal of Mathematical Imaging and Vision*, vol. 44, no. 1, pp. 65–79, 2012.

[240] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[241] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. on Image Proc.*, vol. 13, no. 4, pp. 600–612, 2004.

[242] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of Machine Learning Research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[243] R. Latała, "Some estimates of norms of random matrices," *Proceedings of the American Mathematical Society*, vol. 133, no. 5, pp. 1273–1282, 2005.

[244] M. Talagrand, "Concentration of measure and isoperimetric inequalities in product spaces," *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, vol. 81, no. 1, pp. 73–205, 1995.