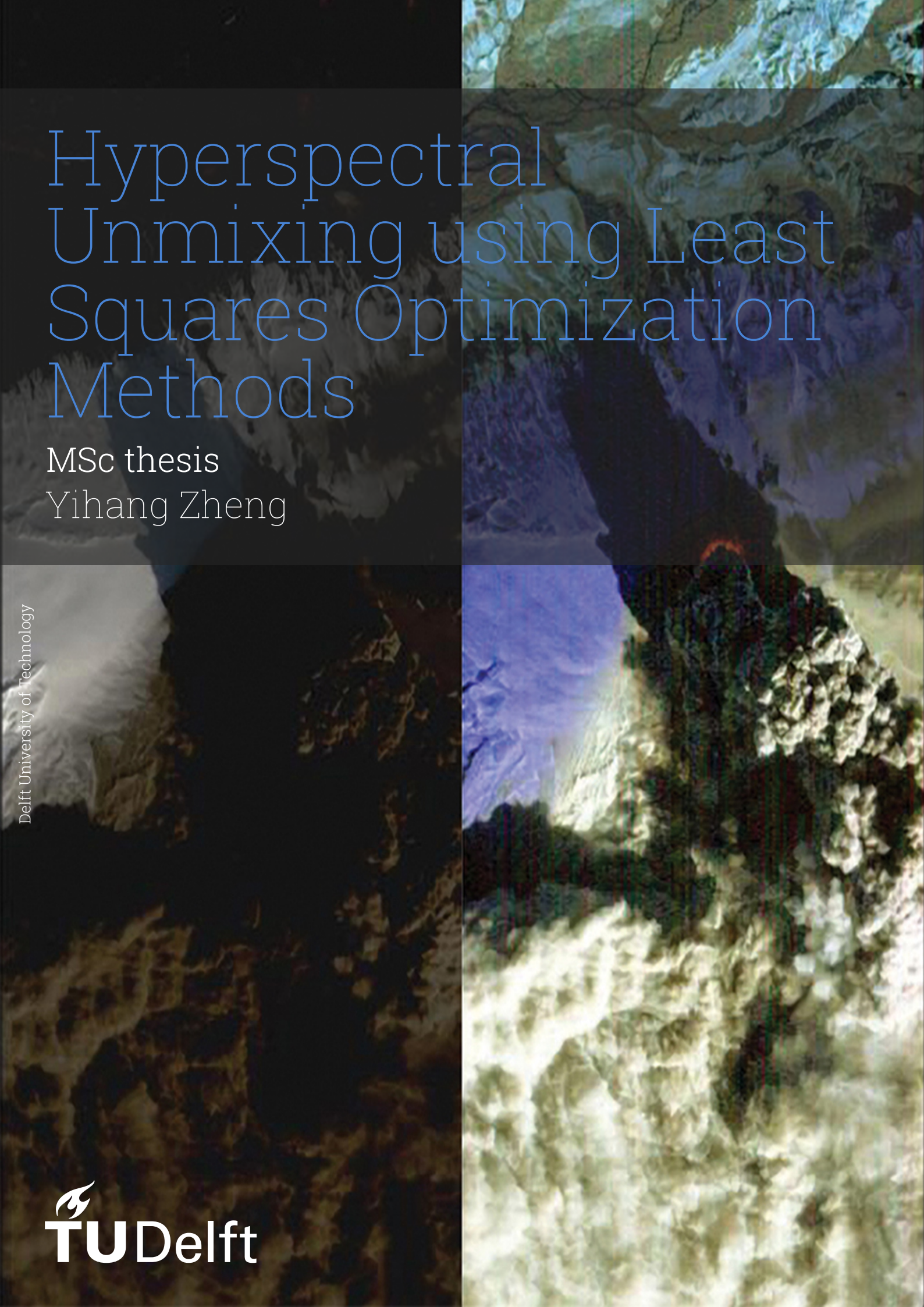


Hyperspectral Unmixing using Least Squares Optimization Methods

MSc thesis

Yihang Zheng

Delft University of Technology

TU Delft

Hyperspectral Unmixing using Least Squares Optimization Methods

by

Yihang Zheng

Thesis Committee:	Dr.A.R (Alireza) Amiri-Simkooei (Daily supervisor) Dr.ir.A.A. (Sandra) Verhagen (Chair) Dr.ir. J. (Joris) Timmermans (Committee)
Student number:	5539099
Project Duration:	Sep, 2023 - June, 2024
Faculty:	Faculty Of Civil Engineering and Geosciences , TUDelft

Cover: Hyperspectral Systems Increase Imaging Capabilities from NASA (modified)

TU Delft

Abstract

Hyperspectral imaging, with its capability of capturing information beyond the visible spectrum, can offer detailed spectral signatures that are critical in various applications, ranging from environmental monitoring to medical diagnostics. However, a significant challenge arises when dealing with hyperspectral data due to the mixed-pixel phenomenon, where a single pixel contains spectral signatures from multiple materials. To solve this problem, hyperspectral unmixing (HU) is used to decompose mixed pixels into their constituent endmembers and their corresponding abundances. This study introduces a novel approach that utilizes least squares optimization methods under various constraints for abundance estimation, specifically using quadratic programming (QP). Additionally, a Principal Component Analysis (PCA) based k-means clustering method is presented for endmember extraction. The research also explores the potential of using Weighted Total Least Squares (WTLS) to refine the estimation process iteratively for the abundance and endmember solutions. The results demonstrate that the type of constraints, whether Weighted Constraints (WC) or Hard Constraints (HC), significantly influences the accuracy of abundance estimation. The QP model, when optimized with appropriate regularization and constraints, showed substantial improvements compared to standard unconstrained least squares methods. The newly proposed PCA method for endmember estimation outperforms traditional methods such as Vertex Component Analysis (VCA). Furthermore, while the WTLS method was sensitive to initial inputs, it showed potential for further enhancing the solutions derived from the QP and PCA methods.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
HU	Hyperspectral Unmixing
NMF	Non-negative Matrix Factorization
QP	Quadratic Programming
WLS	Weighted Least Squares
WTLS	Weighted Total Least squares
EIV	Error-in-variable
WC	Weighted Constraint
HC	Hard Constraint
VCA	Vertex component analysis
SVD	Singular value decomposition
PCA	Principal component analysis
PCs	Principal components
RMSE	Root mean square error
SAD	Spectral angle distance

1

Introduction

Hyperspectral imaging, a branch within remote sensing that is rapidly developing, captures and processes information across the electromagnetic spectrum beyond the visible range. Unlike traditional imaging techniques which typically capture three spectrum bands (red, green, blue), hyperspectral imaging captures hundreds of contiguous bands. This results in a more detailed spectral signature for each pixel, which helps in identifying and differentiating various materials and objects (Bioucas-Dias et al., 2012). Applications to this technology span from precision agriculture to monitor crop health, mineral exploration, medical diagnostics for tissue characterization (G. Lu & Fei, 2014), and environmental monitoring to detect changes in ecosystems (Chang, 2003).

One of the primary challenges in using hyperspectral image data is the mixed-pixel problem. Due to the relatively low spatial resolution of hyperspectral sensors, individual pixels frequently encompass spectral signatures from multiple materials, resulting in homogenized mixtures in the imagery (Keshava & Mustard, 2002). For instance, in precision agriculture, this might result in a single pixel capturing both healthy and diseased plant material which complicates accurate health assessments (Guerri et al., 2024). In mineral exploration, one pixel may blend signatures from different materials, making it difficult to assess the quantity and quality of the resources (Neville et al., 2003). Hyperspectral unmixing (HU), therefore, plays a crucial role in decomposing these mixed pixels into their constituent materials, known as endmembers, and quantifying their proportions, known as abundances.

To tackle the mixed-pixel problem, unmixing methods can generally be categorized into physical-based and data-driven models (Heylen et al., 2014). Physical-based models, such as the linear mixing model and non-linear mixing models, rely on established physical principles to simulate how light interacts with materials. The linear mixing model assumes that the observed spectral signature is a linear combination of endmember spectra weighted by their respective abundances. This simplicity and interoperability make the linear mixing model a widely used approach when it comes to HU.

Specific examples of physical-based methods include geometric-based and statistical-based approaches. Geometric-based methods, such as Vertex Component Analysis (VCA) and N-FINDR, identify endmembers by embedding data points in a lower-dimensional simplex in the feature space and finding the vertices (Gholinejad & Amiri-Simkooei, 2024). These methods are effective only for endmember extractions, thus abundances have to be determined separately.

In contrast, statistical-based methods can estimate both endmembers and abundance simultaneously. A prominent example is Non-negative Matrix Factorization (NMF). NMF decomposes the hyperspectral data matrix into two non-negative matrices representing endmembers and their corresponding abundances. The non-convex nature of the NFM problem allows for the incorporation of various constraints or regularization terms to guide the solutions toward more meaningful local minima. Since then, different variations of the NMF methods have been proposed (Feng et al., 2022).

However, despite the popularity of the linear mixing models, the linear assumption does not always hold, especially in cases involving multi-scattering effects or intimate mixtures of pixels (Heylen et al., 2014). To address these limitations, non-linear mixing models have been developed which account for more complex interactions between light and matter at a cost of increased computational complexity (Dobigeon et al., 2014). Such examples include bilinear models which account for pairwise interaction between endmembers (Heylen et al., 2014), and polynomial post-nonlinear mixing models, which incorporate polynomial terms for higher-order interactions (Altmann et al., 2012).

While physical models are the traditional approach for unmixing, data-driven models leveraging machine learning and deep learning have also gained significant attention (Bhatt & Joshi, 2020). These methods excel in scenarios where underlying physical interactions are complex or poorly understood. However, training these models often requires large labeled datasets and substantial computational resources (Audebert et al., 2019; Paoletti et al., 2019).

This thesis explores advanced methods for extracting both abundances and endmembers in hyperspectral imaging. The primary objective is to develop a fast and computationally efficient approach to address the mixed-pixel problem. Given its simplicity and effectiveness, a linear mixing model-based method will be introduced for abundance estimation, along using least squares optimization. Although there are existing studies that use the least squares method for abundance computation (Deborah et al., 2021; Heinz & Chang, 2001; Menon et al., 2016; Pu et al., 2015), they do not investigate how individual constraints and non-linear constraints affect the estimation process. This study aims to delve deeper by applying least squares under various constraints and systematically investigate how different constraints and regularization affect the abundance estimation process. Additionally, a new data-driven approach will be utilized for endmember extraction, the method combines Principal Component Analysis (PCA) with k-means clustering. Finally, a Weighted Total Least Squares (WTLS) algorithm will be investigated to assess its potential to improve the solutions of the newly proposed models.

1.1. Research questions

The main research question for this research can thus be formulated as follows: **How can least squares methods be used to accurately and efficiently estimate abundance and endmembers for hyperspectral unmixing?**

To be able to answer the main research question, the following sub-questions will be addressed:

1. How does the solution differ when considering different constraints especially when dealing with non-linear constraints for abundance estimation?
2. How do constrained least squares affect the estimation of abundances when compared to the unconstrained approach?
3. How does regularization affect abundance and endmember estimation?
4. How does the endmember matrix influence the abundance matrix estimation? Can we develop a fast and accurate method for endmember estimation?
5. How can the abundance and endmember solutions be further enhanced?

1.2. Thesis outline

Chapter 2 of this thesis presents the fundamental background theory of the least squares methods that will be employed in this study. In Chapter 3, the detailed methodology for developing the models will be explained. Chapter 4 discusses the results and evaluates the solution quality of the new models using various real hyperspectral datasets. Finally, Chapter 5 concludes the study by addressing the research questions and providing recommendations for future research.

2

Background Theory

This chapter presents the fundamental principles and mathematical framework for this research, which serves as the bedrock of our analysis.

2.1. Least squares solutions

2.1.1. Objective function formulation

In many scientific and engineering applications, the primary challenge is to interpret observed data in a way that can accurately reflect the underlying physical reality. This often involves solving an optimization problem, where the goal is to minimize the discrepancy between the observed data and its estimation based on a predefined linear model. One of the most commonly used methods to address this optimization challenge is the least squares method.

The least squares approach is particularly effective in linear scenarios. This method aims to minimize the squares differences (residuals) between observed data and the estimated data by the linear model. Mathematically, the objective is formulated as follows:

$$\begin{aligned} &\text{minimize } \phi(x) \text{ with} \\ &\qquad \phi(x) = \frac{1}{2}\|y - Ax\|^2_{Q_y^{-1}} = \frac{1}{2}\|e\|^2_{Q_y^{-1}} \\ &\qquad\quad = \frac{1}{2}(y - Ax)^T Q_y^{-1}(y - Ax) \\ &\qquad\quad = \frac{1}{2}(y^T Q_y^{-1} y - y^T Q_y^{-1} Ax - x^T A^T Q_y^{-1} y + x^T A^T Q_y^{-1} Ax) \end{aligned} \tag{2.1}$$

where $y \in \mathbb{R}^m$ is the vector of observations, $x \in \mathbb{R}^n$ is the vector of unknown parameters, $A \in \mathbb{R}^{m \times n}$ the design matrix and $Q_y \in \mathbb{R}^{m \times m}$ the covariance matrix of observations. The constant term $\frac{1}{2}$ often appears in optimization problems for mathematical conveniences when computing the critical points. Note that since $y^T Q_y^{-1} Ax$ and $x^T A^T Q_y^{-1} y$ are both scalars and transposes of each other, it follows that:

$$\phi(x) = \frac{1}{2} y^T Q_y^{-1} y - y^T Q_y^{-1} Ax + \frac{1}{2} x^T A^T Q_y^{-1} Ax$$

Omitting the term $y^T Q_y^{-1} y$ that is not a function of x, for simplicity, the objective function is then:

$$\phi(x) = \frac{1}{2} x^T N x - u^T x \tag{2.2}$$

where $N = A^T Q_y^{-1} A$ and $u = A^T Q_y^{-1} y$. To be able to obtain solutions for the least squares method, consider the following two cases that are presented in the next subsections.

2.1.2. Unconstrained least squares solution

When dealing with least squares methods, there are generally 2 cases to be considered: Constrained and Unconstrained solutions.

The unconstrained least squares approach is straightforward and fundamental in its own premise. It does not impose any additional conditions or restrictions on the parameters. In this case, the solution seeks to find the best fit for the observed data within the frame of the linear model. This unconstrained scenario is essential for understanding the core principles of the least squares method and serves as a foundation for more complex, constrained models.

The unconstrained least squares solution is rather trivial. To find the minimizer $\hat{x}$, the first and second derivatives of $\phi(x)$ must be computed.

Let N and u be defined as previously, the first derivative of ϕ with respect to x is given by:

$$\begin{aligned}\frac{\partial \phi}{\partial x} &= \partial_x(\frac{1}{2}y^T Q_y^{-1} y - y^T Q_y^{-1} A x + \frac{1}{2}x^T A^T Q_y^{-1} A x) \\ &= -A^T Q_y^{-1} y + A^T Q_y^{-1} A x \\ &= Nx - u\end{aligned} \tag{2.3}$$

The second derivative of $\phi(x)$ is given by:

$$\begin{aligned}\frac{\partial^2 \phi}{\partial x^2} &= \partial_x(-A^T Q_y^{-1} y + A^T Q_y^{-1} A x) \\ &= A^T Q_y^{-1} A \\ &= N\end{aligned} \tag{2.4}$$

The second derivative is positive definite if A has full column rank and Q_y^{-1} is positive definite. This ensures that the solution $\hat{x}$ has a global minimum for $\phi(x)$. Thus by setting the first derivative to 0, solution $\hat{x}$ will be obtained as:

$$\begin{aligned}-A^T Q_y^{-1} y + A^T Q_y^{-1} A x &= 0 \\ A^T Q_y^{-1} A x &= A^T Q_y^{-1} y \\ \hat{x} &= (A^T Q_y^{-1} A)^{-1} A^T Q_y^{-1} y \\ &= N^{-1} u\end{aligned} \tag{2.5}$$

2.1.3. Constrained least squares solution

The unconstrained least squares aims to minimize the objective function without any restriction on the parameter x. However, while this is straightforward, in many practical scenarios, parameters must satisfy certain physical constraints. To incorporate these physical constraints, the constrained least squares can be considered.

The constrained least squares solution differs from the unconstrained least squares solution. As shown in (Amiri-Simkooei, 2019), let the weighted or hard linear(ized) constraints be defined as:

$$c = C^T x + e_c, \quad D(c) = Q_c \tag{2.6}$$

where C^T is the given matrix with constraints, c a random vector with variance matrix Q_c, and D the dispersion operator which measures the spread or variability of the random vector c that is equal to Q_c. The solution with constraints can be obtained by expanding the objective function along the constraint matrix 2.6 with the observation equations. To be more specific, consider the following systems of linear equations:

$$\begin{aligned}y &= Ax + e \\ c &= C^T x + e_c\end{aligned} \tag{2.7}$$

The objective function thus becomes:

$$\begin{aligned}
&\text{minimize } \phi(x) \text{ with} \\
&\phi(x) = \frac{1}{2}\|y - Ax\|^2_{Q_y^{-1}} + \frac{1}{2}\|c - C^T x\|^2_{Q_c^{-1}} \\
&= \frac{1}{2}\|e\|^2_{Q_y^{-1}} + \frac{1}{2}\|e_c\|^2_{Q_c^{-1}} \\
&= \frac{1}{2}(y_c - A_c x)^T Q_{y_c}^{-1}(y_c - A_c x) \\
&= \frac{1}{2}y_c^T Q_{y_c}^{-1} y_c - y_c^T Q_{y_c}^{-1} A_c x + \frac{1}{2}x^T A_c^T Q_{y_c}^{-1} A_c x
\end{aligned} \tag{2.8}$$

where $y_c = \begin{bmatrix} y \\ c \end{bmatrix}$, $A_c = \begin{bmatrix} A \\ C^T \end{bmatrix}$ and $Q_{y_c} = \begin{bmatrix} Q_y & 0 \\ 0 & Q_c \end{bmatrix}$. Moreover, as $\frac{1}{2}y_c^T Q_{y_c}^{-1} y_c$ is not a function of x, one has:

$$\phi(x) = \frac{1}{2}x^T A_c^T Q_{y_c}^{-1} A_c x - y_c^T Q_{y_c}^{-1} A_c x \tag{2.9}$$

note that equation (2.9) has the same general form as presented in equation 2.2 if $N_c = A_c^T Q_{y_c}^{-1} A_c$ and $u_c = A_c^T Q_{y_c}^{-1} Y$.

By computing N_c explicitly:

$$N_c = \begin{bmatrix} A^T & C \end{bmatrix} \begin{bmatrix} Q_y^{-1} & 0 \\ 0 & Q_c^{-1} \end{bmatrix} \begin{bmatrix} A \\ C^T \end{bmatrix} = A^T Q_y^{-1} A + C Q_c^{-1} C^T \tag{2.10}$$

similarly, for u_c:

$$u_c = \begin{bmatrix} A^T & C \end{bmatrix} \begin{bmatrix} Q_y^{-1} & 0 \\ 0 & Q_c^{-1} \end{bmatrix} \begin{bmatrix} y \\ c \end{bmatrix} = A^T Q_y^{-1} y + C Q_c^{-1} c \tag{2.11}$$

Following the same argument to the unconstraint least squares, combining equation 2.10 and 2.11, the solution of least squares with constraint thus becomes:

$$\begin{aligned}
\hat{x} &= N_c^{-1} u_c \\
&= (A^T Q_y^{-1} A + C Q_c^{-1} C^T)^{-1}(A^T Q_y^{-1} y + C Q_c^{-1} c) \\
&= (N + C Q_c^{-1} C^T)^{-1}(u + C Q_c^{-1} c)
\end{aligned} \tag{2.12}$$

where $N = A^T Q_y^{-1} A$ and $u = A^T Q_y^{-1} y$.

The above solution (2.12) is a general form when considering weighted or hard linear(ized) constraints. In some cases, more advanced constraints may be introduced, including non-linear constraints. These non-linear constraints can significantly increase computational complexity, sometimes making it difficult for standard solvers to find solutions.

2.2. Weighted total least squares solutions

As stated previously, the objective of normal least squares is to minimize the sum of squared residuals between observed dependent variables and the estimated solution. This approach assumes that errors are only present in the vector of observations, while the independent variables are considered to be exact. However, there might be cases where the errors can also be present in the design matrix. Golub and Van Loan (Golub & Van Loan, 1980) introduced total least squares with an error-in-variable (EIV) model that considers errors both in the observation and design matrix. In the following, a comprehensive overview of the solution methodology for WTLS will be provided.

2.2.1. Univariate WTLS solution

Consider the standard univariate EIV model:

$$y = (A - E_A)x + e_y \tag{2.13}$$

with stochastic properties:

$$e = \begin{bmatrix} e_y \\ e_A \end{bmatrix} = \begin{bmatrix} e_y \\ \mathrm{vec}(E_A) \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \sigma_0^2 \begin{bmatrix} Q_y & 0 \\ 0 & Q_A \end{bmatrix} \right) \tag{2.14}$$

where $E_A \in \mathbb{R}^{m \times n}$ and $e_y \in \mathbb{R}^m$ correspond to random errors of the design matrix A and the observation vector y, respectively. $Q_A \in \mathbb{R}^{mn \times mn}$ is the covariance matrices of the design matrix and σ_0^2 is the (un)known variance factor of the unit weight (Gholinejad & Amiri-Simkooei, 2023).

Thus the minimization problem can be formulated as:

$$\begin{aligned} &\text{Objective: minimize} \mathcal{J}_u = e_y^T Q_y^{-1} e_y + e_A^T Q_A^{-1} e_A \\ &\text{subject to: } y - e_y = (A - E_A)\, x \end{aligned} \tag{2.15}$$

Let $\mathcal{L}_u$ be the target Lagrange function (Schaffrin and Wieser, 2007; Shen et al., 2010) :

$$\mathcal{L}_u(e_y, e_A, \lambda, x) = e_y^T Q_y^{-1} e_y + e_A^T Q_A^{-1} e_A + 2\lambda^T \left(y - Ax - e_y + \left(x^T \otimes I_m \right) e_A \right) \tag{2.16}$$

where I_m is an identity matrix of size m and $\lambda \in \mathbb{R}^m$ is the vector of unknown Lagrange multipliers. The stationary points have to be computed, this is done by computing the partial derivatives of the target Lagrange function with respect to the unknown parameters e_y, e_A, λ and x are:

$$\frac{1}{2} \frac{\partial \mathcal{L}_u}{\partial e_y^T} = Q_y^{-1} \tilde{e}_y - \hat{\lambda} = 0 \Longrightarrow \tilde{e}_y = Q_y \hat{\lambda} = \mathrm{vec}(\tilde{E}_y) \tag{2.17}$$

$$\frac{1}{2} \frac{\partial \mathcal{L}_u}{\partial e_A^T} = Q_A^{-1} \tilde{e}_A + (\hat{x} \otimes I_m) \hat{\lambda} = 0 \Longrightarrow \tilde{e}_A = -Q_A (\hat{x} \otimes I_m) \hat{\lambda} = \mathrm{vec}(\tilde{E}_A) \tag{2.18}$$

$$\frac{1}{2} \frac{\partial \mathcal{L}_u}{\partial \lambda} = y - A\hat{x} - \tilde{e}_y + (\hat{x}^T \otimes I_m) \tilde{e}_A = 0 \tag{2.19}$$

$$\frac{1}{2} \frac{\partial \mathcal{L}_u}{\partial x^T} = -A^T \hat{\lambda} + \tilde{E}_A^T \hat{\lambda} = 0 \tag{2.20}$$

where tildas refer to the 'predicted' vectors, and hats refer to 'estimated' vectors. The estimated values are generally obtained by minimizing the objective function, whereas the predicted values are obtained after estimating the model parameters. After some simple mathematical substitution, one obtains

$$\hat{\lambda} = Q_{\tilde{y}}^{-1} (y - A\hat{x}) \tag{2.21}$$

with

$$Q_{\tilde{y}} = Q_y + (\hat{x}^T \otimes I_m) Q_A (\hat{x} \otimes I_m) \tag{2.22}$$

where $Q_{\tilde{y}}$ the covariance matrix of predicted observations for $\tilde{y} = y - \tilde{E}_A \hat{x}$. Thus the minimizer $\hat{x}$ can be formulated as:

$$\hat{x} = ((A - \tilde{E}_A)^T Q_{\tilde{y}}^{-1} A)^{-1} (A - \tilde{E}_A)^T Q_{\tilde{y}}^{-1} y \tag{2.23}$$

However the normal matrix in equation 2.23 is not symmetric and positive-definite. By letting $A = \tilde{A} + \tilde{E}_A$, the new $\hat{x}$ can be reformulated similarly to the standard least squares solution form (Amiri-Simkooei and Jazaeri, 2012; Gholinejad and Amiri-Simkooei, 2023):

$$\hat{x} = (\tilde{A}^T Q_{\tilde{y}}^{-1} \tilde{A})^{-1} \tilde{A}^T Q_{\tilde{y}}^{-1} \tilde{y} \tag{2.24}$$

2.2.2. Multivariate WTLS solution

The multivariate WTLS solution can be computed similarly to the univariate case, as it considers multiple observation vectors at once (Gholinejad & Amiri-Simkooei, 2023). The multivariate EIV model can be formulated as follows:

$$Y - E_Y = (A - E_A)X \tag{2.25}$$

where $Y \in \mathbb{R}^{m\times g}$ is the vectors of observations with g groups of data, $X \in \mathbb{R}^{n\times g}$ is the vectors of unknowns and $E_Y \in \mathbb{R}^{m\times g}$ the random errors. The new objective function thus can be restated as:

$$\begin{aligned} &\text{Objective: minimize} \mathcal{J}_w = e_y^T Q_Y^{-1} e_y + e_A^T Q_A^{-1} e_A \\ &\text{subject to: } y - E_Y = (A - E_A)\, x \end{aligned} \tag{2.26}$$

where $e_y = e_{\text{vec}(Y)} \in \mathbb{R}^{m\times g}$ and $e_a = \text{vec}(E_A) = e_{\text{vec}(A)} \in \mathbb{R}^{m\times n}$. By assuming $a = \text{vec}(A)$, $y = \text{vec}(Y)$, $x = \text{vec}(X)$ and , $\lambda = \text{vec}(\Lambda)$ with identities $\text{vec}(abc) = (c^T \otimes a)\text{vec}(b)$ and $\text{tr}(ab^T) = \text{vec}(a)^T\text{vec}(b)$. The target Lagrange function $\mathcal{L}_m$ for the multivariate case is:

$$\begin{aligned} \mathcal{L}_m(e_y, e_a, \lambda, X) &= e_y^T Q_Y^{-1} e_y + e_a^T Q_A^{-1} e_a + 2\,\text{tr}(\Lambda^T (Y - AX - E_Y + E_A X)) \\ &= e_y^T Q_Y^{-1} e_y + e_a^T Q_A^{-1} e_a + 2\lambda^T (y - \text{vec}(AX) - e_y + (X^T \otimes I_m) e_a) \end{aligned} \tag{2.27}$$

Analogous to the univariate case, the partial derivatives of the Lagrange function with respect to the unknown parameters e_y, e_a, λ and X are:

$$\frac{1}{2}\frac{\partial \mathcal{L}_m}{\partial e_y^T} = Q_Y^{-1}\tilde{e}_y - \hat{\lambda} = 0 \Longrightarrow \tilde{e}_y = Q_Y\hat{\lambda} = \text{vec}(\tilde{E}_Y) \tag{2.28}$$

$$\frac{1}{2}\frac{\partial \mathcal{L}_m}{\partial e_a^T} = Q_A^{-1}\tilde{e}_a + (\hat{X} \otimes I_m)\hat{\lambda} = 0 \Longrightarrow \tilde{e}_a = -Q_A(\hat{X} \otimes I_m)\hat{\lambda} = \text{vec}(\tilde{E}_A) \tag{2.29}$$

$$\frac{1}{2}\frac{\partial \mathcal{L}_m}{\partial \lambda} = y - \text{vec}(A\hat{X}) - \tilde{e}_y + (\hat{X}^T \otimes I_m)\tilde{e}_a = 0 \tag{2.30}$$

$$\frac{1}{2}\frac{\partial \mathcal{L}_m}{\partial X^T} = -A^T\hat{\Lambda} + \tilde{E}_A^T\hat{\Lambda} = 0 \tag{2.31}$$

By doing substitution and letting $\hat{E} = Y - A\hat{X}$ being the estimated total residual matrix, one obtains:

$$\hat{\lambda} = Q_{\tilde{Y}}^{-1}\text{vec}(\hat{E}) = Q_{\tilde{Y}}^{-1}\hat{e} \tag{2.32}$$

where $\hat{e} = \text{vec}(\hat{E})$ with

$$Q_{\tilde{Y}} = Q_Y + (\hat{X}^T \otimes I_m)Q_A(\hat{X} \otimes I_m) \tag{2.33}$$

The above formulation can be further simplified by assuming $Q_A = \Sigma_A \otimes Q$ and $Q_Y = \Sigma_Y \otimes Q$ where $Q \in \mathbb{R}^{m\times m}$ and $\Sigma_A \in \mathbb{R}^{n\times n}$, $\Sigma_Y \in \mathbb{R}^{g\times g}$ being positive definite matrices that are elements of the covariance matrices which explains the cross-covariances between A and Y. Therefore:

$$\begin{aligned} Q_{\tilde{Y}} &= Q_Y + (\hat{X}^T \otimes I_m)Q_A(\hat{X} \otimes I_m) \\ &= \Sigma_Y \otimes Q + (\hat{X}^T \otimes I_m)(\Sigma_A \otimes Q)(\hat{X} \otimes I_m) \\ &= \Sigma_Y \otimes Q + \hat{X}^T\Sigma_A\hat{X} \otimes Q \\ &= \hat{\Sigma} \otimes Q \end{aligned} \tag{2.34}$$

where $\hat{\Sigma} = \hat{\Sigma}_Y + \hat{X}^T\Sigma_A\hat{X}$. This also yields that:

$$\begin{aligned} \hat{\lambda} &= (\hat{\Sigma}^{-1} \otimes Q^{-1})\text{vec}(\hat{E}) \\ &= \text{vec}(Q^{-1}\hat{E}\hat{\Sigma}^{-1}) \\ &\Longrightarrow \hat{\Lambda} = \text{inverse vec}(\hat{\lambda}) = Q^{-1}\hat{E}\hat{\Sigma}^{-1} \end{aligned} \tag{2.35}$$

Substituting everything back to equation 2.31 and to make a symmetric positive-definite normal matrix, let $A = \tilde{A} + \tilde{E}_A$, the minimizer $\hat{X}$ is given as:

$$\begin{aligned} \hat{X} &= ((A - \tilde{E}_A)^T Q^{-1} A)^{-1} (A - \tilde{E}_A)^T Q^{-1} Y \\ &\implies \hat{X} = (\tilde{A}^T Q^{-1} \tilde{A})^{-1} \tilde{A}^T Q^{-1} \tilde{Y} \end{aligned} \tag{2.36}$$

Lastly, $\tilde{e}_y$ and $\tilde{e}_a$ can be simplified to:

$$\begin{aligned} \tilde{e}_y &= \text{vec}(\tilde{E}_Y) = Q_Y \hat{\lambda} \\ &= (\Sigma_Y \otimes Q)(\hat{\Sigma}^{-1} \otimes Q^{-1}) \text{vec}(\hat{E}) \\ &= \text{vec}(\hat{E}\hat{\Sigma}^{-1}\Sigma_Y) \\ &\implies \hat{E}_Y = \text{inverse vec}(\tilde{e}_y) = \hat{E}\hat{\Sigma}^{-1}\Sigma_Y \end{aligned} \tag{2.37}$$

$$\begin{aligned} \tilde{e}_a &= \text{vec}(\tilde{E}_A) = -Q_A(\hat{X} \otimes I_m)\hat{\lambda} \\ &= -(\Sigma_A \otimes Q)(\hat{X} \otimes I_m)\text{vec}(Q^{-1}\hat{E}\hat{\Sigma}^{-1}) \\ &= -(\Sigma_A \hat{X} \otimes Q)\text{vec}(Q^{-1}\hat{E}\hat{\Sigma}^{-1}) \\ &= \text{vec}(\hat{E}\hat{\Sigma}^{-1}\hat{X}^T\Sigma_A) \\ &\implies \tilde{E}_A = \text{inverse vec}(\tilde{e}_a) = -\hat{E}\hat{\Sigma}^{-1}\hat{X}^T\Sigma_A \end{aligned} \tag{2.38}$$

3

Methodology

This chapter outlines the methodology employed in this research. Section 3.1 explains the methods used for abundance computation, an important step for accurately interpreting hyperspectral data. Section 3.2 details the endmember extraction techniques, which are essential for material identification and have a direct influence on the accuracy of abundance estimation. Finally, Section 3.3 discusses the motivation and implementation of the WTLS methods, for potential enhancement of the solutions derived from earlier stages.

3.1. Abundance computation

Historically, linear hyperspectral unmixing has been the standard method for resolving the mixed-pixel challenge (Heinz & Chang, 2001). This approach is encapsulated in the following system of observation equations:

$$y = Ax + e, \quad D(y) = Q_y \tag{3.1}$$

where y is the observed spectrum of the pixels; A is the matrix that contains the spectral signatures of the endmembers. Each column of matrix A represents the spectrum of one endmember, describing how each endmember absorbs or reflects light across various wavelengths; x is the abundance matrix that indicates the proportion of each endmember within a pixel; e is the error or noise and D is the dispersion operator with Q_y being the variance matrix of y. The above model (3.1) is the foundation for various unmixing techniques. The aim is to accurately estimate both the abundance matrix x and the endmember matrix A given the observed data y. It reflects the inherent nature of hyperspectral data, where each pixel's spectrum is a composite of material contributions with given distribution.

Given this model, the hyperspectral unmixing (HU) problem can be formulated as a least squares problem. To be able to accurately estimate the abundance solution, selecting a computationally efficient and fast solver is crucial. Initially, two solvers were considered: a nonlinear programming solver and a quadratic programming (QP) solver. The nonlinear programming solver has the advantage of handling a wide variety of objective functions and constraints, including nonlinear and non-convex ones. However, this flexibility comes at the cost of efficiency and speed, resulting in high computational complexity.

In contrast, general QP problems are considered to be NP-hard (Ye & Tse, 1989), which indicates that they fall into a category where no efficient algorithms are known to solve these problems within polynomial time. However, this does not apply if the QP problem is convex. For convex QP problems, polynomial time interior-point method algorithm can be used to provide unique solutions. The objective function of the least squares problem formulated is convex, making the QP solver a better choice. It offers robustness, computational efficiency, and predictability compared to the general nonlinear solver. Therefore, the QP solver was selected for this research due to its performance in solving convex least squares problems with lower computational cost and more reliable behavior.

To make the abundance estimation more meaningful, nonnegativity, sum-to-one, and sparsity constraints

were considered for this optimization procedure:

- Nonnegativity Constraint: Ensures that the estimated abundances are non-negative, reflecting the physical reality that negative abundances do not make sense in the context of material composition.
- Sum-to-One Constraint: Imposes that the sum of the abundance for each pixel equals to one. Representing a fractional abundance where the total proportion of materials in each pixel must account for the entire pixel.
- Sparsity Constraint: Encourages solutions where most abundances are zero. Promoting a sparse representation that aligns with the assumption that each pixel is composed of a few endmembers rather than a mix of many.

Several studies have been conducted to design new algorithms for solving HU problems. However, there are almost no studies or little that explicitly investigate how these constraints directly influence the results. Most studies consider either the sum-to-one constraint or the sparsity constraint, but rarely both together. Often, the sparsity constraint is treated as a regularization term in the objective function rather than as an individual constraint (Qian et al., 2011). In this research, instead of incorporating the sparsity constraint as a regularization term in the objective function, L_q-norm will be directly applied as a regularizer to the sparsity constraint to enforce sparsness. To thoroughly investigate how these constraints affect abundance estimation, the model for the constrained least squares problem will be formulated as follows:

$$\begin{aligned} \text{Objective}: \quad & \text{minimize } \phi = \frac{1}{2} x^T N x - u^T x \\ \text{Subject to}: \quad & \sum_{i=1}^{n} x_i = 1 \\ & \sum_{i=1}^{n} x_i^2 = \|x\|^2 = 1 \\ & x \geq 0 \end{aligned} \tag{3.2}$$

when one also takes the Lq-norm into consideration:

$$\begin{aligned} \text{Objective}: \quad & \text{minimize } \phi = \frac{1}{2} x^T N x - u^T x \\ \text{Subject to}: \quad & \sum_{i=1}^{n} x_i = 1 \\ & \sum_{i=1}^{n} x_i^q = \|x\|^q = 1 \\ & x \geq 0 \end{aligned} \tag{3.3}$$

or alternatively, the most general form:

$$\begin{aligned} \text{Objective}: \quad & \text{minimize } \phi = \frac{1}{2} x^T N x - u^T x \\ \text{Subject to}: \quad & \sum_{i=1}^{n} x_i = 1 \\ & \sum_{i=1}^{n} x_i^q = \|x\|^q \leq \mu \\ & x \geq 0 \end{aligned} \tag{3.4}$$

where $0 < q \leq 1$ and $\mu \in \mathbb{R}_0^+$.

One might already notice that the sparsity constraints being considered are non-linear. In general, non-linear constraints in QP can potentially lead to a non-convex problem. For instance, if one considers more complex constraints such as the manifold constraint (Feng et al., 2022; X. Lu et al., 2013,

2020), which exploits the underlying geometrical structure of the data under the assumption that the hyperspectral data lies on a low-dimensional manifold, the feasible region of the optimization problem might lie outside of the convex region. To ensure that our problem remains convex, it is imperative to linearize the constraints.

3.1.1. Constraints implementation

The linearization of the constraint is achieved by using the first-order Taylor series expansion of a function $f(x)$ around an initial point x_0. The linearized expression is given by

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \tag{3.5}$$

For the case of sparsity constraint $\sum_{i=1}^{n} x_i^2 \leq \mu$, the function $f(x_i) = x_i^2$ is considered. The derivative of $f(x_i)$ with respect to x_i is $f'(x_i) = 2x_i$. At the point $x_{0,i}$, this derivative becomes $f'(x_{0,i}) = 2x_{0,i}$. Expanding $f(x_i)$ around $x_{0,i}$ yields:

$$x_i^2 \approx x_{0,i}^2 + 2x_{0,i}(x_i - x_{0,i}) \tag{3.6}$$

This simplifies to:

$$x_i^2 \approx 2x_{0,i}x_i - x_{0,i}^2 \tag{3.7}$$

Consequently, the constraint becomes:

$$\sum_{i=1}^{n}(2x_{0,i}x_i - x_{0,i}^2) \leq \mu \tag{3.8}$$

The linearization process for the general sparsity constraint is analogous to the sparsity constraint, let $f(x_i) = x_i^q$, taking the derivative at $x_{0,i}$, one obtains: $f'(x_{0,i}) = qx_{0,i}^{q-1}$. Expanding around $x_{0,i}$:

$$x_i^q \approx x_{0,i}^q + qx_{0,i}^{q-1}(x_i - x_{0,i}) \tag{3.9}$$

Which leads to the linearized form:

$$\sum_{i=1}^{n} x_i^q \approx \sum_{i=1}^{n} qx_{0,i}^{q-1}x_i + (1-q)\sum_{i=1}^{n} x_{0,i}^q \leq \mu \tag{3.10}$$

Moreover, for the constrained least squares case, the constraints can be enforced in a way either as weighted constraints (WC) or hard constraints (HC) (Amiri-Simkooei, 2017). The difference between the two lies in the strictness with which the constraints are applied to the QP model. This is controlled by Q_c, the variance matrix for the constraints. A larger value of Q_c makes the constraints more weighted, while a smaller value makes them harder. In this research, a scaled diagonal variance matrix for Q_c will be considered. This approach is chosen not only for its simplification and computational efficiency but also based on the assumption that the different bands are uncorrelated or independent. While this may not always be true in practice, it is a reasonable approximation when the bands are sufficiently separated in the electromagnetic spectrum. Additionally, the noise in different bands will be assumed to be independent, which naturally leads to a scaled diagonal variance matrix containing noises.

Regularization of the sparsity constraint also plays a crucial role in hyperspectral unmixing, as it promotes sparsity and stabilizes the solution. Investigating the effect of regularization can help improve the robustness and accuracy of the solution. Different regularizers will be investigated to see the effect of abundance estimation, e.g: $0 < L \leq 2$, $0 < a \leq 1$. It is worth stating that when the sum-to-one constraint is being considered, the L_1 (q=1) does not significantly enhance sparsity, as L_1 and the sum-to-one constraint are related to each other (Qian et al., 2011). The sum-to-one constraint inherently imposes a form of restriction on the total abundance, which can lead to a natural, albeit less explicit, sparsity. This overlap with the L_1 regularization in the constraint and the sum-to-one constraint can lead to a case where the explicit sparsity-inducing effect of L_1 regularization is subdued. In other words, while L_1 regularization seeks to minimize the number of contributing endmembers, the sum-to-one constraint simultaneously ensures that the total contribution remains constant. This dual requirement can lead to solutions where the sparsity is not as pronounced as it might be with L_1 regularization alone.

3.1.2. Theoretical computational complexity for the QP model

One of the research questions aims to investigate how constraints affect abundance estimation. To address this, the problem is divided into four specific cases, each with potentially different computational complexities. The computational complexity of each case can be generally described in terms of the operations involved in solving the QP problem, considering factors such as the number of pixels r, the number of bands m, and the number of endmembers n. As each case involves different constraints with varying dependent variables, to be able to find the most optimal solution, hyperparameter tuning for each variable or case will be conducted for fair abundance computation. The theoretical computational complexities for each case based on the QP solver are shown below:

1. Case 1 (Non-negativity constraint): In the first case, the optimization framework is exclusively bounded by a non-negativity constraint. The interior point method implemented in the QP solver gives a computational complexity of $O(r\, n^3)$.
2. Case 2 (Non-negativity and Sum-to-one constraint): Implementing a sum-to-one constraint alongside non-negativity adds a layer of computational complexity without fundamentally changing the overall complexity compared to case 1, which remains $O(r\, n^3)$.
3. Case 3 (Non-negativity, sum-to-one and sparsity constraint): Implementing non-negativity, sum-to-one, and non-linear sparsity constraints directly can increase the computational complexity, potentially exceeding $O(r\, n^{3.5})$. This increase accounts for an estimated factor of $n^{0.5}$ additional iteration due to non-linear constraints. However, linearizing these constraints through Taylor series expansion can reduce the complexity to $O(r\, n^3)$. This transition highlights the computational benefits of linearization.
4. Case 4 (Non-negativity, sum-to-one and generalized sparsity constraint): Implementing a generalized sparsity constraint with extensive hyperparameter tuning is computationally challenging, with complexity expected to exceed $O(r\, n^{3.5})$. However by linearizing the problem, one can reduce this complexity to $O(H\, r\, n^3)$, where H represents the number of hyperparameter configurations explored and $H \in \mathbb{Z}^+$, $H \geq 1$.

3.2. Endmember extraction methods

Endmember estimation also plays an important role in the HU problem, as more accurate endmember estimations lead to better solutions for the abundances. This is directly reflected in the minimizer $\hat{x}$ computed in Section 2 for the constrained least squares, where the endmember matrix A is a dependent variable for computing the abundances.

There are various methods for extracting endmembers, with geometric-based methods being the most well-known. For example, the Pixel Purity Index (PPI) identifies pure pixels by projecting data onto random unit vectors and counting the number of times each pixel is considered extreme in these projections. However, since this method is not iterative, improving the solution requires increasing the number of random projection vectors, leading to high computational complexity (Boardman, 1993; Dixit & Agarwal, 2021). Another iterative method, N-FINDR, seeks to find the simplex of maximum volume within the data. It iteratively examines pixel combinations and selects the one forming the largest simplex, corresponding to the purest spectral signature. This method can also be computationally demanding, especially with large datasets (Du et al., 2008; Winter, 1999).

To incorporate a robust and computationally efficient geometric method for endmember extraction, Vertex Component Analysis (VCA) was chosen for comparison against the newly proposed PCA-based k-means clustering method. VCA is widely used and computationally cheaper compared to the PPI and N-FINDR methods mentioned above. Additionally, since VCA utilizes dimensional reduction techniques such as Singular value decomposition (SVD) or PCA, it serves as a fair benchmark for comparisons. The detailed procedure of how VCA works is shown in the following subsection with slight adjustment to the original paper so it aligns with a pixel-by-pixel manner (Nascimento & Dias, 2005).

3.2.1. Vertex component analysis

The vertex component analysis (VCA) is an unsupervised algorithm used for endmember extraction. This algorithm works under the assumption that in a linear mixing model, the spectral signature of each pixel can be represented as a linear combination of the spectra of the endmembers present in the scene.

The VCA algorithm primarily investigates two key aspects: first, whether the endmembers can be identified as the vertices of simplexes in the data space, and second, whether the affine transformation of each simplex remains simplex. The algorithm begins with the assumption that endmembers are presented within the data, then it iteratively projects the sample data vectors onto directions orthogonal to the subspace spanned by the previously identified endmembers. The new endmembers are identified based on the minimal or maximal extreme projections, and this iterative process continues until all potential endmembers have been evaluated for the dataset.

To have a better understanding of how VCA works, consider the following:

Assume in a linear mixing, in a pixel-by-pixel manner, the observed spectral vector is given as:

$$y_i = \alpha_i + e_i = A\gamma_i x_i + e_i = As_i + e_i \qquad \text{for} \quad i = 1, ...r \tag{3.11}$$

Where r is the number of pixels, y is an m-vector with m number of bands for each pixel, e_i the additive noise vector, $x_i = [x_{i,1}, x_{i,2}...x_{i,n}]^T$ the abundance vector which contains the fractions of each endmember for each pixel with n endmembers, $A = [a_1, a_2, ...a_n]$ a $m \times n$ mixing matrix which maps the abundance vector x_i to the true spectral vector α_i with endmember spectral vector a_n, and lastly γ_i is a scale factor modeling illumination variability due to surface topography.

The main goal of VCA is to find the abundance vector x_i with its corresponding endmember vector a_n. Now note that in this linear mixing model, one also has to consider the physical constraints for the abundance fractions, specifically two constraints are being implemented, the nonnegativity $x \geq 0$ with positivity constraint $1^T x = 1$ with 1 a $n \times 1$ vector of ones (Nascimento & Dias, 2005). Every pixel can be considered as a sample vector within the m-dimensional Euclidean space.

Now since the set $\{x \in \mathbb{R}^n : 1^T x = 1, x \geq 0\}$ is simplex, the set $S_\alpha = \{\alpha \in \mathbb{R}^m : \alpha = Ax, 1^T x = 1, x \geq 0\}$ is also simplex. The proof of the above statement is rather straightforward: A simplex in a n-dimensional space is a convex hull of $n + 1$ vertices, where no vertices are colinear, forming the simplest possible polytope in any given dimension. Note that as A consists of n linearly independent endmembers /columns, the set of all linear combinations of $\alpha = Ax$ under the constraint $1^T x = 1$ and $x \geq 0$ forms a convex polytope. The vertices of this polytope are the columns of A. Since x enforces a combination of these vertices where the coefficients sum to one are non-negative, S_α forms a simplex, as it represents a convex hull of these vertices under given constraints.

Consider now the scaling factor γ with $e = 0$, the observed vector set then belongs to the convex cone $C_n = \{y \in \mathbb{R}^m : y = A\gamma x, 1^T x = 1, x \geq 0, \gamma \geq 0\}$. To prove that it indeed belongs to the convex cone C_n, by definition of a convex cone, it has to satisfy both the convexity and cone property, which can be shown as follows:

1. (Convexity) For any two vectors $y_1, y_2 \in C_n$, with any scalar t where $0 \leq t \leq 1$, the combination of $ty_1 + (1 - t)y_2$ also belongs to C_n. This is due to the linear nature of y with respect to x and γ, and the constraint of x and γ ensures that any linear combination remains in the cone.
2. (Cone property) Let any vector $y \in C_n$ and any scalar $\zeta \geq 0$, the vector ζy also belongs to C_n. This directly follows from the definition of C_n as γ can be scaled by ζ, showing that scaling y by any non-negative scaler keeps it within C_n.

The crucial aspect of C_n is its ability to model the geometric structure of the data in the presence of scale variability.

Lastly, a projective projection of the convex cone C_n onto a chosen hyperplane transforms it into another simplex S_n, where the projection is defined as: $S_n = \{z \in \mathbb{R}^m : z = y/(y^T u), y \in C_n\}$ and u is chosen such that no vector y in C_n is parallel to the hyperplane defined by $y^T u = 1$. This condition

ensures that the division by $y^T u$ is always possible, mapping each point y to new point z, and its infinite extension is mapped to a bounded region on S_n.

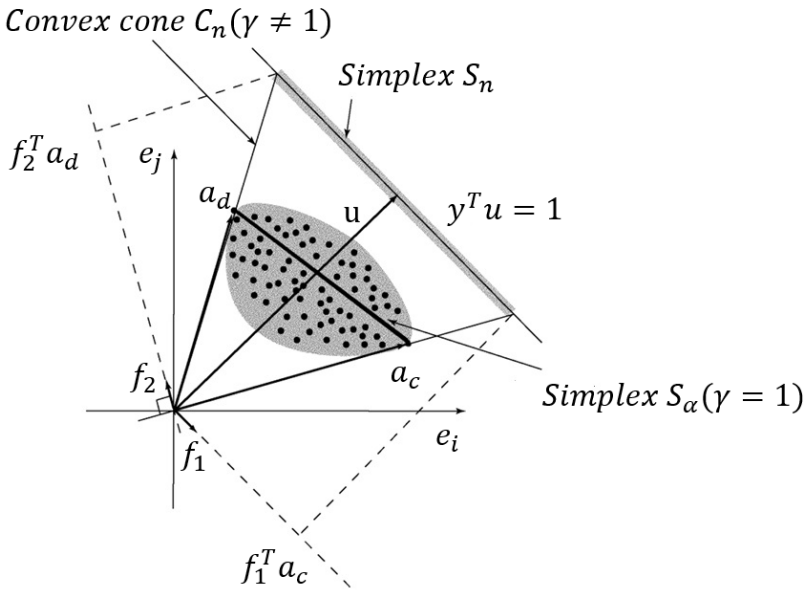


Figure 3.1: VCA algorithm illustration (Nascimento & Dias, 2005)

With the given simplex S_n, the VCA algorithm iteratively projects the data onto a direction orthogonal to the subspace spanned by the endmembers that have been identified. Figure 3.1 shows an illustration of two iterations for the VCA algorithm to the simplex S_n for a mixture of two endmembers. The data first are projected along direction f_1 where the extreme point a_c of this projection corresponds to the first endmember. Subsequently, the projection along direction f_2, which is orthogonal to a_c, identifies the second endmember at the extreme point a_d. The algorithm continues this iterative process, identifying new endmembers by projecting the data onto directions orthogonal to the subspace defined by all previously identified endmembers, until the total number of endmembers is determined. The pseudo-code for the VCA algorithm used in this research is shown in Appendix 2, more details about the code itself can be found in the paper (Nascimento & Dias, 2005).

To address the computational complexity, particularly given that most of the spectral information (the 'subspace signal') is contained within a far smaller space than the total number of spectral bands ($n \ll N$), different dimensional reduction techniques such as PCA or SVD can be employed to VCA algorithm to narrow down the data onto subspace signal based on the signal-noise-ratio values.

3.2.2. Computational complexity for VCA

The theoretical computational complexity for VCA can become significant if the number of endmembers is large. Consider the algorithm shown in Appendix 2. Let m be the number of spectral bands, r be the number of pixels, and n be the number of endmembers. The dimensional reduction step in the VCA for a $m \times r$ matrix has $O(m\, r^2)$ floating point operations (flops). In the iterative step where data is projected onto a lower-dimensional subspace to identify vertices of the convex hull, has a complexity of $O(n\, r)$ flops per iteration. Since there are n endmembers, there are n iterations. Thus the total flops for the iterative step is $O(n^2\, r)$. Combining these steps, the total computational complexity of the VCA algorithm is $O(max(m\, r^2, n^2\, r))$.

3.2.3. The PCA method

In addition to VCA, a newly developed PCA-based k-means clustering method is introduced in this research. For the remainder of this thesis, this method will be referred to as **the PCA method**. The inspiration for this method came from the paper (Amiri-Simkooei et al., 2011), where a similar approach

was used for seafloor classification. This method is simple and intuitive. It combines the advantages of PCA with k-means clustering, which can simply the endmember extraction process while maintaining computational efficiency and robustness for complex datasets. To have a better understanding of how the method works, in the following PCA and K-means will be explained in detail:

Principal component analysis

Principal component analysis (PCA) is a statistical technique that uses an orthogonal transformation to convert a set of possibly correlated observations into a set of linearly uncorrelated variables known as principal components (PCs). In this transformation, the first principal component is defined to capture the maximum variance from the data, then each subsequent component accounts for the highest variance possible while being orthogonal to all previously defined components. The resulting PCs form an uncorrelated orthogonal basis set, which is also sensitive to the relative scaling of the original variables. To address the challenges with high-dimensional feature spaces, which usually have complicated visualization and interpretation, PCA reduces the dimensionality while also retaining the most feature variations. This reduction is achieved by identifying the direction of PCs where data variation is maximized. When visualizing complex datasets as coordinates in a high-dimensional data space, PCA usually provides a lower-dimensional picture, often referred to as the 'shadow' of the object. This shadow is viewed from the most informative angle that can capture the essential structure of the data, however, it is important to note that, some parts of the data might be lost during the PCA process (Jolliffe, 2002; Amiri-Simkooei et al., 2011).

The full PCA process is as follows: Given a data matrix $Y \in \mathbb{R}^{m \times r}$ such that $Y = [y_1\ y_2\ ...\ y_r]$ where r is the number of data points or sample size and m be the spectral bands or features, the data first have to go through standardization that involves adjusting each variable to have zero mean and unit variance to ensure that each variable have the same weight and contributes equally to the PCA process. Then the $m \times m$ covariance matrix Σ can be computed as

$$\Sigma = \frac{YY^T}{r-1} \tag{3.12}$$

where the diagonal elements represent the variances of the feature and off-diagonal elements are the correlation coefficients of the features.

An eigenvalue decomposition of the positive-definite covariance also has to be performed

$$\Sigma = U\Omega U^T \tag{3.13}$$

where U is the matrix whose columns are the eigenvectors of Σ, and Ω is a diagonal matrix with eigenvalues $v_i, i = 1, ...\ r$ of Σ on the diagonal. The eigenvectors (principal axes directions) define the direction of the new feature space, and the eigenvalues define their magnitude (the variance explained by the principal axes).

Lastly, the eigenvectors are ordered by decreasing eigenvalues, a choice can be made on the number of PCs to retain. To be more specific, U can be partitioned as $U = [U_1; U_2]$, where U_1 contains the first $n \leq m$ eigenvectors of Σ with the largest eigenvalues in Ω, and U_2 contains the remaining eigenvectors that have the smallest eigenvalues of Σ. The data then can be projected onto the new feature space

$$P = U_1^T Y \tag{3.14}$$

where P is the new coordinates of data Y in the PC space.

To quantify the relative importance of each PCs, the percentage of the total variance explained by each component is computed as

$$\%i = \frac{v_i}{\sum_{k=1}^{m} v_k} \times 100 \tag{3.15}$$

and $v_k = trace(\Omega) = trace(\Sigma)$ is the total variance (Amiri-Simkooei et al., 2011).

K-means clustering

K-means clustering is a widely used method for partitioning a dataset into k distinct, non-overlapping subgroups (clusters). It aims to minimize the clustering error, which is defined as the sum of the squared distances between the data points and their respective cluster centroids (MacQueen, 1967; Sinaga and Yang, 2020).

To be more specific, let $y_i \in \mathbb{R}^m$ and $Y = [y_1\ y_2\ \ldots\ y_r]$ be the dataset consisting r points in m-dimensional space. By letting $P = \{P_1, P_2, \ldots P_k\}$ be the distinct subsets, the main objective is to partition the observations into k sets. The objective function thus can be formulated as:

$$\arg\min_{P} \sum_{i=1}^{k} \sum_{y_i \in P_i} \|y_i - c_i\|^2 \tag{3.16}$$

where c_i is the cluster centroid, and is defined as:

$$c_i = \frac{1}{|P_i|} \sum_{y_i \in P_i} y_i \tag{3.17}$$

where $|P_i|$ is the number of data points in the i-th subset, and the equation calculates the mean of these points that represents the center of the cluster in the feature space.

With the help of equations 3.16 and 3.17, the k-means clustering algorithm works as follows: k-means clustering algorithm is an iterative process, first an initialization step has to be taken, this step selects the k initial centroids, either randomly or by some heuristics. Each data point y_i has to be assigned to the nearest centroid to form k clusters, this is usually based on the minimum distance criteria:

$$P_i^{(\eta)} = \left\{ y_i \in Y : \left\| y_i - c_i^{(\eta)} \right\|^2 \leq \left\| y_i - c_j^{(\eta)} \right\|^2 \forall j, 1 \leq j \leq k \right\} \tag{3.18}$$

where η denotes the iteration number. The new centroid of the cluster then will be recalculated:

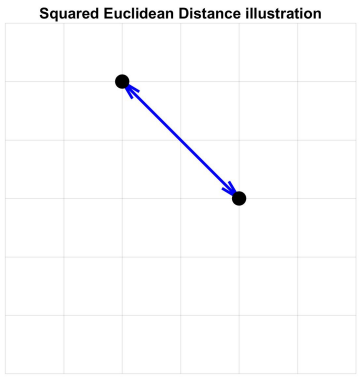
$$c_i^{(\eta+1)} = \frac{1}{\left| P_i^{(\eta)} \right|} \sum_{y_i \in P_i^{(\eta)}} y_i \tag{3.19}$$

This process takes place until the centroids are stabilized, which indicates a convergence.

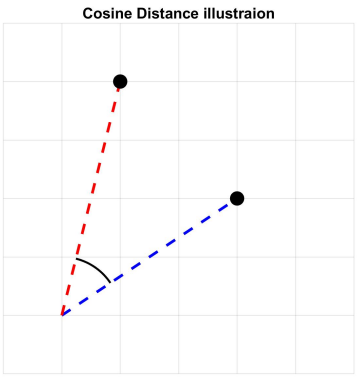
K-means clustering offers significant advantages, as it is a fast, robust method with high computational efficiency. However, it also has downsides. One major limitation is that it does not always lead to a global minimum solution for the clustering function (Steinley, 2008). Instead, it often converges to a local minimum, which means the final solution may depend heavily on the initial selection of the centroids, that can lead to a suboptimal outcome. Furthermore, the use of Euclidean distance measures can lead to unequal weights for underlying factors. In high-dimensional space, the distance can become dominated by features with larger scales, and the square differences can over-penalize outliers. Thus to address some of the limitations, an alternative metric cosine distance can be used, as it measures the similarity in terms of the angle between points rather than their absolute positional differences. Which can be defined as:

$$P_i^{(\eta)} = \left\{ y_i \in Y : \left(1 - \frac{y_i \cdot c_i^{(\eta)}}{\|y_i\| \left\| c_i^{(\eta)} \right\|} \right) \leq \left(1 - \frac{y_i \cdot c_j^{(\eta)}}{\|y_i\| \left\| c_j^{(\eta)} \right\|} \right) \forall j, 1 \leq j \leq k \right\} \tag{3.20}$$

The cosine distance is particularly useful when the orientation matters more than the magnitudes. Figure 3.2a and 3.2b shows an example of the differences when considering the two distance metrics.



(a) Example of squared euclidean distance metric



(b) Example of cosine distance metric

Figure 3.2: Distance metrics examples

Integrating PCA and k-means clustering shown above, the PCA method proceeds as follows:

The process begins with PCA, which reduces the dimensionality of the hyperspectral data by retaining the most significant components. This step involves standardizing each spectral band to have zero mean and unit variance, followed by computing the covariance matrix and performing SVD. The number of PCs retained is determined based on the cumulative variance, which in this case is set to capture over 99.5% of the variance. This criterion ensures that the most critical information is retained, with the first two PCs capturing the majority of the variance.

After dimensionality reduction, k-means clustering is applied to the data transformed by PCA. The k-means algorithm partitions the data into k clusters, each representing a potential endmember, which means in our case $k = n$. During the k-means clustering process, various distance metrics can be used to measure the similarity between data points and cluster centroids. In this research, metrics such as squared Euclidean distance and cosine distance were utilized to explore different clustering characteristics.

In summary, the steps for this method work as:

1. PCA step:
    - Standardize each spectral bands to have zero mean and unit variance.
    - Compute the covariance matrix of the standardized data.
    - Perform SVD on the covariance matrix to obtain eigenvectors and eigenvalues.
    - Retain PCs which captures over 99.5% of the cumulative variance.
2. k-means clustering step:
    - Apply k-means clustering to the data transformed by PCA.
    - Apply distance metrics to partition data into n clusters where each n corresponds to the number of endmembers.
    - The cluster centroid then represents a potential endmember.

3.2.4. Computational complexity for the PCA method

The computational complexity for the PCA method can potentially be lower than VCA if the number of PCs retained after PCA and the number of k-means iterations are kept low. Let m be the number of spectral bands, r be the number of pixels, η be the number of iterations, n be the number of clusters/endmembers and p be the retained PCs. The computational complexity of computing the covariance matrix of an $m \times r$ data matrix is $O(m\, r^2)$, performing SVD on the covariance matrix typically has a complexity of $O(m^3)$. However, when $m \ll r$, the complexity is dominated by the matrix

multiplication, the complexity then remains with $O(m\,r^2)$ flops. Each iteration of the k-means clustering step involves assigning r pixels to n centroid in p-dimensional space and updating the centroid, with η iteration, the complexity is $O(\eta\,n\,p\,r)$. Combining these steps the total computational complexity of the PCA method is $O(max(m\,r^2, \eta\,n\,p\,r))$, which indicates the PCA method is more computationally efficient when p and η are low.

3.3. WTLS algorithm formalization

In the field of HU, many methods estimate the endmember matrix and the abundance matrix simultaneously rather than computing them separately. Examples include various non-negative matrix factorization (NMF) methods (Feng et al., 2022; Gholinejad & Amiri-Simkooei, 2024; Khoshsokhan et al., 2019). These methods iteratively estimate both the endmember and abundance matrices by decomposing the observed data into non-negative components, ensuring that the solution remains physically meaningful.

Inspired by the iterative nature of NMF, the WTLS algorithm can be adapted to perform a similar joint estimation, potentially enhancing the solution quality for the endmembers. Unlike NMF, WTLS inherently accounts for error in both the observation y and the design endmember matrix A. This duel error considerations makes the WTLS particularly suitable for HU where measurement noise and model inaccuracies can impact the estimation process. The WTLS algorithm used in this research is adapted from (Gholinejad & Amiri-Simkooei, 2023) with some modification. It uses the output from the QP model and endmember estimation from the VCA and PCA methods as an initial input. The adapted algorithm 1, outputs $\tilde{A}$, which is the new estimated endmember matrix.

Algorithm 1 Weighted Total Least Squares (WTLS) Algorithm

Inputs: A, Y, X_{optimal} (Solution from QP)
Initialize: $\hat{X} = X_{\text{optimal}}$
$\hat{E} \leftarrow Y - A\hat{X}$
$\hat{\Sigma}^{-1} \leftarrow \text{inv}\left(\Sigma_Y + \hat{X}^T\Sigma_A\hat{X}\right)$
$\tilde{E}_A \leftarrow -\hat{E}\hat{\Sigma}^{-1}\hat{X}^T\Sigma_A$
$\tilde{A} \leftarrow A - \tilde{E}_A$
Output: $\tilde{A}$

3.3.1. Computational complexity for WTLS algorithm

The computational complexity for the full WTLS algorithm can be found in paper (Gholinejad & Amiri-Simkooei, 2023). Although the WTLS algorithm was modified, Σ^{-1} remains as one of the most computationally heavy steps as it is required to compute the inverse of a $g \times g$ matrix, which leads to $O(g^3)$. As for the minimizer $\hat{X}$, it is dependent on the QP solutions. The specific computational complexities for each case of the QP solution are listed in subsection 3.1.2. These complexities vary based on the constraints applied towards the abundance estimation. Thus considering the most computationally heavy QP solution, the complexity for the adjusted WTLS is $O(max(g^3, H\,r\,n^3))$.

3.4. Evaluation metrics

The quality of the HU method proposed above will be assessed using two evaluation metrics: Root Mean Square Error (RMSE) and Spectral Angle Distance (SAD). These metrics provide a quantitative measure of the accuracy and reliability of the estimated abundance and endmember matrices.

The RMSE is computed as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_r - \hat{c}_r^2)} \tag{3.21}$$

where c_r and $\hat{c}_r$ represent the true and estimated spectral signatures of the r-th endmember, respectively. RMSE is a widely used metric for quantifying the difference between the estimated values and the true

values. It provides a straightforward measure of the magnitude of the error.

SAD is calculated as:

$$\text{SAD} = \arccos\left(\frac{c_r^T \hat{c}_r}{\|c_r\| \|\hat{c}_r\|}\right) \tag{3.22}$$

SAD measures the spectral similarity between the true and estimated spectral signatures by calculating the angle between them. Unlike RMSE, SAD is invariant to scaling, making it particularly useful in hyperspectral applications where the spectral shape is more important than the absolute magnitude.

4

Results and Discussions

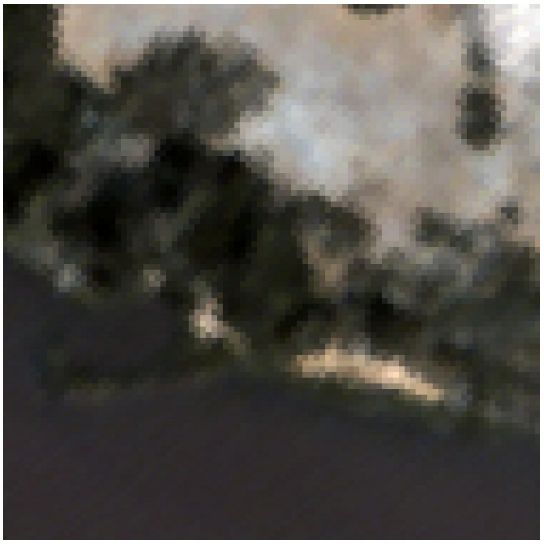
To evaluate the proposed QP model, endmember matrix estimation, and the effects of using WTLS algorithm, a series of experiments have been conducted across different cases using three distinct datasets: the Samson dataset, the Jasper Ridge dataset, and the Urban dataset. The Samson dataset comprises 3 endmembers, while the Jasper Ridge and Urban datasets each contain 4 endmembers. The three datasets contain diversified endmembers, which can test how the proposed method performs for various spectral signatures. Moreover, all datasets include observed spectra y and a known endmember matrix A, enabling experiments to proceed under the assumption that these variables are pre-determined. However, this assumption may not always hold in practical applications.

The specific settings for each case are listed in Subsection 3.1.2. To be able to conduct a comprehensive analysis of our QP model, the true abundance with unconstrained least squares solution will also be included in the comparison. In each case, different parameters significantly influence the outcomes of the QP model, thus to optimize results across various conditions, hyperparameter tuning will be applied to each parameter. A grid search method will be implemented to find the best hyperparameter values for this study (Zhu, Wang, Fan, et al., 2014; Zhu, Wang, Xiang, et al., 2014), and the specific parameters to be tuned are as follows: Q_c, the variance matrix which assigns weights to the corresponding constraints; q, the regularization factor that modulates the sparsity of the solution; and μ, a constant that influences a certain degree of sparseness in the solution through the regularization term involving q, that ensures the solution remains in a practical bound.

This process will also involve an exploration of both weighted constraints (WC) and hard constraints (HC) to determine their ideal configurations. By systematically adjusting and evaluating these parameters, the analysis ensures the most accurate result of the QP model for diverse datasets. Moreover, to evaluate the performance under hyperparameter tuning, evaluation metrics such as RMSE and SAD will be used to determine the performance of the QP model.

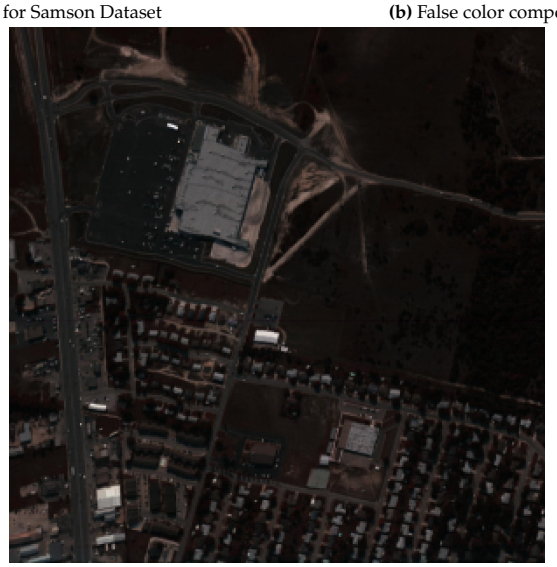
It is also worth stating that for the WC, the hyperparameter tuning for the variance matrix will have the value $Q_c \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3\}$, whereas for the HC the variance matrix will be restricted to $0 < Q_c \leq 1$, specifically $Q_c \in \{10^{-3}, 10^{-2}, 10^{-1}\}$, as it implies that the constraints are forced to be implemented to the QP calculation. Note that as case 3 is equivalent to L_2 regularization for the sparsity constraint, and when $q = 1$ it implies L_1 regularization which duplicates the sum-to-one constraint, investigation for $L_{0<q<1}$ will be done together with $0 < \mu \leq 1$. This adopted range of values for q and μ will only be used in case 4, as it represents the most generic form for the QP model.

After the analysis of the QP model, the effectiveness and accuracy of the estimated endmember matrix will be further investigated through comparison with the VCA and the new PCA methods. Lastly, to explore the potential of enhancing these results, WTLS algorithm will be employed.



(a) False color composite for Samson Dataset

(b) False color composite Jasper Ridge Dataset



(c) False color composite for Urban Dataset

Figure 4.1: False color composite for each dataset

4.1. Samson Dataset

The Samson Dataset is the first hyperspectral dataset that will be tested. The data originally is very large which has 952×952 pixels with 156 bands. However, for this study, a sub-image will be considered and now the size is reduced to 95×95 pixels. As stated for this dataset the endmember signature and the abundance are all available, so the quality of the QP model can be evaluated. Further details of the dataset can be found in (Zhu, 2017).

The false color composite for the Samson Dataset can be found in Figure 4.1a, where it contains 3 endmembers, namely: 'Rock', 'Tree' and 'Water'.

4.1.1. Abundace solutions

The following Table 4.1 shows the optimal hyperparameter tuning values for the WC and HC under each case. With these parameter settings, the visualization of the abundance maps using QP can be found in Figure 4.2 and Figure 4.3 for the WC and HC respectively. Moreover, the quality of each endmember compared to the true abundance solution is examined in terms of RMSE and SAD, which the results are presented in Table 4.2 and 4.3.

	Parameters	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Q_{c1}	-	-	1000	1000	1000
	Q_{c2}	-	-	-	1000	10
	μ	-	-	-	1	0.7
	q	-	-	-	2	0.7
HC	Q_{c1}	-	-	0.1	0.001	0.1
	Q_{c2}	-	-	-	0.1	0.1
	μ	-	-	-	1	0.6
	q	-	-	-	2	0.9

Table 4.1: Weighted Constraints (WC) and Hard Constraints (HC) optimal parameter values via hyperparameter tuning for Samson Dataset

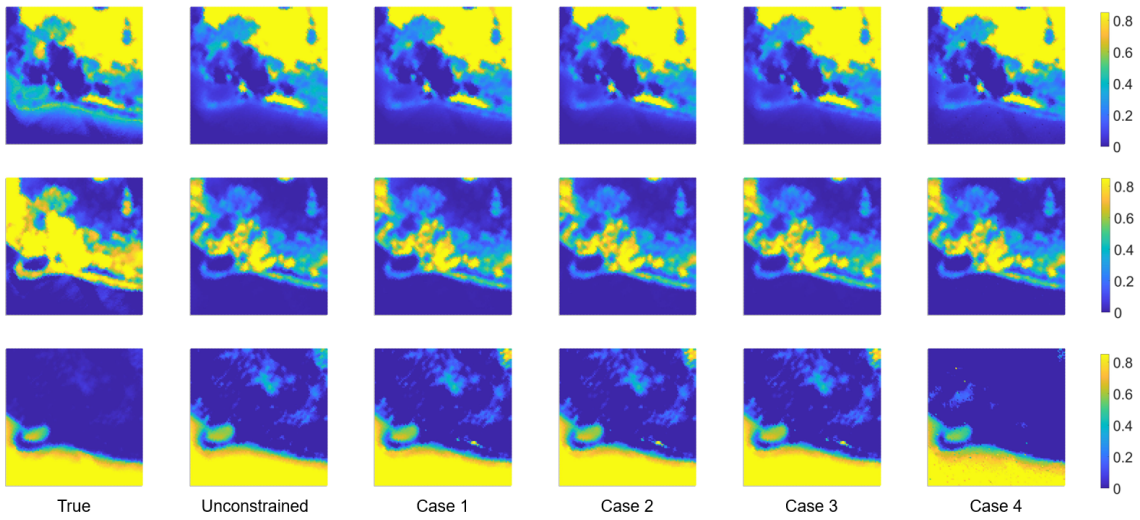


Figure 4.2: Weighted-constrained abundance maps for Samson Dataset with true and unconstrained abundance maps. Rows from top to bottom represent rock, tree and water respectively.

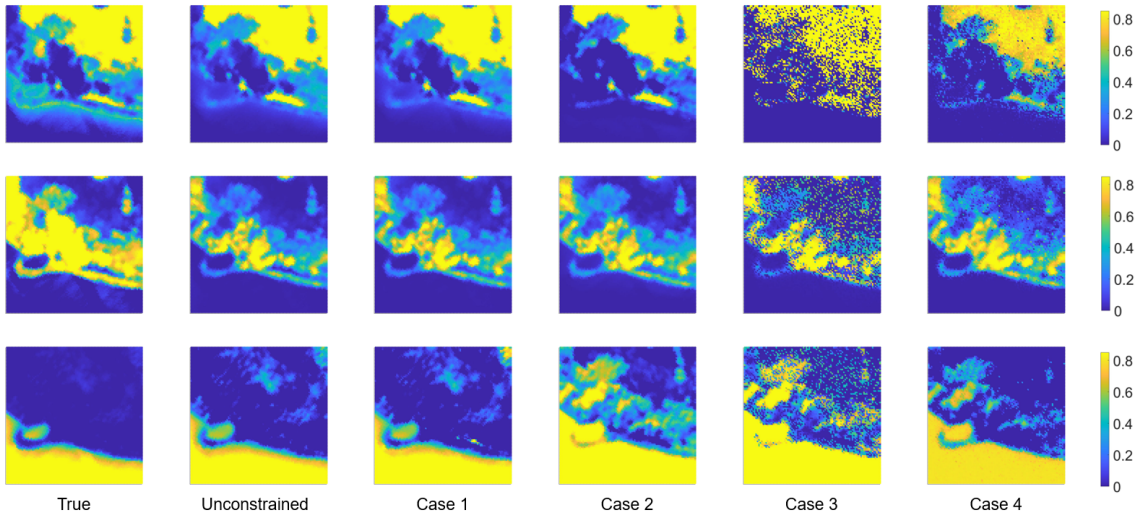


Figure 4.3: Hard-constrained Samson abundance maps for Samson Dataset with true and unconstrained abundance maps. Rows from top to bottom represent rock, tree and water respectively.

	Endmember	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Rock	0.1452	0.1369	0.1374	0.1353	0.1305
	Tree	0.2546	0.2483	0.2482	0.2481	0.2496
	Water	0.1663	0.0999	0.1005	0.1222	0.0589
	WC Average	0.1887	0.1617	0.1620	0.1685	**0.1463**
HC	Rock	0.1452	0.1369	0.1789	0.3505	0.1924
	Tree	0.2546	0.2483	0.2302	0.3311	0.2442
	Water	0.1663	0.0999	0.3473	0.3646	0.2013
	HC Average	0.1887	**0.1617**	0.2521	0.3487	0.2126

Table 4.2: Weighted constraint (WC) and Hard Constrained (HC) RMSE values for each case under optimal parameter settings with Samson Dataset

	Endmember	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Rock	16.0701°	15.3170°	15.3723°	15.1319°	14.5865°
	Tree	20.4673°	19.8965°	19.8704°	19.8307°	20.2670°
	Water	18.0215°	10.7151°	10.7467°	14.0768°	6.3698°
	WC Average	18.1863°	15.3095°	15.3298°	16.3465°	**13.7411**°
HC	Rock	16.0701°	15.3170°	20.1884°	38.8075°	20.3512°
	Tree	20.4673°	19.8965°	17.8041°	37.6717°	21.8771°
	Water	18.0215°	10.7151°	33.3821°	34.3561°	25.6258°
	HC Average	18.1863°	**15.3095**°	23.7915°	36.9451°	22.6180°

Table 4.3: Weighted Constrained (WC) and Hard Constrained (HC) SAD values for each case under optimal parameter settings with Samson Dataset

As one can see from the Table 4.2, the solution for the WC tends to have better solutions compared to the HC solution. This is reasonable as in HC cases setting extra restrictions can lead to a worse solution, since the solution can lie outside the proper solution space after adding constraints. Moreover, while the unconstrained weighted least squares case already has a quite decent solution when comparing the abundance map to the true solution, WC case 4 has an even better solution. As for the HC cases, case 1 produces the best solution, although this is the case, the nonnegativity constraint itself is not enough as it falls short of encapsulating the full physical essence of the unmixing process. Thus it is important to consider the trade-off between the solution accuracy and the integrity of the physical interpretation of HU.

The solution from Table 4.3 shows a similar trend to the RMSE solutions meaning they are somewhat correlated. Higher RMSE values show higher SAD values and vice versa. Based on Figure 4.3 shows that most cases have difficulty estimating endmember 'water' for HC cases, which might be the illumination effect based on the dataset itself.

4.1.2. Endmember matrix estimations

As the endmember matrix plays a crucial role in abundance estimation for the QP model, the following Figure 4.4 shows the result of using the VCA and PCA methods. These methods are categorized into two distinct groups. In the VCA method, 'VCA single runs' execute the algorithm only once to deduce endmembers from the hyperspectral dataset y, while 'VCA bundle runs' perform multiple iterations to enhance the robustness of the extraction. Conversely, the PCA method is divided based on the 2 metrics when clustering the data, specifically squared Euclidean distance and cosine distance, as they can affect the final solution performances. The quality of clustering achieved by the two PCA methods is illustrated in Figure 4.5. These plots show how the data points which represent different spectral signatures are distributed along PCs. Effective clustering would be indicated by well-separated clusters with minimal overlap, suggesting that the PCA method has successfully distinguished different endmembers. Although the first two PCs in already captures 99.5% of the cumulative variance which reflects the most critical variations within data, other PCs comparisons are also shown to reveal additional data structure and to examine the consistency of clustering across different dimensions.

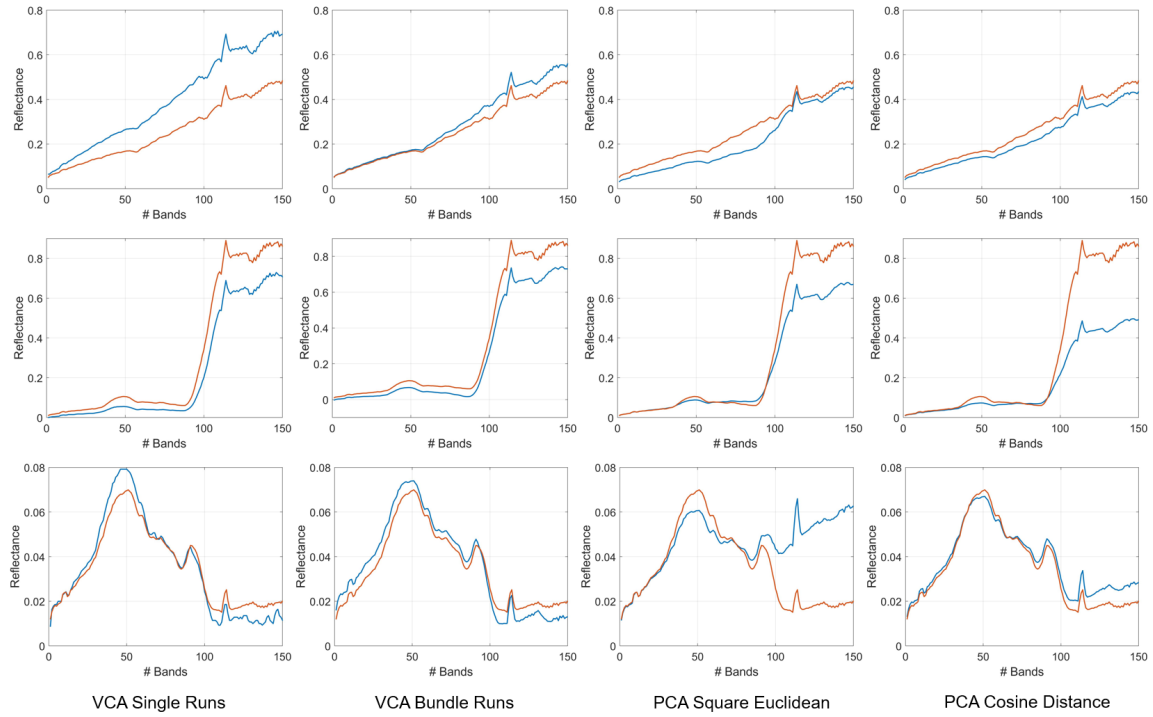
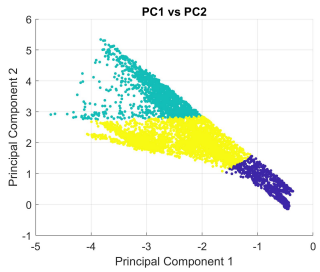
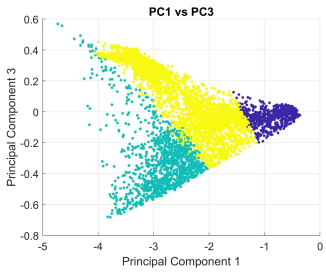


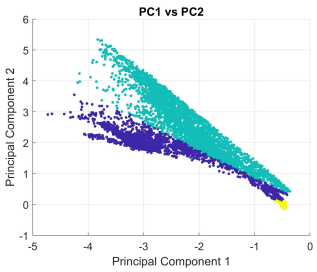
Figure 4.4: Estimated endmembers (blue) and true endmembers (red) for Samson Dataset. Rows from top to bottom represent rock, tree and water respectively



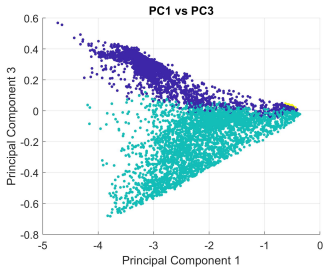
(a) PC1 vs PC2 using squared euclidean metric for clustering



(b) PC1 vs PC3 using squared euclidean metric for clustering



(c) PC1 vs PC2 using cosine distance metric for clustering



(d) PC1 vs PC3 using cosine distance metric for clustering

Figure 4.5: PCs visualization for Samson Dataset, each color represents a potential endmember

	Endmember	VCA Single	VCA Bundle	PCA Square Euclidean	PCA Cosine
RMSE	Rock	0.1516	0.0419	0.0502	0.0373
	Tree	0.1068	0.0863	0.1148	0.2133
	Water	0.0055	0.0050	0.0226	0.0046
	Average	0.0880	**0.0444**	0.0625	0.0851
SAD	Rock	2.3168°	1.7947°	7.3738°	2.1329°
	Tree	4.0840°	2.6828°	2.8025°	2.8025°
	Water	7.4718°	6.4537°	27.0267°	6.2479°
	Average	4.6242°	**3.6437°**	12.4010°	3.7278°

Table 4.4: RMSE and SAD values for endmember matrix estimation (Samson Dataset)

From Table 4.4, 'VCA bundle runs' achieve the lowest values in terms of both RMSE and SAD. However, a closer inspection of the second column in Figure 4.4 reveals that the 'VCA bundle runs' generate negative values. In the context of endmember estimation, negative values are not physically meaningful and should generally be avoided.

In comparison, the 'PCA Square Euclidean' and 'PCA Cosine' methods provide the next best solutions based on RMSE and SAD values, respectively. Although their RMSE values are relatively close, there are notable differences in their SAD values. Specifically, the 'PCA Square Euclidean' shows a relatively poor estimation for the endmember 'water' compared to the 'PCA Cosine' method. This indicates that when the dataset includes the endmember 'water', the 'PCA Cosine' approach might be more appropriate due to its superior performance in capturing the spectral characteristics of water in terms of SAD. Since there are no significant differences in RMSE values between the two PCA methods, the better SAD performance of the 'PCA Cosine' method makes it a preferable choice over the 'PCA Square Euclidean' method.

4.1.3. WTLS solutions

The solutions from the QP model and the endmember matrices will serve as inputs for the WTLS simulations. For these tests, only case 4 will be examined for both WC and HC scenarios, as it is the most generic case encompassing all constraints. Additionally, the QP solutions will be valued across each endmember method that was proposed to assess the potential improvement in solutions.

To ensure efficient and effective convergence, a stopping criterion was set at 10^{-6} with a maximum of 50 iterations, since this process can be time consuming when the datasets get larger. Figures 4.6 and 4.7 illustrate the endmember solutions using the WC and HC solutions from the QP model, applied to each endmember extraction method. Figures 4.8 and 4.9 demonstrate how the RMSE and SAD values for endmembers and abundances change after implementing the WTLS algorithm.

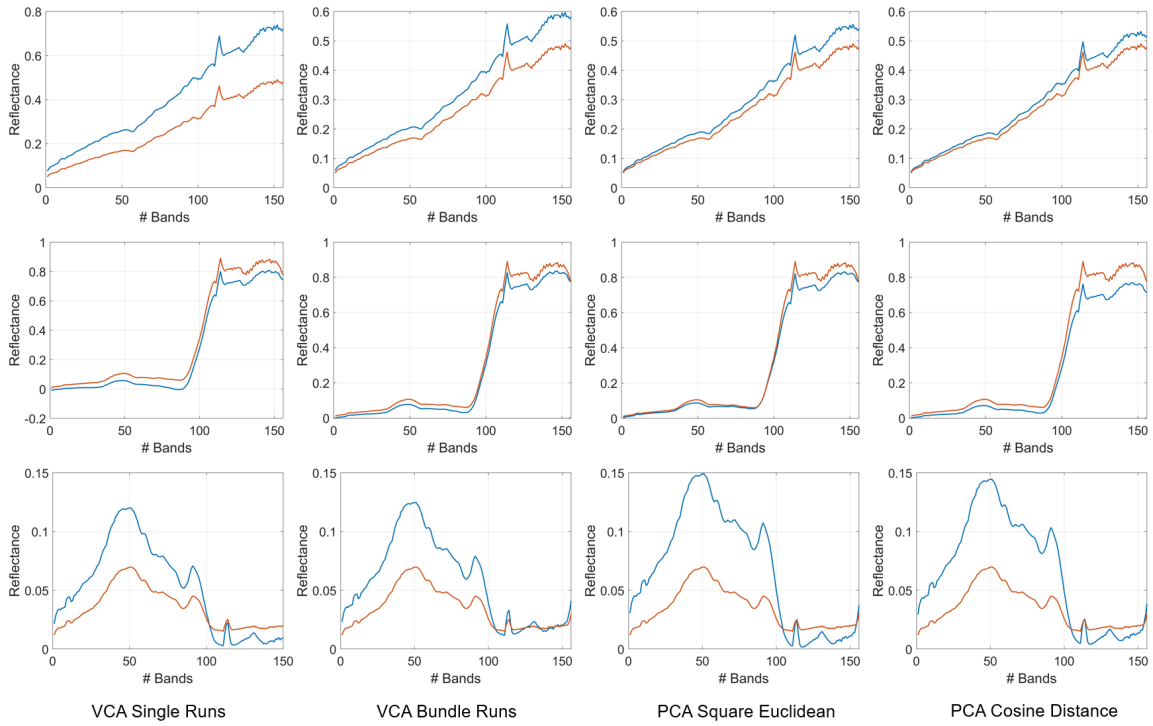


Figure 4.6: WTLS estimated endmembers (blue) and true endmembers (red) for Samson Dataset with WC settings. Rows from top to bottom represent rock, tree and water respectively

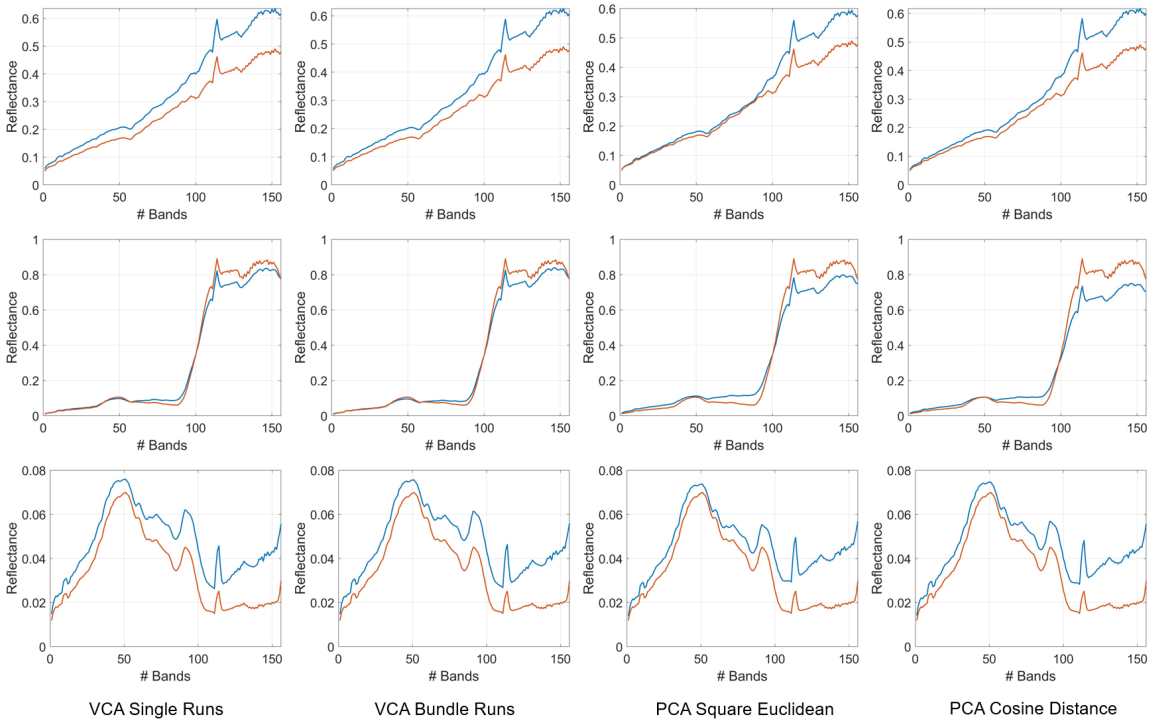


Figure 4.7: WTLS estimated endmembers (blue) and true endmembers (red) for Samson Dataset with HC settings. Rows from top to bottom represent rock, tree and water respectively

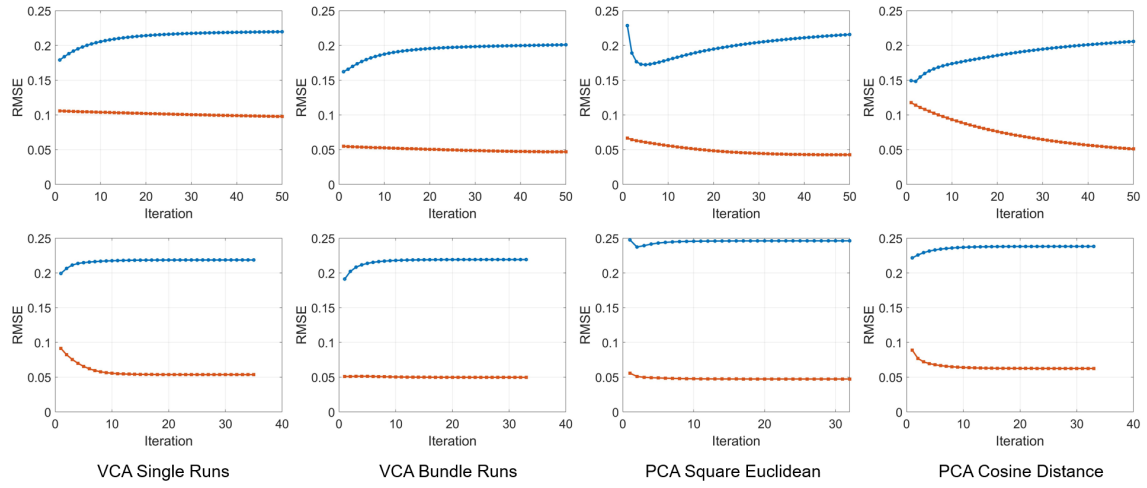


Figure 4.8: Abundance RMSE (blue) and endmember RMSE (red) comparison with different input endmember matrices using WTLS algorithm. The first row represents the RMSE after using WC abundance solutions, and the second row represents the RMSE after using HC abundance solutions from QP respectively. (Samson Dataset)

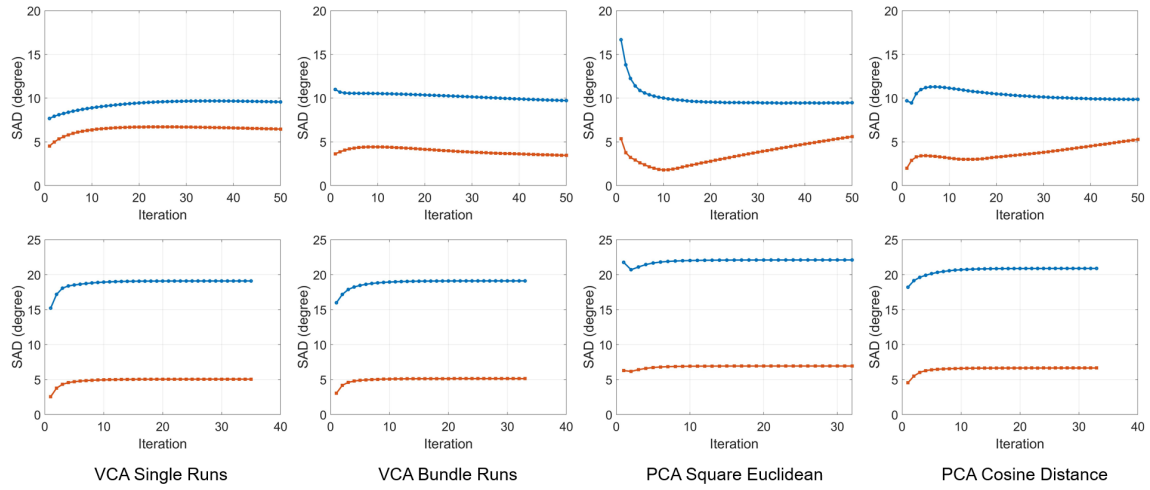


Figure 4.9: Abundance SAD (blue) and endmember SAD (red) comparison with different input endmember matrices using WTLS algorithm. The first row represents the SAD after using WC abundance solutions, and the second row represents the SAD after using HC abundance solutions from QP respectively. (Samson Dataset)

From Figure 4.6, it can be observed that 'VCA Single Runs' produce negative values for the endmember 'tree' under WC cases. For the other three cases, there is an improvement in the endmember 'tree' compared to the original endmember extraction results in Figure 4.4, but generally, worse results for the endmember 'water'. The first row of Figure 4.8 also shows an inversely proportional relationship between the abundance and endmember RMSE: as the endmember RMSE improves, the abundance RMSE worsens. Notably, the endmember RMSE for the PCA case seems to improve after 50 iterations.

When the abundance is set to HC solutions, all new endmember solutions become positive for all methods. Significant improvement is observed for the endmember 'tree' in all cases, along with a direct improvement in the endmember 'water' estimation. A closer examination, considering the average RMSE values from Table 4.4 and the second row of Figure 4.8, indicates improvement in the endmember solution for 'VCA Single Runs', 'PCA Square Euclidean', and 'PCA Cosine Distance'. However, the RMSE value for abundances worsens compared to the original value of 0.2126 for abundance estimation alone.

Regarding the solutions in terms of SAD, using the WC QP solution, it can be seen that the SAD values using WTLS can lead to an improvement in abundance results in terms of SAD compared to the original value of 13.7411°. There is a significant improvement in the 'PCA Square Euclidean' SAD value compared to 12.4010°, but the SAD values for the other methods worsen slightly. Under HC QP solutions, there are substantial improvements in the endmember estimation for 'PCA Square Euclidean' SAD values, with slight improvements in abundance SAD and a slight worsening of endmember SAD compared to their original values.

4.2. Jasper Ridge Dataset

Jasper Ridge is the second hyperspectral dataset that is going to be analyzed as it is popular in various HU studies. The data is captured via AVIRIS (Airborn visible/infrared imaging spectrometer) sensor from the Jet Propulsion Laboratory. The original dataset consists of 512×614 pixels with 224 bands. As there is no available ground truth for the whole surface of this hyperspectral image, a sub-image of 100×100 will be considered. Moreover, due to atmospheric effects and dense water vapor, spectral bands 1-3, 108-112, 154-266, and 220-224 have been removed which remains 198 bands for this study. The false color composite of this dataset is shown in Figure 4.1b. The Jasper Ridge Dataset consists of 4 endmembers, namely: 'tree', 'dirt', 'water', and 'road'. Further details can be found in (Zhu, 2017; Zhu, Wang, Xiang, et al., 2014).

4.2.1. Abundace solutions

Table 4.5 shows the optimal hyperparameter tuning values for the weighted and hard constraints under each case. The illustration of the abundance maps with their optimal setting for WC and HC are shown in Figure 4.10 and 4.11. Furthermore, the RMSE and SAD values obtained from QP are presented in Table 4.6 and 4.7 respectively.

	Parameters	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Q_{c1}	-	-	0.1	0.1	1
	Q_{c2}	-	-	-	1000	1
	μ	-	-	-	1	0.8
	q	-	-	-	2	0.9
HC	Q_{c1}	-	-	0.1	0.1	0.1
	Q_{c2}	-	-	-	0.1	0.1
	μ	-	-	-	1	1
	q	-	-	-	2	0.9

Table 4.5: Weighted Constraints (WC) and Hard Constraints (HC) optimal parameter values via hyperparameter tuning for Jasper Ridge Dataset

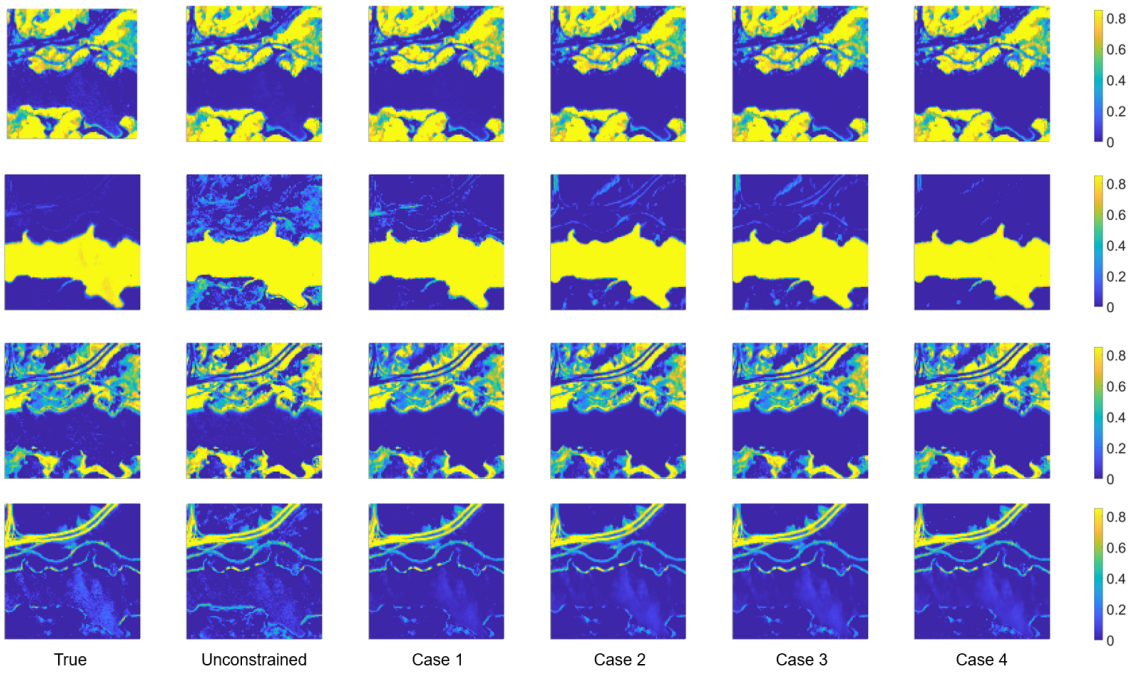


Figure 4.10: Weighted-constrained abundance maps for Jasper Ridge Dataset with true and unconstrained abundance maps. Rows from top to bottom represent tree, water, dirt and road respectively.

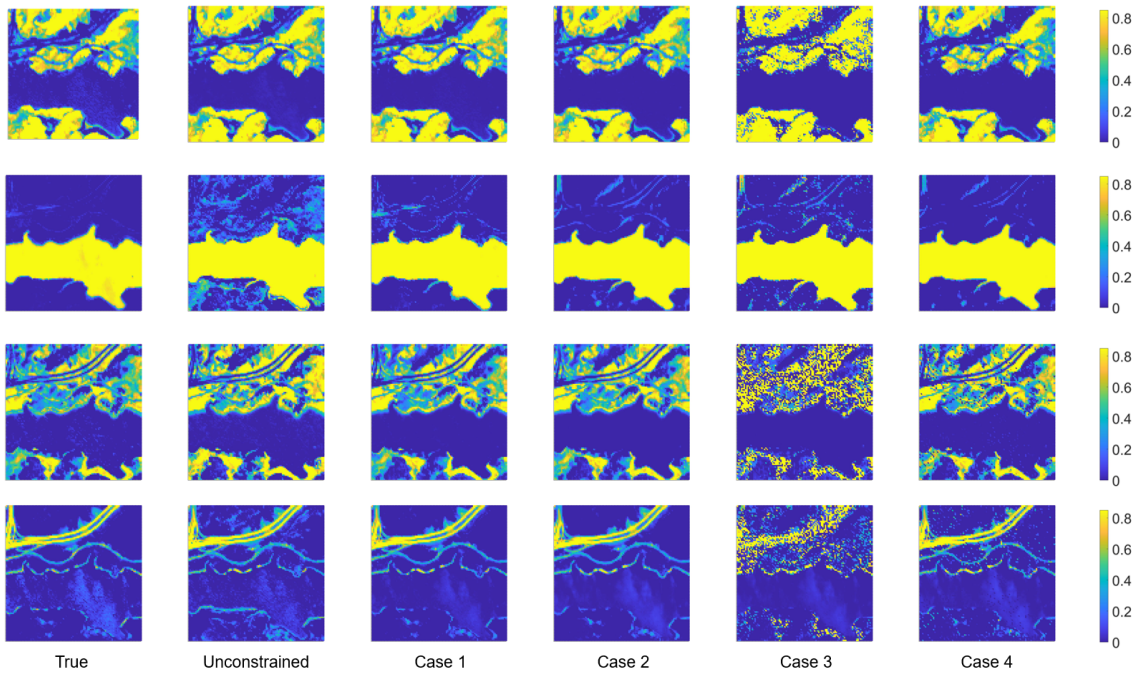


Figure 4.11: Hard-constrained abundance maps for Jasper Ridge Dataset with true and unconstrained abundance maps. Rows from top to bottom represent tree, water, dirt and road respectively.

	Endmember	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Tree	0.1332	0.0874	0.0574	0.0515	0.0658
	Water	0.2337	0.1172	0.0819	0.0819	0.0518
	Dirt	0.1726	0.0642	0.0703	0.0720	0.0789
	Road	0.1213	0.0482	0.0585	0.0581	0.0528
	WC Average	0.1652	0.0793	0.0670	0.0659	**0.0623**
HC	Tree	0.1332	0.0874	0.0574	0.1862	0.0641
	Water	0.2337	0.1172	0.0819	0.1437	0.0695
	Dirt	0.1726	0.0642	0.0703	0.2844	0.1141
	Road	0.1213	0.0482	0.0585	0.2154	0.0858
	HC Average	0.1652	0.0793	**0.0670**	0.2074	0.0833

Table 4.6: Weighted constraint (WC) and Hard Constrained (HC) RMSE values for each case under optimal parameter settings with Jasper Ridge Dataset

	Endmember	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Tree	9.9748°	7.5454°	5.9926°	5.7490°	6.9218°
	Water	19.8804°	9.1917°	7.6856°	7.6886°	5.2339°
	Dirt	16.9321°	8.6099°	9.3201°	9.8059°	10.6867°
	Road	26.9809°	12.2069°	14.7565°	14.7203°	13.2609°
	WC Average	18.4421°	9.3885°	9.4387°	9.4910°	**9.0258°**
HC	Tree	9.9748°	7.5454°	5.9926°	18.3050°	7.2957°
	Water	19.8804°	9.1917°	7.6856°	13.2120°	6.5636°
	Dirt	16.9321°	8.6099°	9.3201°	41.0465°	15.6960°
	Road	26.9809°	12.2069°	14.7565°	47.7749°	20.8773°
	HC Average	18.4421°	**9.3885°**	9.4387°	30.0846°	12.6081°

Table 4.7: Weighted Constrained (WC) and Hard Constrained (HC) SAD values for each case under optimal parameter settings with Jasper Ridge Dataset

From table 4.5, 4.6 and 4.7, one can see that for the WC case, case 4 produces the best result in terms of RMSE and SAD, and as for the HC case, case 2 leads to the lowest RMSE values and case 1 produces the lowest SAD. This solution improvement is also translated into the abundance map, as in endmember 'dirt' there are direct improvements across various cases.

By taking a closer look at case 3, it can be seen that the value Q_{c2}, which corresponds to the weight of the sparsity constraint, does not improve the solution as much for the WC case, and even produces worse results in the HC case compared to case 2. This indicates that sparsity constraint and sum-to-one constraint share similar properties when being considered together (Bruckstein et al., 2008, Qian et al., 2011).

4.2.2. Endmember matrix estimations

The endmember matrix estimation exhibits greater variation when comparing solutions to the Samson Dataset. Figure 4.12 illustrates the endmember estimation using VCA and PCA methods, while Figure 4.13 demonstrates the clustering quality for the PCA methods. Table 4.8 provides the RMSE and SAD values for the endmember estimations.

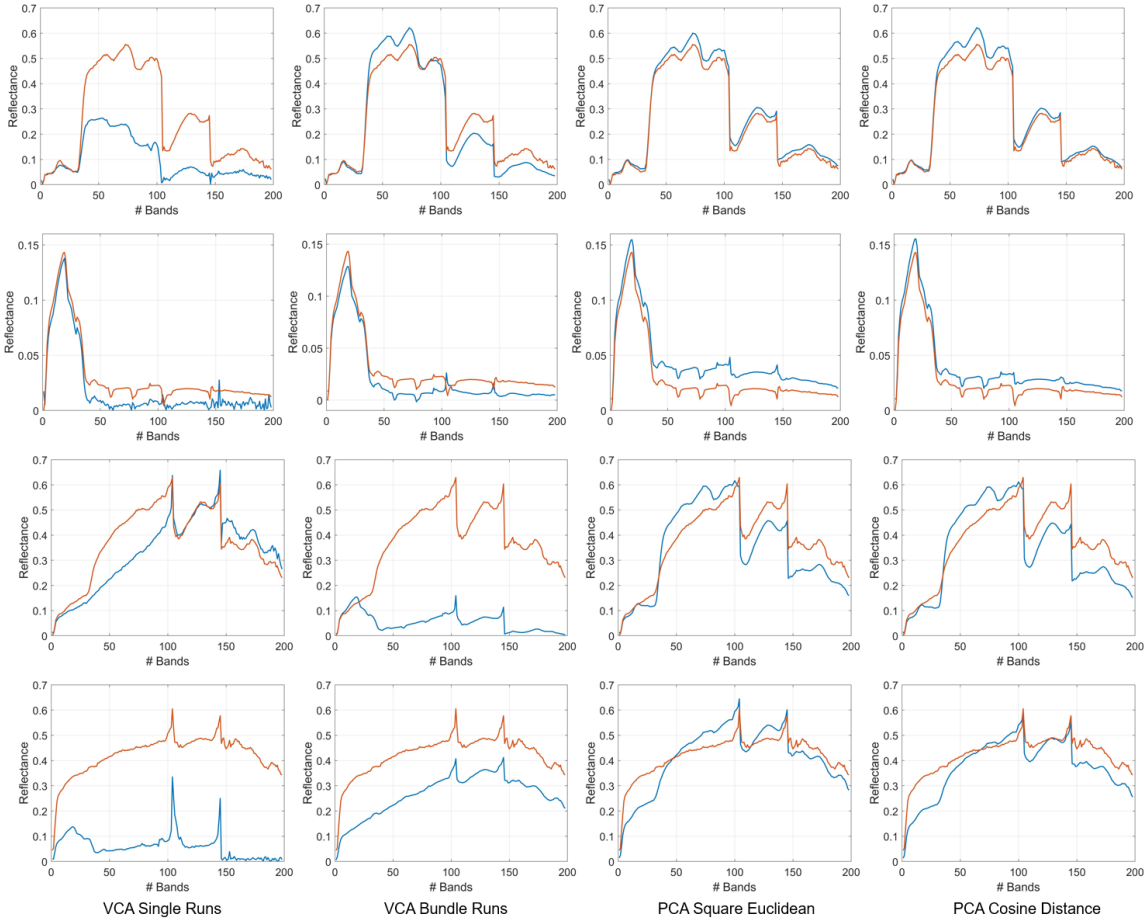
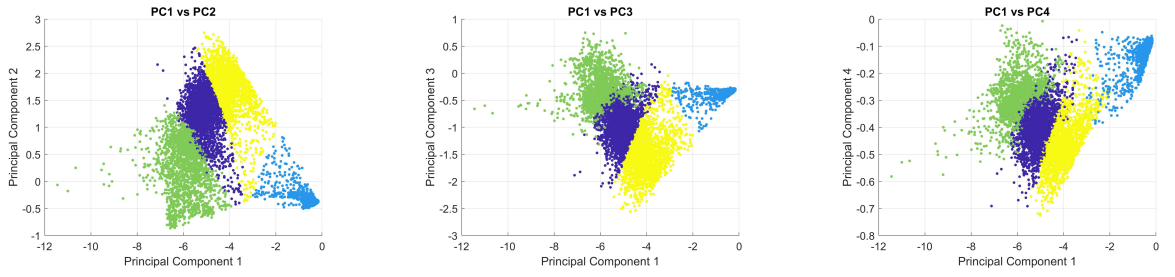


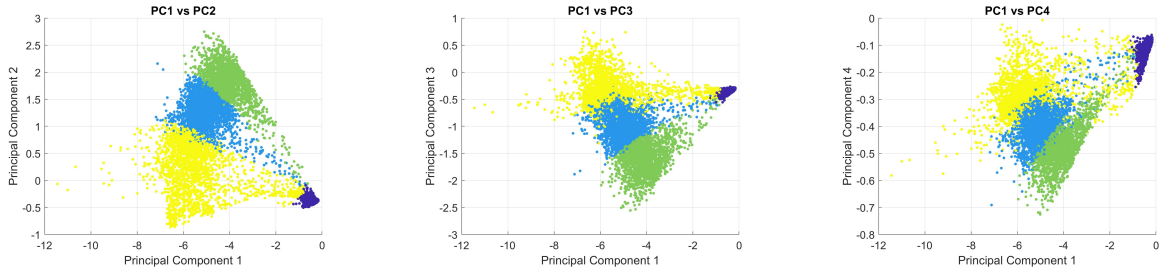
Figure 4.12: Estimated endmembers (blue) and true endmembers (red) for Jasper Ridge dataset. Rows from top to bottom represent tree, water, dirt and road respectively



(a) PC1 vs PC2 using squared euclidean metric for clustering

(b) PC1 vs PC3 using squared euclidean metric for clustering

(c) PC1 vs PC4 using squared euclidean metric for clustering



(d) PC1 vs PC2 using cosine distance metric for clustering

(e) PC1 vs PC3 using cosine distance metric for clustering

(f) PC1 vs PC4 using cosine distance metric for clustering

Figure 4.13: PCs visualization for Jasper Ridge Dataset, each color represents a potential endmember

	Endmember	VCA Single	VCA Bundle	PCA Square Euclidean	PCA Cosine
RMSE	Tree	0.1919	0.0530	0.0243	0.0308
	Water	0.0123	0.0115	0.0148	0.0095
	Dirt	0.0952	0.3508	0.0790	0.0834
	Road	0.3736	0.1600	0.0623	0.0695
	Average	0.1683	0.1438	**0.0451**	0.0483
SAD	Tree	17.2757°	9.4105°	1.6098°	1.1857°
	Water	14.5512°	13.1135°	12.0531°	6.9352°
	Dirt	13.2047°	39.9091°	11.4405°	12.0477°
	Road	37.9603°	7.9833°	8.3121°	8.0858°
	Average	20.7480°	17.6041°	8.3539°	**7.06336**°

Table 4.8: RMSE and SAD values for endmember matrix estimation (Jasper Ridge Dataset)

Based on Table 4.8, the 'PCA Square Euclidean' method produces the best result in terms of RMSE, while the 'PCA Cosine' method yields the best SAD value. 'VCA bundle runs' still produce negative values, particularly for the endmember 'water'. As previously mentioned, VCA methods are sensitive to illumination and atmospheric effects, which might lead to negative estimations for endmembers due to the properties of the dataset.

The PCA methods demonstrate remarkable results compared to the VCA methods, with nearly three times the improvement. Although there are no significant differences between RMSE and SAD values overall, a closer examination reveals that the 'PCA Cosine' method estimates the endmember 'water' especially well compared to the 'PCA Square Euclidean' method similar to the Samson Dataset. This is likely due to the cosine distance metric, which measures the similarity in orientation between vectors rather than their absolute differences. Since the spectral signature of water can vary due to factors like illumination and viewing angle, the cosine metric is more effective in capturing these variations, making 'PCA Cosine' a better choice when the dataset contains endmember 'water'.

4.2.3. WTLS solutions

The endmember simulation using WTLS for the Jasper Ridge Dataset with WC and HC settings are shown in Figures 4.14 and 4.15, respectively. Figures 4.16 and 4.17 illustrate how the RMSE and SAD values change when implementing different endmember extraction methods together with the QP solutions.

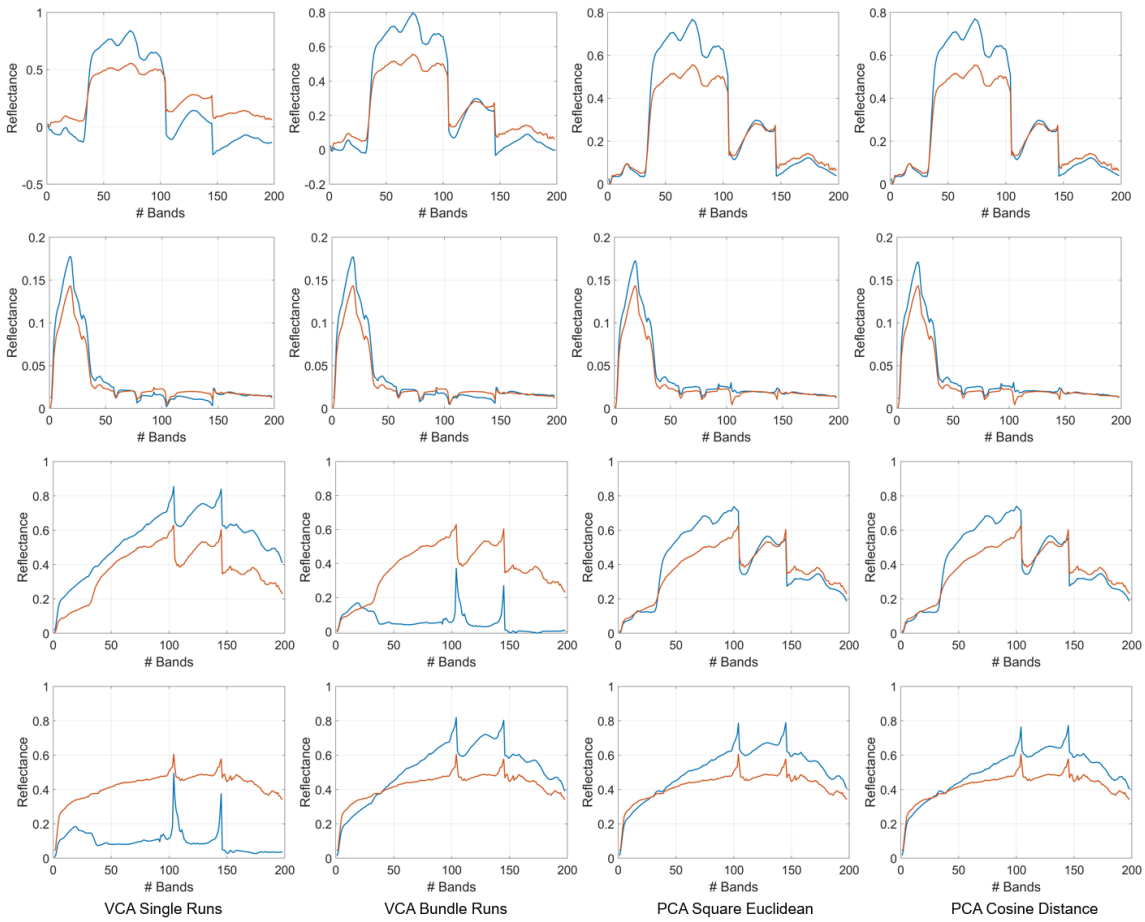


Figure 4.14: WTLS estimated endmembers (blue) and true endmembers (red) for Jasper Ridge dataset with WC settings. Rows from top to bottom represent tree, water, dirt and road respectively

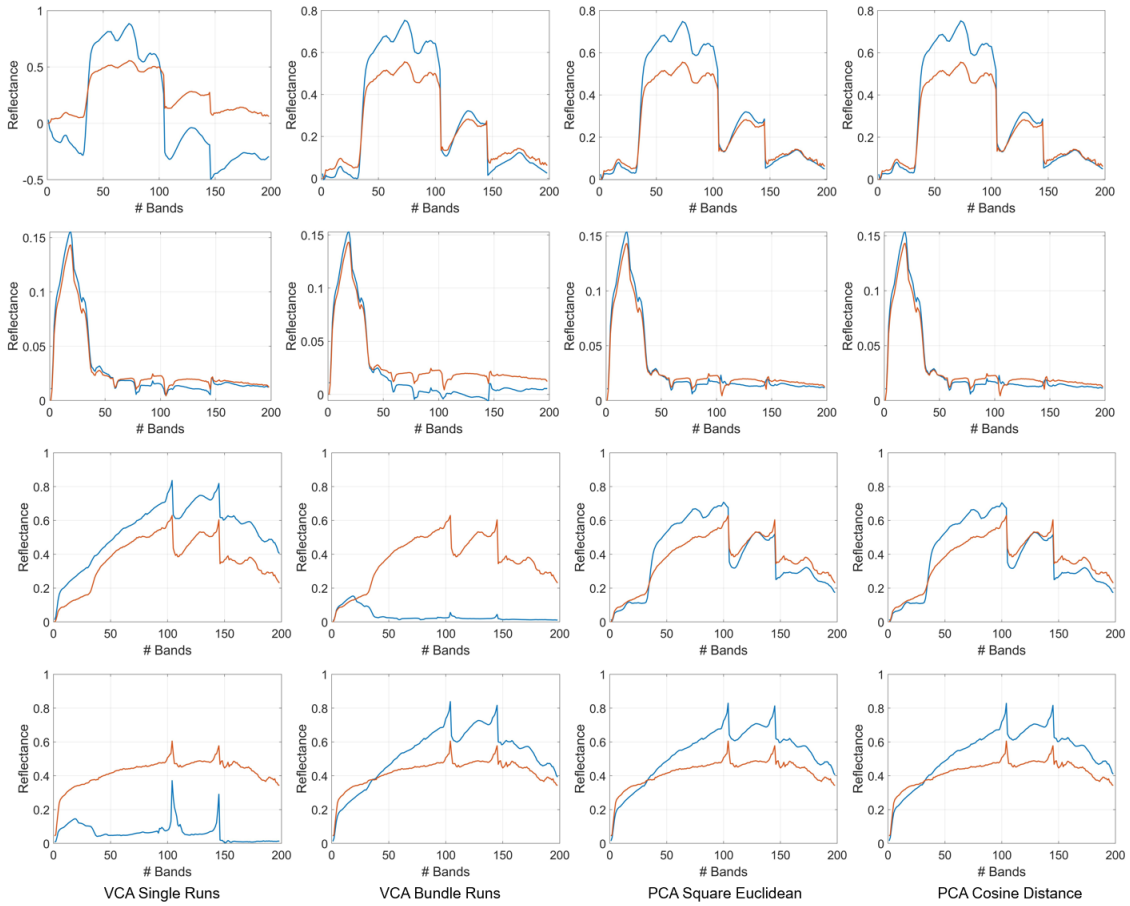


Figure 4.15: WTLS estimated endmembers (blue) and true endmembers (red) for Jasper Ridge dataset with HC settings. Rows from top to bottom represent tree, water, dirt and road respectively

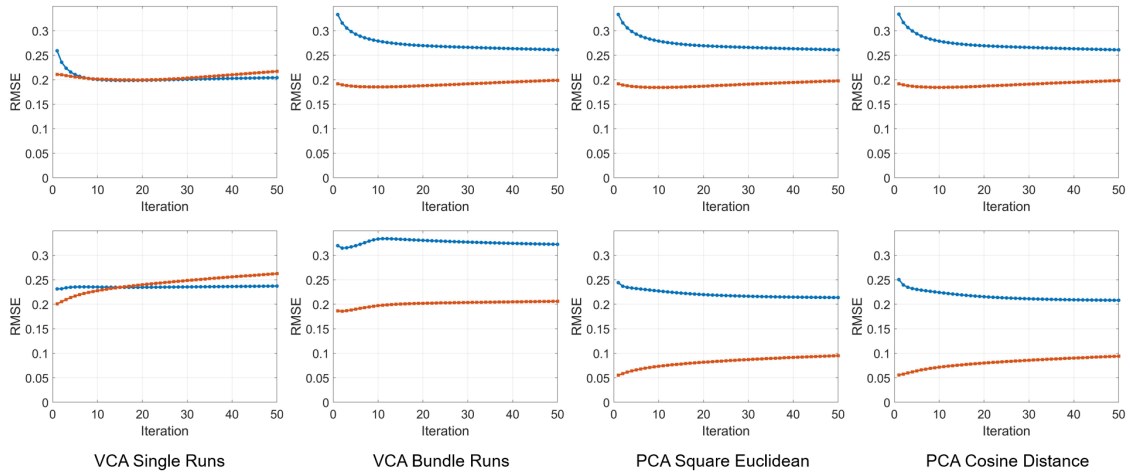


Figure 4.16: Abundance RMSE (blue) and endmember RMSE (red) comparison with different input endmember matrices using WTLS algorithm. The first row represents the RMSE after using WC abundance solutions, and the second row represents the RMSE after using HC abundance solutions from QP respectively. (Jasper Ridge Dataset)

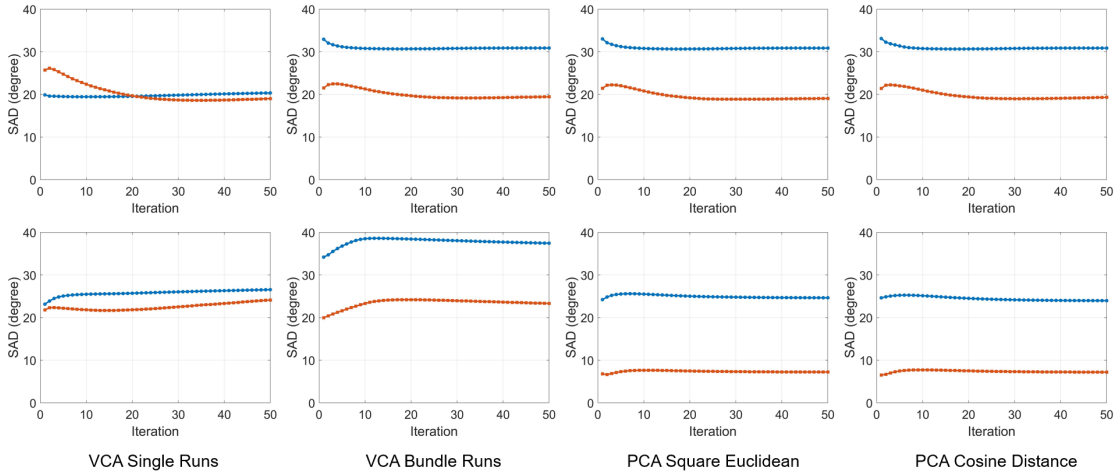


Figure 4.17: Abundance SAD (blue) and endmember SAD (red) comparison with different input endmember matrices using WTLS algorithm. The first row represents the SAD after using WC abundance solutions, and the second row represents the SAD after using HC abundance solutions from QP respectively. (Jasper Ridge Dataset)

From Figure 4.14, it can be seen that the VCA method produces negative values under WC settings. Since VCA does not perform well in endmember extraction, using WTLS worsens the solution, indicating that the WTLS algorithm is quite sensitive to the initial input. By applying HC constraints to the abundance solution, as shown in Figure 4.15, 'VCA Bundle Runs' produce positive values, but 'VCA Single Runs' still yield negative values for the endmember 'Tree'.

The PCA method remains positive for both WC and HC abundance solutions, but its performance deteriorates compared to its initial results from endmember extraction alone. This decline in solution quality is reflected in Figures 4.16 and 4.17, where RMSE and SAD values for both abundance and endmember solutions worsen compared to their original values. This is somewhat expected since introducing more errors to more variables can worsen the results.

Additionally, the relationship between abundance and endmember solutions still shows an inverse trend, however, in this case, the abundance values improve while the endmember solutions deteriorate in terms of RMSE values and in the first few iterations for the SAD values. This suggests that the endmember solutions are compensating for the abundance solutions until convergence is reached.

4.3. Urban Dataset

The Urban Dataset is the last hyperspectral dataset that will be investigated in this study. It is one of the most widely used hyperspectral images since its high spectral resolution together with its intricate patterns of land coverage. The data was recorded by the HYDICE (Hyperspectral Digital Image Collection Experiment) sensor at $2 \times 2m^2$ spacial resolution consisting of 307×307 pixels in October 1995, Copperas Cove, TX, US. The image consists of 210 spectral bands ranging from 400 - $2500nm$, providing a spectral resolution at a 10 nm scale. Due to atmospheric effects and dense water vapor, bands 1-4, 76,87,101-111,136-153, and 198-210 were removed leaving 162 bands for this experiment.

This dataset features four primary endmembers: 'asphalt', 'grass', 'tree', and 'roof', each representing a distinct surface type within the urban landscape. The false color composite for this dataset is illustrated in figure 4.1c. Further details of the dataset can be found in (Liu et al., 2011; Zhu, 2017).

4.3.1. Abundance solutions

Table 4.9 shows the optimal hyperparameter tuning values for the weighted and hard constraints for each case. The abundance maps with their optimal setting for WC and HC are shown in Figure 4.18 and 4.19. Lastly, Table 4.10 and 4.11 demonstrate the RMSE and SAD values obtained from QP respectively.

	Parameters	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Q_{c1}	-	-	10	10	10
	Q_{c2}	-	-	-	10	1000
	μ	-	-	-	1	1
	q	-	-	-	2	0.9
HC	Q_{c1}	-	-	0.1	0.1	0.1
	Q_{c2}	-	-	-	0.1	0.1
	μ	-	-	-	1	1
	q	-	-	-	2	0.9

Table 4.9: Weighted Constraints (WC) and Hard Constraints (HC) optimal parameter values via hyperparameter tuning for Urban Dataset

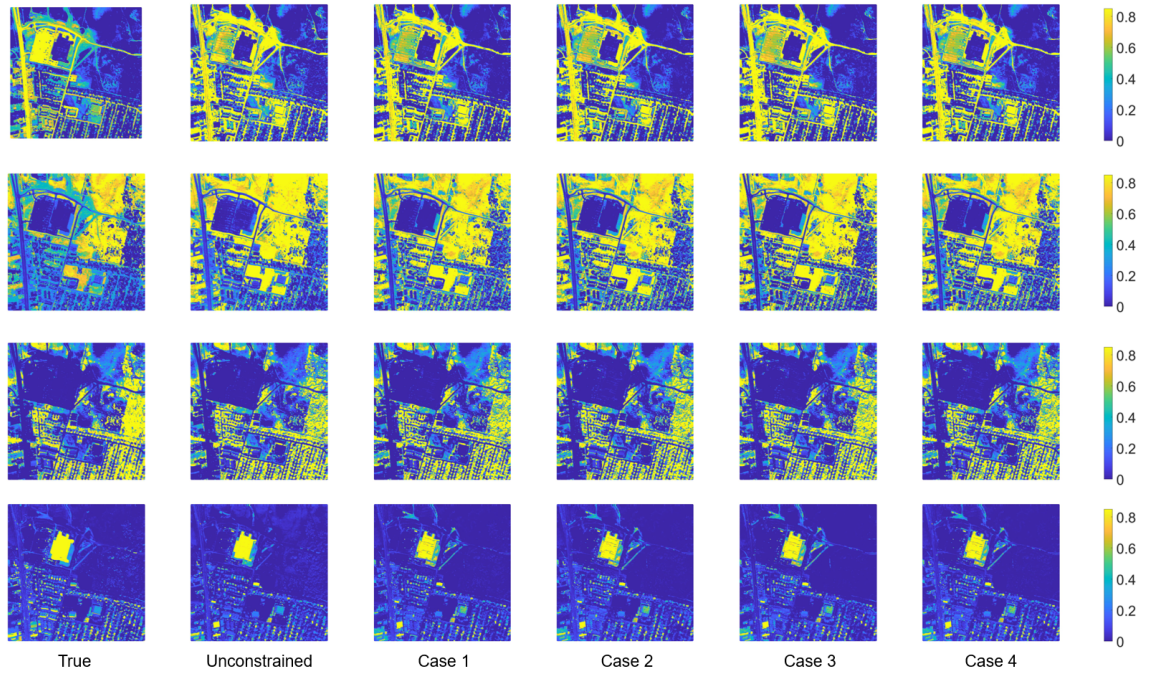


Figure 4.18: Weighthed-constrained abundance maps for Urban Dataset with true and unconstrained abundance maps. Rows from top to bottom represent asphalt, grass, tree and roof respectively.

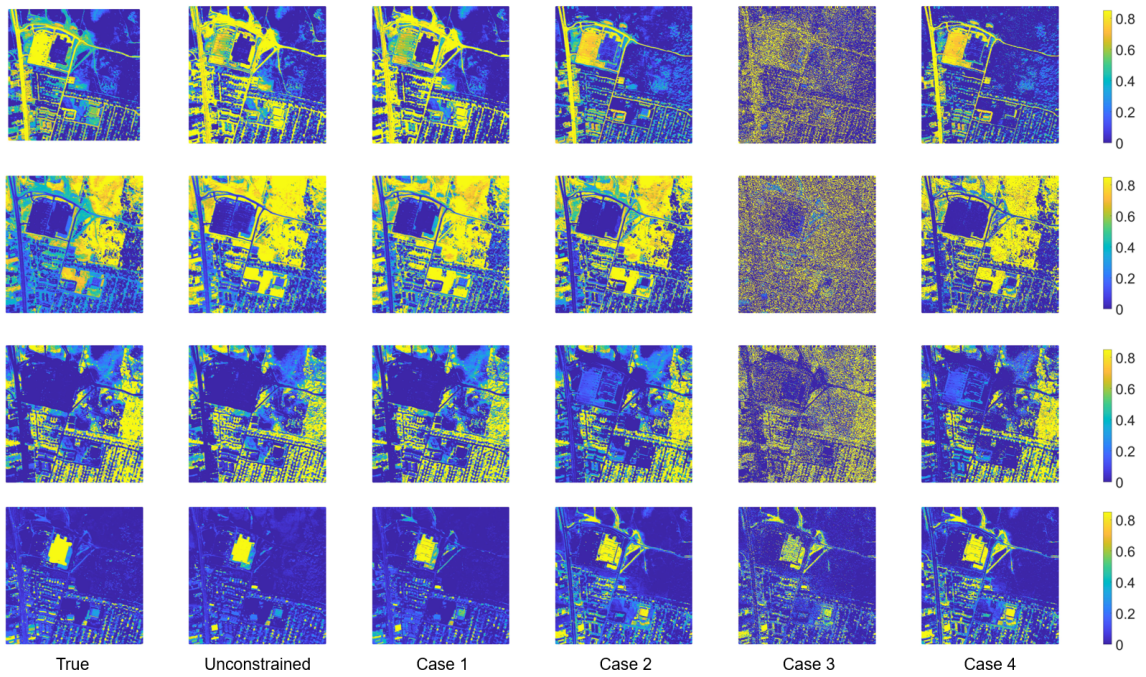


Figure 4.19: Hard-constrained abundance maps for Urban Dataset with true and unconstrained abundance maps. Rows from top to bottom represent asphalt, grass, tree and roof respectively.

	Endmember	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Asphalt	0.3963	0.2277	0.2152	0.1854	0.1892
	Grass	0.2003	0.1875	0.1990	0.1991	0.2046
	Tree	0.2518	0.1611	0.1481	0.1346	0.1403
	Roof	0.1472	0.1280	0.1318	0.1383	0.1339
	WC Average	0.2489	0.1761	0.1735	**0.1644**	0.1670
HC	Asphalt	0.3963	0.2277	0.2082	0.4343	0.2425
	Grass	0.2003	0.1875	0.1766	0.4989	0.2279
	Tree	0.2518	0.1611	0.1299	0.4147	0.1720
	Roof	0.1472	0.1280	0.2614	0.2842	0.2804
	HC Average	0.2489	**0.1761**	0.1940	0.4073	0.2307

Table 4.10: Weighted Constraint (WC) and Hard Constrained (HC) RMSE values for each case under optimal parameter settings with Urban Dataset

	Endmember	Unconstrained WLS	Case 1	Case 2	Case 3	Case 4
WC	Asphalt	33.5732°	24.1975°	23.1738°	19.9418°	21.4963°
	Grass	17.3710°	16.6498°	17.4633°	16.8690°	18.1270°
	Tree	27.6680°	20.4719°	19.0141°	17.4742°	18.3255°
	Roof	36.2513°	31.9628°	33.0220°	34.7200°	33.5475°
	WC Average	28.7159°	23.3205°	23.1683°	**22.2513**°	22.8740°
HC	Asphalt	33.5732°	24.1975°	29.8737°	54.9721°	35.4025°
	Grass	17.3710°	16.6498°	18.7891°	56.3311°	24.1759°
	Tree	27.6680°	20.4719°	16.9978°	49.0940°	22.7913°
	Roof	36.2513°	31.9628°	55.3782°	61.9204°	56.3308°
	HC Average	28.7159°	**23.3205**°	30.2497°	55.5794°	34.6751°

Table 4.11: Weighted Constrained (WC) and Hard Constrained (HC) SAD values for each case under optimal parameter settings with Urban Dataset

For the Urban Dataset, both the WC and HC cases reveal notable differences in performance, with case 3 excelling in the WC cases and case 1 outperforming in the HC cases. Although L_2 regularization promotes model generalization by penalizing the coefficients for the sparsity constraint, for all 3 datasets it produces the worst results (case 3) under WC cases. L_2 and L_1 (a special case of case 3) translate their straightforward penalization based on the current solution estimate after linearizing the constraint. The exploration of $L_{0<q<1}$ (case 4) induces sparsity constraint more aggressively, but it aims to leverage the benefits of both L_1 and L_2, as it allows for a more continuous and adjustable transition between the sparsity level of L_1 and smoothness of solution without enforcing too harshly on sparsity in L_2.

Most literature considers the sparsity constraint regularization in the objective function, where $L_{1/2}$ regularizer tends to be the optimal choice (Qian et al., 2011; X. Lu et al., 2013; Wang et al., 2016; Feng et al., 2022). However as they use different methods and our regularization is implemented in the constraint, the regularizer choice of the QP model is $L_{0.9}$ while enforcing sum-to-one and sparsity constraints to a certain level for all 3 datasets. The differences arise because the QP model tends to prefer simpler solutions to reduce overfitting. Conversely, it directly limits model parameters to a predefined domain where the parameter 'μ' was defined, enforcing specific structural properties like sparsity or bounds on coefficients.

4.3.2. Endmember matrix estimations

The superior performance of the PCA method for endmember estimation is further demonstrated using the Urban Dataset. Figure 4.20 illustrates the endmember estimation, while Figure 4.21 shows the clustering results. Table 4.12 provides the RMSE and SAD values for the endmember matrix estimation.

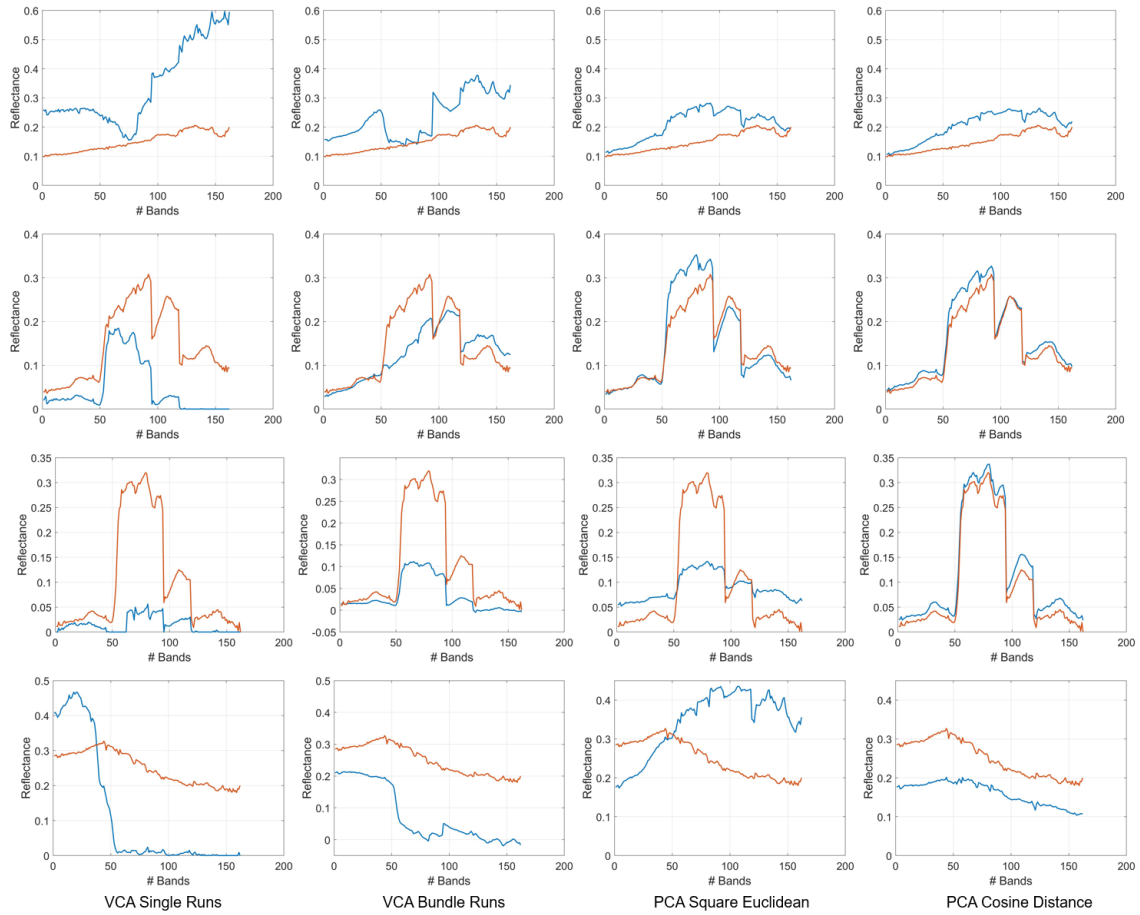
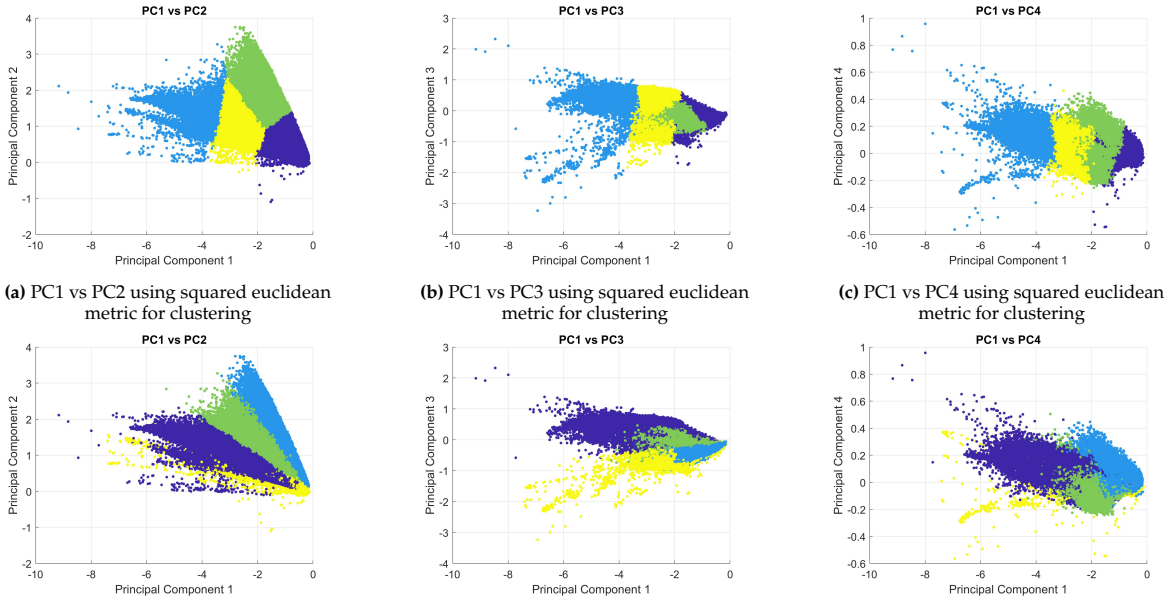


Figure 4.20: Estimated endmembers (blue) and true endmembers (red) for Urban dataset. Rows from top to bottom represent asphalt, grass, tree and roof respectively



(a) PC1 vs PC2 using squared euclidean metric for clustering

(b) PC1 vs PC3 using squared euclidean metric for clustering

(c) PC1 vs PC4 using squared euclidean metric for clustering

(d) PC1 vs PC2 using cosine distance metric for clustering

(e) PC1 vs PC3 using cosine distance metric for clustering

(f) PC1 vs PC4 using cosine distance metric for clustering

Figure 4.21: PCs visualization for Urban Dataset, each color represents a potential endmember

	Endmember	VCA Single	VCA Bundle	PCA Square Euclidean	PCA Cosine
RMSE	Asphalt	0.2194	0.1007	0.0712	0.0615
	Grass	0.1172	0.0568	0.0371	0.0212
	Tree	0.1308	0.0988	0.0837	0.0202
	Roof	0.1997	0.1830	0.1472	0.0925
	Average	0.1668	0.1098	0.0848	**0.0489**
SAD	Asphalt	13.0373°	9.8501°	10.7305°	7.0806°
	Grass	37.1779°	18.2019°	10.1899°	3.6495°
	Tree	30.3539°	9.9970°	30.7708°	5.2747°
	Roof	49.6782°	40.8743°	21.3841°	3.2529°
	Average	32.5618°	19.7308°	18.2688°	**4.8144°**

Table 4.12: RMSE and SAD values for endmember matrix estimation (Urban Dataset)

From Table 4.12, it is clear that the 'PCA Cosine' method produces the best values in terms of both RMSE and SAD. Remarkably, it achieves RMSE values that are twice as good as the 'PCA Square Euclidean' method and SAD values that are almost four times better. In contrast, the VCA methods yield relatively poor solutions compared to the PCA methods, with 'VCA bundle runs' still producing negative values.

A closer examination of the PCA methods reveals that 'PCA Cosine' excels at estimating the endmembers 'tree' and 'roof'. Given that the Urban dataset is relatively larger compared to the other two datasets, it becomes apparent that for larger datasets, 'PCA Cosine' tends to deliver superior solutions in terms of both RMSE and SAD compared to 'PCA Square Euclidean' and the VCA methods. For smaller datasets, the differences in magnitudes can be more accurately captured by the square Euclidean distance. However, as the size of the dataset increases, the orientation of the data points becomes more meaningful. While the RMSE values do not show significant differences for all three datasets, the SAD values are always better for the 'PCA Cosine'. Considering all trade off between the two PCA cases, 'PCA Cosine' would be a better choice when comes to endmember estimation, as it highlights the effectiveness of capturing the spectral characteristics more accurately when it comes to individual endmembers.

4.3.3. WTLS solutions

The results of the WTLS algorithm for endmember estimation on the Urban dataset are shown in Figure 4.22 for the WC setting and figure 4.23 for the HC setting. Figure 4.24 and Figure 4.25 correspond to the RMSE and SAD values for the abundance and endmember after implementing the WTLS algorithm using endmember extraction methods and QP solutions.

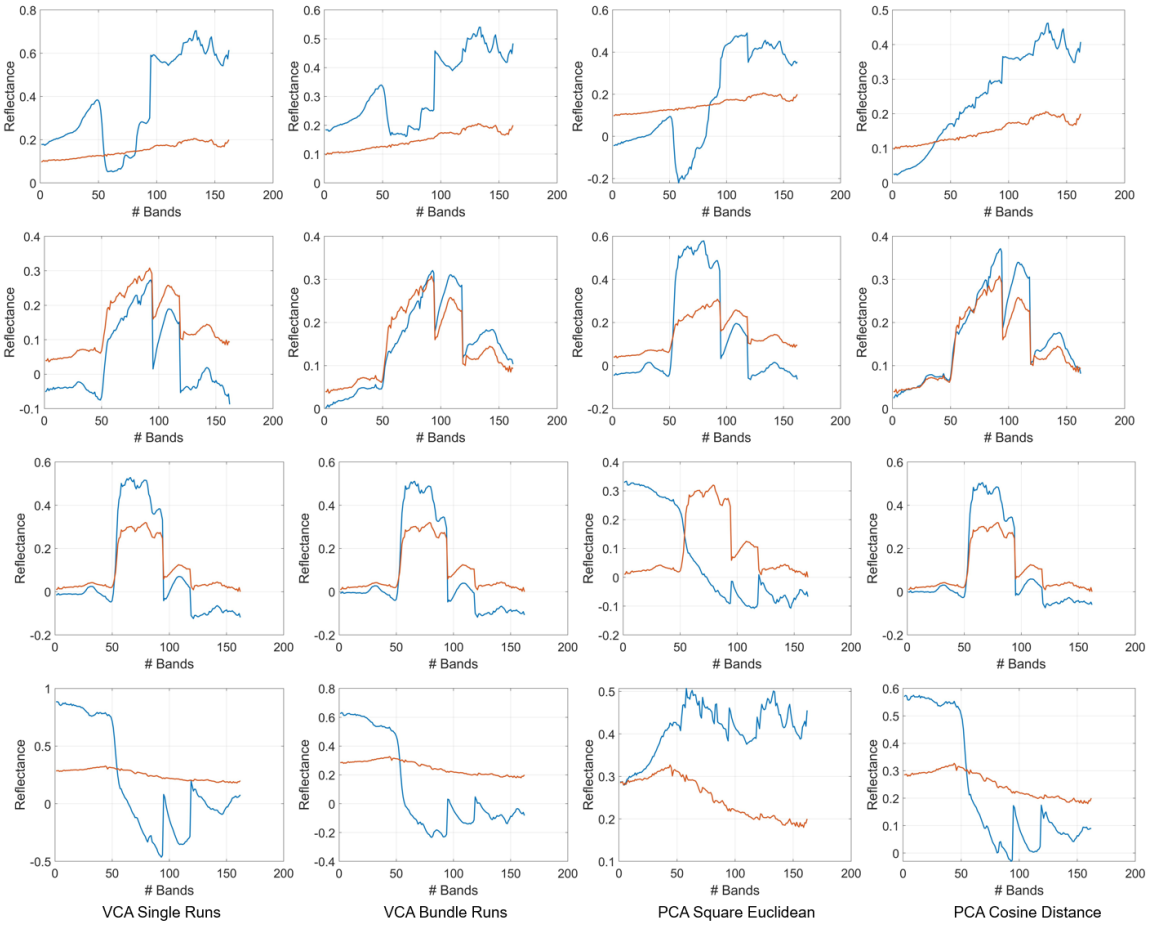


Figure 4.22: WTLS estimated endmembers (blue) and true endmembers (red) for Urban dataset with WC settings. Rows from top to bottom represent asphalt, grass, tree and roof respectively

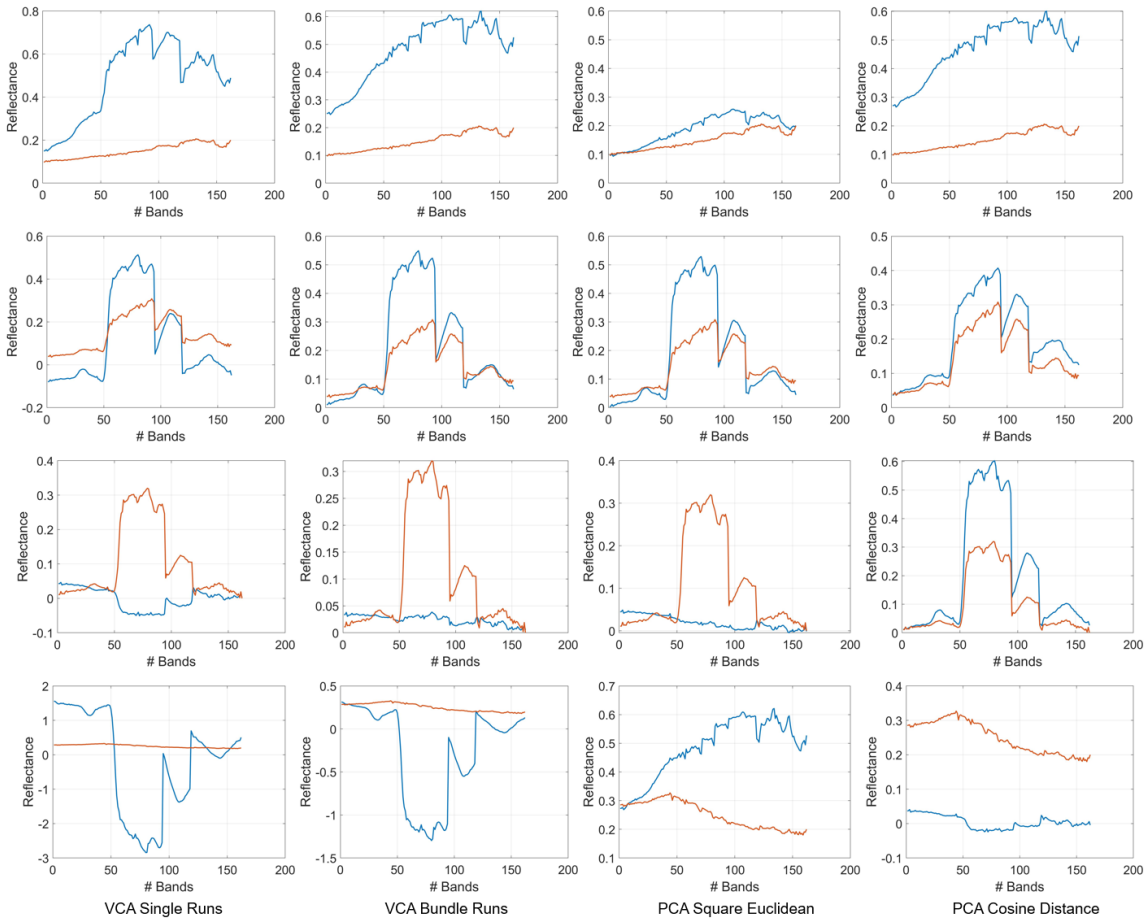


Figure 4.23: WTLS estimated endmembers (blue) and true endmembers (red) for Urban dataset with HC settings. Rows from top to bottom represent asphalt, grass, tree and roof respectively

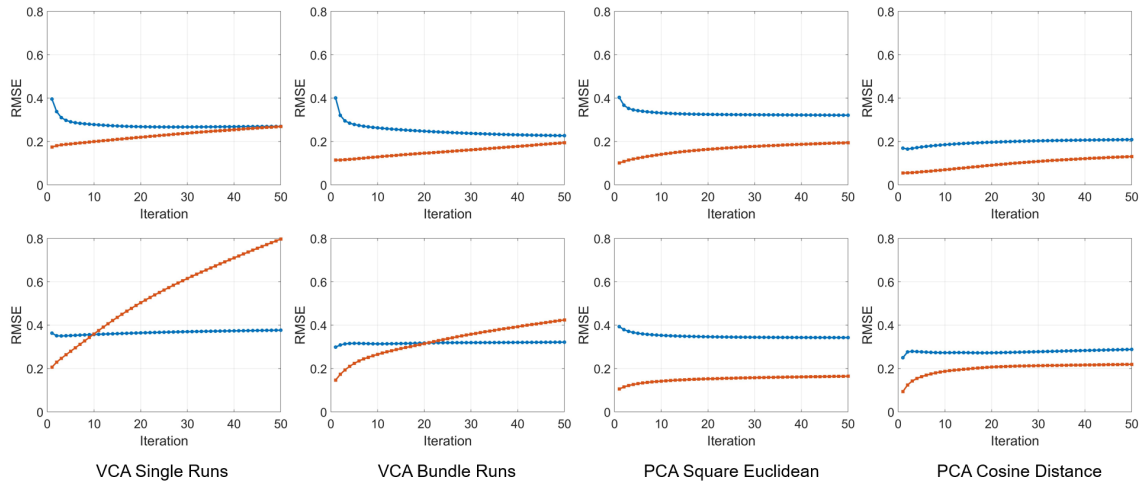


Figure 4.24: Abundance RMSE (blue) and endmember RMSE (red) comparison with different input endmember matrices using WTLS algorithm. The first row represents the RMSE after using WC abundance solutions, and the second row represents the RMSE after using HC abundance solutions from QP respectively. (Urban Dataset)

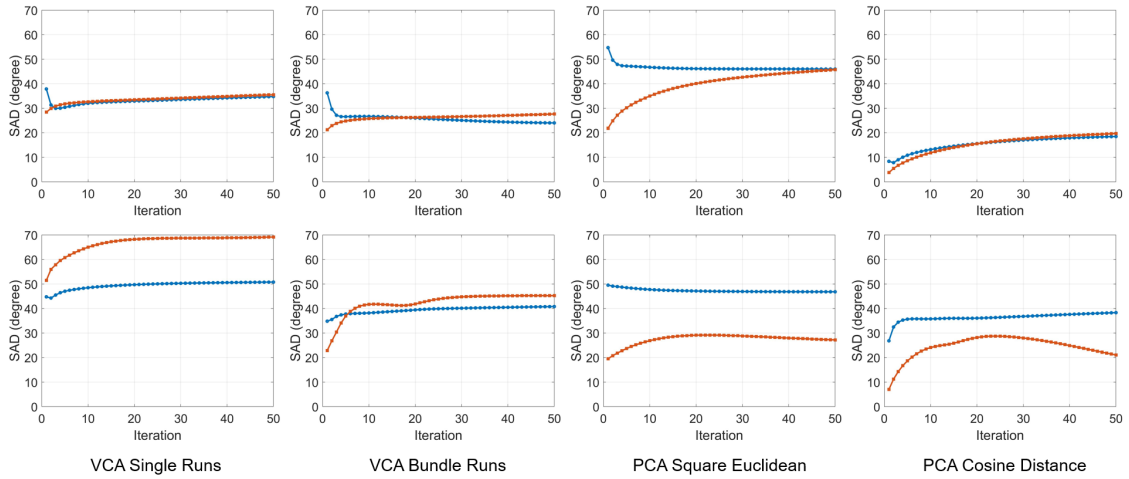


Figure 4.25: Abundance SAD (blue) and endmember SAD (red) comparison with different input endmember matrice using WTLS algorithms. The first row represents the SAD after using WC abundance solutions, and the second row represents the SAD after using HC abundance solutions from QP respectively. (Urban Dataset)

The WTLS simulations encounter significant difficulty when dealing with the Urban dataset. As shown in Figures 4.22 and 4.23, all results produce negative values for endmember estimation. This outcome highlights the sensitivity of the WTLS algorithm to the initial inputs, either the endmember matrix or the abundance matrix. Given that the Urban dataset is substantially larger than the other two datasets, Figure 4.20 clearly demonstrates the shortcomings of the VCA method in computing accurate endmember solutions.

Although the PCA method yields relatively good results for endmember estimation, 'PCA Square Euclidean' struggles to accurately estimate the endmembers 'tree' and 'roof'. The 'PCA Cosine Distance' provides the best results, which is reflected in the WTLS estimations. As seen in Figures 4.22 and 4.23, column four, the general shape of the endmembers remains recognizable, further indicating the sensitivity of the algorithm to initial inputs. However, the presence of negative values in the endmember solutions is physically meaningless, meaning that the WTLS algorithm failed to provide accurate estimations for the Urban dataset.

It is also interesting to note that since the WTLS algorithm itself does not incorporate constraints, for testing purposes, the parameters Q_{c1} and Q_{c2} were set to harder conditions. By reducing these values, it is possible to force the new endmember estimations using the WTLS algorithm to become positive. This suggests a potential research direction for further optimizing the dependent variables in the QP model to improve the robustness and accuracy of the WTLS algorithm's results.

4.3.4. Summary

The results obtained from the three datasets highlight various aspects of the HU process using the proposed methods. By comparing these datasets, one can identify trends and anomalies that provide insight into the performance and limitations of the models.

1. The QP model shows comparable results for all three datasets. Jasper Ridge Dataset showed the best performance when it came to abundance estimation in terms of RMSE and SAD. Although the expectation was that the smaller dataset with fewer endmember should perform the best solution, the Urban dataset with the largest data size did perform the worst out of the three. The WC cases outperform the HC cases for all three datasets. The flexibility provided by WC allows the model to better accommodate variations within the data, leading to more accurate abundance estimations. In contrast, HC can overly restrict the solution space, resulting in suboptimal solutions.
2. The PCA method outperforms the VCA method in both the Jasper Ridge and Urban datasets. The use of a cosine distance metric in PCA resulted in superior solutions in terms of SAD. While the differences in RMSE values were negligible for the Samson and Jasper Ridge datasets, the

cosine distance metric significantly outperformed both RMSE and SAD values for the Urban dataset, making it a preferable choice for endmember estimation. This performance underscores the robustness of the PCA method in capturing spectral characteristics more effectively than the VCA method.

3. The WTLS algorithm showed a high sensitivity to the initial inputs across all three datasets. Specifically, when the VCA method was used as the initial input, the algorithm tended to produce negative endmember values. In contrast, initializing with the PCA method generally resulted in positive endmember values for the Sasom and Jasper Ridge datasets. However, it only improved the solutions during certain iterations, indicating that the initial choice significantly influences the algorithm's performance.

5

Conclusion and Recommendations

This chapter concludes all the findings of this research, the responses to each research question will be provided, followed by recommendations for further studies.

5.1. Conclusion

How can least squares methods be used to accurately and efficiently estimate abundance and endmembers for hyperspectral unmixing?

In prior research, a QP model was developed based on the least squares theory, specifically for abundance estimation. Although there are existing studies that use least squares methods to solve the unmixing problem (Deborah et al., 2021; Heinz & Chang, 2001; Menon et al., 2016; Pu et al., 2015), they often do not consider nonlinear constraints such as the sparsity constraint, as well as how individual constraints affects the abundance solution. Our research incorporated these constraints by linearizing the non-linear constraint and adding regularization effects, optimizing distinct cases with their hyperparameters to achieve the best possible solution for abundance estimation. Moreover, a modified WTLS algorithm was proposed, extending the standard least squares theory. This algorithm iteratively computes abundance and endmember solutions using initial inputs from the QP model and the newly proposed PCA method. However, the algorithm's sensitivity to initial inputs limited its effectiveness in further enhancing the abundance and endmember solutions produced by the QP model and PCA method.

1. How does the solution differ when considering different constraints, especially when dealing with non-linear constraints for abundance estimation?

The inclusion of different constraints significantly impacts the quality of the abundance solutions. The performance of the QP model is data-dependent and also relies heavily on the configuration of the variance matrix, which determines whether the model applies WC or HC. The results indicate that WC generally outperforms HC across all datasets. WC offers more flexibility by not enforcing constraints as strictly as HC, allowing the model to accommodate variations in the data better. However, the results also highlight that incorporating non-linear sparsity constraints without regularization leads to a significant downgrade in solution quality. In contrast, when optimal hyperparameters are used to regulate these sparsity constraints, the overall solution quality improves, particularly in the WC case. This demonstrates that while non-linear constraints can enhance the model's performance, their effectiveness is highly dependent on appropriate regularization and hyperparameter tuning.

2. How do constrained least squares affect the estimation of abundances when compared to the unconstrained approach?

Constrained least squares methods generally provide more accurate abundance estimations compared to the unconstrained approach. This is shown in the results, with only a few exceptions noted in

the Samson Dataset. By incorporating constraints, the solution space is restricted to a more feasible region, reducing the likelihood of obtaining physically meaningless solutions. The results consistently show that WC cases yield lower RMSE and SAD values than the unconstrained least squares approach, highlighting the critical role of constraints in achieving reliable abundance estimations.

3. How does regularization affect abundance and endmember estimation?

Regularization plays a crucial role in abundance estimation by preventing overfitting and ensuring the stability of the solution. Introducing regularization terms directly into the sparsity constraint helps achieve sparser solutions. However, the choice of regularizer in this research differs from existing literature (Qian et al., 2011), which typically incorporates the regularization term in the objective function rather than the constraint itself. Moreover, the choice of tuning variables significantly impacts the solution, potentially leading to different optimal regularizers. Regularization does not directly influence endmember estimation methods like VCA or the new PCA method, as these methods depend primarily on the observation matrix rather than the abundance solution. However, when using iterative methods such as the proposed WTLS algorithm, regularization tends to improve the overall solution quality. This is because regularization enhances the accuracy of the abundance estimation, which in turn improves the endmember estimation in iterative methods.

4. How does the endmember matrix influence the abundance matrix estimation? Can we develop a fast and accurate method for endmember estimation?

The quality of the endmember matrix directly influences the accuracy of abundance estimation. In this research, the datasets provided included known endmember matrices, but in practical scenarios, these matrices are often unknown. This underscores the necessity of accurate endmember extraction methods to ensure reliable abundance estimations. This research also demonstrated a newly proposed PCA method specifically for endmember extraction. The quality of solutions obtained using the PCA method surpassed from the well-known VCA method. By combining PCA and k-means clustering, the new method is both fast and accurate, benefiting from being computationally efficient. Furthermore, the choice of evaluation metric significantly impacts solution quality. The results showed that using the cosine distance metric leads to better solutions, especially with larger datasets.

5. How can the abundance and endmember solutions be further enhanced?

There are several aspects that can improve the accuracy of abundance and endmember solutions. One key aspect is the quality of the dataset itself. Having a sensor with a higher resolution can lead to a more accurate observation matrix, which in turn enhances the overall unmixing process. In this research, a WTLS algorithm was proposed aiming at enhancing both abundance and endmember estimations. However, the results indicated that the WTLS algorithm is highly sensitive to initial inputs. While it can improve solutions to a certain extent, in some cases, it can produce worse results than the initial input.

5.2. Recommendations

There are several aspects that can be further explored in this research, recommendations for further investigation are summarised below.

Constraints and regularization investigation for the QP model

The regularization term should be applied not only to the sparsity constraint but can also be applied to the sum-to-one constraint. Evaluation of the performance impact of adding regularization terms to the objective function could also provide valuable insights. Additionally, more advanced constraints can be considered for potential enhancement of the model's robustness. Developing an adaptive constraint mechanism that can dynamically adjust based on the dataset characteristics is another promising area. This could involve real-time adjustment of constraint parameters to better fit varying data conditions, which can improve the flexibility and generalizability of the QP model.

Parameter tuning improvement for the QP model

While this research used a grid search method for hyperparameter tuning, future studies could consider using smaller grid steps to achieve more precise optimization. Additionally, investigating the relationships between variables could help identify which parameters have the most significant impact on optimization, allowing for more targeted adjustments. Moreover, adopting more sophisticated hyperparameter tuning techniques, such as Bayesian optimization or genetic algorithm, could lead to more efficient and accurate identification of optimal parameter settings for abundance estimation.

Additional metrics evaluation for the PCA method

The current research only considered two evaluation metrics. Exploring different metrics such as Minkowski distance (De Amorim, 2012), which is a generalization of both Euclidean and Manhattan distances, can better capture the geometry of the data. Additionally, Mutual Information (Knops et al., 2006) can be used to measure the amount of information obtained from one variable to another, capturing non-linear dependencies between spectral bands.

Replacement of k-means clustering with spectral clustering for the PCA method

Instead of using k-means clustering, employing spectral clustering might be a suitable choice. Spectral clustering uses the eigenvalues of the similarity matrix of the data to perform dimensional reduction before clustering (Zhao et al., 2019). This technique is particularly effective for handling more complex data and can potentially lead to better separation of endmembers.

WTLS algorithm improvement

As discussed in the results section, although the WTLS algorithm did not meet its expectations for solution enhancement, there is still significant potential in this algorithm. The lack of constraints led to negative values in the WTLS solutions. Enforcing constraints to ensure physically meaningful solutions should be considered. Subsequently, hyperparameter tuning can be performed in conjunction with the QP model parameters to obtain the most optimal settings.

Original WTLS algorithm investigation

The proposed WTLS was an adjusted version of the original algorithm (Gholinejad & Amiri-Simkooei, 2023). Future research could explore using the full process of the original WTLS algorithm, which iteratively produces abundance and endmember solutions rather than relying strictly on the initial input. Alternatively, finding more robust initialization techniques for the adjusted WTLS algorithm could also be a viable option.

Dataset investigation

Further investigation into the real datasets could provide deeper insights and improvement for the model. Techniques such as applying Fourier transform within the frequency and spatial domain followed by noise reduction can be explored. This approach can potentially help in identifying and mitigating various types of noises and distortions presented in the hyperspectral dataset. Additionally, studying the effects of different types of preprocessing sets such as normalization can lead to more accurate and reliable solutions.

References

Altmann, Y., Halimi, A., Dobigeon, N., & Tourneret, J. (2012). Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery. *IEEE Transactions on Image Processing*, *21*(6), 3017–3025. https://doi.org/10.1109/TIP.2012.2187668

Amiri-Simkooei, A. R. (2019). Unified least-squares formulation of a linear model with hard constraints. *Journal of Surveying Engineering-ASCE*. https://doi.org/https://doi.org/10.1061/(ASCE)SU.1943-5428.0000290

Amiri-Simkooei, A. R. (2017). Weighted total least squares with singular variance matrices subject to weighted and hard constraints. *Journal of Surveying Engineering-ASCE*, *143*(4). https://doi.org/10.1061/(ASCE)SU.1943-5428.0000239

Amiri-Simkooei, A. R., & Jazaeri, S. (2012). Weighted total least squares formulated by standard least squares theory. *Journal of Geodetic Science*, *2*(2), 113–124. https://doi.org/10.2478/v10156-011-0036-5

Amiri-Simkooei, A. R., Snellen, M., & Simons, D. G. (2011). Principal component analysis of single-beam echo-sounder signal features for seafloor classification. *IEEE Journal of Oceanic Engineering*, *36*(2), 259–272. https://doi.org/10.1109/JOE.2011.2122630

Audebert, N., Saux, B. L., & Lefèvre, S. (2019). Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geoscience and Remote Sensing Magazine*, *7*(2), 159–173. https://doi.org/10.1109/MGRS.2019.2912563

Bhatt, J. S., & Joshi, M. V. (2020). Deep learning in hyperspectral unmixing: A review. *IEEE International Symposium on Geoscience and Remote Sensing*. https://doi.org/10.1109/IGARSS39084.2020.9324546

Bioucas-Dias, J. M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., & Chanussot, J. (2012). Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *5*(2), 354–379. https://doi.org/https://doi.org/10.48550/arXiv.1202.6294

Boardman, J. W. (1993). Automating spectral unmixing of aviris data using convex geometry concepts. *JPL, Summaries of the 4th Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*.

Bruckstein, A. M., Elad, M., & Zibulevsky, M. (2008). On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Transactions on Information Theory*, *54*(11), 4813–4820. https://doi.org/10.1109/TIT.2008.929920

Chang, C. I. (2003). *Hyperspectral imaging: Techniques for spectral detection and classification*. Springer. https://doi.org/https://doi.org/10.1007/978-1-4419-9170-6

De Amorim, R. C. (2012). Constrained clustering with minkowski weighted k-means. *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*. https://doi.org/10.1109/CINTI.2012.6496753

Deborah, H., Ulfarsson, M. O., & Sigurdsson, J. (2021). Fully constrained least squares linear spectral unmixing of the scream (verso, 1893) [Conference date: March 24, 2021]. *IEEE Whispers*. https://doi.org/10.1109/WHISPERS52202.2021.9484037

Dixit, A., & Agarwal, S. (2021). Non-linear spectral unmixing of hyperspectral data using modified ppnmm. *Applied Computing and Geosciences*, *9*, 100053. https://doi.org/10.1016/j.acags.2021.100053

Dobigeon, N., Tourneret, J., Richard, C., Bermudez, J. C. M., McLaughlin, S., & Hero, A. O. (2014). Nonlinear unmixing of hyperspectral images: Models and algorithms. *IEEE Signal Processing Magazine*, *31*(1), 82–94. https://doi.org/10.1109/MSP.2013.2279274

Du, Q., Raksuntorn, N., Younan, N. H., & King, R. L. (2008). Variants of n-findr algorithm for endmember extraction. *Proceedings of SPIE, the International Society for Optical Engineering*. https://doi.org/10.1117/12.799361

Feng, X., Li, H., Wang, R., Du, Q., Jia, X., & Plaza, A. (2022). Hyperspectral unmixing based on nonnegative matrix factorization: A comprehensive review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 4414–4436. https://doi.org/10.1109/JSTARS.2022.3175257

Gholinejad, S., & Amiri-Simkooei, A. (2024). Combinatorial non-negative matrix-tensor factorization for hyperspectral unmixing using a general ℓ_q norm regularization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1–16. https://doi.org/10.1109/JSTARS.2024.3392497

Gholinejad, S., & Amiri-Simkooei, A. R. (2023). Multivariate weighted total least squares based on the standard least-squares theory. *Journal of Surveying Engineering-ASCE*, *149*(4). https://doi.org/https://doi.org/10.1061/JSUED2.SUENG-1424

Golub, G. H., & Van Loan, C. F. (1980). An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, *17*(6), 883–893. https://doi.org/10.1137/0717073

Guerri, M. F., Distante, C., Spagnolo, P., Bougourzi, F., & Taleb-Ahmed, A. (2024). Deep learning techniques for hyperspectral image analysis in agriculture: A review. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, *12*, 100062. https://doi.org/10.1016/j.ophoto.2024.100062

Heinz, D. C., & Chang, C. I. (2001). Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *39*(3), 529–545. https://doi.org/10.1109/36.911111

Heylen, R., Parente, M., & Gader, P. (2014). A review of nonlinear hyperspectral unmixing methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(6), 1844–1868. https://doi.org/10.1109/JSTARS.2014.2320576

Jolliffe, I. T. (2002). *Principal component analysis*. Springer. https://doi.org/10.1007/b98835

Keshava, N., & Mustard, J. F. (2002). Spectral unmixing. *IEEE Signal Processing Magazine*, *19*(1), 44–57. https://doi.org/10.1109/79.974727

Khoshsokhan, S., Rajabi, R., & Zayyani, H. (2019). Clustered multitask non-negative matrix factorization for spectral unmixing of hyperspectral data. *Journal of Applied Remote Sensing*, *13*(02), 1. https://doi.org/10.1117/1.JRS.13.026509

Knops, Z., Maintz, J., Viergever, M., & Pluim, J. (2006). Normalized mutual information based registration using k-means clustering and shading correction. *Medical Image Analysis*, *10*(3), 432–439. https://doi.org/10.1016/j.media.2005.03.009

Liu, X., Xia, W., Wang, B., & Zhang, L. (2011). An approach based on constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(2), 757–772. https://doi.org/10.1109/TGRS.2010.2068053

Lu, G., & Fei, B. (2014). Medical hyperspectral imaging: A review. *Journal of Biomedical Optics*, *19*(1), 010901. https://doi.org/10.1117/1.JBO.19.1.010901

Lu, X., Dong, L., & Yuan, Y. (2020). Subspace clustering constrained sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, *58*(5), 3007–3019. https://doi.org/10.1109/TGRS.2019.2946751

Lu, X., Wu, H., Yuan, Y., Yan, P., & Li, X. (2013). Manifold regularized sparse nmf for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, *51*(5), 2815–2826. https://doi.org/10.1109/TGRS.2012.2213825

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *1*, 281–297.

Menon, V., Du, Q., & Fowler, J. E. (2016). Random-projection-based nonnegative least squares for hyperspectral image unmixing. *IEEE Xplore*. https://doi.org/10.1109/WHISPERS.2016.8071796

Nascimento, J. V. D., & Dias, J. (2005). Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, *43*(4), 898–910. https://doi.org/10.1109/TGRS.2005.844293

Neville, R. A., Lévesque, J., Staenz, K., Nadeau, C., Hauff, P., & Borstad, G. A. (2003). Spectral unmixing of hyperspectral imagery for mineral exploration: Comparison of results from sfsi and aviris. *Canadian Journal of Remote Sensing*, *29*(1), 99–110. https://doi.org/10.5589/m02-085

Paoletti, M. E., Haut, J. M., Plaza, J., & Plaza, A. (2019). Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *158*, 279–317. https://doi.org/10.1016/j.isprsjprs.2019.09.006

Pu, H., Chen, Z., Wang, B., & Xia, W. (2015). Constrained least squares algorithms for nonlinear unmixing of hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(3), 1287–1303. https://doi.org/10.1109/TGRS.2014.2336858

Qian, Y., Jia, S., Zhou, J., & Robles-Kelly, A. (2011). Hyperspectral unmixing via $l_1/2$ sparsity-constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(11), 4282–4297. https://doi.org/10.1109/TGRS.2011.2144605

Schaffrin, B., & Wieser, A. (2007). On weighted total least-squares adjustment for linear regression. *Journal of Geodesy*, *82*(7), 415–421. https://doi.org/10.1007/s00190-007-0190-9

Shen, Y., Li, B., & Yi, C. (2010). An iterative solution of weighted total least-squares adjustment. *Journal of Geodesy*, *85*(4), 229–238. https://doi.org/10.1007/s00190-010-0431-1

Sinaga, K. P., & Yang, M. (2020). Unsupervised k-means clustering algorithm. *IEEE Access*, *8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Steinley, D. (2008). Stability analysis in k-means clustering. *British Journal of Mathematical and Statistical Psychology*, *61*(2), 255–273. https://doi.org/10.1348/000711007X184849

Wang, W., Qian, Y., & Tang, Y. Y. (2016). Hypergraph-regularized sparse nmf for hyperspectral unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(2), 681–694. https://doi.org/10.1109/JSTARS.2015.2508448

Winter, M. E. (1999). N-findr: An algorithm for fast autonomous spectral end-member determination in hyperspectral data. *Proceedings of SPIE, the International Society for Optical Engineering*. https://doi.org/10.1117/12.366289

Ye, Y., & Tse, E. (1989). An extension of karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, *44*(1-3), 157–179. https://doi.org/10.1007/BF01587086

Zhao, Y., Yuan, Y., & Wang, Q. (2019). Fast spectral clustering for unsupervised hyperspectral image classification. *Remote Sensing*, *11*(4), 399. https://doi.org/10.3390/rs11040399

Zhu, F. (2017). Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey. http://export.arxiv.org/pdf/1708.05125

Zhu, F., Wang, Y., Fan, B., Xiang, S., Meng, G., & Chen, P. (2014). Spectral unmixing via data-guided sparsity. *IEEE Transactions on Image Processing*, *23*(12), 5412–5427. https://doi.org/10.1109/TIP.2014.2363423

Zhu, F., Wang, Y., Xiang, S., Fan, B., & Chen, P. (2014). Structured sparse method for hyperspectral unmixing. *Isprs Journal of Photogrammetry and Remote Sensing*, *88*, 101–118. https://doi.org/10.1016/j.isprsjprs.2013.11.014

6

Appendix A

Algorithm 2 Vertex Component Analysis (VCA)

Input: $n, \mathbf{Y} \equiv [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r]$
$\text{SNR}_{\text{th}} = 15 + 10\log_{10}(k)\text{dB}$
if SNR > SNR_{th} **then**
  $d := n$
  $\mathbf{X} := \mathbf{U}_d^{\mathrm{T}}\mathbf{Y}$ {$\mathbf{U}_d$ obtained by SVD}
  $\mathbf{u} := \text{mean}(\mathbf{X})$ {$\mathbf{u}$ is a $1 \times d$ vector}
  $[\mathbf{Z}]_{:,j} := [\mathbf{X}]_{:,j} / \left([\mathbf{X}]_{:,j}^T \mathbf{u} \right)$ {projective projection}
else
  $d := n - 1$
  $[\mathbf{X}]_{:,j} := \mathbf{U}_d^{\mathrm{T}} \left([\mathbf{Y}]_{:,j} - \overline{\mathbf{y}} \right)$ {$\mathbf{U}_d$ obtained by PCA}
  $c := \arg\max_{j=1\ldots n} \|[\mathbf{X}]_{:,j}\|$
  $\mathbf{c} := [c|c|\ldots|c]$ {$\mathbf{c}$ is a $1 \times n$ vector}
  $\mathbf{Z} := \begin{bmatrix} \mathbf{X} \\ \mathbf{c} \end{bmatrix}$
end if
$\mathbf{A} := [\mathbf{e}_u|0|\ldots|0]$ {$\mathbf{e}_u = [0, \ldots, 0, 1]^T$ and $\mathbf{A}$ is a $n \times n$ auxiliary matrix}
for $i := 1$ **to** n **do**
  $\mathbf{w} := \text{randn}(0, \mathbf{I}_n)$ {$\mathbf{w}$ is a zero-mean random Gaussian vector of covariance $\mathbf{I}_n$}
  $\mathbf{f} := \left(\left(\mathbf{I} - \mathbf{A}\mathbf{A}^{\#} \right) \mathbf{w} \right) / \left(\| \left(\mathbf{I} - \mathbf{A}\mathbf{A}^{\#} \right) \mathbf{w} \| \right)$ {$\mathbf{f}$ is a vector orthonormal to the subspace spanned by $[\mathbf{A}]_{:,1:i}$}

  $\mathbf{v} := \mathbf{f}^T\mathbf{Z}$
  $o := \arg\max_{j=1,\ldots,n} |[\mathbf{v}]_{:,j}|$ {find the projection extreme}
  $[\mathbf{A}]_{:,i} := [\mathbf{Z}]_{:,o}$
  $[\text{indices}]_i := o$ {stores the pixel index}
end for
if SNR > SNR_{th} **then**
  $\widehat{\mathbf{M}} := \mathbf{U}_d[\mathbf{X}]_{:,\text{indices}}$ {$\widehat{\mathbf{M}}$ is a $m \times n$ estimated mixing matrix}
else
  $\widehat{\mathbf{M}} := \mathbf{U}_d[\mathbf{X}]_{:,\text{indices}} + \overline{\mathbf{y}}$ {$\widehat{\mathbf{M}}$ is a $m \times n$ estimated mixing matrix}
end if
Output: $\widehat{\mathbf{M}}$