# Endmember Extraction from Hyperspectral Image

Zhifei Zhang
School of Mechanical, Industrial and Manufacturing Engineering
*zhanzhif@onid.orst.edu*

Yawei Zhang
School of Electrical Engineering and Computer Science
*zhanyawe@onid.orst.edu*

Bin Zhang
School of Electrical Engineering and Computer Science
*zhangbin@onid.orst.edu*

***Abstract*—For a single pixel in a hyperspectral image, its spectrum is a mixture of several spectra from different materials. Therefore, a hyperspectral image can be seen as highly mixed data. Thus the common problem is how to decompose these mixed pixels into endmembers and their corresponding proportions. Each endmember presents a material, and its proportion shows its percentage among other materials. We will apply a convex optimization way to achieve this goal. In the paper, a objective function with minimum volume constraint (MVC) is set up, and then the convexity of the objective function is analyzed . Gradient descent method is applied to solved this problem. In order to speed up the searching of optimal point, conjugate gradient method is used as well. Experiment results are given at last to illustrate the performance of the algorithm applied in this paper. Detailed comparison with vertex component analysis (VCA) is presented to prove the high efficiency of MVC.**

## I. Introduction

Nowadays, satellite imagery analysis plays an important role in different areas, such as mineral seeking, military surveillance, space exploration, etc. A long term problem which has bothered researchers is the wide existence of mixed pixels [1]. Mixed pixels include more than one type of material. A mixed pixel includes different spectra. The measured spectrum of a mixed pixel is called *endmembers*. In practice, one has to decompose the mixed pixels to get endmember signatures and corresponding proportions. This process requires two steps: 1)identify the endmember signatures; 2)estimate the proportions of different endmembers.

In this project, we focus on endmembers identification. A large number of algorithm have been applied in a few past decades. Projections pursuit approaches including principal component analysis (PCA) [2], independent component analysis (ICA) [3], and singular value decomposition (SVD) [4]. These methods have a same problem that the extracted endmembers are not guaranteed to be nonnegative.

Another type of algorithm explores the connection between the theory convex geometry and the linear mixture model. In [5], for example, the extraction of endmembers is equivalent to finding the vertices of simplex that encloses the data cloud. In order to speed up the extraction, some algorithms [6] assume the exist of pure pixels, within which only one type of endmember is present. However, under some situations, such as the data is from specific ground covers [7], it is not reliable to make the assumption.

Recent years, an new method nonnegative matrix factorization (NMF) has been introduced into hyperspectral data unmixing. It approximates the original data through linear combinations and describes them as a set of non-negative basis vectors. The vectors are viewed similar as endmembers. Standard NMF algorithms do not apply any constraint on the vectors except nonnegative. In order to reduce the problem to be a more well-defined problem, a few different algorithms add some more constraints. In [8], the algorithm add smoothness constraint for estimating the endmembers. Another algorithm introduces a volume constraint into the NMF formulation thus it can integrate the least squares analysis and convex optimization. In this project, we will apply the same constraint to the problem and use gradient descent and conjugate gradient descent to solve the problem.

The rest of the report is organized as follows. In section II, we introduce the notations and formulate the problem. Section III we talk about the convexity of the problem. After that, in section IV we describe our two algorithms, including gradient descent and conjugate gradient descent.

## II. Background

Simplicity, a mixed pixel in a hyperspectral image is assumed as linear combination, which can be presented as

$$\mathbf{p} = \mathbf{A}\mathbf{s}, \tag{1}$$

where $\mathbf{p} \in \mathbf{R}_+^m$ is an observation vector of a single pixel with $m$ spectral bands. $\mathbf{A} \in \mathbf{R}_+^{m\times r}$ is the endmember matrix, whose columns $\{\mathbf{A}_j\}_{j=1,\cdots,r} \in \mathbf{R}_+^m$ correspond to $r$ endmembers (different materials). $\mathbf{s} \in \mathbf{R}_+^r$ is the proportion vector, and $\sum_{j=1}^r s_j = 1$. Thus, the original mixed pixel can be decomposed into linear combination of $r$ bases.

For all pixel, we can get the expression

$$\mathbf{P} = \mathbf{A}\mathbf{S}, \tag{2}$$

where $\mathbf{P} \in \mathbf{R}_+^{m\times n}$ is an observation matrix containing $m$ observations. $\mathbf{S} \in \mathbf{R}_+^{r\times m}$ is the corresponding proportion matrix.

So far, many algorithms can solve the decomposition problem, such as principle component analysis (PCA), singular value decomposition (SVD), independent component analysis (ICA), minimum volume transform (MVT), vertex component analysis (VCA) — one of the most advanced method.

We will apply convex optimization related algorithm to present the problem. Since the ultimate goal is to find out the endmember matrix $\mathbf{A}$, as well as their corresponding proportion $\mathbf{S}$, which can lead minimum error between real

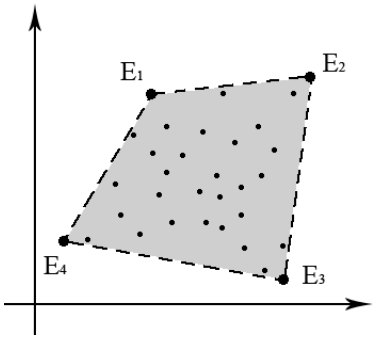


Fig. 1. Illustration of endmembers in geometric theory

observation and decomposed combination. The optimization problem can be written as

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{P}-\mathbf{A}\mathbf{S}\|_F^2 \\ \text{s.t.} \quad & \mathbf{A} \succeq 0,\ \mathbf{S} \succeq 0,\ \mathbf{1}_r^T\mathbf{S} = \mathbf{1}_m^T \end{aligned} \tag{3}$$

$\|\cdot\|$ is the Frobenius norm, and the symbol $\succeq$ denotes component-wise inequality (i.e., $\mathbf{A} \succeq 0$ means all elements in matrix $\mathbf{A}$ are not less then zero). From the theory of convex geometry, the objective function is to fund a convex hull with the vertices (endmembers), which can circumscribe all of the raw pixel points as illustrated in Fig. 1. At the same time, we wish the convex hull to be as small as possible. Therefore, the final optimization problem can be expressed as

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{P}-\mathbf{A}\mathbf{S}\|_F^2 + \alpha V(\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A} \succeq 0,\ \mathbf{S} \succeq 0,\ \mathbf{1}_r^T\mathbf{S} = \mathbf{1}_m^T \end{aligned} \tag{4}$$

where $V(\mathbf{A})$ is the penalty function, which calculates the volume of the convex hull determined by the endmembers. $\alpha \in \mathbf{R}_+$ is a regularization parameter used to adjust the trade-off between accurate decomposition and the convex hull volume.

A normal way to calculate the volume is

$$V(\mathbf{A}) = \frac{1}{(r-1)!}\left|\det\left([\mathbf{A}_2-\mathbf{A}_1,\cdots,\mathbf{A}_r-\mathbf{A}_1]\right)\right| \tag{5}$$

Equivalently,

$$V(\mathbf{A}) = \frac{1}{(r-1)!}\left|\det\left(\begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{A}_1 & \cdots & \mathbf{A}_r \end{bmatrix}\right)\right| \tag{6}$$

where $\mathbf{A}_j,\ j=1,\cdots,r$ is the $j$th column of the endmember matrix $\mathbf{A}$. Here, we assume that the relation between the number of endmembers $r$ and the dimension of pixels is $m = r-1$, because in geometric theory, $n$-dimension points can be circumscribed by a convex hull with at least $n+1$ points. We always want to pick as less endmembers as possible to decompose the mixed pixels, so assuming there are $r = m+1$ endmembers. In practice, it happens that $r < m+1$. If so, we may apply PCA or some other algorithms to reduce the dimension of $\mathbf{A}$ from $m$ to $r-1$.

Generally, the matrix $\mathbf{A}$ is not an ideal $(r-1)\times r$ matrix, thus following the way in [9], we can transfer matrix $\mathbf{A} \in \mathbf{R}^{m\times r}$ to the matrix $\widetilde{\mathbf{A}} \in \mathbf{R}^{(r-1)\times r}$.

$$\widetilde{\mathbf{A}} = \mathbf{U}^T(\mathbf{A}-\boldsymbol{\mu}\mathbf{1}_r^T) \tag{7}$$

where $\mathbf{U} \in \mathbf{R}^{m\times(r-1)}$ is formed by $r-1$ columns significant principle components of data $\mathbf{P}$ through PCA, and the vector $\boldsymbol{\mu}$ is the mean of data. Therefore, the penalty function can be written as

$$V(\mathbf{A}) = \frac{1}{2(r-1)!}\det{}^2\left(\begin{bmatrix} \mathbf{1}_r^T \\ \widetilde{\mathbf{A}} \end{bmatrix}\right) \tag{8}$$

In order to formulate the penalty function as a function of $\mathbf{A}$, let

$$\mathbf{Z} = \begin{bmatrix} \mathbf{1}_r^T \\ \widetilde{\mathbf{A}} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_r^T \\ \mathbf{0}_{(r-1)\times r} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{r-1}^T \\ \mathbf{I}_{(r-1)\times(r-1)} \end{bmatrix}\widetilde{\mathbf{A}} \tag{9}$$

The corresponding matrices are denoted by

$$\mathbf{C} = \begin{bmatrix} \mathbf{1}_r^T \\ \mathbf{0}_{(r-1)\times r} \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} \mathbf{0}_{r-1}^T \\ \mathbf{I}_{(r-1)\times(r-1)} \end{bmatrix} \tag{10}$$

Then we have

$$\mathbf{Z} = \mathbf{C} + \mathbf{B}\mathbf{U}^T(\mathbf{A}-\boldsymbol{\mu}\mathbf{1}_r^T) \tag{11}$$

Therefore, we can get the final objective function

$$f(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{P}-\mathbf{A}\mathbf{S}\|_F^2 + \frac{\tau}{2}\det{}^2\left(\mathbf{C}+\mathbf{B}\mathbf{U}^T(\mathbf{A}-\boldsymbol{\mu}\mathbf{1}_r^T)\right) \tag{12}$$

where $\tau = \frac{\alpha}{(r-1)!}$. Given observed data $\mathbf{P}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{U}$, and $\boldsymbol{\mu}$ are all constants.

## III. Convexity Analysis

The objective function consists of two parts, which can be denoted as

$$f_1(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{P}-\mathbf{A}\mathbf{S}\|_F^2 \tag{13}$$

$$f_2(\mathbf{A}) = \frac{\tau}{2}\det{}^2\left(\mathbf{C}+\mathbf{B}\mathbf{U}^T(\mathbf{A}-\boldsymbol{\mu}\mathbf{1}_r^T)\right) \tag{14}$$

We can check the second derivative of Eqs. 13 and 14 to determine whether they are convex. For Eq. 13, we can Hessian matrix

$$\mathcal{H}(f_1) = \begin{bmatrix} \dfrac{\partial f_1^2}{\partial \mathbf{A}^2} & \dfrac{\partial f_1^2}{\partial \mathbf{A}\partial \mathbf{S}} \\ \dfrac{\partial f_1^2}{\partial \mathbf{S}\partial \mathbf{A}} & \dfrac{\partial f_1^2}{\partial \mathbf{S}^2} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^T\mathbf{S} & 2(\mathbf{A}\mathbf{S})^T \\ 2\mathbf{A}\mathbf{S} & \mathbf{A}\mathbf{A}^T \end{bmatrix} \tag{15}$$

Under the constrains that $\mathbf{A} \succeq 0,\ \mathbf{S} \succeq 0$, it is obvious that $\mathcal{H}(f_1) \succeq 0$, but it does not guarantee that $\mathcal{H}(f_1)$ is positive semidefinite. So, $f_1$ is non-convex.

Next, let's figure out whether $f_2$ is convex. Since $f_2$ is much more complicate, we can solve its first derivative as beginning

$$\triangledown f_2(\mathbf{A}) = \tau\det(\mathbf{Z})\frac{\partial\det(\mathbf{Z})}{\partial\mathbf{A}} \tag{16}$$

where $\mathbf{Z}$ is defined in Eq. 11. $\frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}}$ can be expressed as

$$\frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}} = \begin{bmatrix} \frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}_{11}} & \cdots & \frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}_{1r}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}_{m1}} & \cdots & \frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}_{mr}} \end{bmatrix} \tag{17}$$

It is known that

$$\frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}_{ij}} = \det(\mathbf{Z}) Tr\left(\mathbf{Z}^{-1}\frac{\partial \mathbf{Z}}{\partial \mathbf{A}_{ij}}\right) \tag{18}$$

where $\mathbf{A}_{ij}$ denotes the $i$th row and $j$th column element of matrix $\mathbf{A}$. Since,

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{A}_{ij}} = \frac{\partial \mathbf{B}\mathbf{U}^T\mathbf{A}}{\partial \mathbf{A}_{ij}} \tag{19}$$

Let $\mathbf{D} = \mathbf{B}\mathbf{U}^T$ and $\bar{\mathbf{Z}} = \mathbf{Z}^{-1}$. It is easy to derive that

$$Tr\left(\mathbf{Z}^{-1}\frac{\partial \mathbf{Z}}{\partial \mathbf{A}_{ij}}\right) = \bar{\mathbf{Z}}_j^T\mathbf{D}_i \tag{20}$$

where $\mathbf{D}_i$ is the $i$th column of $\mathbf{D}$, and $\bar{\mathbf{Z}}_j^T$ is the $j$th row of $\bar{\mathbf{Z}}$. Finally, we have

$$\frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}} = \det(\mathbf{Z})(\bar{\mathbf{Z}}\mathbf{D})^T = \det(\mathbf{Z})\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T \tag{21}$$

Therefore,

$$\bigtriangledown f_2(\mathbf{A}) = \tau \det{}^2(\mathbf{Z})\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T \tag{22}$$

With the similar method, we can get the second derivative

$$\begin{aligned} \bigtriangledown^2 f_2(\mathbf{A}) = \quad & 2\tau \det(\mathbf{Z})\frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}}\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)^T \\ & +\tau \det{}^2(\mathbf{Z})\mathbf{U}\mathbf{B}^T\frac{\partial (\mathbf{Z}^{-1})^T}{\partial \mathbf{A}} \\ = \quad & 2\tau \det{}^2(\mathbf{Z})\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)^T \\ & -\tau \det{}^2(\mathbf{Z})\mathbf{U}\mathbf{B}^T\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\mathbf{Z}^{-1}\right)^T \\ = \quad & \tau \det{}^2(\mathbf{Z})\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)^T \end{aligned} \tag{23}$$

We can find that $\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)\left(\mathbf{U}\mathbf{B}^T(\mathbf{Z}^{-1})^T\right)^T$ is positive semidefinite. So, $f_2(\mathbf{A})$ is convex. Therefore, $f(\mathbf{A},\mathbf{S})$ is non-convex.

## IV. Optimization Algorithm

### A. Initialization

In the objective function, $\mathbf{P} \in \mathbf{R}^{m\times n}$ is the given data, from which columns of matrix $\mathbf{A}$ are randomly selected. For example, we can choose the first $r$ columns of $\mathbf{P}$ to build the matrix $\mathbf{A}$. How to determine the value of $r$ is an important issue, which will be discussed later. For the matrix $\mathbf{S}$, it can be initialized as a zero matrix.

### B. Lagrange dual

There is an equality (sum-to-one) and two inequality constraints in the optimal problem that can be dealt with using Lagrange dual. In practice, however, it is really hard to compute the Lagrange dual since $\mathbf{S}$ and $\mathbf{A}$ are both component-wise positive. This means there will be many independent parameters for each inequality constraint and a vector parameter for the equality constraint. The Lagrange dual is expressed as following

$$\begin{aligned} g(\mathbf{A},\mathbf{S},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}) = \inf_{\mathbf{A},\mathbf{S}} \Big[ & \frac{1}{2}\|\mathbf{P}-\mathbf{A}\mathbf{S}\|_F^2 + \frac{\tau}{2}\det{}^2(\mathbf{Z}) \\ & - \sum_{i=1}^{m}\sum_{j=1}^{r}\lambda_{ij}\mathbf{A}_{ij} - \sum_{i=1}^{r}\sum_{j=1}^{n}\mu_{ij}\mathbf{S}_{ij} \\ & + \boldsymbol{\nu}(\mathbf{1}_r^T\mathbf{S} - \mathbf{1}_n^T)\Big] \end{aligned} \tag{24}$$

In Eq. 24, there are $m \times r$ $\lambda$'s and $r \times n$ $\mu$'s. It will result in a big mess if we try to solve this Lagrange dual problem. Therefore, we need to figure out a more direct way to solve the optimal problem.

### C. Equality Constraint Elimination

Here we use a simple but effective method to achieve the equality constraint. A row of constant is added to matrix $\mathbf{P}$ and $\mathbf{A}$ as following.

$$\mathbf{P} = \begin{bmatrix} \mathbf{P} \\ \beta\mathbf{1}_n^T \end{bmatrix},\ \mathbf{A} = \begin{bmatrix} \mathbf{A} \\ \beta\mathbf{1}_r^T \end{bmatrix} \tag{25}$$

where $\beta \in \mathbf{R}_{++}$ is to control the effect of the sum-to-one constraint. As $\beta$ increases, the columns of matrix $\mathbf{S}$ is forced to approach to sum-to-one constraint. In order to balance between the estimation accuracy and the convergence rate.

### D. Gradient descent method

This objective function is a combinatorial optimization problem because $\mathbf{A}$ and $\mathbf{S}$ are independent. We can treats the problem as two sub-problems. In each sub-problem, one matrix is seen as the variable, and the other is fixed. Thus the problem is

$$\mathbf{A}^{k+1} = \arg\min_{\mathbf{A}} f(\mathbf{A},\mathbf{S}^k) \le f(\mathbf{A}^k,\mathbf{S}^k) \tag{26}$$

$$\mathbf{S}^{k+1} = \arg\min_{\mathbf{S}} f(\mathbf{A}^{k+1},\mathbf{S}) \le f(\mathbf{A}^{k+1},\mathbf{S}^k) \tag{27}$$

In the inequality Eq. 26, $\mathbf{S}^k$ is fixed, and we need to update $\mathbf{A}$ iteratively. Similarly, in the inequation(27), $\mathbf{A}$ is fixed, and $\mathbf{S}$ is updated iteratively. The update rule is expressed as

$$\begin{aligned} \mathbf{A}^{k+1} &= \mathbf{A}^k - \alpha^k \bigtriangledown_{\mathbf{A}} f(\mathbf{A}^k,\mathbf{S}^k) \\ \mathbf{S}^{k+1} &= \mathbf{S}^k - \beta^k \bigtriangledown_{\mathbf{S}} f(\mathbf{A}^{k+1},\mathbf{S}^k) \end{aligned}$$

where the parameters $\alpha^k$ and $\beta^k$ are stepsizes and $-\bigtriangledown_{\mathbf{A}} f(\mathbf{A}^k,\mathbf{S}^k)$ and $-\bigtriangledown_{\mathbf{S}} f(\mathbf{A}^{k+1},\mathbf{S}^k)$ are descent direction. Let $\alpha_o$ be the initial stepsize, and $\rho \in (0,1)$ be the scaling factor that reduce the stepsize iteratively. The stepsize can be expressed as $\alpha^k = \beta^{m_k}\alpha_o$, where $m_k$ is the first integer such that

$$f(\mathbf{A}^{k+1},\mathbf{S}^k) - f(\mathbf{A}^k,\mathbf{S}^k) \le \sigma\rho^{m_k}\alpha_o \bigtriangledown f(\mathbf{A}^k,\mathbf{S}^k)^T(\mathbf{A}^{k+1}-\mathbf{A}^k)$$

where $\sigma \in (0, \frac{1}{2})$ is the tolerance. Similarly, $\beta^k$ is obtained from the same process.

The first order derivative $\bigtriangledown_{\mathbf{S}} f(\mathbf{A}, \mathbf{S})$ can be easily found,

$$\bigtriangledown_{\mathbf{S}} f(\mathbf{A}, \mathbf{S}) = \mathbf{A^T}(\mathbf{AS} - \mathbf{X}) \tag{28}$$

The first order derivative $\bigtriangledown_{\mathbf{A}} f(\mathbf{A}, \mathbf{S})$ can be obtained from previous derivation in convexity analysis.

$$\bigtriangledown_{\mathbf{A}} f(\mathbf{A}, \mathbf{S}) = (\mathbf{AS} - \mathbf{X})\mathbf{S^T} + \tau \det(\mathbf{Z}) \frac{\partial \det(\mathbf{Z})}{\partial \mathbf{A}} \tag{29}$$

where $\mathbf{Z}$ is defined in Eq. 11. Finally, we have

$$\bigtriangledown_{\mathbf{A}} f(\mathbf{A}, \mathbf{S}) = (\mathbf{AS} - \mathbf{X})\mathbf{S^T} + \tau \det{}^2(\mathbf{Z})\mathbf{UB^T}(\mathbf{Z}^{-1})^T \tag{30}$$

Now, we need to find the stop condition. After infinite iteration, some specific requirement are satisfied. The stop condition is that the number of successive increasing steps is over a predefined value.

**Gradient descent algorithm with BTLS**

**Data**: Non-negative mixture data $\mathbf{P} \in \mathbf{R}^{m \times n}$ with each column begin an observation vector.
**Result**: Two non-negative matrices $\mathbf{A} \in \mathbf{R}^{m \times r}$ and $\mathbf{S} \in \mathbf{R}^{r \times n}$ with sum-to-one constraint $\mathbf{1}_r^T \mathbf{S} = 1_n^T$
**Initialization**: $\mathbf{S} = \mathbf{0}_{r \times n}$; iteration index $k = 0$;
**repeat BTLS**
$\mathbf{A}^{k+1} = \mathbf{A}^k - \alpha^k \bigtriangledown_{\mathbf{A}} f(\mathbf{A}^k, \mathbf{S}^k)$
$\mathbf{S}^{k+1} = \mathbf{S}^k - \beta^k \bigtriangledown_{\mathbf{S}} f(\mathbf{A}^{k+1}, \mathbf{S}^k)$
increase k by 1
**until** stop condition is satisfied

Here, the stop condition could be a limit of iteration number or error between two steps. Since we cannot proof the convexity of the objective function, we have to see it as non-convex. So, in order to avoid tripping into local minimum, the gradient search should run multiple times with random initial matrices $\mathbf{A}$ and $\mathbf{S}$. Also, we may apply simulated annealing or genetic algorithm to ensure a global minimum. However, considering computation complexity, simulated annealing should be better.

### *E. Conjugate gradient method*

Conjugate gradient method is expected faster than the gradient descent method. For each step, the searching direction will be decided according to both current gradient and last searching direction. Specifically, the steps of conjugate gradient method is shown as following, which only updates $\mathbf{S}$ as an example.

**Conjugate gradient method (update S only)**

**Data**: Non-negative mixture data $\mathbf{P} \in \mathbf{R}^{m \times n}$ with each column begin an observation vector.
**Result**: Two non-negative matrices $\mathbf{A} \in \mathbf{R}^{m \times r}$ and $\mathbf{S} \in \mathbf{R}^{r \times n}$ with sum-to-one constraint $\mathbf{1}_r^T \mathbf{S} = 1_n^T$
**Initialization**: $\mathbf{S} = \mathbf{0}_{r \times n}$; initial searching direction $\mathbf{d}^0$; iteration index $k = 0$;
**repeat**

$$\alpha^k = -\frac{\text{diag}\left(\bigtriangledown_{\mathbf{S}} f^T(\mathbf{A}, \mathbf{S}^k)\mathbf{d}^k\right)}{\text{diag}\left((\mathbf{d}^k)^T \mathbf{A}^T \mathbf{A} \mathbf{d}^k\right)}$$

$$\mathbf{S}^{k+1} = \mathbf{S}^k + \alpha^k \mathbf{d}^k$$

**stop** if condition is satisfied

$$\beta^k = \frac{\text{diag}\left(\bigtriangledown_{\mathbf{S}} f^T(\mathbf{S}^{k+1})(\bigtriangledown_{\mathbf{S}} f(\mathbf{S}^{k+1}) - \bigtriangledown_{\mathbf{S}} f(\mathbf{S}^k))\right)}{\text{diag}\left(\bigtriangledown_{\mathbf{S}} f^T(\mathbf{A}, \mathbf{S}^k) \bigtriangledown_{\mathbf{S}} f(\mathbf{A}, \mathbf{S}^k)\right)}$$

$$\mathbf{d}^{k+1} = -\bigtriangledown_{\mathbf{S}} f(\mathbf{A}, \mathbf{S}^{k+1}) + \beta^k \mathbf{d}^k$$

$k = k + 1$

Note that, the equation of $\alpha^k$ gives a optimal step size regardless the constraint on $\mathbf{S}$. When the updated $S^{k+1}$ is out of the feasible set, we will track back to the feasible set. If current point is on the boundary, or there is no available $\alpha^k$ could result in a available $\mathbf{S}$, we'll project it back to the feasible set, which means setting $\alpha^k$ as zero or setting those negative element in $\mathbf{S}$ as zeros. Thus, we could ensure the $\mathbf{S}$ satisfies the constraint. Similarly, the make operation is taken when searching $\mathbf{A}$.

The stop condition in the conjugate gradient method could be error tolerance of maximal iteration step. In this paper, maximal iteration step is used in order to speed up the computation time.

## V. Experiment Result

In order to estimate the optimal algorithms, we need to generate a matrix of mixed pixels $\mathbf{X}$, and then compare the true endmember matrix $\mathbf{A}$ with the one obtained from the optimization problem.

### *A. Generation of mixed pixels*

The mixed pixels can be generated from a series of given endmembers. For example, we already have $r$ endmember with $m$ dimension for each, which is noted by $\mathbf{A}_{true}$. Then $n$ pixels can be generated by multiplying $\mathbf{A}_{true}$ with a series of random factors whose sum approaches to one. In order to simulate real pixels, noise is add to the generated pixels.

In practice, a set of spectral reflections are selected from the USGS digital spectral library. So, we have the true endmember matrix $\mathbf{A}_{true}$ as shown in Fig. 2.

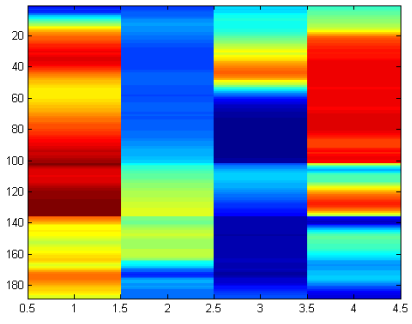


Fig. 2. The true endmember

A random proportional matrix $\mathbf{S} \in \mathbf{R}_{++}^{r \times n}$ with sum-to-one constraint can be easily generated through fully constrained least square method. After adding noise (around 20dB), we

get a generated highly mixed pixel matrix $\mathbf{P}$. Generated $\mathbf{P}$ is shown in Fig. 3, totally over 3000 pixels.

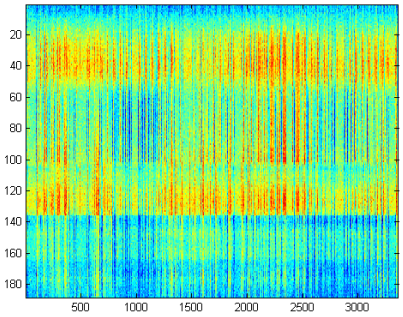


Fig. 3. Generated highly mixed pixels

### B. Algorithm evaluation

The first step of dealing with a real pixel matrix is denoising. Here, SVD is applied to denoise the generated $\mathbf{P}$. Fig. 4 shown the denoised $\mathbf{P}$.

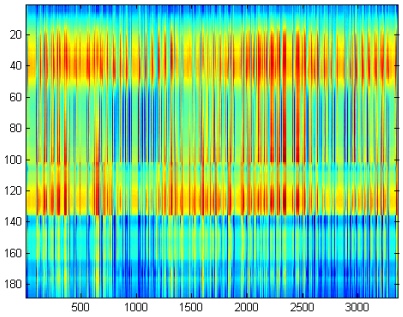


Fig. 4. Denoised mixed pixels

Selection of initial points is important for this optimal problem since the objective function is non-convex. One way to selection the initial points is random selection, but this way might trip us into local optimum. So, the algorithm need to repeat from different random initial points in order to achieve the global optimum. Obviously, this way is time-consuming. A smarter way is to find some suboptimal optimal points, which have already been closed to the real optimal value. Fortunately, the vertex component analysis (VCA), one of the most advanced convex geometry-based endmember detection algorithm, can help give the suboptimal points. Based on the suboptimal point, we can run conjugate gradient method to search the optimal solution. Finally, we get the optimized endmember matrix $\mathbf{A}$ as shown in Fig. 5. Compared with the true endmember in Fig. 2, the optimized endmembers are similar in corresponding column pairs, which means the algorithm used in this paper can successfully find the true endmembers within a finite error. Note that the VCA is already a successful method to find the endmembers, in order to highlight the algorithm used in this paper, we will compare it with VCA in performance analysis.

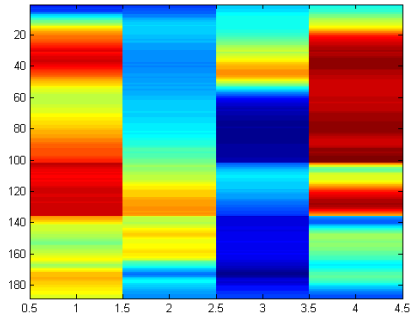


Fig. 5. Optimized endmember

### C. Convergence rate analysis

Mathematical convergence rate analysis can be easily found from plenty of literature, so no more derivation here. An intuitive illustration of convergence rate is shown in Fig. 6. Compared with gradient descent method, the conjugate gradient method converges faster in each step $k$, but both methods could approach the optimal value in finite steps. Specifically, the conjugate method has been closed to optimal value after 10 steps.

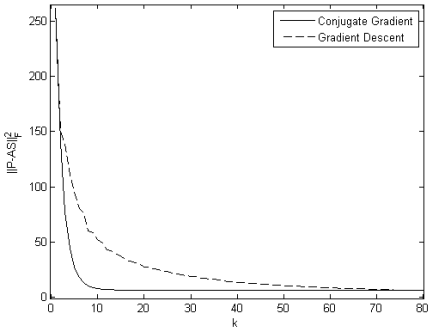


Fig. 6. Comparison of convergence rate on each step

Another criterion we are interested in is the computation time of obtaining a satisfied optimal value. From Fig. 7, the conjugate gradient method finished the searching within 8 second, while the gradient descent method consumes over half a minute.

In order to compare the performance of two method, the same initial point is used. Here, the initial point is obtained from VCA, which can give the endmembers closed to the true endmembers globally. Therefore, based on the close-to-optimal initial point, we could improve it using the MVC algorithm. In another word, a global optimum could be guaranteed in our algorithm.

Moreover, in the process of searching the optimal solution, it is allowed that some continuous steps make the value of

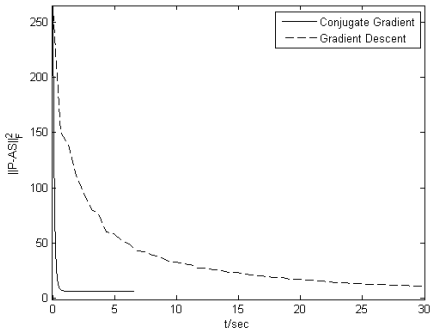


Fig. 7. Comparison of convergence rate based on computation time

objective function increase. This may help avoid trapped in local optimal.

### D. Parameter effect

There a free parameter $\tau$ in the objective function. A larger $\tau$ will increase the strength of volume constraint. Since we want to make the volume as small as possible, so a larger $\tau$ is better. However, a larger $\tau$ will result in slower convergence rate as shown in Fig. 8. So, a trade-off problem–looser and faster or stronger and slower–need to tackle with. Empirically, we choose $\tau = 0.01$.

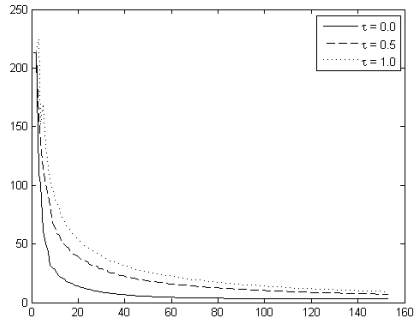


Fig. 8. Comparison of convergence rate for different value of $\tau$

### E. Performance analysis

In the section, we will compare the performance of the algorithm used in this paper when select different values of $\tau$. Performance between our algorithm and the VCA will be compared as well. Before this, we need to define some metrics to quantitatively evaluate the performance.

A widely used metric, called spectral angle distance (SAD), is to measure the shape similarity between the true endmember $\mathbf{a}$ and the estimated one $\hat{\mathbf{a}}$. Here, $\mathbf{a}$ and $\hat{\mathbf{a}}$ are high dimensional vectors. The SAD can be expressed as

$$SAD = \arccos\left(\frac{\mathbf{a}^T\hat{\mathbf{a}}}{\|\mathbf{a}\|\|\hat{\mathbf{a}}\|}\right) \tag{31}$$

Another metric is spectral information divergence (SID), which is like the expression of information entropy as shown in following.

$$SID = \sum_{i=1}^{m}\left(p_i \log\left(\frac{p_i}{\hat{p}_i}\right) + \hat{p}_i \log\left(\frac{\hat{p}_i}{p_i}\right)\right) \tag{32}$$

$$p_i = \frac{a_i}{\sum_{j=1}^{m} a_j},\ \hat{p}_i = \frac{\hat{a}_i}{\sum_{j=1}^{m} \hat{a}_j} \tag{33}$$

where $a_i$ and $\hat{a}_i$ are the $i$th element of vector $\mathbf{a}$ and $\hat{\mathbf{a}}$ respectively.

Using above two metrics, we can test the performance with different $\tau$. Table I shown the metric values in detail. The initial point is the same for different $\tau$, and smaller metric value means better performance. Therefore, if $\tau$ is too large, the performance will become worse. From experiment, we know that the performance of VCA and our algorithm is close when the value of $\tau$ is around 0.05.

TABLE I
COMPARISON OF PERFORMANCE WITH DIFFERENT $\tau$

| $\tau$ | 0.0 | 0.5 | 1.0 |
|---|---|---|---|
| *SAD* | 6.5685 | 11.2716 | 12.4341 |
| *SID* | 0.0430 | 0.0999 | 0.1174 |

In practice, we choose $\tau = 0.015$. Then the comparison of two method is shown in Table II, where MVC is the short for minimum volume constrained. Therefore, MVC can achieve better performance compared with VCA.

TABLE II
PERFORMANCE COMPARISON OF TWO METHOD

| | MVC | VCA |
|---|---|---|
| *SAD* | 7.5774 | 9.1541 |
| *SID* | 0.0453 | 0.0614 |

## VI. CONCLUSION AND FUTURE WORK

This paper, a minimum volume constrained (MVC) optimal problem is set up to to estimate the endmember of hyperspectral pixels. The objective function can be seen as a combination of the error function and penalty function, which aims to minimize the volume of the convex hull built by the endmembers. With this constrain, we can ensure to find the minimum convex hull that involve all data points.

Proved that the objective function is non-convex, and there are two free variables $\mathbf{A}$ and $\mathbf{S}$. However, the objective function is convex when fix either variable. Therefore, those convex optimal method can be used here. Conjugate gradient method is applied to solve this problem since it could converge faster than gradient method, as well as many other similar method. Specifically, we search the optimal value by fixing one variable and modifying another, then iterate alternatively. With finite iterations, an optimal solution can be obtained, which is better than the solution calculated by one of the most advanced method vertex component analysis (VCA). In order to avoid trapped in local optimal, multiple initial point can be selected randomly to repeat the conjugate method. But in

practice, we used the result of VCA as initial point. This way could significantly reduce the computation time.

Compared with the true endmembers, the estimated members have the similar pattern. Using two criteria spectral angle distance (SAD) and spectral information divergence (SID), we can compare the performance of MVC with VCA. From convergence analysis, conjugate gradient method converges faster in each step compared with gradient descent method. In the aspect of computation time, conjugate gradient method runs much faster.

Currently, selection of the parameter $\tau$ only depends on experiments. A theoretical way need to be developed to choose a optimal parameter. In this paper, we only compare the performance of MVC and VCA. Actually, other more advanced method should be introduced in the comparison, which could make the MVC method more convicted.

## REFERENCES


[1] G. M. Foody, *Remote Sensing Image Analysis: Including the Spatial Domain*. Kluwer Academic Publishers, 2004, ch.3, pp 37-49.

[2] M. O. Smith, P. E. Johnson, and J. B. Adams, "Quantitative determination of mineral types and abundances from reflectance specrtral using principal components analysis," *J. Geophys. Res.,* no.90, pp. C797-C804, 1985.

[3] J. Bayliss, J. A. Gualtieri, and R. Cromp, ""Analyzing hyperspectral data with independent component analysis" in *Proc, of SPIE* vol. 3240, 1997, pp.133-143.

[4] G. Healey and D. Slater, "Models and methods for automated material identificatin in hyperspectral imagery acquired under unknown illunmination and atmospheric conditions," *IEEE Trans Geosci. Remote Sensing,* vol.32, no.3 pp. 2706-2717, 1999.

[5] M. D. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Trans Geosci. Remote Sensing,* vol.32, no.3 pp. 542-552, May 1994.

[6] J. M. P. Nascimento, and J. M. B. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data" *IEEE Trans Geosci. Remote Sensing,* vol.43, no.4 pp. 898-910, Apr 2005.

[7] M. Berman, H. Kiiveri, R. Lagerstrom, R. Dunne, and J. F. Huntington,"Ice: A statistical approach to identifying endmembers in hyperspectral images," *IEEE Trans Geosci. Remote Sensing,* vol.42, no.10 pp. 2085-2095, Oct. 2004.

[8] V.P. Paura, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *To appear in Linear Algebra and Applications,* 2006.

[9] Lidan Miao, Hairong Qi, *Endmember Extraction from Highly Mixed Data Using Minimum Volume Constrained Non-negative Matrix Factorization*. Geoscience and Remote Sensing, IEEE Transactions on 45.3 (2007): 765-777.