ABSTRACT

CONVEX OPTIMIZATION TECHNIQUES AND THEIR

APPLICATION IN HYPERSPECTRAL

VIDEO PROCESSING

By

Torin Gerhart

December 2013

Many problems in image and video processing may be formulated in the language of constrained optimization. Algorithms for solving general constrained optimization problems may not guarantee solutions or be computationally efficient, particularly if the problem is nonlinear or non-convex. Oftentimes these constrained optimization problems may be relaxed into the form of a convex problem. This allows for the use of convex solvers such as the Augmented Lagrangian method and the Split Bregman iteration. In this thesis, we will study the advantages of incorporating convexity into constrained optimization problems. These problems will be motivated from the standpoint of hyperspectral image processing, particularly the detection and identification of airborne chemicals in gas cloud releases.

CONVEX OPTIMIZATION TECHNIQUES AND THEIR APPLICATION IN HYPERSPECTRAL VIDEO PROCESSING

A THESIS

Presented to the Department of Mathematics and Statistics

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Applied Mathematics

Committee Members:

Jen-Mei Chang, Ph.D.
Tangan Gao, Ph.D.
William Ziemer, PhD.

College Designee:

Robert Mena, Ph.D.

By Torin Gerhart

B.S. Mathematics, 2010, California State University, Northridge

December 2013

WE, THE UNDERSIGNED MEMBERS OF THE COMMITTEE,

HAVE APPROVED THIS THESIS

CONVEX OPTIMIZATION TECHNIQUES AND THEIR

APPLICATION IN HYPERSPECTRAL

VIDEO PROCESSING

By

Torin Gerhart

COMMITTEE MEMBERS

Jen-Mei Chang, Ph.D. — Mathematics & Statistics

Tangan Gao, Ph.D. — Mathematics & Statistics

William Ziemer, PhD. — Mathematics & Statistics

ACCEPTED AND APPROVED ON BEHALF OF THE UNIVERSITY

Robert Mena, Ph.D.
Department Chair, Department of Mathematics and Statistics

California State University, Long Beach

December 2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Many of the recent developments in the field of optimization have led to fast, efficient algorithms for solving certain types of constrained optimization problems. These problems arise naturally in image processing; examples include deblurring and denoising. Efficient convex solvers have become of particular interest due to the development of advanced imaging technologies. Modern spectral imaging produces large, and complex data sets that hold a substantial amount of information about the scene. Extracting this information requires complicated mathematical models, oftentimes in the form of constrained optimization problems, such as the minimization of a functional subject to some constraints.

This work will focus on two of these models with applications to hyperspectral image and video processing. One model involves the addition of ℓ_1 regularizer and a type of augmented Lagrangian to solve overdetermined systems very quickly. This type of problem is related to hyperspectral unmixing, where abundances of chemical signatures are determined in a given image. The other model involves finding a low-rank and sparse decomposition of a given matrix. This method stems from the convexification of a non-convex problem that would otherwise be very difficult to solve. The convexified problem is solved easily by a alternating Lagrange multiplier method. These low-rank and sparse decompositions are used in video processing as a means of background subtraction. The incorporation of regularizers and convexity into these constrained problems

provides means of obtaining solutions under a variety of conditions that may not be obtainable otherwise.

In chapter 2 ideas from constrained optimization will be discussed, starting from the basic form of constrained problems and ending with numerical convex solvers. Chapter 3 will provide background information on hyperspectral imaging and how certain image processing problems are formulated on these data sets. This section will include specific information about the data set used throughout this work. Chapter 4 will introduce the problem of hyperspectral unmixing and discuss a numerical method of obtaining sparse solutions to a constrained optimization problem. Low-rank and sparse matrix decompositions will be explored in chapter 5, along with their relation to principle components analysis. Their connection to matrix recovery problems will be shown through the numerical methods used to solve these types of decompositions. Results of background subtraction using low-rank and sparse decompositions will be shown. The final section will discuss recent developments, current trends, and future work in these fields.

The field of hyperspectral imaging has been around for several decades, however hyperspectral video sequences are a relatively new type of dataset. In addition to this, the hyperspectral video datasets containing chemical plume releases used in this work were only recently declassified and not well understood. As a consequence, the low-rank and sparse decomposition results presented in chapter 5 are novel. These results contribute to a better understanding of the detection of chemical plumes as well as the collection and analysis of hyperspectral video data.

CHAPTER 2

OPTIMIZATION AND BASIS PURSUIT PROBLEMS

This chapter outlines many of the important concepts from constrained optimization problems that appear in subsequent chapters. It starts with the general form of a constrained optimization problem. The Lagrangian form is presented, along with ideas of convexity and bounds on optimal solutions. Basis pursuit problems are then introduced. These problems have important connections to rank minimization problems that are analyzed in chapter 5. Two different solution methods for constrained optimization problems are derived in section 2.3, the Augmented Lagrangian method and the Alternating Direction Method of Multipliers, respectively. These algorithms will be used to solve two different constrained optimization problems from hyperspectral image processing in chapters 4 and 5.

Constrained Optimization

The general formulation of a constrained mathematical optimization problem is of the following form:

$$\begin{aligned} &\underset{x}{\text{minimize}}\ f_0(x) \\ &\text{subject to } f_i(x) \leq 0,\ i = 1, \ldots, m \\ &\qquad\qquad h_j(x) = 0,\ j = 1, \ldots, p \end{aligned} \tag{2.1}$$

where, $x \in \mathbb{R}^n$ is the *optimization variable*, f_0 is called the *objective function*, and the f_i's and h_j's are *constraint functions*. This form includes the possibility of inequality and equality constraints. Notice that each equality constraint $h_j(x) = 0$

may be replaced by two inequality constraints, $h_j(x) \leq 0$, and $-h_j(x) \leq 0$, which imposes additional structure on each $h_j(x)$. If all the objective and constraint functions satisfy

$$\begin{aligned} f_i(\alpha x + \beta y) &= \alpha f_i(x) + \beta f_i(y) \\ h_j(\alpha x + \beta y) &= \alpha h_j(x) + \beta h_j(y) \end{aligned} \tag{2.2}$$

for all $x, y \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$, then problem (2.1) is referred to as a *linear program*. If the constraint and objective functions f_i have a slightly more general structure defined by,

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{2.3}$$

for all $x, y \in \mathbb{R}^n$, for any $\alpha, \beta \in \mathbb{R}_+$ such that $\alpha + \beta = 1$, and the equality constraint functions are affine, then the problem is called *convex*. Convexity of the equality constraint functions $h_j(x)$ is a byproduct of the requirement to be affine (linear plus a constant). That is, the only way for $h_j(x)$ to satisfy $h_j(x) = 0$ and also be convex is to be affine. A function is called *strictly convex* if equality in the above formula is never achieved. Convexity incorporates a number of useful properties into problem (2.1). Most importantly it guarantees the existence of a global minimum if a local minimum exists. If each f_i is strictly convex then the uniqueness of the minimizer of each function f_i (if a minimum exists) is also guaranteed. Also, note that all norms on $\mathbf{R}^n$ are convex by the triangle inequality. Writing the primal Lagrangian of (2.1), we get

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x) \tag{2.4}$$

such that $\lambda_i, \nu_i \in \mathbb{R}, \lambda_i \geq 0$ for all i. The variables λ_i and ν_j are referred to as the

Lagrange multipliers associated with the inequality and equality constraint $f_i(x)$, and $h_j(x)$, respectively. The vectors λ and ν are called *dual variables*. The *Lagrange dual function* is defined as

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x \in D} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_j(x) \right) \\ &= \inf_{x \in D} \mathcal{L}(x, \lambda, \nu) \end{aligned} \tag{2.5}$$

For certain values of x, λ, and ν the value of g can go to $-\infty$. Parameters $\lambda \geq 0, \nu$, for which $g > -\infty$ are called *dual feasible*. The important thing to notice here is since the dual Lagrangian function is a pointwise infimum, it is always a concave function of λ and ν, even if the primal is not convex [2]. This dual function holds important information about the lower bound of the optimal solution of (2.1). Suppose $\tilde{x}$ is in the *feasable set*, that is $\tilde{x}$ satisfies the constraints of (2.1). So $f_i(\tilde{x}) \leq 0$ and $h_j(\tilde{x}) = 0$ and $\lambda_i \geq 0$ which implies

$$\begin{aligned} \mathcal{L}(\tilde{x}, \lambda, \nu) &= f_0(\tilde{x}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{j=1}^{n} \nu_j h_j(\tilde{x}) \\ &\leq f_0(\tilde{x}) \end{aligned} \tag{2.6}$$

then,

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x \in D} \mathcal{L}(x, \lambda, \nu) \\ &\leq \mathcal{L}(\tilde{x}, \lambda, \nu) \\ &\leq f_0(\tilde{x}) \end{aligned} \tag{2.7}$$

Since equation (2.7) holds for any x in the feasible set, it must hold for the optimal point as well. This gives the lower bound on the optimal value $p^\star$,

$$g(\lambda, \nu) \leq p^\star \tag{2.8}$$

This optimal lower bound is given by the solution to the dual problem,

$$\begin{aligned} \underset{\lambda,\nu}{\text{maximize}} \quad & g(\lambda,\nu) \\ \text{subject to} \quad & \lambda_i \geq 0,\ i = 1,\ldots,m \end{aligned} \tag{2.9}$$

which is convex. Let $d^\star$ be the optimal value of the dual problem. *Weak duality* is the case when $d^\star \leq p^\star$, and *strong duality* is when equality holds. Strong duality will hold when the primal problem is convex and the constraints are affine. Since dual problems are often easier to solve if the original problem is convex, this gives a fast solution method. Namely, solve the dual problem for $\lambda^\star$, and $\nu^\star$ and convert to the primal domain to obtain the solution $x^\star$[2].

Basis Pursuit

Another important type of optimization problem that will be related to topics in later chapters is known as the *basis pursuit problem*,

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \|u\|_1 \\ \text{subject to} \quad & Au = f \end{aligned} \tag{2.10}$$

where f is the original signal and A is an $m \times n$ matrix, and $\|\cdot\|_1$ is the ℓ_1 norm. This problem seeks u with minimal norm, that best recreates the given signal f, with respect to the basis A. Since the ℓ_1 norm is the sparsity inducing norm on $\mathbb{R}^n$ solutions to (2.10) are guaranteed to be the sparsest (containing the most zeros) relative to the provided basis[12]. These types of problems are common in the field of compressed sending where a basis is sought to achieve some amount of sparsity when representing the signal f. The constrained basis pursuit problem is often relaxed to an unconstrained problem of the form [2],

TABLE 1. Related Ideas from Vector (left) and Matrix (right) Minimization

Parsimony concept	Cardinality	Rank
Hilbert space norm	Euclidean	Frobenius
Sparsity inducing norm	ℓ_1	nuclear
Dual norm	ℓ_∞	operator
Convex optimization	Linear programming	Semidefinite programming

$$\underset{u}{\text{minimize}} \quad \mu\|u\|_1 + \frac{1}{2}\|Au - f\|_2^2 \tag{2.11}$$

In this form, this as an unconstrained least squares problem with an ℓ_1 regularizer. The underdetermined case ($m < n$) is common in sparse coding and compressed sensing. Chapter 4 will consider the overdetermined case ($m > n$) which occurs in hyperspectral image unmixing.

These ideas may be further extended to matrices, where minimization is performed over $\mathbb{R}^{m\times n}$. This comes from relating the ideas of cardinality and rank. Table 1 extends ideas of minimizing vectors to minimizing matrices. It is from reference [12], which also contains a detailed discussion relating these concepts. The parallel between the introduction of sparsity for vectors and matrices comes up in chapter 5 when the low-rank and sparse matrix decomposition problem is analyzed.

Solution Methods

Augmented Lagrangian methods transform a constrained optimization problem into a series of unconstrained problems. This is done by writing the unconstrained objective function as the Lagrangian of the constrained problem and adding an additional term. The additional term is designed to act similar to a

Lagrange multiplier [2]. Specifically, a problem of the form,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) = 0,\ i = 1, \ldots, n \end{aligned} \tag{2.12}$$

is transformed into a series of unconstrained problems. The Lagrangian of the above problem would be

$$\begin{aligned} \mathcal{L}(x, \lambda) &= f(x) + \sum_{i=1}^{n} \lambda_i g_i(x) \\ &= f(x) + \lambda^T g(x) \end{aligned} \tag{2.13}$$

Notice that if $x^\star$ solves (2.12) then $\mathcal{L}(x^\star, \lambda) = f(x^\star) + \lambda^T g(x^\star) = f(x^\star) = \textit{const.}$ Now, the *augmented Lagrangian* is given by adding a penalty term to Lagrangian,

$$\begin{aligned} \mathcal{L}_A(x, \lambda, \mu_n) &= f(x) + \frac{\mu_n}{2} \sum_{i=1}^{n} g_i^2(x) + \sum_{i=1}^{n} \lambda_i g_i(x) \\ &= f(x) + \frac{\mu_n}{2} \|g(x)\|_2^2 + \lambda^T g(x) \end{aligned} \tag{2.14}$$

this additional term shows preference to solutions that make $\|g(x)\|_2^2$ small. The strategy for minimization of this function is to fix λ and μ_n and minimize $\mathcal{L}_A(x, \lambda, \mu_n)$ with respect to x. Then, update λ based on the new x and re-solve the problem with a larger value μ_n. That is, $\{\mu_n\}$ is a sequence of increasing values. Larger values of μ_n increase the emphasis on the additional penalty term. The updates for x and λ are given by the iterations,

$$\begin{aligned} x_{k+1} &= \underset{x}{\text{argmin}}\ f(x) + \frac{\mu_n}{2} \|g(x)\|_2^2 + \lambda^T g(x) \\ \lambda_{k+1} &= \lambda_k - \mu_n g_i(x_{k+1}) \end{aligned} \tag{2.15}$$

Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers is for convex optimization problems with an objective function that has a special type of separable structure associated with it. Consider the problem,

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & J(u) \\ \text{subject to} \quad & Ku = f \end{aligned} \tag{2.16}$$

where $J(u)$ may be written as $J(u) = H(u) + \sum_{i=1}^{n} G_i(A_i u + b_i) = H(u) + F(z)$, where H and G_i are convex and $z_i = A_i u + b_i$. The separable structure is what allows J to be split into two convex functionals [10]. This means that problem (2.16) may be re-written as,

$$\underset{z,u:Au+Bz=c}{\text{minimize}} \; F(z) + H(u) \tag{2.17}$$

where,

$$F(z) = \sum_{i=1}^{n} G_i(z_i), \; B = \begin{pmatrix} -I \\ 0 \end{pmatrix}, \; A = \begin{pmatrix} A_1 \\ \vdots \\ A_N \\ K \end{pmatrix}, \text{ and } c = \begin{pmatrix} -b_1 \\ \vdots \\ -b_N \\ f \end{pmatrix}$$

The Augmented Lagrangian of problem (2.17) is

$$\mathcal{L}_A(u, z, \lambda) = F(z) + H(u) + \langle \lambda, c - Au - Bz \rangle + \frac{\alpha}{2} \|c - Au - Bz\|_2^2 \tag{2.18}$$

Then the objective function is then minimized by first fixing z and minimizing

$\mathcal{L}_A(u, z, \lambda_i)$ over u, then fixing this new u and minimizing $\mathcal{L}_A(u, z, \lambda_i)$ over z, and then updating λ_i. These iterations are given by [10],

$$\begin{aligned} z^{k+1} &= \underset{z}{\operatorname{argmin}}\ \mathcal{L}_A(z, u^k, \lambda^k) \\ u^{k+1} &= \underset{u}{\operatorname{argmin}}\ \mathcal{L}_A(z^{k+1}, u, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \alpha(c - Au^{k+1} - Bz^{k+1}) \end{aligned} \tag{2.19}$$

Connections to Bregman Iteration

Bregman iterations are a class of algorithms related to Alternating Direction Method of Multipliers and similar Lagrangian based methods involving sums of convex functionals subject to equality constraints [10]. Variations on the Bregman iteration have led to fast solvers for convex programs. These algorithms require some additional assumptions about the objective function. Specifically, the function must have some separable structure that allows the problem to be split into a series of subproblems. These algorithms involve generalized notions of the derivative, known as *subdifferentials*, or *subgradients*. It has been shown that classical Bregman iteration is analogous to augmented Lagrangian methods when the constraints are linear [15]. Essentially, these connections show that the augmented Lagrangian, up to an additive constant, is equivalent to the objective function used in the Bregman iteration. A more recent variant of Bregman iteration, the split Bregman algorithm, was shown to be closely related to ADMM [10] [16]. In this work, the constraints on the considered minimization problems will all be linear. The solutions methods will be presented from the standpoint of augmented Lagrangian methods since these methods are equivalent to Bregman iteration and avoid many complicated notions of generalized derivatives. However,

the reader should note that many of the references used herein refer to these algorithms as Bregman methods.

CHAPTER 3

HYPERSPECTRAL IMAGING

Hyperspectral imaging extends traditional imaging techniques, and allows for a more complete depiction of the electromagnetic spectrum. This chapter is intended to provide a background to those unfamiliar with hyperspectral images and outline the plume detection problem that will be investigated further in later chapters. A major part of the plume detection problem has to do with the way material signatures mix in the scene. Understanding the mixture models discussed in section 3.3 will provide better insight into the methods discussed in chapter 4. Detailed information on the particular dataset used in this work may be found in section 3.2.

Background

RGB images consist of three layers, where each layer corresponds to an intensity value of the red, green, and blue components of each image. These three layers each represent a particular wavelength of the electromagnetic spectrum. Hyperspectral images extend this idea into hundreds of dimensions, capturing information about many different frequencies of light. This allows for the identification of particular materials in each image based on what frequencies are absorbed and which are reflected.

Hyperspectral imaging aims to view a large portion of the radiated electromagnetic radiation of an object with high spectral resolution. This requires a very fine discretization of the electromagnetic spectrum. This is often limited to a narrow spectral band in order to reproduce a more continuous spectrum. By

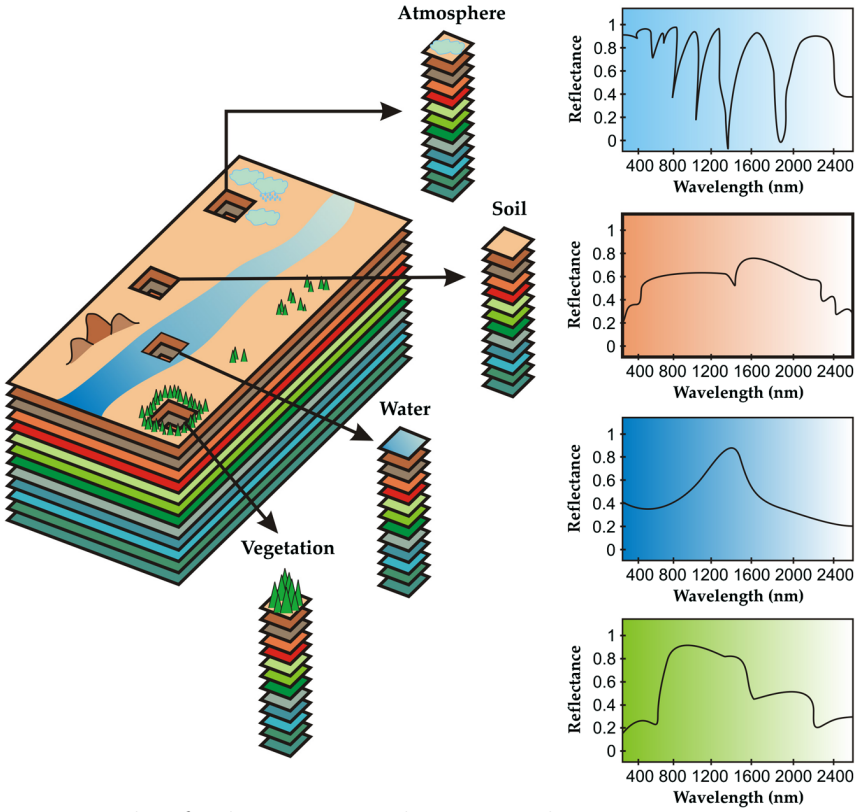

FIGURE 1. Example of a hyperspectral image. The image contains many layers that capture different wavelengths of reflected light. Each pixel contains information about the materials present in the scene.

imaging the light that is absorbed and reflected it is possible to identify particular materials present in the image. Figure 1 shows how pixels in a hyperspectral image contain spectral information that may be used to distinguish between vegetation, water, soil, and atmosphere.

Hyperspectral video sequences serve as an excellent example of a problem from modern imaging science that is closely tied to the analysis of large data sets. Each hyperspectral video frame is an individual data cube, which means these video sequences require a large amount of memory. The size of these videos may prohibit loading the entire video into memory without sufficient preprocessing or dimension reduction. Hyperspectral imaging was originally developed to be used

for geology and mining applications. The imaging of core samples would detect the presence of particular mineral deposits and could aid in finding oil [1]. Today, hyperspectral imaging is used in many different areas from surveillance and defense to ecology and agriculture. One major area of interest in hyperspectral video is the detection, identification, and tracking of gas plumes. This problem has major applications in defense, security, and environmental safety [13].

Dugway Proving Ground Dataset

The hyperspectral data set analyzed for this project was provided by the Applied Physics Laboratory at Johns Hopkins University as part of a Defense Threat Reduction Agency (DTRA) research grant. It consists of a series of video sequences recording the release of chemical plumes into the atmosphere. Figure 2 shows the three long wave infrared spectrometers (named Romeo, Victory and Tango) placed at different locations to track the release of known chemicals. The sensors capture one frame every five seconds consisting of 2 spatial dimensions and one spectral dimension. The spatial dimension of each of these data cubes is 128 $\times$ 320 pixels, while spectral dimension measures 129 different wavelengths in the long wave infrared (LWIR) portion of the electromagnetic spectrum. Each layer in the spectral dimension depicts a particular frequency starting at 7,830 nm and ending with 11,700 nm.

Mixing Models

The three-layer model is a simple method to describe the different components that comprise the spectral radiance measurement for each pixel in the long wave infrared hyperspectral image. Figure 3 illustrates the different objects, or layers, that contribute to the spectral radiance measurement of the long wave infrared spectrometer. For the chemical plumes released in this data set, the three layers are the background, the chemical plume and the atmosphere. Each pixel has

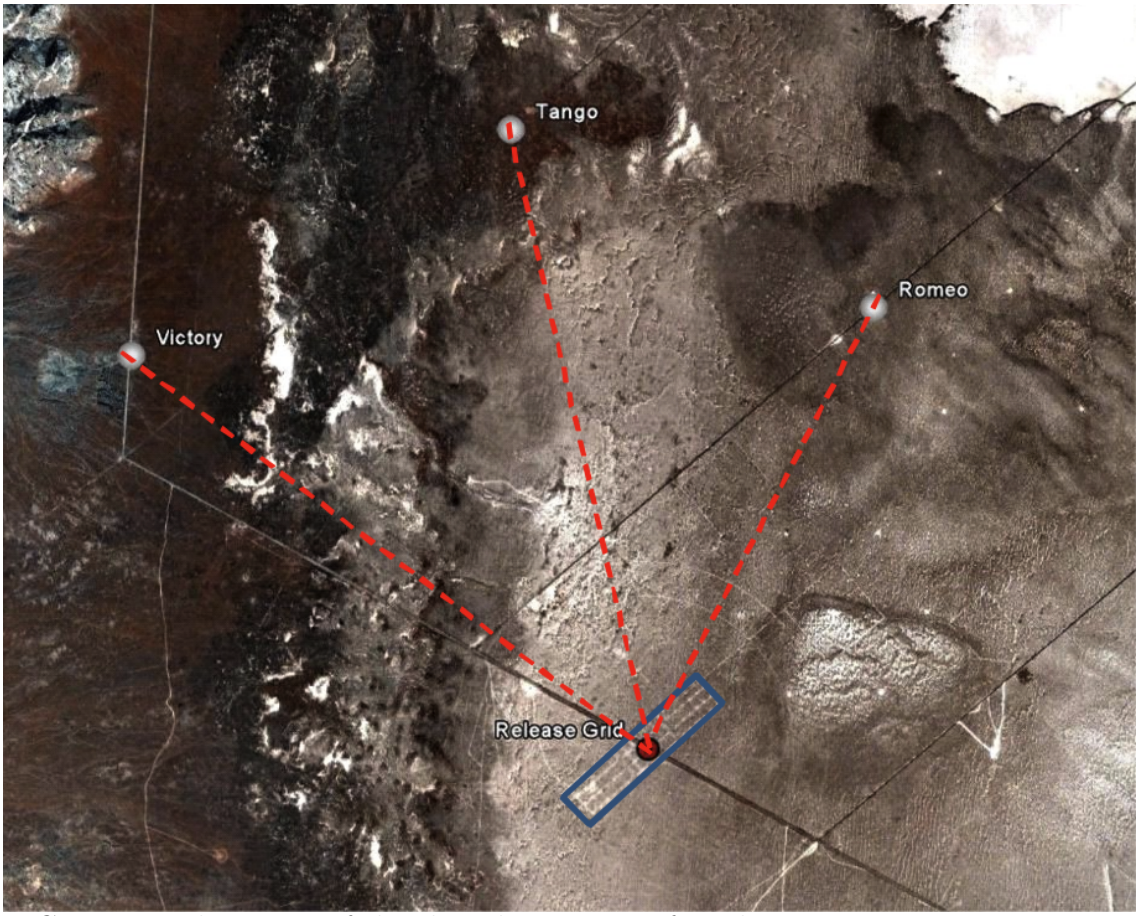

FIGURE 2. Placement of the three long wave infrared spectrometers.

its own radiance $L(\nu)$, and transmittance $\tau(\nu)$. The transmittance is the ratio of light leaving a surface relative to the amount of light entering the medium. Both the background and plume spectral radiances must pass through other mediums before reaching the long wave infrared spectrometer. Therefore, the spectral radiance measurement of the sensor can be represented as

$$L(\nu) = \tau_{atm}(\nu)L_p(\nu) + \tau_p(\nu)\tau_{atm}(\nu)L_b(\nu) + L_{atm}(\nu) \tag{3.1}$$

The subscripts *atm, p* and *b* in Equation (3.1) refer to the atmosphere, plume, and background, respectively. This model can be further simplied for the given data

set. Since the ground based long wave infrared spectrometers are placed within two kilometers of the chemical plume release site, the spectral radiance of the atmosphere is very small in comparison to the other spectral radiance terms, and therefore can be dropped from this equation. In addition, it is assumed that the atmospheric transmittance does not significantly affect the spectral radiance because of the short path length, allowing most of the signal to pass through. These assumptions reduce equation (3.1) into the two layer model equation

$$L(\nu) = \tau_p(\nu)L_b(\nu) + L_p(\nu) \tag{3.2}$$

According to this model the spectral radiance of the scene measured by the LWIR sensors is a sum of the light emitted by the chemical plume and background mediums at 129 wavelengths in the electromagnetic spectrum.

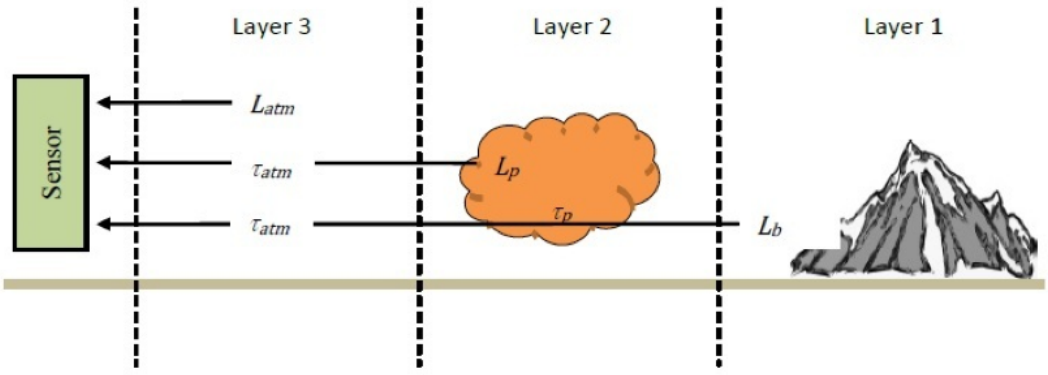

FIGURE 3. Three-layer model depicting the spectral radiance.

Problem Outline

The ultimate goal of this line of research is to accurately detect, identify, and track the releases of chemical gas plumes. In this work, more focus will be placed on detection and identification methods. The detection aspect is concerned

with finding moving objects within each video sequence. Identification matches the spectral signatures from pixels in each frame to the signatures of known materials. Once the chemical cloud is detected and identified, the video tracking algorithms could be incorporated to track and possibly predict where the chemicals disburse. In this work we will focus on the detection and identification problems presented in the context of constrained optimization. The method of identification will use a sparsity inducing form of hyper spectral unmixing in order to determine the presence of a target signature. Low-rank and sparse matrix decompositions will be investigated as part of the detection problem.

CHAPTER 4

CONVEX OPTIMIZATION AND HYPERSPECTRAL UNMIXING

The purpose of unmixing is to calculate the abundance signatures in each pixel of an image. Let's assume that some information is known about the signatures present in the scene and are stored as columns of an $m \times n$ matrix A. This matrix A is commonly referred to as the *dictionary*. Then a pixel, f, maybe represented as $f = Ax + \epsilon$, where ϵ is an error term, that measures how accurately the linear combination of signatures from A recreates f. In other words, the solution to the linear program

$$\begin{aligned} &\underset{x}{\text{minimize}} && \|Ax - f\|_2^2 \\ &\text{subject to} && x_i \geq 0,\ i = 1, \ldots, n. \end{aligned} \tag{4.1}$$

is the vector of abundances that minimizes the error of how accurately Ax depicts f. This type of linear program comes up in many areas of mathematics and is called the *non-negative linear least squares problem*. In practice, this dictionary matrix A is often overdetermined ($m \geq n$) and therefore exact solutions may not exist.

A very distant object or the presence of very diffuse gas will have a much weaker signal than close, dense objects. Therefore small concentrations may be ignored, or set to zero. This line of reasoning is motivated by wanting to incorporate *sparsity* into the solution. Sparse signal representations have been the topic of much current research. In order to capture sparsity in the solution of (4.1), a term called a *regularizer* will be added to the objective function:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|Ax - f\|_2^2 + \eta\|x\|_1, \text{ where } \eta > 0 \\ \text{subject to} \quad & x_i \geq 0,\ i = 1, \ldots, n. \end{aligned} \tag{4.2}$$

The regularizer that was added is the ℓ_1-norm, $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. This term rewards sparse solutions, and penalizes solutions that have many non-zero entries [5]. The parameter η determines how much emphasis on sparsity is desired. The problem stated in (4.2) may be rewritten in the equivalent form,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|Ax - f\|_2^2 + \eta \sum_{i=1}^{n} x_i \\ \text{subject to} \quad & x = P(y) \end{aligned} \tag{4.3}$$

where $P(y)$ is the projection onto $\mathbb{R}_+^n$ and is defined element wise on the vector y. That is, $P(y) = \max(0, y_i) = \max(0, x_i - b_i)$. This operator is sometimes referred to as the *shrink* operator in some literature [15]. The b_i's are introduced to convert the inequality to equity constraints. The augmented Lagrangian of (4.3) is given by,

$$\mathcal{L}_A(x, y, b) = \underset{x,y}{\text{argmin}}\ \lambda\|Ax - f\|_2^2 + \lambda\eta \sum_{i=1}^{n} x_i + \|x - P(y) - b\|_2^2 \tag{4.4}$$

As shown in chapter 2, problem (4.3) may be solved be the classical augmented Lagrangian method [5]. Replacing (4.3) with a sequence of unconstrained problems,

$$\begin{aligned} (x^{k+1}, y^{k+1}) &= \underset{x,y}{\text{argmin}}\ \lambda\|Ax - f\|_2^2 + \lambda\eta \sum_{i=1}^{n} x_i + \|x - P(y) - b^k\|_2^2 \\ b^{k+1} &= b^k + P(y^{k+1}) - x^{k+1} \end{aligned} \tag{4.5}$$

Notice that the first line in (4.5) may be split into two subproblems by holding one variable fixed and solving for the other and vice versa,

$$\begin{aligned} y^{k+1} &= \underset{y}{\operatorname{argmin}}\ \lambda\|Ax^k - f\|_2^2 + \lambda\eta\sum_{i=1}^{n} x_i^k + \|x^k - P(y) - b^k\|_2^2 \\ x^{k+1} &= \underset{x}{\operatorname{argmin}}\ \lambda\|Ax - f\|_2^2 + \lambda\eta\sum_{i=1}^{n} x_i + \|x - P(y^{k+1}) - b^k\|_2^2 \\ b^{k+1} &= b^k + P(y^{k+1}) - x^{k+1} \end{aligned} \tag{4.6}$$

Now, solve each subproblem for its minimizer. Starting with y, differentiating the first equation with respect to y and setting it equal to zero:

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}y}\left(\lambda\|Ax^k - f\|_2^2 + \lambda\eta\sum_{i=1}^{n} x_i + \|x^k - P(y) - b^k\|_2^2\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}y}(-(x^k)^T P(y) + (b^k)^T P(y) - P(y)^T(x^k - P(y) - b^k)) \\ &= \frac{\mathrm{d}}{\mathrm{d}y} - 2(x^k - b^k)^T P(y) + \frac{\mathrm{d}}{\mathrm{d}y}\|P(y)\|_2^2 \\ &= -2P(x^k - b^k) + 2P(y) \\ &\implies P(x^k - b^k) = P(y) = y \\ &\implies y^{k+1} = P(x^k - b^k) \end{aligned} \tag{4.7}$$

That is, the minimum y is found by projecting $x^k - b^k$ into $\mathbb{R}_+^n$. Now, for the x variable. Differentiating the second equation with respect to x and setting equal to

zero:

$$
\begin{aligned}
0 &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\lambda\|Ax-f\|^2+\lambda\eta\sum_{i=1}^{n}x_i+\|x-P(y^{k+1})-b^k\|^2\right)\\
&= \frac{\mathrm{d}}{\mathrm{d}x}\left(\lambda(Ax-f)^T(Ax-f)+\lambda\eta\sum_{i=1}^{n}x_i+x^T(x-2P(y^{k+1})-2b^k)\right)\\
&= 2\lambda A^TAx-2\lambda A^Tf+\lambda\eta\mathbf{1}+2x-2P(y^{k+1})-2b^k\\
&\implies (\lambda A^TA+I)x = P(y^{k+1})+b^k-\lambda\eta\mathbf{1}+\lambda A^Tf\\
&\implies x^{k+1} = (\lambda A^TA+I)^{-1}(P(y^{k+1})+b^k-\lambda\eta\mathbf{1}+\lambda A^Tf)
\end{aligned}
\tag{4.8}
$$

The most expensive part of this calculation is finding $(\lambda A^TA+I)^{-1}$, but never needs to be calculated explicitly. Now that the exact solutions for each subproblem are known, the iterations become [5]

$$
\begin{aligned}
y^{k+1} &= P(x_k-b_k)\\
x^{k+1} &= (\lambda A^TA+I)^{-1}(P(y^{k+1})+b^k-\lambda\eta\mathbf{1}+\lambda A^Tf)\\
b^{k+1} &= b^k+P(y^{k+1})-x^{k+1}
\end{aligned}
\tag{4.9}
$$

Results

In order to perform target detection using ℓ_1-unmixing the dictionary matrix was generated using PCA from frames known to not contain any chemical plume in order to provide a general estimate of background signatures. Then, a known target signature was added to the dictionary. The results of unmixing an entire frame, using the estimated background signatures may be seen in figure 4. The top left frame of figure 4 correctly identifies the signature of the chemical release. Notice in the top right image of figure 4 the algorithm identifies the foreground, the middle right image identifies the distant mountains, and the bottom images identify atmospheric signatures from the sky. Figure 5 shows the

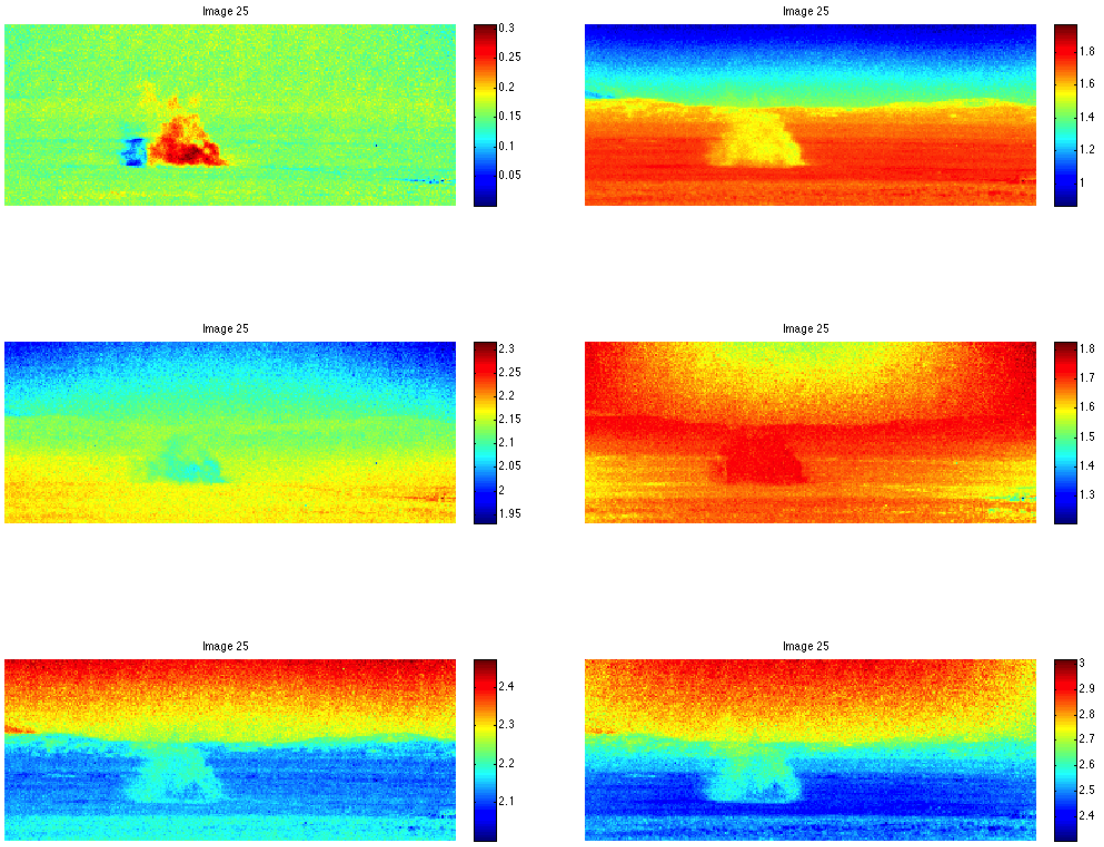

FIGURE 4. The results of ℓ_1-unmixing on a single frame of the video sequence. The top left image identifies the target chemical signature. The remaining five images show other components of the scene, such as foreground (top right, middle left), mountains (middle right), and sky (bottom).

results of using ℓ_1-unmixing to detect a chemical release across multiple frames of a hyper spectral video sequence. The unmixing process averaged around 20 seconds per frame using a 3 GHz Intel i7.

Remarks

Sparse representations, such as those given by equation (4.2), come up in a variety of other situations when the incorporation of sparsity is desired. For example, in non-negative matrix factorization (NNMF). Given a data matrix X, such that $X_{i,j} \geq 0$, NNMF seeks $A, S \geq 0$ such that $X \approx AS$. If the approximation

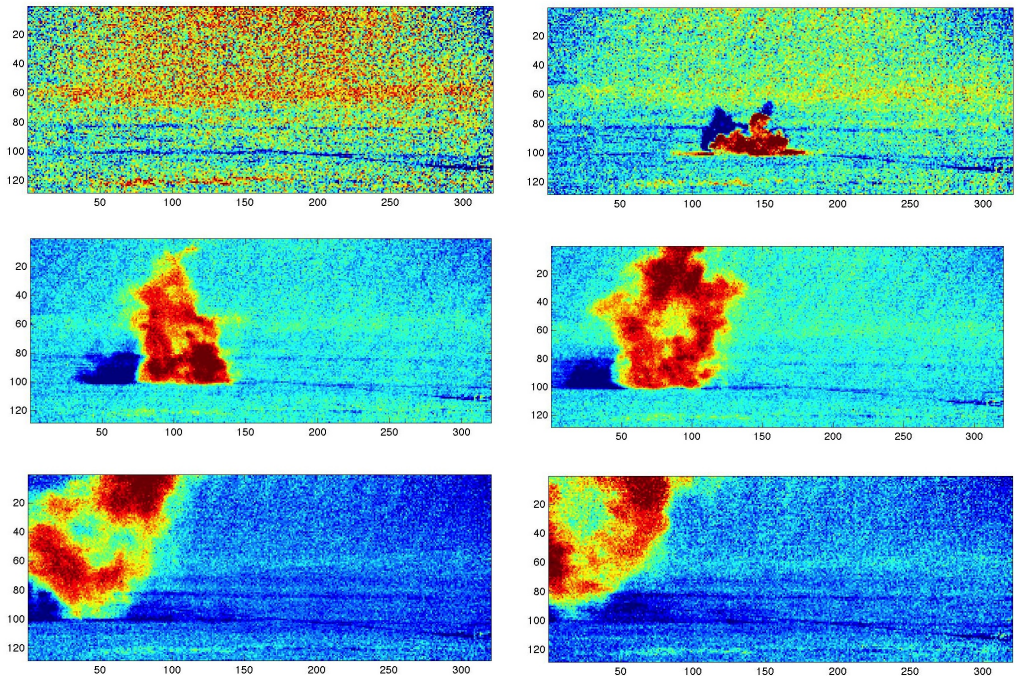

FIGURE 5. The results of ℓ_1-unmixing detecting the target signature in frames 19, 23, 28, 31, 36, and 60 of the video sequence.

involves minimizing the 2-norm, this problem is in some sense solving a non-negative linear least squares problem for every column of X. Thinking of the factor A as the dictionary matrix from section 4, NNMF provides a means of unmixing entire hyperspectral images rather than unmixing pixel-wise, where columns of S represent chemical abundance's. For reasons stated previously, it is natural to want to impose a sparsity constraint on the factor S. So instead of solving non-negative least squares problems for each column of X, one would be solving problem (4.2) at each column.

CHAPTER 5

ROBUST PRINCIPAL COMPONENTS ANALYSIS

The previous chapter showed how sparsity may be incorporated into solutions of a constrained problem by the addition of particular regularizer to the objective function. The focus of this chapter will be extending the ideas from the vector case to solve constrained optimization problems with matrix variables. The minimization of particular matrix norms, such as the nuclear norm has been the focus of much recent research. The nuclear norm is defined to be the sum of the singular values of a matrix, and is involved in a number of low-rank recovery and decomposition problems. Minimization of the nuclear norm was shown to be able to recover low-rank matrices from only a small number of its entries. Nuclear norm minimization is also used to solve low-rank and sparse decompositions where a matrix X is decomposed into $X = L + S$, where L is low-rank, and S is sparse. This particular matrix decomposition is closely related to Principle Components Analysis, and in many ways relaxes several assumptions from classical PCA.

Formulated as a basis pursuit problem, the low-rank and sparse decomposition is able to recover the data represented in the principle component basis of a matrix that is corrupted with arbitrarily large amounts of noise and/or missing entries. The principle components would not be recoverable using the standard method of SVD, as the noise would obscure the true basis.

Low-rank and sparse decompositions have applications to background estimation and motion detection in video processing. An augmented Lagrangian

multiplier approach to low-rank and sparse decompositions will be derived, then applied to false-color RGB videos made from the DTRA chemical plume dataset.

Rank Minimization and the Nuclear Norm

A number of real world applications may be formulated as rank minimization problems. These problems aim to recover a matrix given some number of samples, with the assumption that the underlying matrix has inherently low rank. Formally, the problem is stated as,

$$\begin{aligned} &\underset{L}{\text{minimize}}\ \text{rank}(L) \\ &\text{subject to } \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(X) \end{aligned} \tag{5.1}$$

where $\mathcal{P}_\Omega$ is the projection onto the sample space Ω [8], and X is the matrix of known samples. That is, $\mathcal{P}_\Omega(X) = X_{i,j}$ for $(i,j) \in \Omega$. The problem stated in this way is NP-hard and no efficient algorithms exist for solving this problem for matrices with rank larger than 10 [7]. The main challenge with this problem is the *rank* function, defined to be the number of positive singular values. This objective function is non-convex and any algorithm solving (5.1) would need to distinguish between local vs global minima. However, by replacing the *rank* function with a convex approximation, such as a norm, the problem becomes much easier to solve.

The nuclear norm is in fact the "best" convex approximation to the *rank* function [12]. It is defined as $\|X\|_\star = \sum_{i=1}^{r} \sigma_i(X)$, where $r = rank(X)$. This is essentially the ℓ_1 norm of the singular values of X. Minimization of this functional seeks a low rank approximation of the data. The basic form of a nuclear norm

minimization problem is,

$$\begin{aligned} &\underset{L}{\text{minimize}} \ \|L\|_\star \\ &\text{subject to } \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(X) \end{aligned} \tag{5.2}$$

where,

$$\|X\|_\star = \sum_{i=1}^{r} \sigma_i(X) \text{ (nuclear norm)}$$

In other words, problem (5.2) seeks a matrix L that has low rank and agrees exactly with X on Ω. The recovery of a low rank matrix from only a subset of its entries solves a problem know as the *Netflix prize*. The company was interested in how to provide better suggestions to its customers, based only on knowing a few movies that a customer rented. The resulting data matrix consists of rows corresponding to customers and columns corresponding to movies. Since each customer has only rented a small number of movies, this matrix is very sparse. Recovering missing entries in this matrix would give a better understanding of a particular type of customers viewing habits, and allow for better suggestions to be made to any particular subset of Netflix customers [7].

The solution to the nuclear norm minimization problem (5.2) is related to the singular value thresholding operator, defined as

$$\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^T = U\text{diag}(\max(\sigma_i - \tau, 0))V^T$$

where $S_\tau = max(0, x_i - \tau)$ is the projection or *shrink* operator that solved (4.2). Since the nuclear norm is non-differentiable the minimizer is derived using a projective subgradient method. A discussion on projective subgradient methods may be found in [14], with methods specifically related to the nuclear norm found

in [8], and [6]. A detailed derivation of this solution using subgradient methods is beyond the scope of what is being discussed here. Instead, the method will be interpreted as Lagrange multiplier method in order to avoid many technicalities in deriving the iterations. The singular value thresholding operator solves a problem that is very close to (5.1),

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & \tau\|L\|_\star + \frac{1}{2}\|L\|_F^2 \\ \text{subject to} \quad & \mathcal{P}_\Omega(L) = \mathcal{P}_\Omega(X) \end{aligned} \tag{5.3}$$

where $\|X\|_F^2 = trace(X^T X)$ is the Frobenius norm. Writing the Lagrangian for this problem gives,

$$\mathcal{L}(L, \Lambda) = \tau\|L\|_\star + \frac{1}{2}\|L\|_F^2 + \langle \Lambda, P_\Omega(X - L)\rangle \tag{5.4}$$

Notice that (5.3) is a convex problem with linear constraints, so strong duality holds. In order to solve this problem, a particular method of Lagrange multipliers is known as *Uzawa's algorithm*, which involves varying step sizes. This method is derived by applying a subgradient method to the dual problem. The dual function of (5.3) is given by $g(\Lambda) = \inf_L \mathcal{L}(L, \Lambda)$. This leads to the iteration,

$$\begin{aligned} \mathcal{L}(L_k, \Lambda_{k-1}) &= \min_L \mathcal{L}(L, \Lambda_{k-1}) \\ \Lambda_{k+1} &= \Lambda_k + \delta_k \mathcal{P}_\Omega(X - L_k) \end{aligned} \tag{5.5}$$

where $\{\delta_k\}_{k\geq 1}$ is a sequence of step sizes. The iteration for Λ is derived as a gradient decent update [11]. It is also known [11] that the operator

$\mathcal{D}_\tau(X) = \underset{L}{\operatorname{argmin}} \frac{1}{2}\|X - L\|_F^2 + \tau\|L\|_\star$ and therefore,

$$\begin{aligned} \underset{L}{\operatorname{argmin}}\ \tau\|L\|_\star + \frac{1}{2}\|L\|_F^2 + \langle \Lambda, \mathcal{P}_\Omega(X - L)\rangle = \\ \underset{L}{\operatorname{argmin}}\ \frac{1}{2}\|\mathcal{P}_\Omega(\Lambda) - L\|_F^2 + \tau\|L\|_\star = \mathcal{D}_\tau(\mathcal{P}_\Omega(\Lambda)) \end{aligned} \tag{5.6}$$

That is, $\mathcal{D}_\tau(\mathcal{P}_\Omega(\Lambda))$ is the minimizer of the dual function $g(\Lambda) = \inf_L \mathcal{L}(L, \Lambda)$. Also, since $\mathcal{P}_\Omega(\Lambda_k) = \Lambda_k$ for all $k \geq 0$, the iterations for L and Λ become

$$\begin{aligned} L_{k+1} &= \mathcal{D}_\tau(\Lambda_k) \\ \Lambda_{k+1} &= \Lambda_k + \delta_k \mathcal{P}_\Omega(X - L_k) \end{aligned} \tag{5.7}$$

The singular value thresholding operator provides a means of solving nuclear norm minimization problems over some set of convex constrains. This method is quite efficient, since the dominant cost of each iteration is an SVD.

Principal Component Pursuit

The focus of this section will be a functional consisting of the nuclear norm and the ℓ_1 norm defined on matrices. The method of minimizing this functional uses the singular value thresholding operator discussed in the previous section, and a thresholding operator similar to the projection onto $\mathbb{R}_+$ operator from section 4. This functional is related to low rank representations of corrupted data and its representation in the principle component basis.

There are a number of assumptions that are present when attempting to reduce the dimension of a dataset. Dimension reduction assumes that the data has some intrinsic low dimensionality associated with it, such as existing on a low dimensional subspace or manifold perturbed by noise. Therefore the data matrix may be decomposed as $X = L + N$, where L is the low rank representation of the

data, and N is a small perturbation matrix. Principal Components Analysis may be formulated as the constrained matrix optimization problem

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & \|X - L\|_2 \\ \text{subject to} \quad & \text{rank}(L) \leq k \end{aligned} \tag{5.8}$$

for data matrix X. Notice that the objective function being minimized is the ℓ_2 matrix norm, which is the largest singular value of the matrix. The classical method of solving problem (5.8) is by singular value decomposition (SVD). This problem may be solved very efficiently under the assumption that the noise is small and i.i.d. Gaussian. However, if the data is corrupted, lossy, or noisy, this method may fail to accurately find this low-rank matrix L.

The *Principal Component Pursuit* (PCP) problem relaxes a number of assumptions from classical PCA. Namely, it allows entries in X to be arbitrarily large, and have unknown, sparse, support. Now, the data matrix may be decomposed as $X = L + S$, where S is a general sparse matrix. PCP aims to *exactly* recover the low-rank and sparse components of this noisy data matrix X by solving

$$\begin{aligned} \underset{L,S}{\text{minimize}} \ & \|L\|_\star + \lambda \|S\|_1 \\ \text{subject to} \ & X = L + S \end{aligned} \tag{5.9}$$

where,

$$\|S\|_1 = \sum_{i,j} |S_{i,j}| \ \ (\ell_1 \text{ norm})$$

Notice that the $\|\cdot\|_1$ norm in this case is different from the matrix 1-norm. It is defined as the ℓ_1 norm of the matrix seen as a long vector. Writing the

augmented Lagranigian [8] of (5.9),

$$\mathcal{L}_A(L, S, Y) = \|L\|_\star + \lambda\|S\|_1 + \langle Y, X - L - S\rangle + \frac{\mu}{2}\|X - L - S\|_F^2 \tag{5.10}$$

This objective function of equation (5.9) has a separable structure associated with it, and therefore makes the Alternating Direction Method of Multipliers (see section 2.4) an efficient means of solution. Proceeding accordingly, minimizing the Lagrangian may be broken into two easily solvable subproblems in a similar manner as was done to equation (4.5). One subproblem holds S fixed and minimizes $\mathcal{L}_A(L, S, Y)$ with respect to L, then holds the new L fixed and minimizes $\mathcal{L}_A(L, S, Y)$ with respect to S. The solution to the subproblem involving minimization of the nuclear norm is given by $\mathcal{D}_\tau$. Since the other subproblem is essentially the ℓ_1 vector norm, the same projection operator that solved (4.2) will solve the subproblem involving the $\|S\|_1$ term.

$$\begin{aligned} L_{k+1} &= \underset{L}{\operatorname{argmin}}\ \mathcal{L}_A(L_k, S_k, Y_k) = \mathcal{D}_\tau(X - S_k + \mu^{-1}Y_k) \\ S_{k+1} &= \underset{S}{\operatorname{argmin}}\ \mathcal{L}_A(L_{k+1}, S_k, Y_k) = \mathcal{S}_{\lambda\mu^{-1}}(X - L_{k+1} + \mu^{-1}Y_k) \\ Y_{k+1} &= Y_k + \mu(X - L_{k+1} - S_{k+1}) \end{aligned} \tag{5.11}$$

In this case the projection operator was rewritten as $S_{\lambda\mu^{-1}} = max(0, x_i - \lambda\mu^{-1})$. In the application to video analysis, columns of the data matrix represent frames, the low rank matrix represents the background, and the perturbation matrix represents any motion between frames.

Results

The Low-Rank Sparse algorithm outlined in (5.11) was applied to a false color RGB video, created by projecting each frame onto the first three principle

components. The movie generated for these results was done in RGB to demonstrate different aspects of the decomposition. Movies with more layers were processed and the results are similar to what is presented here. In early frames, the low rank approximation is able to capture the background very well. After the plume is released, the sparse component captures the movement of the plume through each channel of the video sequence. Applying this method to the original (non-reduced) video sequence results in the background matrix approximating stationary signals and the sparse component showing moving signals and noise. Each frame contained 128×320 pixels, with 3 layers. By concatenating the frames into long vectors in $\mathbb{R}^{128 \cdot 320 \cdot 3}$, a movie of 40 frames was made into the $128 \cdot 320 \cdot 3 \times 40$ matrix M. The low rank approximation L had rank 12.

Motion was captured on each RGB layer in the sparse component S. The red component of the resulting decomposition may be seen in figures 6 and 7. The first half of the video sequence is shown in figure 6. The frames of the low rank decomposition are virtually identical, and noise may be seen in the sparse component. The second half of the video sequence, seen in figure 7, shows the release of the plume. Its motion is captured throughout the sparse component, however some anomalies appear in the low rank component towards the end of the sequence in areas where the plume has traveled. The green and blue components of the resulting decomposition may be seen in figures 8, 9, 10, and 11, respectively. The final resulting RGB movies may be seen in figures 12, and 13.

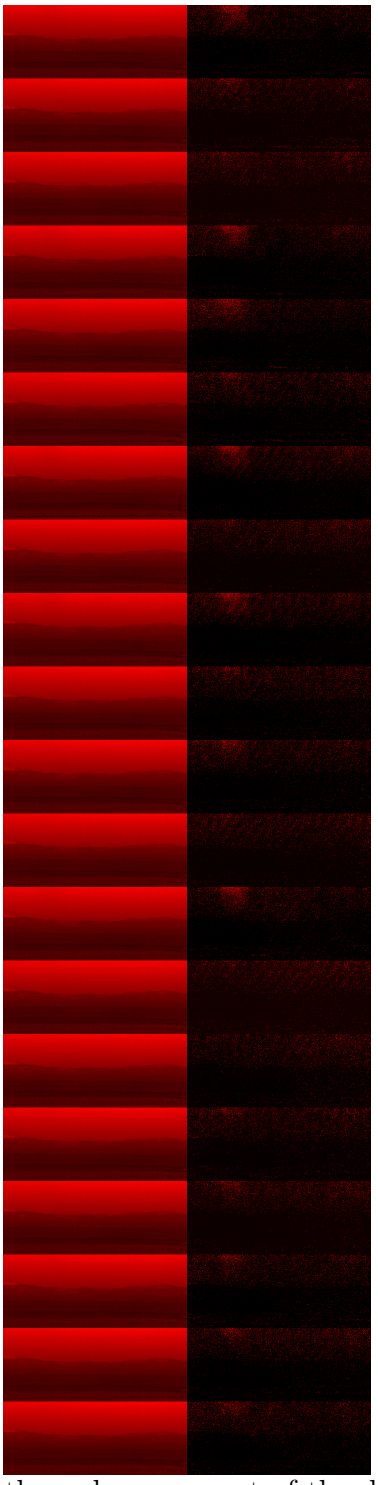

FIGURE 6. Frames 1-20 of the red component of the decomposed video sequence. The left hand side is the low rank approximation, and the right hand side is the sparse component.

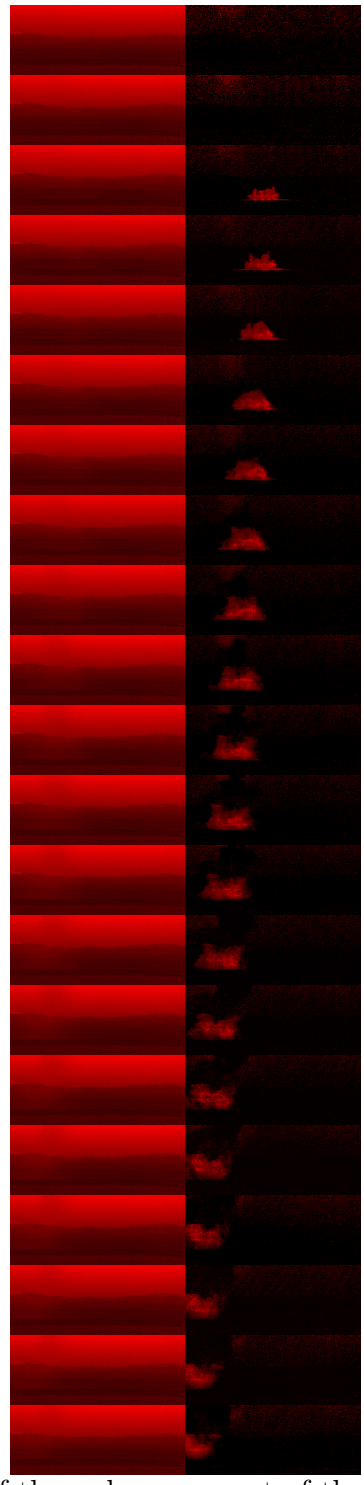

FIGURE 7. Frames 21-40 of the red component of the decomposed video sequence. Notice the motion of the plume captured in the sparse component of the right hand side.

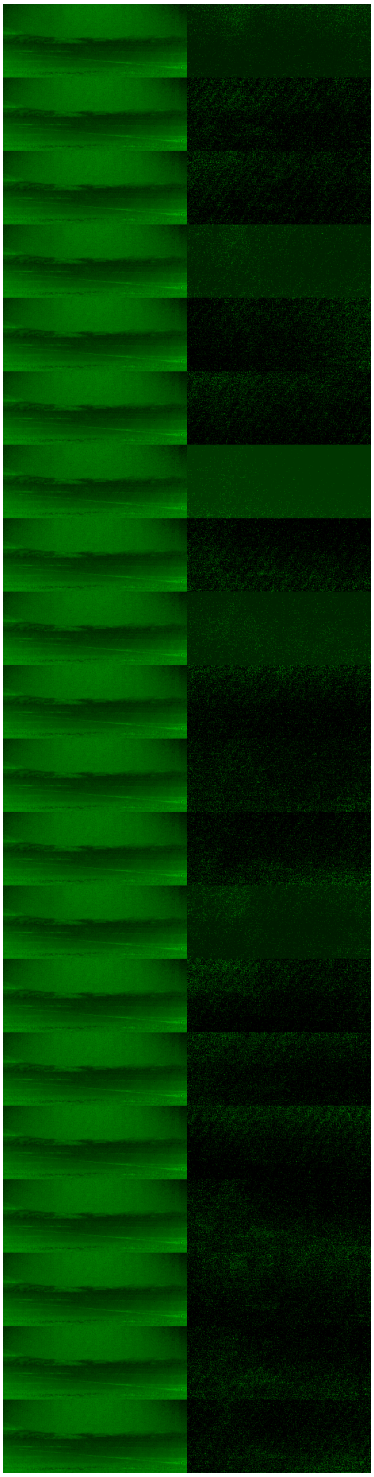

FIGURE 8. The first half of the green component of the resulting decomposed video sequence.

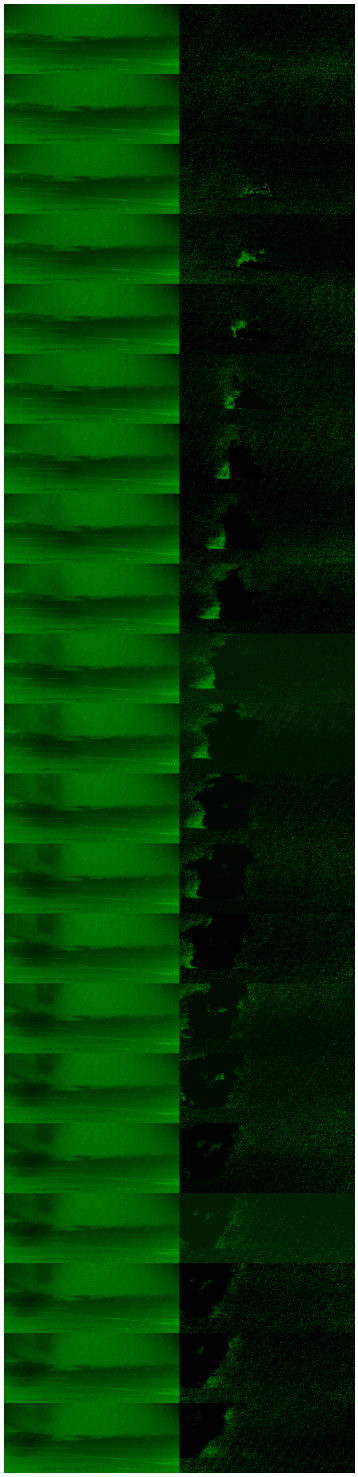

FIGURE 9. Frames 21-40 of the green component of the decomposed video sequence.

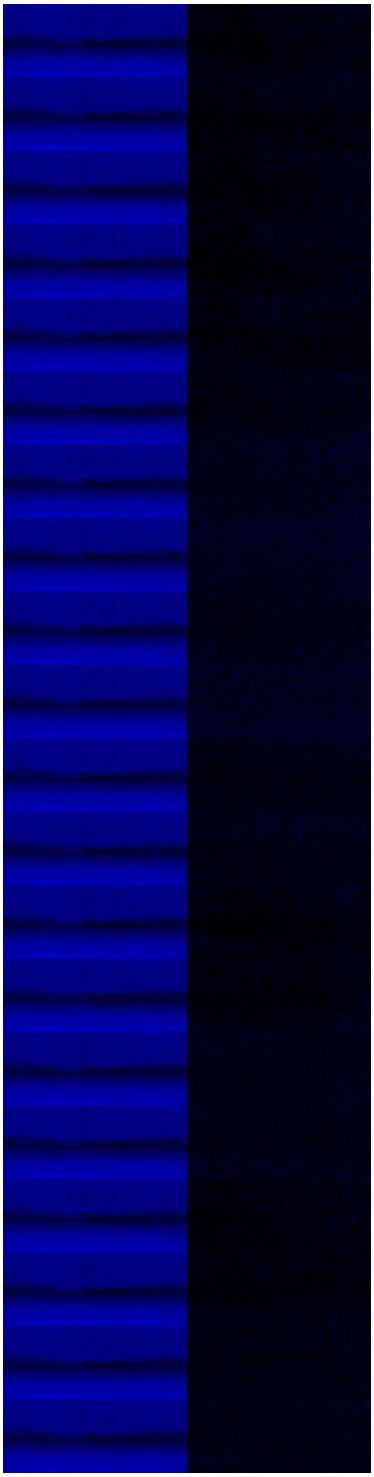

FIGURE 10. Frames 21-40 of the blue component of the decomposed video sequence.

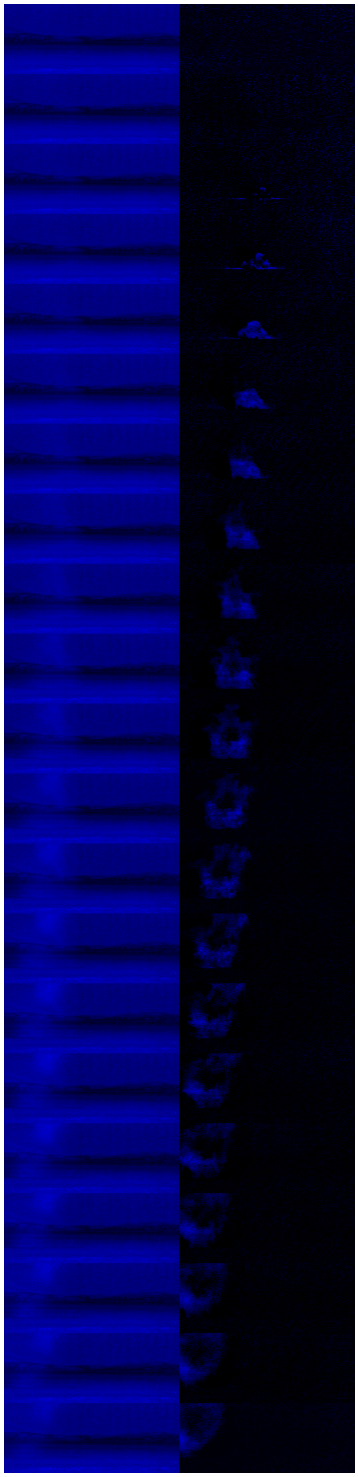

FIGURE 11. Frames 21-40 of the green component of the decomposed video sequence.

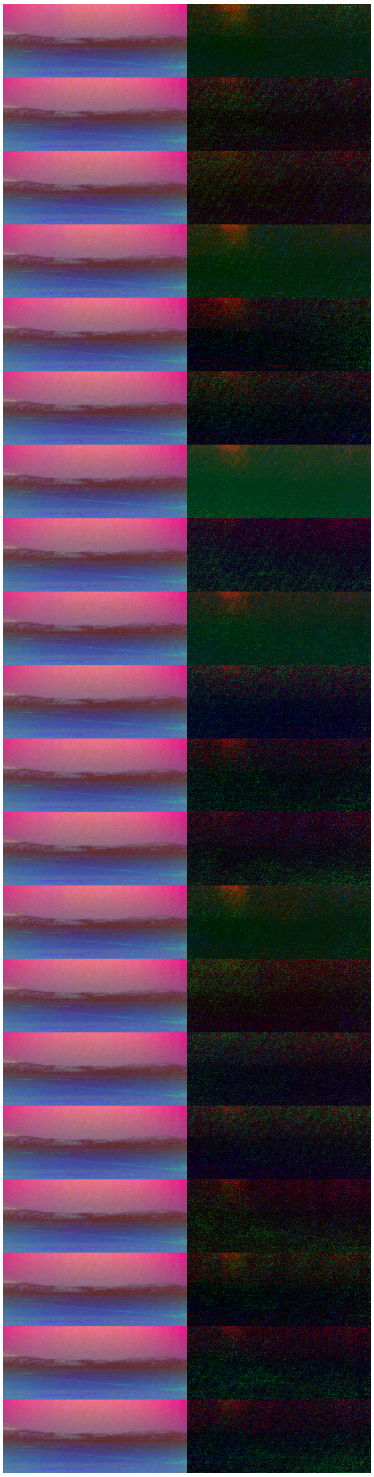

FIGURE 12. Frames 1-20 of the red, green, and blue decompositions combined. Lighting fluctuations and noise may be seen in the sparse component.

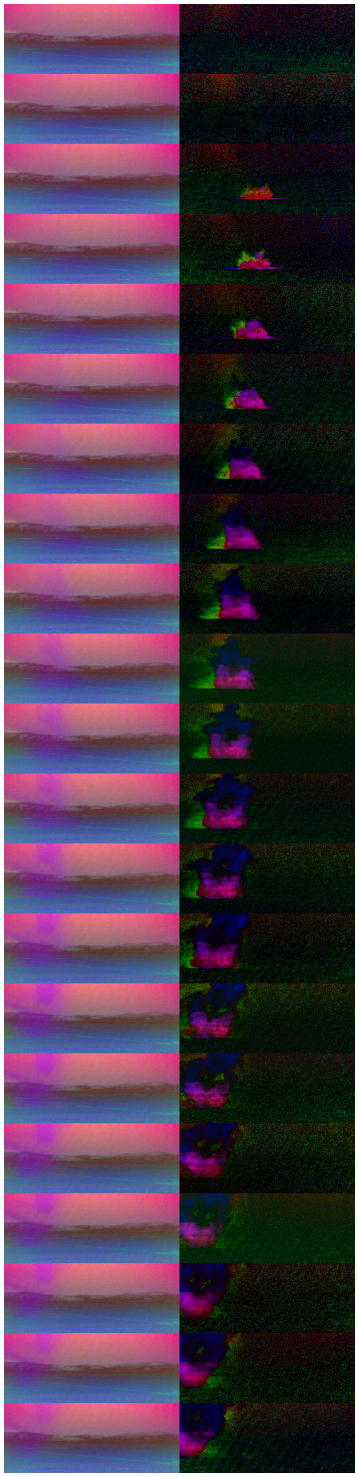

FIGURE 13. Frames 21-40 of the red, green, and blue decompositions combined. Notice the diffuse, but noticeable presence of the plume in the low rank component.

CHAPTER 6

CONCLUSION

Advances in fast, efficient algorithms for solving constrained optimization problems have made contributions to many diverse fields. This work has examined methods for solving certain types of constrained problems from hyperspectral image and video analysis. Both of the solution methods considered here incorporated convexity into the original problem in some way. By doing so the problem became easier to solve and certain desirable properties were included in the solution. This technique of convexification has been used to derive efficient solutions to many problems that were previously thought to difficult.

The results presented here show that it is possible to detect and isolate chemical releases from hyperspectral video sequences. However, a better understanding of the diffusion processes associated with the chemicals in the atmosphere is needed for a practical real-time detection and tracking system for toxic chemical releases. In order to provide the accuracy and reliability required for such a system in the real world, many more physical and computational experiments would be needed.

The applications shown here are from hyperspectral video analysis, but these methods have applications in many different fields. Low-rank sparse matrix decompositions have applications in document classification and many different medical imaging problems. An amazing aspect of image processing as a field, is the ability of an algorithm, or improvement to an existing algorithm to affect a variety of different fields. Even though the problem of detecting chemical releases

seems quite difficult now, advances in computing ability and algorithms may provide solutions sooner than we think.

APPENDICES

APPENDIX A
MATLAB CODES

```
% L1 Unmixing
%
%      This code solves the minimization probem:
%
%   minimize    mu * || u ||_1 + 0.5 * || Au - f ||^2
%   subject to  u ≥ 0
%
% where ||.||   denotes the l2 norm
%       ||.||_1 denotes the l1 norm
%
% Inputs:   Pixels - N x M matrix where columns are pixels,
%                    each column will become the 'f' in the
%                    above fuctional
%           Dictionary - N x K matrix where columns are signals
%           mu - Weights the sparse term
%           lambda - Weights the fitting term
%           epz - Stopping criterion
%
% Outputs: Pics - K x M matrix
%
%
% Torin Gerhart
% 9/12/2012
function [ Pics ] = L1Unmixing(Pixels, Dictionary, mu, lambda, epz)

    if nargin < 5
        epz = 1e-6;
    end

    [ R C ] = size(Pixels);
    [ R N ] = size(Dictionary);

    Pics = zeros(N, C);

    invrs = (lambda * (Dictionary' * Dictionary)) + eye(N);

    ln = lambda * mu * ones(N, 1);

    for i=1:C

        vk = zeros(N, 1);
        bk = zeros(N, 1);

        f  = Pixels(:,i);

        error = epz + 1;

        while ( error > epz )

            vo = vk;
```

```
51
52              dk = max(vk - bk, 0);
53              vk = invrs \ (max(dk,0) + bk + (lambda * ...
                    (Dictionary' * f)) - ln);
54              bk = bk + max(dk,0) - vk;
55
56              vk = max(vk, 0);
57
58              error = (vo - vk)' * (vo - vk);
59          end
60
61          Pics(:,i) = vk;
62
63      end
64
65  end
```

```
1  % Low Rank + Sparse Decomposition (Robust PCA)
2  %
3  %      This code solves the Principal Component
4  % Pursuit problem by means of alternating directions.
5  % This solves the minimization problem:
6  %
7  %   minimize     ||L||_* + lambda ||S||_1
8  %   subject to      M = L + S
9  %
10 % where ||.||_* denotes the nuclear norm (sum of singular ...
       values), and
11 %       ||.||_1 denotes the l_1 norm (of the matrix as a long ...
       vector)
12 %
13 % Inputs:  M   - data matrix with columns as data points
14 %          err - error (optional, default is 10^-7)
15 %
16 % Outputs: L   - low rank matrix
17 %          S   - sparse matrix
18 %
19 % Torin Gerhart
20 % 9/12/2012
21 function [ L S ] = LowRankSparse(M, err)
22
23      [ n1 n2 ] = size(M);
24
25      % Initialize variables
26      S = zeros(n1, n2);
27      L = zeros(n1, n2);
28      Y = zeros(n1, n2);
29
30      % Calculate the parameters
```

```
31      mu      = n1 * n2 / ( 4 * sum(abs(M(:))));
32      lambda = 1 / sqrt(max(n1, n2));
33      error  = 1;
34
35      if nargin == 2
36          stop = err * norm(M, 'fro');
37      else
38          stop = 1e-7 * norm(M, 'fro');
39      end
40
41      while error > stop
42
43          L = SVT(M - S + (1/mu)*Y, 1/mu);
44
45          X = M - L + (1/mu)*Y;
46          S = sign(X) .* max(abs(X) - (lambda / mu), 0);
47
48          Y = Y + mu * (M - L - S);
49
50          error = norm(M - L - S, 'fro');
51      end
52  end
53
54  % Singular Value Thresholding
55  function d_nu = SVT ( temp_nu, threshold )
56
57      if isreal(temp_nu)
58
59          [ u ss v ] = svd(temp_nu, 0);
60
61          id = (ss > 0);
62          ss(id) = sign(ss(id)) .* max( abs(ss(id)) - threshold, 0);
63
64          d_nu = u * ss * v';
65
66      else
67
68          % REAL
69          [ u ss v ] = svd(real(temp_nu), 0);
70
71          id = (ss > 0);
72          ss(id) = sign(ss(id)) .* max( abs(ss(id)) - threshold, 0);
73          tempr = u * ss * v';
74
75          % IMAGINGARY
76          [ u ss v ] = svd(imag(temp_nu), 0);
77
78          id = (ss >0);
79          ss(id) = sign(ss(id)) .* max( abs(ss(id)) - threshold, 0);
80          tempi = u * ss * v';
81
```

```
82            d_nu = complex(tempr, tempi);
83        end
84
85    end
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] J. Ellis. "Searching for oil seeps and oil-impacted soil with hyperspectral imagery." *Earth Observation Magazine*, vol. 10, no. 1, pp. 1058–1064, 2001.

[2] S. Boyd, and L. Vandenberghe. *Convex Optimization*, Cambridge: Cambridge Press, 2004.

[3] J.A. Tropp. "Algorithms for simultaneous sparse approximation." *Signal Processing*, vol. 86,no. 3, pp. 589–602, 2006.

[4] A. Szlam, Z. Guo, and S. Osher. "A split Bregman method for non-negative sparsity penalized least squares with applications to hyperspectral demixing." *ICIP*, vol. 17, pp. 1917–1920, 2010.

[5] T. Goldstein, and S. Osher. "The split Bregman method for L1-regularized problems." *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323-343, 2009.

[6] E.J. Candès, and B. Recht. "Exact matrix completion via convex optimization." *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[7] E.J. Candès, and T. Tao. "The power of convex relaxation: Near-optimal matrix completion." *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[8] E.J. Candès, and T. Tao. "Robust principal component analysis?." *arXiv preprint arXiv:0912.3599*, 2009.

[9] P.J. Martínez, R.M. Pérez, A. Plaza, P.L. Aguilar, M.C. Cantero, and J. Plaza. "Endmember extraction algorithms from hyperspectral images." *Annals Of Geophysics*, vol. 49, no. 1, pp. 93–101, 2006.

[10] E. Esser. *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, CAM Report 9. Los Angeles, CA: University of California Los Angeles, 2009.

[11] 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.

[12] B. Recht, M. Fazel, and P.A. Parrilo "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization.", *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[13] J.B. Broadwater, D. Limsui, and A.K. Carr, *A primer for chemical plume detection using LWIR senators*, Technical Report. Laurel, MD: National Security Technology Department, Johns Hopkins University. 2011.

[14] S. Boyd, L. Xiao, and A. Mutapcic. "Subgradient methods." Lecture notes of EE3920, Stanford University, Autumn Quarter, 2003.

[15] W. Yin, S. Osher,D. Goldfarb, and J. Darbon. "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing." *SIAM Journal on Imaging Sciences* , vol. 1, no. 1, pp. 143–168, 2008.

[16] C. Wu, and X. Tai. "Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models." *SIAM Journal on Imaging Sciences* , vol. 3, no. 3, pp. 300-339, 2010.