A Quantitative and Comparative Analysis of Different Implementations of N-FINDR: A Fast Endmember Extraction Algorithm

Maciel Zortea and Antonio Plaza, *Senior Member, IEEE*

Abstract—The N-FINDR algorithm is one of the most widely used and successfully applied methods for automatically determining endmembers in hyperspectral image data without using *a priori* information. The algorithm attempts to automatically find the simplex of maximum volume that can be inscribed within the hyperspectral data set. Due to the intrinsic complexity of remotely sensed scenes, the final volume-based solution provided by N-FINDR may be not the global maximum. In addition, the final results provided by the algorithm are typically dependent of its initialization. In this letter, we explore the aforementioned issues and conduct a quantitative and comparative analysis of different (available and new) strategies for the implementation of N-FINDR. Our experimental evaluation and comparison are conducted using two well-known hyperspectral scenes collected by the National Aeronautics and Space Administration Jet Propulsion Laboratory's Airborne Visible Infrared Imaging Spectrometer.

Index Terms—Endmember extraction, hyperspectral imaging, N-FINDR algorithm, spectral unmixing.

I. INTRODUCTION

ENDMEMBER extraction is the process of selecting a collection of pure signature spectra of the materials present in a remotely sensed hyperspectral scene [1]. Let us assume that a remotely sensed hyperspectral scene with n bands is denoted by $\mathbf{I}$, in which the pixel at the discrete spatial coordinates (i, j) of the scene is represented by a feature vector $\mathbf{X}(i, j) = [x_1(i, j), x_2(i, j), \dots, x_n(i, j)] \in \Re^n$, where $\Re$ denotes the set of real numbers corresponding to the pixel's spectral response $x_k(i, j)$ at sensor channels $k = 1, \dots, n$. Under the linear mixture model assumption, each pixel vector in the original scene can be modeled using the following:

$$\mathbf{X}(i, j) = \sum_{z=1}^{p} \Phi_z(i, j) \cdot \mathbf{E}_z + \mathbf{n}(i, j) \qquad (1)$$

where $\mathbf{E}_z$ denotes the spectral response of endmember z, $\Phi_z(i, j)$ is a scalar value designating the fractional abundance of the endmember z at the pixel $\mathbf{X}(i, j)$, p is the total number of endmembers, and $\mathbf{n}(i, j)$ is a noise vector. The solution of the linear spectral mixture problem described in (1) relies on a successful estimation of how many endmembers p

are present in the input hyperspectral scene $\mathbf{I}$ and also on the correct determination of a set $\{\mathbf{E}_z\}_{z=1}^{p}$ of endmembers.

Over the last decade, several algorithms have been developed for automatic or semiautomatic extraction of spectral endmembers [2], [3]. One of the most widely used and successfully applied methods for this purpose is Winter's N-FINDR algorithm [4]. This approach attempts to automatically find the simplex of maximum volume that can be inscribed within the hyperspectral data set. In order to accomplish this task, the N-FINDR algorithm typically starts with a set of candidates which are randomly selected from the hyperspectral scene. Due to the intrinsic complexity of remotely sensed scenes, the final volume-based solution provided by N-FINDR may be not the global maximum, and this final solution is typically dependent on the algorithm's initialization [5]. This suggests the need for different implementation strategies [6].

In this letter, we develop a quantitative and comparative analysis of different implementations of N-FINDR. First, we use the original implementation of the algorithm [4], which tests each pixel in each endmember position and recalculates the volume, retaining the pixel combinations that result in the maximum volume. Then, we consider a second implementation [5], in which the order of the two main loops of the original algorithm is switched. Finally, we propose optimizations for both versions. The optimization for the first version is based on selecting the pixels to be tested in each endmember position randomly, until all the pixels in the scene have been exhausted. On the other hand, the optimization for the second version is based on dividing all the input pixels present in the scene into a few subblocks made up of mutually exclusive subsets of pixels, randomly generated from the original scene, so that the union of all the subblocks provides the original scene. The optimized version is then run sequentially on each subblock in a cascade fashion, so that N-FINDR is initialized in each step with the solution obtained for the previous subblock.

The remainder of the letter is organized as follows. Section II describes the considered implementations of N-FINDR in an algorithmic fashion. In Section III, the performance discrepancy among these implementations is investigated using two well-known hyperspectral scenes covering different application domains. Finally, Section IV provides some concluding remarks.

II. IMPLEMENTATIONS OF N-FINDR

In this section, we describe four different implementations of the N-FINDR algorithm. The first implementation corresponds to the original algorithm proposed by Winter, and the second implementation is a modification of this algorithm. In addition,

The authors are with the Department of Technology of Computers and Communications, University of Extremadura, 10071 Cáceres, Spain (e-mail: mzortea@unige.it; aplaza@unex.es).

two optimizations for the aforementioned implementations are proposed. It should be noted that the proposed optimizations have not been designed with the goal of reducing the computational complexity of N-FINDR (as it is the case with other efforts in the literature). Instead, the optimizations aim at improving the endmember extraction capability of N-FINDR, but keeping its computational complexity unchanged (a detailed study of the computational complexity of N-FINDR is available in [7]). To achieve this, we focus on two main issues: 1) the initialization of the algorithm and 2) the order in which the pixels are processed. Despite previous efforts in the literature, which analyze the first issue [5], the second issue has not been studied in detail as of yet. However, according to our experience, this issue has a significant impact on the final results provided by the algorithm. An algorithmic description of the considered implementations of N-FINDR follows.

A. Implementation #1

The original N-FINDR algorithm developed by Winter [4] can be summarized by the following steps.

1) *Feature reduction*. Apply a dimensionality reduction transformation such as the minimum noise fraction (MNF) [8] or the principal component analysis (PCA) [1] to reduce the dimensionality of the data from n to $p - 1$, where p is an input parameter to the algorithm (number of endmembers to be extracted).

2) *Initialization*. Let $\{\mathbf{E}_1^{(0)}, \mathbf{E}_2^{(0)}, \ldots, \mathbf{E}_p^{(0)}\}$ be a set of endmembers randomly extracted from the input data.

3) *Volume calculation*. At iteration $k \geq 0$, calculate the volume defined by the current set of endmembers as follows:

$$V\left(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)}\right) = \frac{\left|\det\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{E}_1^{(k)} & \mathbf{E}_2^{(k)} & \cdots & \mathbf{E}_p^{(k)} \end{bmatrix}\right|}{(p-1)!}.$$
(2)

4) *Replacement*. For each pixel vector $\mathbf{X}(i, j)$ in the input hyperspectral data, recalculate the volume by testing the pixel in all p endmember positions, i.e., first calculate $V(\mathbf{X}(i, j), \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)})$, then $V(\mathbf{E}_1^{(k)}, \mathbf{X}(i, j), \ldots, \mathbf{E}_p^{(k)})$, and so on until $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \ldots, \mathbf{X}(i, j))$. If none of the p recalculated volumes is greater than $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)})$, then no endmember is replaced. Otherwise, the combination with maximum volume is retained. Let us assume that the endmember absent in the combination resulting in the maximum volume is denoted by $\mathbf{E}_j^{(k+1)}$. In this case, a new set of endmembers is produced by letting $\mathbf{E}_j^{(k+1)} = \mathbf{X}(i, j)$ and $\mathbf{E}_i^{(k+1)} = \mathbf{E}_i^{(k)}$ for $i \neq j$. The replacement step is repeated for all the pixel vectors in the input data until all the pixels have been exhausted.

It is important to emphasize that, in step 4) of the aforementioned implementation, we assume that the pixel vectors in the input data are processed in a predefined row–column order, i.e., from the first pixel in the first image row to the last pixel in the last image row. This approach seems the most intuitive for software implementation, but there is a high probability that each pair of consecutive sample pixels processed by the algorithm is spectrally similar due to spatial correlation.

B. Implementation #2

In this section, we develop an alternative implementation of N-FINDR (tested originally in [5]) in which the orders of the two main loops in step 4) of implementation #1 are switched. In this case, the first three steps are identical to those in implementation #1, and only step 4) is rewritten as follows:

1) *Replacement*. Recalculate the volume by testing all pixel vectors in the input data (processed in a row–column order) in the first endmember position, i.e., first calculate $V(\mathbf{X}(1, 1), \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)})$, then $V(\mathbf{X}(1, 2), \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)})$, and so on until $V(\mathbf{X}(r, c), \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)})$, where r is the number of rows and c is the number of columns. If none of the $r \times c$ recalculated volumes is greater than $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \ldots, \mathbf{E}_p^{(k)})$, then the endmember $\mathbf{E}_1^{(k)}$ is not replaced. Otherwise, the combination with maximum volume is retained. Let us assume that the pixel which replaced $\mathbf{E}_1^{(k)}$ in this combination is denoted by $\mathbf{X}(i, j)$. In this case, a new set of endmembers is produced by letting $\mathbf{E}_1^{(k+1)} = \mathbf{X}(i, j)$ and $\mathbf{E}_i^{(k+1)} = \mathbf{E}_i^{(k)}$ for $1 < i \leq p$. The replacement step is now repeated by testing all pixel vectors in the input data in the second endmember position, then in the third endmember position, and so on, retaining the combinations which result in the maximum value until all pixel vectors in the input data have been tested in each endmember position.

C. Randomization: An Optimization for Implementation #1

In this section, we introduce a simple yet effective optimization for implementation #1 of N-FINDR. The optimization is performed in step 4) (*replacement*) and consists of selecting the pixels to be tested in each endmember position randomly (instead of assuming that the pixels are processed in a predefined row–column order) until all the pixels in the hyperspectral scene have been exhausted. This is done by randomly exchanging the spatial coordinates (i, j) of all pixel vectors in the input scene prior to execution of implementation #1. The main assumption behind our proposed optimization is that a random strategy in the selection of pixel vectors to be processed by the algorithm is more likely to better approximate the statistics of the original image with a smaller percentage of sample pixels than the standard row–column strategy, in which consecutive samples may be spatially correlated. As a result, the proposed optimization is expected to conduct a more effective endmember search and to better adjust endmember candidates in all p endmember positions, thus increasing the chances to improve the final volume-based solution. This approach is shown in Fig. 1.

D. Sequential Blocks: An Optimization for Implementation #2

In this section, we introduce a sequential block optimization for implementation #2 of N-FINDR. The basic idea is to divide the input hyperspectral scene into a set of k subblocks made up of mutually exclusive subsets of pixels, which are randomly generated from the original data cube so that the union of all the subblocks provides the original cube. Implementation #2 is then run sequentially on each subblock, in a cascade fashion, so that the algorithm is initialized in each step with the solution
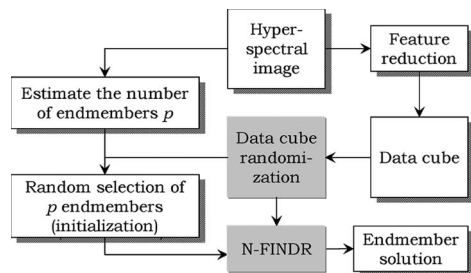
Fig. 1.   Flowchart of the optimization for N-FINDR implementation #1.

Fig. 2.   Flowchart of the optimization for N-FINDR implementation #2.

obtained for the previously analyzed subblock. This approach is shown in Fig. 2.

It is worth mentioning that the sequential block optimization proposed for implementation #2 is an adaptation of the randomization optimization proposed for implementation #1. The latter cannot improve implementation #2 since, in this case, all pixels are tested in each single endmember position before testing all pixels again in a different endmember position. In the following section, we evaluate the two considered implementations of N-FINDR and their respective optimizations in different application domains.

III. EXPERIMENTAL RESULTS

A. Experimental Setup

The two considered implementations of N-FINDR and their respective optimized versions are validated and intercompared using two well-known hyperspectral scenes collected by the airborne visible infrared imaging spectrometer (AVIRIS) over the Indian Pines region in northwest Indiana[1] and the Cuprite mining district in Nevada.[2] A library of reference spectral signatures collected by the U.S. Geological Survey (USGS) is available for the Cuprite scene.[3] Prior to data analysis, water absorption and noisy channels have been removed from both scenes, leaving a total of 190 radiance channels for the Indian Pines scene and 192 reflectance channels for the Cuprite scene.

In our experiments, the *feature reduction* step of N-FINDR has been conducted using both PCA and MNF. The *initialization* step has been performed so that each implementation is run 50 times (where the four considered implementations receive exactly the same 50 sets of randomly chosen endmember pixels to start the process), and the mean volume-based scores provided by each implementation after the 50 runs are reported. Finally, it is worth noting that the estimation of the number of endmembers p to be extracted by each implementation has been conducted using the virtual dimensionality concept [9]. This approach uses a Neyman–Pearson detector as a decision maker based on a prescribed false alarm probability P_F. According to recent studies [5], a reasonable empirical choice is $P_F = 10^{-4}$, which resulted in estimates of $p = 18$ for the 145×145 pixel Indian Pines scene and $p = 15$ for the 250×200 pixel Cuprite scene. These values of p will be used for the different endmember extraction methods applied to the same data set.
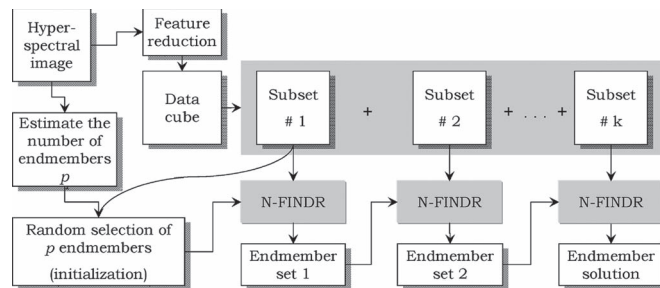
B. Experiment 1: Analysis of Volume Estimations

Our first experiment analyzes the volume of the simplex provided for implementations #1 and #2, as well as their respective optimizations. Fig. 3 shows the box and whisker plots summarizing the results obtained for the 50 algorithm runs conducted for each implementation using both MNF and PCA as feature reduction techniques. In the plots, implementation #1 is denoted by "1" and implementation #2 is denoted by "2," while the randomization optimization for implementation #1 is denoted by "R" and the sequential block optimization for implementation #2 is denoted by "B." In the latter optimization, we have empirically set parameter k (number of blocks in which the original scene is randomly divided) to $k = 8$.

In the box and whisker plots shown in Fig. 3, the boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the boxes to show the extent of the rest of the data. Outliers are data with values beyond the end of the whiskers (taken as 1.5 times the interquartile range of the data). In a notched box plot, the notches represent a robust estimate of the uncertainty about the medians for box-to-box comparison. Boxes whose notches do not overlap indicate that the medians of the two groups differ at the 5% significance level [10]. From Fig. 3, it can be observed that implementation #1 outperforms implementation #2 in all considered data sets, while the two proposed optimizations outperform both implementations #1 and #2 in terms of volume.

For illustrative purposes, the scatter plots in Fig. 4 show—for each run of an original versus an optimized implementation—the volume of the simplex obtained by the original algorithm versus the volume obtained by the proposed optimization (using the same initial conditions in both cases). The MNF transform was used for feature reduction purposes. The plots in Fig. 4 indicate how many times (out of 50) the proposed optimizations are able to outperform their respective counterparts when both start from the same initial condition (see data points above the diagonal in the scatter plots). As shown in Fig. 4, both optimizations often lead to a significant increase in the final volume of the simplex that may well reach the same order of magnitude as the volume variability observed due to the random choice of the initial set of endmembers. This indicates that, without increasing the computational complexity of the original implementations, the proposed optimizations are likely to provide a final solution with higher volume when processing the two considered hyperspectral data sets.

An important issue for the sequential block optimization of implementation #2 is the choice of the number of blocks k to randomly divide the original scene (see Fig. 2). In order to complete our analysis in terms of volume estimations, it is desirable to evaluate the impact of this parameter on the

---

[1]http://cobweb.ecn.purdue.edu/biehl/MultiSpec/

[2]http://aviris.jpl.nasa.gov/html/aviris.freedata.html
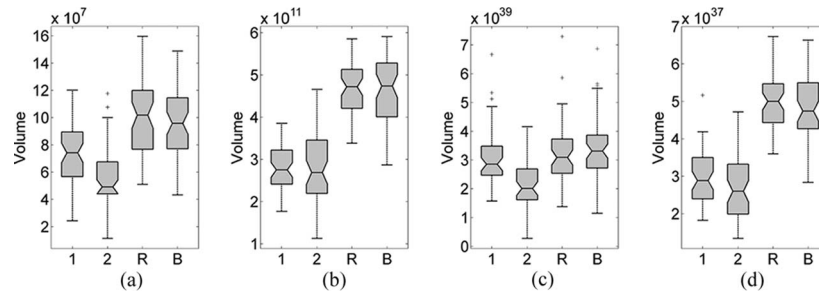
[3]http://speclab.cr.usgs.gov

Fig. 3.   Experiment 1. Volume of the simplex obtained for the AVIRIS Indian Pines (AVIP) and Cuprite (AVCP) scenes by the different implementations of N-FINDR, using both the MNF and PCA transforms for feature reduction. (a) AVIP (MNF). (b) AVCP (MNF). (c) AVIP (PCA). (d) AVCP (PCA).
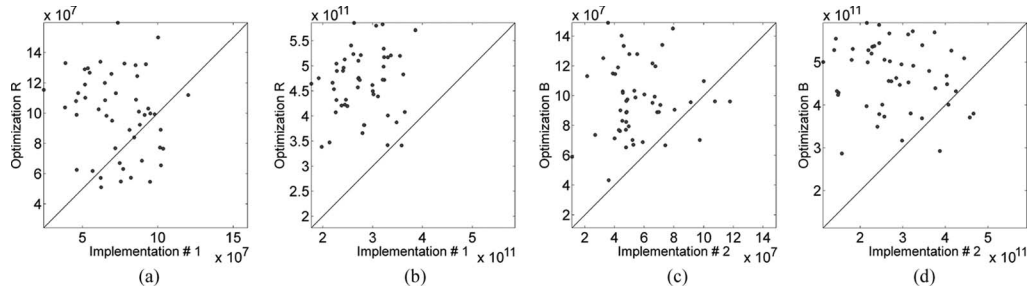

Fig. 4.   Experiment 1. Scatter plots showing the volume of the simplex obtained for the AVIP and AVCP scenes after 50 random initializations of the original implementations versus their corresponding optimizations. (a) AVIP (implementation #1). (b) AVCP (implementation #1). (c) AVIP (implementation #2). (d) AVCP (implementation #2).
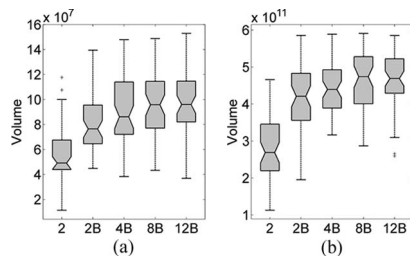

Fig. 5.   Experiment 1. Volume of the simplex obtained for the AVIP and AVCP scenes by using implementation #2 and the proposed sequential block optimization, denoted by "kB," where k is the number of blocks (values of $k = 2$, $k = 4$, $k = 8$, and $k = 12$ are tested). (a) AVIP. (b) AVCP.
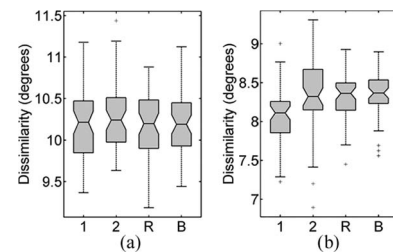

Fig. 6.   Experiment 2. Average SAD between the full sets of endmembers extracted from the AVIP and AVCP scenes by the different implementations of N-FINDR. (a) AVIP. (b) AVCP.

final volume-based results. In our previous experiments, we arbitrarily chose $k = 8$. In this experiment, we evaluate this parameter by focusing our analysis on the proposed sequential block partitioning algorithm using MNF for feature reduction. Fig. 5 shows the box and whisker plots depicting the volume of the simplex generated by the endmember solution obtained after setting $k = 2$, $k = 4$, $k = 8$, and $k = 12$. As Fig. 5 shows, all experimental choices of k resulted in a remarkable increase of the volume of the generated simplex when compared to implementation #2. The difference is already remarkable for $k = 2$ and further increases progressively for the larger values of k tested. The results in Fig. 5 suggest that, for the range of values tested, each particular data set may have a slightly different optimal number of block partitions. In any event, all tested values of k resulted on proper choices. The visual analysis of the overall performances reported in the box and whisker plots in Fig. 5 reveals that a safe empirical recommendation for this parameter would be in the range $k = 4$ to $k = 8$.

C. Experiment 2: Analysis of Extracted Endmembers

Our second experiment analyzes both the spectral distinctiveness and purity of the endmembers extracted by

implementations #1 and #2 and their respective optimizations. Spectral distinctiveness is assessed by a simple spectral dissimilarity score, defined as the average spectral angle distance (SAD) between the full sets of extracted endmembers [3]. Ideally, the extracted endmembers should be as distinct as possible (higher dissimilarity). As shown in Fig. 6, the two proposed optimizations provide endmember sets with average SAD scores which are comparable (Indian Pines scene) or higher (Cuprite scene) than their respective implementations. On the other hand, spectral purity is assessed in Table I by reporting the SAD scores obtained after comparing the USGS library signatures of the main five minerals present in the Cuprite scene with the corresponding endmembers extracted by the different N-FINDR implementations, where each library signature was matched to one of the endmembers extracted by a certain algorithm in terms of the smaller SAD value observed across the full endmember set. It should be noted that the SAD values reported in Table I correspond to the average (and standard deviation) of 50 random initializations for each algorithm. As shown by Table I, the volume increase observed for the proposed optimizations in the Cuprite scene results in a better match between library and extracted endmembers (lower SAD scores) without increasing the computational complexity.

Mineral	1	2	R	B
Alunite	$6.21°(0.45°)$	$6.21°(0.40°)$	$6.12°(0.26°)$	$6.17°(0.30°)$
Buddingtonite	$4.33°(0.56°)$	$4.71°(0.79°)$	$4.23°(0.41°)$	$4.34°(0.56°)$
Calcite	$5.69°(0.14°)$	$5.64°(0.14°)$	$5.66°(0.12°)$	$5.65°(0.13°)$
Kaolinite	$8.99°(0.92°)$	$8.39°(0.39°)$	$8.41°(0.65°)$	$8.35°(0.42°)$
Muscovite	$4.97°(0.79°)$	$5.41°(0.56°)$	$5.43°(0.46°)$	$5.32°(0.56°)$
Average	$6.04°(1.73°)$	$6.07°(1.35°)$	$5.97°(1.43°)$	$5.97°(1.40°)$

D. Experiment 3: Analysis of Abundance Estimations

In our third experiment, we use the linear mixture model in (1) to validate the quality of fractional abundance estimations (performed using a standard least squares approach) resulting from the endmembers provided by the different implementations of N-FINDR tested in this letter. Two physical constraints are generally imposed on the linear mixture model described in (1): the abundance nonnegativity constraint (ANC), i.e., $\Phi_z(i,j) \geq 0$, and the abundance sum-to-one constraint (ASC), i.e., $\sum_{z=1}^{p} \Phi_z(i,j) = 1$. In this letter, we have conducted our least squares-based estimations of fractional abundances without imposing such constraints. Our assumption is that, if the endmembers are properly selected, the ASC and ANC should naturally hold in the resulting estimations. However, when an unconstrained model is applied, this may not be the case, and the resulting estimations may actually comprise negative fractions or sets of fractions summing less than or above one for a certain pixel. To account for these effects, a new error measure is introduced to evaluate the quality of fractional abundance estimations on a pixel-by-pixel basis as follows:

$$\epsilon_\Phi = \frac{1}{r \times c \times p} \sum_{i=1}^{r} \sum_{j=1}^{c} \left| \left(1 - \sum_{z=1}^{p} |\Phi_z(i,j)| \right) \right|. \tag{3}$$

It is worth noting that the error measure defined by (3) does not require ground-truth information in order to provide an idea about the quality of the fractional abundance of endmembers. Instead, ϵ_Φ measures the quality of the fractional abundance estimations by applying an unconstrained linear mixture model using the endmembers provided by the different methods tested and then comparing how much the fractional abundance estimates provided by the unconstrained model deviate from those provided by a fully constrained linear mixture model using exactly the same set of endmembers.

Fig. 7 shows the box and whisker plots corresponding to the fractional abundance estimation errors measured by (3) for the different implementations of N-FINDR. In all cases, the MNF transform is used for feature reduction. From Fig. 7, it can be observed that the proposed optimizations may also reduce the fractional abundance estimation error quantified by (3). This is particularly the case for the Cuprite scene, in which the proposed optimizations provide significantly lower fractional abundance estimation errors (the optimizations also provided higher volume estimations and better spectral purity scores in previous experiments). On the other hand, the reduction in abundance estimation errors is not as apparent for the Indian Pines scene. This may be related with the known predominance of mixed pixels in this scene due to the early growth stage of agricultural features in the area, which complicates the selection of pure spectral signatures from the scene.
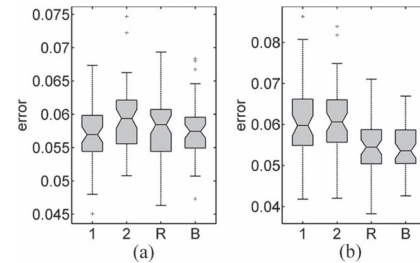

Fig. 7. Experiment 3. Errors ϵ_Φ in fractional abundance estimation for the AVIP and AVCP scenes by the different implementations of N-FINDR. (a) AVIP. (b) AVCP.

IV. CONCLUSION

In this letter, we have evaluated the performance of two different implementations of the N-FINDR algorithm and further proposed optimizations for improving endmember extraction performance. The proposed optimizations do not increase the computational complexity of the algorithm. Our experimental results indicate that the proposed optimizations can provide solutions with larger volume, higher spectral purity, and better representation of physical closure constraints in fractional abundance estimation using a linear mixture model.

REFERENCES

[1] R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing,* 2nd ed. New York: Academic, 1997.
[2] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 650–663, Mar. 2004.
[3] Q. Du, N. Raksuntorn, N. Younan, and R. King, "End-member extraction for hyperspectral image analysis," *Appl. Opt.*, vol. 47, no. 28, pp. 77–84, Oct. 2008.
[4] M. Winter, "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," *Proc. SPIE*, vol. 3753, pp. 266–270, Oct. 1999.
[5] A. Plaza and C.-I Chang, "Impact of initialization on design of endmember extraction algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3397–3407, Nov. 2006.
[6] X. Tao, B. Wang, and L. Zhang, "Orthogonal bases approach for the decomposition of mixed pixels in hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 219–223, Apr. 2009.
[7] J. M. P. Nascimento and J. M. Bioucas-Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.
[8] A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geosci. Remote Sens.*, vol. 26, no. 1, pp. 65–74, Jan. 1988.
[9] C.-I Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 608–619, Mar. 2004.
[10] R. McGill, J. Tukey, and W. Larsen, "Variations of box plots," *Amer. Stat.*, vol. 32, no. 1, pp. 12–16, Feb. 1978.