

A New Elbow Estimation Method for Selecting the Best Solution in Sparse Unmixing

Risto Sarjonen  and Tomi Rätty, *Member, IEEE*

Abstract—The goal of hyperspectral image analysis is often to determine which materials, out of a given set of possibilities, are present in each pixel. As hyperspectral data are being gathered in rapidly increasing amounts, automatic image analysis is becoming progressively more important. Automatic identification of materials from a mixed pixel is possible with 1) Bayesian unmixing algorithms and 2) multiobjective sparse unmixing algorithms when a method such as elbow estimation is used to select the best solution from the set of Pareto-optimal solutions. We develop a new elbow estimation method called termination condition adaptive elbow (TCAE) for selecting the best solution from the set of Pareto-optimal solutions to a biobjective unmixing problem. Specifically, the two objectives are assumed to be the sparsity level of the fractional abundance vector and the reconstruction error. We conduct experiments with real-world unmixing applications in mind, and TCAE performs significantly better than a state-of-the-art elbow estimation method when they are both used to select the best solution from the sequence of fractional abundance vectors generated by iterative spectral mixture analysis (ISMA). Furthermore, the combination of ISMA and TCAE is able to identify endmembers from mixed pixels several times faster and with higher F1-score than the two Bayesian unmixing algorithms used as a reference. We conclude that the combination of ISMA and TCAE facilitates automatic, reliable, and rapid identification of endmembers from mixed pixels.

Index Terms—Hyperspectral imaging, remote sensing, sparse unmixing, spectral mixture analysis.

I. INTRODUCTION

HYPERSPECTRAL cameras are used for close-range imaging in addition to remote sensing from satellites and manned aircraft. Specifically designed instruments may also be mounted on unmanned aerial vehicles. As a result, hyperspectral imaging sees wide use in many different fields, including geology, precision agriculture, food quality control, medical diagnosis, forensics, and military applications [1], [2], [3]. Furthermore, the use of supercontinuum lasers for illumination has been studied in the recent decade, and in the future, they may enable more widespread use of hyperspectral imaging in underground mines and other applications that have challenging illumination conditions [4].

Manuscript received 21 December 2022; revised 13 February 2023 and 10 March 2023; accepted 8 April 2023. Date of publication 19 April 2023; date of current version 15 May 2023. This work was supported by the Business Finland projects Real-Time AI-Supported Ore Grade Evaluation for Automated Mining (RAGE) and Next Generation Mining (NGMining). (*Corresponding author: Risto Sarjonen.*)

Risto Sarjonen is with the VTT Technical Research Centre of Finland Ltd, 02044 Espoo, Finland (e-mail: risto.sarjonen@vtt.fi).

Tomi Rätty is with the Microsoft, Redmond, WA 98052 USA.
Digital Object Identifier 10.1109/JSTARS.2023.3267466

Numerous methods for hyperspectral image analysis have been presented [5]. Classification algorithms assign a single label to every pixel. Thus, materials are left unidentified in the case of mixed pixels. Furthermore, a classification algorithm may yield a label that does not correspond to any of the endmembers occupying a mixed pixel, as it is difficult to classify a mixed pixel by comparing its spectral signature with the individual endmember spectra [5].

Unmixing methods have the advantage that they can identify multiple endmembers from a single pixel and estimate their abundances [6], [7], [8], [9]. Abundance estimates tend to be the most accurate when the endmembers participating in the given pixel have been correctly identified [10]. As the endmember spectra are assumed to be known in semisupervised unmixing, identifying the endmembers from a pixel equates to determining the smallest subset of endmembers with which all essential features of the pixel spectrum y can be modeled.

Several different approaches have been developed for finding the subset of endmembers that neither overfits nor underfits the pixel spectrum y . Greedy and relaxation-based unmixing algorithms select the best subset with the help of hyperparameters whose values need to be determined by a human expert. However, the optimal values of the hyperparameters are typically data dependent and difficult to find [11], [12], [13], [14]. Furthermore, the need for expert intervention reduces the speed at which a large number of pixels can be analyzed.

Automatic identification of endmembers from a mixed pixel is possible with multiobjective sparse unmixing algorithms when a method such as elbow estimation is used to select the best solution from the set of Pareto-optimal solutions [15], [16]. In addition, Bayesian unmixing algorithms can automatically identify which materials occupy a mixed pixel [9].

Angle-based elbow estimation (ABEE) [14] has been used to select the best solution from the set of Pareto-optimal solutions in many studies, e.g., [17], [18], [19], [20]. However, ABEE was criticized in [11] by stating that it sometimes selects wrong solutions or fails to unambiguously identify the best solution. Typical downsides of Bayesian unmixing algorithms include computational heaviness and general complexity [21], [22]. As hyperspectral data are being gathered in rapidly increasing amounts [23], automatic identification of materials from mixed pixels warrants further research.

The main contribution of this article is a method that enables automatic, reliable, and rapid identification of materials from mixed pixels. Specifically, we have developed a new elbow estimation method called termination condition adaptive elbow

(TCAE), which has been combined with iterative spectral mixture analysis (ISMA) [10] in this article. However, TCAE can be used to select the best solution from the set of Pareto-optimal solutions generated by any unmixing algorithm, whereby TCAE is widely usable in sparse unmixing.

The rest of this article is organized as follows. Section II describes the theoretical and methodological background of our work, while Section III presents TCAE. Section IV describes the chosen performance metrics, while Section V focuses on experiments with synthetic data. Sections VI–VIII cover real-data experiments. Section IX examines the performance of TCAE with an ablation experiment and presents a runtime comparison between the algorithms. Section X discusses the novelty and wider applicability of TCAE in addition to suggesting future research directions. Finally, Section XI concludes this article.

II. BACKGROUND

The linear mixing model may be written as

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{n} \quad (1)$$

in which $\mathbf{y} \in \mathbb{R}^m$ is the mixture spectrum, the dictionary $\Phi \in \mathbb{R}^{m \times n}$ contains the endmember spectra as columns, $\mathbf{x} \in \mathbb{R}^n$ is the fractional abundance vector, and $\mathbf{n} \in \mathbb{R}^m$ represents noise. Only nonnegative abundances are physically meaningful, whereby the abundance nonnegativity constraint (ANC) is typically imposed. In addition, it is sometimes required that the components of \mathbf{x} must sum to 1. The use of the abundance sum-to-one constraint (ASC) has been criticized because it is rarely satisfied in other than simulation data [7], [8].

Deviations from the linear mixing model arise if photons scatter several times before being detected. Such multiple scattering is especially common in the case of *intimate mixtures*, in which the materials are mixed on spatial scales smaller than the path length of photons in the mixture. Spectral preprocessing methods [24] and nonlinear mixing models, such as bilinear models, can be used to deal with the effects that arise from multiple scattering. We will return to multiple scattering in Section VI.

In many unmixing algorithms, each target material is represented by a single endmember spectrum. However, in reality, the spectral signature of a particular material may vary within a single scene [25]. One way to account for such *endmember variance* is to represent each target material with a set of spectra instead of one spectrum. For example, the MESMA algorithm and its variations are based on this principle [13], [26]. We will return to endmember variance in Section X-A.

A. Sparse Unmixing

In terms of the linear mixing model (1), the goal in sparse unmixing is to find such sparse \mathbf{x} that $\Phi \mathbf{x} \approx \mathbf{y}$. Thus, sparse unmixing has two fundamental questions. The first question is how accurately the sparse approximation $\Phi \mathbf{x}$ must match the mixture spectrum \mathbf{y} . Subsequently, one must find the smallest subset of endmembers that gives the desired accuracy.

In the equation form, the second step may be written as

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \|\mathbf{y} - \Phi \mathbf{x}\| \leq \delta \quad (2)$$

in which $\|\cdot\|_0$ yields the number of nonzero components in the argument vector, $\|\cdot\|$ is the Euclidean norm, and $\delta \geq 0$ is the largest allowed reconstruction error. Although ANC and ASC could be imposed in (2), we have omitted them here for simplicity. Since (2) is a combinatorial problem for a given δ , finding the exact solution to (2) is computationally unaffordable even for moderately large dictionaries Φ [13]. To obtain useful solutions in reasonable time, computationally lighter methods have been developed for approximately solving (2).

While greedy and relaxation-based algorithms can be used to approximately solve (2), they contain hyperparameters whose values must be set before the estimated fractional abundance vector is obtained. Chen and Zhang [27] have derived a formula that yields the optimal value of the regularization parameter in nonconvex relaxation of (2). Specifically, the formula yields the optimal value of the regularization parameter as a function of signal-to-noise ratio (SNR). However, we believe that the applicability of the formula is limited mainly to simulations, because SNR is usually not known in real-world measurements. Thus, it remains problematic that the values of the hyperparameters in greedy and relaxation-based unmixing algorithms must be set before the estimated fractional abundance vector is obtained.

To avoid the problem posed by hyperparameters, sparse unmixing methods based on multiobjective optimization have been developed [11], [12], [14]. Considering the sparsity level $\|\mathbf{x}\|_0$ and the squared reconstruction error $\|\mathbf{y} - \Phi \mathbf{x}\|^2$ as the objectives, sparse unmixing may be formulated as a biobjective optimization problem

$$\min_{\mathbf{x}} \left(\|\mathbf{x}\|_0, \|\mathbf{y} - \Phi \mathbf{x}\|^2 \right). \quad (3)$$

Since the second objective tends to increase when the first objective is decreased, the biobjective minimization problem (3) does not have a unique solution. A solution \mathbf{x} is called Pareto-optimal if neither of the objectives $\|\mathbf{x}\|_0$ and $\|\mathbf{y} - \Phi \mathbf{x}\|^2$ can be decreased without increasing the other. The solution to the biobjective minimization problem (3) is the set of all Pareto-optimal solutions \mathbf{x} .

The task of choosing the best solution from the set of Pareto-optimal solutions may sometimes be left for a human expert [12], while automatic selection is usually based on elbow estimation [15], [16]. The reason for using elbow estimation is that the Pareto-optimal front of a biobjective minimization problem usually features a region, where a small decrease in one objective causes a large increase in the other objective [16].

B. Iterative Spectral Mixture Analysis

ISMA is a greedy sparse unmixing algorithm that uses backward elimination to find the best subset of endmembers [10]. Algorithm 1 presents the details of ISMA in pseudocode. The first input of ISMA is the mixture spectrum $\mathbf{y} \in \mathbb{R}^m$. The second input is the dictionary $\Phi \in \mathbb{R}^{m \times n}$, which has the n endmember

Algorithm 1: Iterative Spectral Mixture Analysis.

Input 1: Mixture spectrum $\mathbf{y} \in \mathbb{R}^m$
Input 2: Dictionary $\Phi \in \mathbb{R}^{m \times n}$ having the endmember spectra as columns.
Input 3: Threshold $\Delta_{th} \in (0, 1]$ and integer $\eta \geq 1$.
Output: Estimated fractional abundance vector \mathbf{x}

- 1: **Part 1:** Define set $S = \{1, 2, \dots, n\}$, in which n is the number of columns in Φ .
- 2: Initialize the while loop iterator by setting $i = 1$.
- 3: **while** $i \leq n$ **do**
- 4: Set $\mathbf{x}^{(i)} = \operatorname{argmin}_{\mathbf{x}} \|\mathbf{y} - \Phi \mathbf{x}\|$ s.t. $x_j = 0 \forall j \notin S$.
- 5: Set $r_i = \|\mathbf{y} - \Phi \mathbf{x}^{(i)}\|$.
- 6: Set $j = \operatorname{argmin}_{k \in S} x_k^{(i)}$. Here, $x_k^{(i)}$ is the k th component of $\mathbf{x}^{(i)}$.
- 7: Set $S = S \setminus \{j\}$.
- 8: Increase the iterator by setting $i = i + 1$.
- 9: **end while**
- 10: Set $r_{n+1} = \|\mathbf{y}\|$.
- 11: Set $\Delta_i = 1 - r_i/r_{i+1}$ for $i \in \{1, 2, \dots, n\}$.
- 12: **Part 2:** Let v be the largest integer for which $\Delta_i < \Delta_{th}$ for all $i \in \{v, \dots, v + \eta - 1\}$.
- 13: The critical iteration $\text{critIT} = v + \eta$.
- 14: Return $\mathbf{x}^{(\text{critIT})}$ as the output.

spectra as columns. The last inputs are the hyperparameters $\Delta_{th} \in (0, 1]$ and $\eta \in \{1, 2, 3, \dots\}$, which determine the sparsity level of the estimated fractional abundance vector \mathbf{x} .

Steps 3–9 of ISMA constitute a while loop that carries out the backward elimination. Specifically, Step 4 fits the mixture spectrum \mathbf{y} using the columns of Φ that are listed in the set S . At the first iteration of the while loop, this amounts to using the whole dictionary Φ , because S was initialized to contain all integers from 1 to n in Step 1.

Step 5 stores the reconstruction error $r_i = \|\mathbf{y} - \Phi \mathbf{x}^{(i)}\|$ for later use in Step 11. Step 6 finds out which component of the fractional abundance vector $\mathbf{x}^{(i)}$ has the smallest value, whereafter Step 7 removes the corresponding index from the set S . Thus, the endmember that had the smallest abundance will not be used for fitting in Step 4 at the subsequent iterations of the while loop.

As a result, endmembers corresponding to negative abundances get eliminated first. Note that this is a desired feature in ISMA, because negative abundances are unphysical, i.e., they are not possible in reality. At some point, the elimination results in a situation in which all abundances are nonnegative. Then, the endmember corresponding to the smallest abundance is likely used to fit noise or is otherwise the least important for a good fit. Thus, the endmembers that have contributed the most to the mixture spectrum \mathbf{y} are likely to get eliminated last.

We note that the solution $\mathbf{x}^{(i)}$ in Step 4 must be unique, because otherwise the endmember selected for elimination in Step 6 will be arbitrary. Consequently, the dictionary Φ must be undercomplete, i.e., it must have less columns than rows.

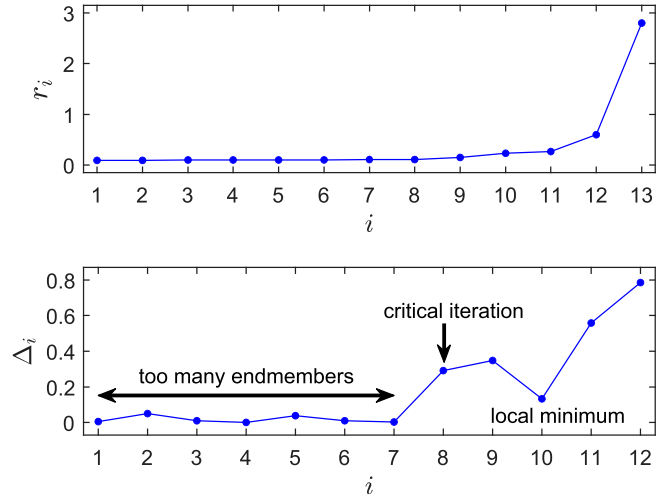


Fig. 1. Upper panel schematically illustrates how the reconstruction error r_i might increase at consequent iterations. The lower panel shows the corresponding Δ_i values that are calculated in Step 11 of ISMA.

The Δ_i values calculated in Step 11 of ISMA describe how the reconstruction error r_i increases over consequent iterations. Specifically, Δ_i indicates how large the relative change in reconstruction error was between iterations i and $i + 1$. For the purposes of calculating Δ_n , we use the definition $r_{n+1} = \|\mathbf{y}\|$ made in Step 10. Small and large values of Δ_i correspond to small and large relative changes, respectively.

The upper panel in Fig. 1 illustrates how the reconstruction error r_i might increase at consequent iterations. The lower panel in Fig. 1 shows the corresponding Δ_i values calculated in Step 11 of ISMA. The total number of endmembers is $n = 12$, while it is assumed that the mixture spectrum \mathbf{y} is a linear combination of five endmember spectra plus noise.

At the early iterations in Fig. 1, the Δ_i values are small, because the endmembers that were eliminated from the set S fitted only noise. The first prominent increase in the Δ_i values occurs when one of the mixture components is eliminated from S (Step 7 at iteration $i = 8$). The reason is that the remaining endmembers can no longer model all major features of the mixture spectrum \mathbf{y} .

A human expert must specify a threshold Δ_{th} that separates the small Δ_i values from the larger values that arise when mixture components are removed from the set S . Furthermore, the human expert must specify a nonnegative integer η , which determines how resistant the search for the critical iteration will be against local minima.

Part 2 of ISMA determines the critical iteration with the help of the hyperparameters Δ_{th} and η . Starting from the last iteration, one finds the first η consecutive Δ_i values that are smaller than Δ_{th} . Assuming that the η th value was found from iteration v , the critical iteration critIT is taken to be $v + \eta$.

The endmembers indexed in S at the beginning of iteration critIT are considered to form the optimal endmember set. ISMA returns the corresponding fractional abundance vector $\mathbf{x}^{(\text{critIT})}$ as the output.

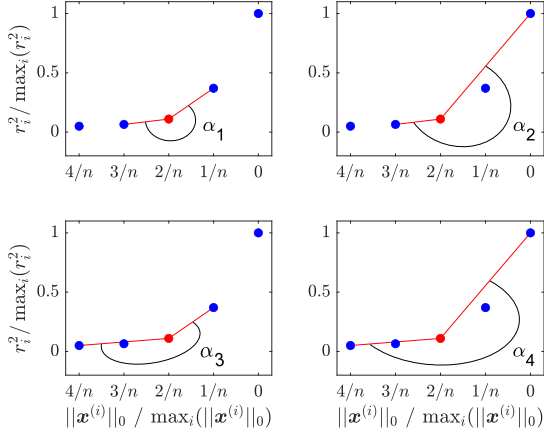


Fig. 2. ABEE uses scaled values of the objectives $\|\mathbf{x}\|_0$ and $\|\mathbf{y} - \Phi\mathbf{x}\|^2$ in the calculation of the angles α_1 , α_2 , α_3 , and α_4 . The definition $r_i = \|\mathbf{y} - \Phi\mathbf{x}^{(i)}\|^2$ is used in the y -axis.

C. Angle-Based Elbow Estimation

As mentioned in Section I, ABEE [14] is widely used for selecting the best solution from the set of Pareto-optimal solutions to (3). The incomparable scales of the objectives $\|\mathbf{x}\|_0$ and $\|\mathbf{y} - \Phi\mathbf{x}\|^2$ in (3) complicate the elbow estimation. To avoid this problem, the values of both objectives are scaled with their maximum values before estimating the location of the elbow. That is, ABEE seeks the elbow from the points

$$P_i = \left(\frac{\|\mathbf{x}^{(i)}\|_0}{\max_i \|\mathbf{x}^{(i)}\|_0}, \frac{\|\mathbf{y} - \Phi\mathbf{x}^{(i)}\|^2}{\max_i \|\mathbf{y} - \Phi\mathbf{x}^{(i)}\|^2} \right) \quad (4)$$

in which $\mathbf{x}^{(i)}$ with $i \in \{1, 2, \dots, n\}$ are the Pareto-optimal solutions of (3).

To find the elbow, the four angles α_1 , α_2 , α_3 , and α_4 defined in [14, Fig. 5] are calculated for each point P_i . The maximum of these four angles is taken to represent the tradeoff of the associated solution $\mathbf{x}^{(i)}$. The solution $\mathbf{x}^{(i)}$ having the largest tradeoff is chosen as the best solution.

In ISMA, ABEE can be used instead of Steps 12 and 13 to determine the critical iteration critIT . To elaborate, let us assume that $\{\mathbf{x}^{(i)} \mid i = 1, 2, \dots, n\}$ is the set of solutions determined by the while loop in ISMA. Furthermore, we define $\mathbf{x}^{(n+1)} = \mathbf{0}$ as the solution corresponding to the reconstruction error $r_{n+1} = \|\mathbf{y}\|^2$ defined in Step 10 of ISMA.

The points P_i with $i \in \{n-3, n-2, \dots, n+1\}$ are schematically illustrated in Fig. 2. In the x -axis, the maximum sparsity level $\max_i(\|\mathbf{x}^{(i)}\|_0)$ is equal to the number of fractional abundance vectors generated by ISMA, namely, n . In the y -axis, $r_i^2 = \|\mathbf{y} - \Phi\mathbf{x}^{(i)}\|^2$ in accordance with Step 5 of ISMA.

For points P_i with $i \in \{3, 4, \dots, n-1\}$, all of the angles α_1 , α_2 , α_3 , and α_4 are calculated. For point P_2 , only angles α_1 and α_2 are calculated, as the calculation of α_3 and α_4 is impossible. Similarly, only angles α_1 and α_3 are calculated for point P_n , because the calculation of α_2 and α_4 is impossible.

According to the principle of ABEE, the maximum of the angles associated with a point P_i is taken to represent the tradeoff of the associated solution $\mathbf{x}^{(i)}$. The value of i corresponding to

the largest tradeoff is taken as the critical iteration critIT . As a result, the solution $\mathbf{x}^{(\text{critIT})}$ returned by ISMA has the largest tradeoff.

D. Spectral Angle Mapper (SAM) Classifier

The spectral angle between two vectors $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^m$ is

$$\theta = \arccos \left(\frac{\sum_{i=1}^m a_i b_i}{\sqrt{\sum_{i=1}^m a_i^2 \sum_{i=1}^m b_i^2}} \right). \quad (5)$$

Let us now denote the angle between a pixel spectrum and the i th endmember spectrum by θ_i . The SAM classifier considers the pixel to consist of the i th endmember if $\theta_i < \theta_{\text{th}}^{(i)}$, in which $\theta_{\text{th}}^{(i)}$ is a user-specified threshold.

Thanks to its simplicity and computational lightness, the SAM classifier is the most commonly used technique for mineral identification from hyperspectral images [28]. Its primary disadvantage is that it requires the estimation of the threshold values $\theta_{\text{th}}^{(i)}$, which is subjective and challenging. Since this article focuses on unmixing algorithms that do not require manual tuning of hyperparameters, we used a simplified version of the SAM classifier that considers a pixel to consist of the k th endmember in which $k = \text{argmin}_i \theta_i$.

E. Nonnegativity Constrained Least Squares (NCLS)

NCLS finds the fractional abundance vector \mathbf{x} that minimizes $\|\mathbf{y} - \Phi\mathbf{x}\|$ subject to ANC. That is, the estimated fractional abundance vector given by NCLS is

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{y} - \Phi\mathbf{x}\| \text{ s.t. } \mathbf{x} \geq 0 \quad (6)$$

in which the greater than or equal to relation is understood componentwise.

We used the MATLAB function `lsqnonneg` as our NCLS implementation. According to MATLAB documentation, `lsqnonneg` uses the NCLS algorithm described in [29].

F. Fully Constrained Least Squares (FCLS)

FCLS finds the fractional abundance vector \mathbf{x} that minimizes $\|\mathbf{y} - \Phi\mathbf{x}\|$ subject to ANC and ASC. That is, the estimated fractional abundance vector given by FCLS is

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{y} - \Phi\mathbf{x}\| \text{ s.t. } \mathbf{x} \geq 0 \text{ and } \mathbf{1}^T \mathbf{x} = 1 \quad (7)$$

in which $\mathbf{1}$ is a column vector of 1s having as many components as \mathbf{x} .

We realized FCLS by using an embedded ASC in our NCLS implementation. Thus, the FCLS algorithm we used is essentially identical with the one that was introduced in [30] and discussed in Section V-A of [31].

G. Sparse Bayesian Learning (SBL) Algorithm

An unmixing algorithm based on SBL was presented in [32]. Henceforth, we will call the algorithm SBL in accordance with [32]. SBL first uses the relevance vector machine (RVM) [33] to select a subset of endmembers, whereafter it

determines the abundances of the selected endmembers with FCLS as RVM does not enforce ANC or ASC.

To discuss how SBL uses RVM to select a subset of endmembers, we denote the fractional abundance vector given by RVM by $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. In contrast to what is stated in [32], the support of \mathbf{x} was taken to be¹

$$S = \{i \mid x_i > \text{sthr}\} \quad (8)$$

with $\text{sthr} = 0.001$. That is, the value of sthr was not zero as implied in [32, p. 645].

After solving the support S , SBL determines the abundances of the endmembers indexed in S with FCLS. That is, FCLS is executed with a dictionary $\tilde{\Phi}$ that contains only those columns of the original dictionary Φ that are indexed in S . The abundances of the endmembers that are not indexed in S are considered to be zero.

However, we have observed instances in which FCLS yields one or more vanishing (i.e., zero) abundances when it is executed with the dictionary $\tilde{\Phi}$. That is, SBL sometimes predicts a vanishing abundance for an endmember indexed in S . Consequently, it is somewhat ambiguous which endmembers SBL actually considers as mixture components. Our choice was to consider all endmembers indexed in S as mixture components. Thereby, we were able to eliminate the effect of FCLS and examine how well the RVM-based basis selection performed individually.

The results presented in [32] were calculated with the RVM implementation developed by Tipping [34]. Following the example of [32], we left all adjustable parameters in the Tipping's implementation to their default values.

H. BI-ICE

BI-ICE is a Bayesian unmixing algorithm [35], which is based on the linear mixing model (1). The point estimate \mathbf{x} for the fractional abundance vector is calculated from [35, eq. (29)]. All components of \mathbf{x} are necessarily positive, because the right-hand side of (29) in [35] is positive for all $\mu^* \in \mathbb{R}$ and $\sigma^* > 0$.

Considering the original purpose of BI-ICE, namely, abundance estimation, it is not problematic that all the components of \mathbf{x} are nonzero. However, an additional rule is needed if one wishes to use BI-ICE for identifying which endmember spectra constitute a given mixture spectrum \mathbf{y} . We constructed the needed rule from the probability distributions that BI-ICE determines for each component of \mathbf{x} . Specifically, [35, eq. (30)] implies that each component $x_i \sim \mathcal{N}_{\mathbb{R}_+^1}(x_i \mid \mu_i^*, \sigma_{ii}^*)$ with μ_i^* and σ_{ii}^* defined in [35, eq. (31) and (32)], respectively. The parameter μ_i^* specifies where the probability density function attains its maximum, whereas $\sqrt{\sigma_{ii}^*}$ describes the width of the distribution.

Accordingly, we considered the i th endmember to be present in the mixture if

$$\mu_i^* > 3\sqrt{\sigma_{ii}^*}. \quad (9)$$

Fundamentally, (9) means that the i th endmember is considered to be present in the mixture if the associated probability distribution attains its maximum sufficiently far from zero in

Algorithm 2: Automatic Iterative Spectral Mixture Analysis.

Input 1: Mixture spectrum $\mathbf{y} \in \mathbb{R}^m$

Input 2: Dictionary $\Phi \in \mathbb{R}^{m \times n}$ having the endmember spectra as columns.

Input 3: String `critName`, which specifies the termination condition that will be used to determine the critical iteration.

Output: Estimated fractional abundance vector \mathbf{x}

- 1: Execute Part 1 of ISMA.
 - 2: Determine critical iteration `critIT` with the termination condition specified in the string `critName`.
 - 3: Return $\mathbf{x}^{(\text{critIT})}$ as the output.
-

comparison to the width of the distribution. Specifically, we chose the constant 3 in (9) in analogy to the three-sigma rule [36].

Themelis et al. [35] reported that establishing a formal convergence criterion for BI-ICE had been cumbersome. We required BI-ICE to execute at least five iterations, whereafter the convergence criterion was that a new iteration did not change any of the fractional abundances by more than 10^{-4} . According to our tests, the chosen value of 10^{-4} should be so small that choosing a smaller value should not have any appreciable effect on the unmixing results.

III. PROPOSED ALGORITHMS

In addition to TCAE, ABEE can be used to automatically determine the critical iteration in ISMA (see Section II-C). Therefore, we need formalism for dealing with algorithms that automatically determine the critical iteration in ISMA.

Automatic ISMA (AISMA) described in Algorithm 2 formally represents the combination of ISMA and some algorithm that automatically determines the critical iteration. For example, we will refer to the combination of ISMA and TCAE as AISMA with TCAE. The first two inputs of AISMA are the mixture spectrum $\mathbf{y} \in \mathbb{R}^m$ and the dictionary $\Phi \in \mathbb{R}^{m \times n}$. The third input specifies which termination condition is used to determine the critical iteration.

Step 1 of AISMA executes Part 1 of ISMA. Thereby, the fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$, reconstruction errors r_1, r_2, \dots, r_{n+1} and fractional changes $\Delta_1, \Delta_2, \dots, \Delta_n$ become defined. Thereafter, Step 2 determines the critical iteration `critIT` by calling the termination condition specified in the third input.

A. Termination Condition Adaptive Elbow

TCAE is the main contribution of our research, and it is described in Algorithm 3. TCAE is called in Step 2 of AISMA to determine the critical iteration `critIT`. To illustrate how TCAE automatically determines the critical iteration `critIT`, we use dictionary Φ_2 . The properties of dictionary Φ_2 have been described in Table I.

¹Additional details were received directly from the first author of [32].

TABLE I
DICTIONARIES USED IN THE SIMULATIONS

	m	n	$\mu(\Phi)$
Φ_1	200	3	0.9817
Φ_2	200	12	0.9916
Φ_3	200	22	0.9995
Φ_4	200	40	0.9995

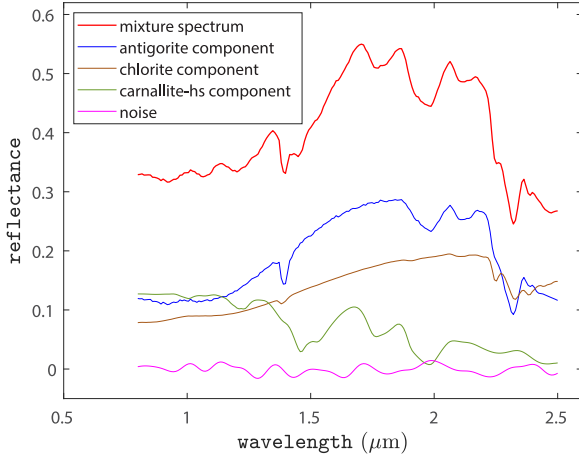


Fig. 3. Red curve is the mixture spectrum \mathbf{y} that we use to illustrate the functionality of AISMA and TCAE. Also shown are the three components of \mathbf{y} and the noise term.

The mixture spectrum \mathbf{y} we use in this illustration is a linear combination of three endmember spectra plus correlated noise. Specifically, the mixture components are antigorite, chlorite, and carnallite-hs with fractional abundances of 55%, 30%, and 15%, respectively. The mixture spectrum \mathbf{y} and its components plus the noise term are shown in Fig. 3.

TCAE takes the fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ and fractional changes $\Delta_1, \Delta_2, \dots, \Delta_n$ determined by ISMA as inputs. The values of $\Delta_1, \Delta_2, \dots, \Delta_{12}$ are shown in Fig. 4. We recall that ISMA calculated each of the Δ_i values by leaving out the mineral that corresponded to the smallest abundance. The name of the excluded mineral is shown at the upper horizontal axis.

The black dots in Fig. 4 indicate the iterations in which excess minerals are included in the unmixing solution. The red triangles indicate the iterations in which all the remaining minerals are represented in the mixture spectrum \mathbf{y} . Note that ISMA removed all other minerals before starting to exclude endmembers which were in the mixture.

The fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(6)}$ determined by ISMA contain negative components. Therefore, one must set $\delta_i = 0$ for $i = 1, 2, \dots, 6$ in Step 1 of TCAE. Subsequently, the $\tilde{\Delta}_i$ values are calculated in Step 2 of TCAE. The δ_i and $\tilde{\Delta}_i$ values are shown in the upper and lower panels of Fig. 5, respectively.

The first iteration of the while loop in TCAE is illustrated in Fig. 6. The upper panel shows the line $L(z)$ defined in Step 5 for the values of $i = 1$ and $j = 12$ set in Steps 12 and 10, respectively. The area shaded in blue is A_L , which is defined

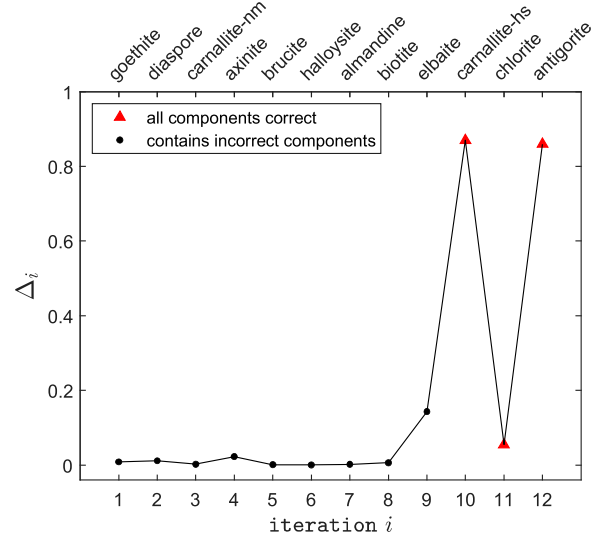


Fig. 4. Iterationwise results from the application of ISMA on the mixture spectrum \mathbf{y} depicted in Fig. 3.

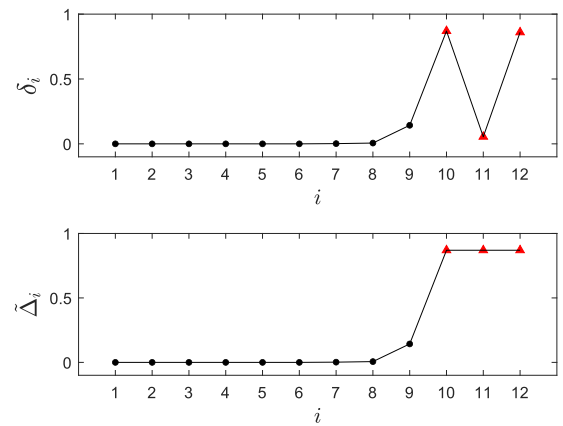


Fig. 5. Upper and lower panels show the δ_i and $\tilde{\Delta}_i$ values calculated in Steps 1 and 2 of TCAE.

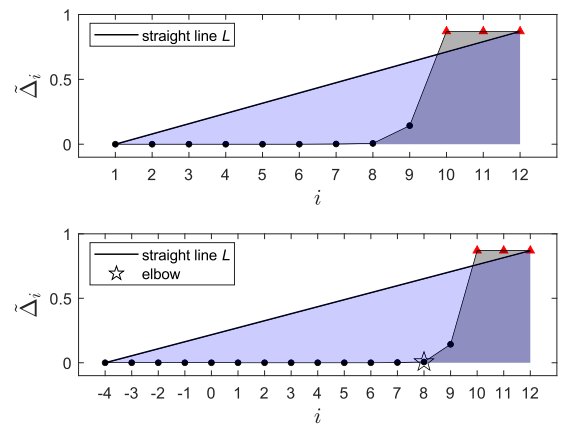


Fig. 6. Upper panel shows the straight line $L(z)$ at the beginning of the first iteration of the while loop in TCAE. The areas shaded in blue and gray are A_L and $A_{\tilde{\Delta}}$, respectively. The lower panel shows the straight line $L(z)$ after Step 14. The location of the elbow determined in Step 18 is shown as a five-pointed star.

Algorithm 3: Termination Condition Adaptive Elbow.

Input 1: Fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$
Input 2: Fractional changes $\Delta_1, \Delta_2, \dots, \Delta_n$
Output: Critical iteration `critIT`

- 1: First define $\delta_i = \Delta_i$ for all $i \in \{1, 2, \dots, n\}$, and then set $\delta_i = 0$ for all i for which $\mathbf{x}^{(i)}$ contains at least one negative component. Thereby, δ_i vanishes whenever the associated abundance vector $\mathbf{x}^{(i)}$ contains unphysical values.
- 2: Define $\tilde{\Delta}_i = \max_{j \in \{1, 2, \dots, i\}} \delta_j$ for $i \in \{1, 2, 3 \dots, n\}$. This may be perceived as smoothing that eliminates local minima.
- 3: Extend the definition of $\tilde{\Delta}_i$ to nonpositive integers i by setting $\tilde{\Delta}_i = 0$ for $i \in \{0, -1, -2, \dots\}$.
- 4: Hereafter it is assumed that the integer valued variables i and j follow the constraint $i < j \leq n$.
- 5: Let us consider a straight line connecting the points $(i, \tilde{\Delta}_i)$ and $(j, \tilde{\Delta}_j)$. The said line $L(z) = \tilde{\Delta}_i + k(z - i)$ with the slope $k = (\tilde{\Delta}_j - \tilde{\Delta}_i)/(j - i)$.
- 6: Let us consider the area of the rectangular triangle having $L(z)$ as the hypotenuse. The area of the said triangle $A_L = (L(j) - L(i))(j - i)/2$.
- 7: Let us consider linear interpolation of the points $\{(i, \tilde{\Delta}_i), (i + 1, \tilde{\Delta}_{i+1}), \dots, (j, \tilde{\Delta}_j)\}$. The area under the said curve is $A_{\tilde{\Delta}} = \sum_{m=i}^{j-1} (\tilde{\Delta}_m + \tilde{\Delta}_{m+1})/2$.
- 8: Define constant $R = 3$. It is used in Steps 13 and 15. For a discussion, see the last three paragraphs of Section III-A.
- 9: Set initial value `critIT` = 1. Thus, all endmembers belong to the initial unmixing solution. The while loop starting from Step 11 will refine this initial solution.
- 10: Set initial value $j = n$. Effectively, this fixes the right end of the line $L(z)$ to the point $(n, \tilde{\Delta}_n)$.
- 11: **while** `critIT` < j **do**
- 12: Set $i = 1$. Effectively, this fixes the left end of the line $L(z)$ to the point $(1, \tilde{\Delta}_1)$.
- 13: **if** $A_L/A_{\tilde{\Delta}} < R$ **then**
- 14: Decrease i until $A_L/A_{\tilde{\Delta}}$ is as close to R as possible. Note that decreasing the value of i may be understood as moving the left end of the line $L(z)$ to left. Thus, decreasing i increases the area of the rectangular triangle having $L(z)$ as hypotenuse.
- 15: **else if** $A_L/A_{\tilde{\Delta}} > R$ **then**
- 16: Increase i until $A_L/A_{\tilde{\Delta}}$ is as close to R as possible. Note that increasing the value of i may be understood as moving the left end of the line $L(z)$ to right. Thus, increasing i decreases the area of the rectangular triangle having $L(z)$ as hypotenuse.
- 17: **end if**
- 18: The values of i and j are now fixed and the line $L(z)$ extends from $(i, \tilde{\Delta}_i)$ to $(j, \tilde{\Delta}_j)$. The line $L(z)$ is used to find the elbow from the values $\tilde{\Delta}_i, \dots, \tilde{\Delta}_j$. Thereby, one obtains `elbow` = $\operatorname{argmax}_{i \leq m \leq j} (L(m) - \tilde{\Delta}_m)$.
- 19: **if** `critIT` < `elbow` + 1 **then**
- 20: Set `critIT` = `elbow` + 1. The iteration following the elbow is taken as the critical iteration. Due to Step 19, the value of `critIT` is updated only if the new value is larger than the old one.
- 21: **end if**
- 22: Set $j = j - 1$. The right end of the line $L(z)$ is moved one step to the left.
- 23: **end while**
- 24: Return `critIT` as the output.

in Step 6 as the area of a rectangular triangle having the line $L(z)$ as the hypotenuse. In comparison, the area shaded in gray is $A_{\tilde{\Delta}}$, which is defined in Step 7 as the area under the linear interpolation of the points $\tilde{\Delta}_m$ with $i \leq m \leq j$.

With the initial values of $i = 1$ and $j = 12$, the ratio $A_L/A_{\tilde{\Delta}} \approx 2.06 < R$, in which $R = 3$ as defined in Step 8. Therefore, Step 14 adjusts $A_L/A_{\tilde{\Delta}}$ to 2.99 by moving the left end of the line $L(z)$ to $i = -4$. The lower panel of Fig. 6 shows $L(z)$ after Step 14. The five-pointed star indicates the location of the elbow determined in Step 18 [15], [37], [38], [39].

Steps 19 and 20 perform the update `critIT` = `elbow` + 1 = 9, because the initial value `critIT` = 1 < `elbow` + 1 = 9. Thus, the unmixing solution determined at the first iteration of the while loop includes one excess mineral, namely elbaite.

The variable j is decreased to $12 - 1 = 11$ in Step 22. The second iteration of the while loop ensues and again the ninth iteration is determined as the critical iteration. Subsequently, j is decreased to $11 - 1 = 10$ in Step 22, and the third iteration of the while loop commences.

The situation at the third iteration of the while loop is illustrated in Fig. 7. The upper panel shows the line $L(z)$ for the initial

values of $i = 1$ and $j = 10$. Since the ratio $A_L/A_{\tilde{\Delta}} \approx 6.68 > R$, Step 16 adjusts $A_L/A_{\tilde{\Delta}}$ to 2.97 by moving the left end of the line $L(z)$ to 6. The lower panel of Fig. 7 shows $L(z)$ after Step 16. The five-pointed star indicates the value `elbow` = 9 determined in Step 18. Since `critIT` = 9 < `elbow` + 1 = 10, Steps 19 and 20 perform the update `critIT` = `elbow` + 1 = 10.

Thereafter, Step 22 decreases j to 9 and the while loop terminates because the condition `critIT` < j is no longer true. As a result, Step 24 returns the tenth iteration as the critical iteration. Thus, AISMA managed to identify all the endmember spectra used to construct the mixture spectrum \mathbf{y} , while no excess endmembers were included in the unmixing solution.

To summarize, TCAE determines the critical iteration with the assistance of elbow point estimation. Elbow detection is an inherently heuristic process, which may be understood as finding the point in which a curve changes from steep to flat [37], [38], [39]. However, the last two $\tilde{\Delta}_i$ values depicted in Fig. 6 do not contribute to the steep part, because they are equal to the third last $\tilde{\Delta}_i$ value. Consequently, we believe that a human would intuitively disregard the last two $\tilde{\Delta}_i$ values when visually searching for the elbow. That is, a human would likely start

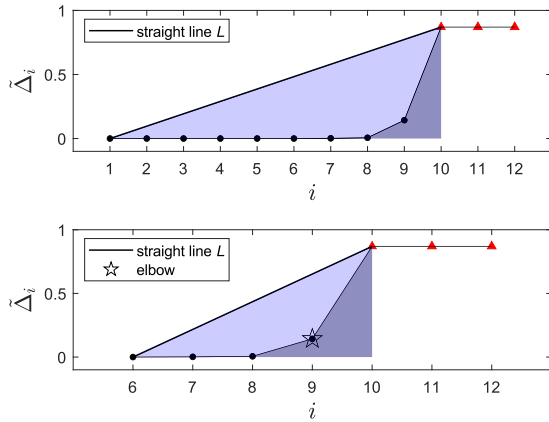


Fig. 7. Upper panel shows the straight line $L(z)$ at the beginning of the third iteration of the while loop in TCAE. The areas shaded in blue and gray are A_L and $A_{\tilde{\Delta}}$, respectively. The lower panel shows the straight line $L(z)$ after Step 16. The location of the elbow determined in Step 18 is shown as a five-pointed star.

searching for the elbow in the way depicted in the upper panel of Fig. 7.

Elbow detection algorithms usually work the best if the steep and flat parts of the curve have roughly equal lengths [38], [39]. Relatedly, we believe that a human intuitively disregards some of the earlier iterations if the tail of small $\tilde{\Delta}_i$ appears to be excessively long. Similarly, we believe that a human mentally extends the tail of small $\tilde{\Delta}_i$ values to the left if a longer tail of small $\tilde{\Delta}_i$ values is needed to make the elbow apparent. In particular, the extension of the tail may be necessary if the dictionary contains only a handful of endmembers and they all have contributed to the given mixture spectrum.

The constant R defined in Step 8 controls the length of the tail formed by the small $\tilde{\Delta}_i$ values. Specifically, large and small values of R favor Steps 14 and 16, respectively. Thus, large and small values of R lead to long and short tails, respectively. We chose to fix $R = 3$, because we believe that the resulting tails of small $\tilde{\Delta}_i$ values are approximately as long as a human would consider when intuitively searching for the elbow.

IV. PERFORMANCE METRICS

We consider the quality of unmixing from two complementary perspectives: 1) quantitative abundance estimation and 2) endmember identification. To discuss these approaches in greater detail, we assume that $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ and $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)^T$ are the ground truth fractional abundance vector and its estimate, respectively.

In quantitative abundance estimation, the estimate $\hat{\mathbf{x}}$ is considered to be better the closer it is to \mathbf{x} in some continuous sense. We measure the goodness of $\hat{\mathbf{x}}$ in quantitative abundance estimation with the relative L^2 error

$$\text{RL2E} = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2}. \quad (10)$$

Note that RL2E is the reciprocal of the square root of the signal-to-reconstruction error used in [7] and [8]. We use RL2E instead

of the root-mean-square error (RMSE) because we believe that a large relative error usually renders an abundance estimate useless even if the absolute error is small.

In endmember identification, the goal is to determine which endmembers are present in a given pixel. That is, the goal is to decipher the support of \mathbf{x} , namely

$$\text{support}(\mathbf{x}) = \{i \mid x_i > 0\} \quad (11)$$

from the mixture spectrum \mathbf{y} [40]. By bearing in mind the linear mixing model (1), one may view endmember identification as basis selection from the dictionary Φ [41].

We use recall, precision, and F_1 -score to measure how well the support of $\hat{\mathbf{x}}$ matches the support of \mathbf{x} . Together, recall and precision give more information than any single metric such as the distance between the supports of $\hat{\mathbf{x}}$ and \mathbf{x} . At the same time, F_1 -score acts as an overall measure of the goodness of $\hat{\mathbf{x}}$ in endmember identification.

The number of true positives

$$\text{TP} = \|\mathbf{x} \circ \hat{\mathbf{x}}\|_0 \quad (12)$$

in which \circ means componentwise product. The number of false positives

$$\text{FP} = \|\hat{\mathbf{x}}\|_0 - \text{TP}. \quad (13)$$

The number of false negatives

$$\text{FN} = \|\mathbf{x}\|_0 - \text{TP}. \quad (14)$$

By using (12)–(14), we can define recall and precision as follows:

Recall

$$\text{REC} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (15)$$

Precision

$$\text{PRE} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (16)$$

The harmonic mean of recall and precision, namely, F_1 -score

$$F_1 = \frac{2}{1/\text{REC} + 1/\text{PRE}}. \quad (17)$$

V. SIMULATIONS

We examined the performance of the unmixing algorithms with simulations similar to [7]. To cover different types of unmixing scenarios, we used four different dictionaries. The endmember spectra included in the dictionaries are mineral spectra from the ECOSTRESS library [42], [43] and the USGS spectral library [44].

The properties of the dictionaries are summarized in Table I. Table I shows that the number of wavelength bands in each dictionary is $m = 200$. The bands equidistantly span the wavelength range from $\lambda = 0.8 \mu\text{m}$ to $\lambda = 2.5 \mu\text{m}$. The third column shows the number of endmembers in the dictionary, whereas the fourth column shows the mutual coherence [7] of the dictionary. We chose to use dictionaries that entail up to 40 endmembers, because the number of endmembers in a real-life unmixing scenario is often small, say less than 20 [7].

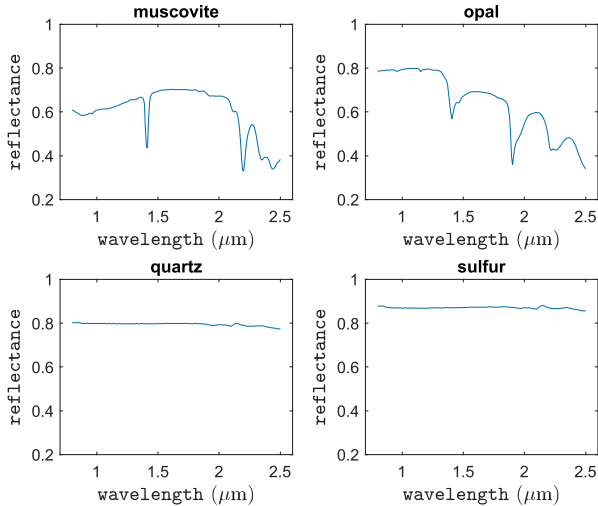


Fig. 8. Spectra of quartz and sulfur are featureless.

TABLE II
SIMULATIONS USED FOR PERFORMANCE COMPARISONS

	dictionary	$\ \mathbf{x}\ _0$	SNR (dB)
simulation 1	Φ_1	1–3	20
simulation 2	Φ_1	1–3	35
simulation 3	Φ_1	1–3	50
simulation 4	Φ_2	1–5	20
simulation 5	Φ_2	1–5	35
simulation 6	Φ_2	1–5	50
simulation 7	Φ_3	1–5	20
simulation 8	Φ_3	1–5	35
simulation 9	Φ_3	1–5	50
simulation 10	Φ_4	1–5	20
simulation 11	Φ_4	1–5	35
simulation 12	Φ_4	1–5	50

The smallest dictionary Φ_1 contains three mineral spectra, namely, the spectra of alunite, muscovite, and zoisite. The dictionary Φ_2 contains the 12 mineral spectra used in [32].

Dictionary Φ_3 contains 22 common rock-forming minerals, which were used in the article that introduced ISMA [10]. However, we did not include the practically featureless spectra of quartz and sulfur in Φ_3 . The reason is that these featureless minerals cannot be detected by hyperspectral imaging [45]. To illustrate the flatness of quartz and sulfur spectra, Fig. 8 shows them along with two mineral spectra included in Φ_3 .

The largest dictionary Φ_4 contains 40 mineral spectra. We required all spectra in Φ_4 to have appreciable absorption features, whereby minerals such as quartz and sulfur were not included in Φ_4 .

We used the 12 simulations detailed in Table II to study the performance of the unmixing algorithms. The second column of Table II shows the dictionary used in the simulation, while the third column shows the sparsity level of the generated fractional abundance vectors \mathbf{x} . We limited the maximum number of mixture components to five, because real-world unmixing scenarios typically entail up to five endmembers per pixel [21], [46].

In each simulation, 1 000 000 fractional abundance vectors \mathbf{x} were generated. For each \mathbf{x} , the number of nonzero components was randomly drawn from the range indicated in the third column of Table II. Thereafter, the indices of the nonzero components were chosen randomly. Finally, the selected components were assigned values drawn from the flat Dirichlet distribution. Hence, each \mathbf{x} satisfies ANC and ASC.

In our view, the use of the flat Dirichlet distribution may be considered analogous to the use of a flat (uninformative) prior in Bayesian analysis. Since we are not aware of any reasons to prefer some specific form of the Dirichlet distribution, we opted to use the flat distribution.

Corresponding to each \mathbf{x} , a mixture spectrum

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{n} \quad (18)$$

was generated. Here, Φ and \mathbf{n} are the dictionary and the noise vector, respectively. We used correlated noise in the simulations because it is a better model of real-world noise than independent and identically distributed (i.i.d.) Gaussian noise [7]. The correlated noise was generated according to [7]: low-pass filtering i.i.d. Gaussian noise with a normalized cutoff frequency of $5\pi/m$, in which m is the number of spectral bands. Since (18) does not entail baseline shifts or other scattering effects, we did not use any spectral preprocessing [47] in the simulations of Table II.

We define the SNR as the power of the signal divided by the power of the noise. That is

$$\text{SNR} = \frac{\|\Phi \mathbf{x}\|^2}{\|\mathbf{n}\|^2}. \quad (19)$$

To adjust the SNR of a mixture spectrum \mathbf{y} , we set $\mathbf{n} = \alpha \tilde{\mathbf{n}}$, with $\alpha \geq 0$ being a parameter and vector $\tilde{\mathbf{n}}$ containing the unscaled noise. The desired SNR was then obtained by setting

$$\alpha = \frac{\|\Phi \mathbf{x}\|}{\|\tilde{\mathbf{n}}\| \cdot \sqrt{\text{SNR}}}. \quad (20)$$

As presented in the fourth column of Table II, we used the SNR values of 20 dB, 35 dB, and 50 dB in our simulations (cf. Section IV-E in [7]).

We note that simulations 1–3 do not belong to the category of sparse unmixing as the other simulations. The reason is that the maximum sparsity level $\|\mathbf{x}\|_0 = 3$ is equal to the number of endmembers in the dictionary Φ_1 . Nevertheless, some hyperspectral images involve only few endmembers, which may all be present in a single pixel. For example, the dataset of [48] contains images in which most or all pixels contain all the three or four possible endmembers. Therefore, we consider simulations 1–3 relevant even though they do not represent sparse unmixing.

A. Simulation Results

The performance metrics describing the quality of endmember identification are shown in Table III. AISMA with TCAE tended to have the largest F_1 -score, and the marginal by which AISMA with TCAE has the largest F_1 -score tends to increase according to SNR.

The critical difference diagram of Fig. 9 shows the average ranks of the unmixing algorithms based on the F_1 -scores shown

TABLE III
AVERAGE ENDMEMBER IDENTIFICATION METRICS IN THE SIMULATIONS

	SAM classifier			FCLS			NCLS			SBL			BI-ICE			AISMA with ABEE			AISMA with TCAE		
	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁
simulation 1	0.60	0.98	0.71	0.96	0.82	0.86	0.96	0.82	0.86	0.94	0.85	0.87	0.93	0.89	0.89	0.61	1.00	0.72	0.74	1.00	0.82
simulation 2	0.60	0.98	0.71	0.99	0.82	0.88	0.99	0.82	0.88	0.99	0.86	0.90	0.99	0.89	0.92	0.61	1.00	0.72	0.94	1.00	0.96
simulation 3	0.60	0.98	0.71	1.00	0.82	0.88	1.00	0.82	0.88	1.00	0.89	0.92	1.00	0.87	0.92	0.61	1.00	0.72	0.99	1.00	0.99
simulation 4	0.40	0.81	0.49	0.87	0.52	0.62	0.84	0.49	0.60	0.77	0.48	0.56	0.77	0.64	0.67	0.55	0.92	0.65	0.61	0.89	0.69
simulation 5	0.40	0.82	0.50	0.97	0.54	0.67	0.96	0.52	0.65	0.94	0.53	0.66	0.94	0.68	0.77	0.59	1.00	0.70	0.88	0.97	0.91
simulation 6	0.40	0.82	0.50	0.99	0.54	0.68	0.99	0.53	0.67	0.99	0.58	0.71	0.99	0.69	0.80	0.59	1.00	0.70	0.97	0.99	0.98
simulation 7	0.34	0.64	0.41	0.72	0.37	0.47	0.66	0.33	0.42	0.61	0.27	0.35	0.55	0.44	0.47	0.44	0.71	0.51	0.44	0.71	0.51
simulation 8	0.36	0.67	0.43	0.92	0.40	0.54	0.90	0.37	0.51	0.86	0.35	0.48	0.85	0.53	0.63	0.58	0.98	0.69	0.79	0.93	0.84
simulation 9	0.36	0.67	0.43	0.98	0.41	0.57	0.98	0.38	0.54	0.96	0.39	0.53	0.95	0.57	0.69	0.59	1.00	0.70	0.95	0.99	0.96
simulation 10	0.32	0.56	0.38	0.67	0.29	0.38	0.62	0.26	0.35	0.58	0.13	0.21	0.50	0.37	0.41	0.42	0.61	0.47	0.42	0.64	0.48
simulation 11	0.33	0.59	0.39	0.90	0.33	0.47	0.89	0.31	0.44	0.85	0.19	0.30	0.83	0.48	0.59	0.64	0.97	0.74	0.79	0.91	0.83
simulation 12	0.33	0.59	0.39	0.98	0.34	0.49	0.98	0.32	0.47	0.96	0.22	0.35	0.93	0.53	0.66	0.66	1.00	0.76	0.95	0.98	0.96

To make a comparison between the sparse unmixing algorithms, the largest value of each metric has been bolded from every row without considering the first three algorithms.

TABLE IV
AVERAGE RELATIVE L^2 ERROR IN THE SIMULATIONS

	full dictionary		ground truth		SBL		BI-ICE		AISMA w/ ABEE		AISMA w/ TCAE	
	FCLS	NCLS	FCLS	NCLS	FCLS	NCLS	ASC	no ASC	FCLS	NCLS	FCLS	NCLS
simulation 1	1.2e-01	1.4e-01	8.7e-02	1.0e-01	1.2e-01	1.4e-01	1.4e-01	1.6e-01	4.3e-01	4.3e-01	2.3e-01	2.4e-01
simulation 2	2.2e-02	2.6e-02	1.6e-02	2.0e-02	2.1e-02	2.5e-02	2.5e-02	2.9e-02	4.1e-01	4.1e-01	2.3e-02	2.6e-02
simulation 3	4.0e-03	4.6e-03	3.0e-03	3.5e-03	3.8e-03	4.5e-03	4.5e-03	5.3e-03	4.1e-01	4.1e-01	3.2e-03	3.8e-03
simulation 4	2.8e-01	3.6e-01	1.2e-01	1.5e-01	3.5e-01	3.9e-01	3.6e-01	4.8e-01	5.3e-01	5.1e-01	4.3e-01	4.2e-01
simulation 5	6.7e-02	9.2e-02	2.5e-02	3.3e-02	6.7e-02	8.5e-02	8.0e-02	1.1e-01	4.2e-01	3.8e-01	6.5e-02	6.9e-02
simulation 6	1.3e-02	1.8e-02	4.6e-03	6.1e-03	1.1e-02	1.5e-02	1.4e-02	2.1e-02	4.1e-01	3.8e-01	6.8e-03	8.2e-03
simulation 7	5.0e-01	7.0e-01	1.7e-01	2.4e-01	6.3e-01	7.2e-01	7.4e-01	9.9e-01	8.0e-01	8.7e-01	7.9e-01	8.3e-01
simulation 8	1.5e-01	2.2e-01	3.8e-02	5.5e-02	1.7e-01	2.0e-01	2.1e-01	3.0e-01	4.5e-01	4.3e-01	1.6e-01	1.6e-01
simulation 9	3.2e-02	4.6e-02	7.1e-03	1.0e-02	5.2e-02	6.1e-02	4.4e-02	8.0e-02	4.3e-01	3.9e-01	1.5e-02	1.7e-02
simulation 10	5.7e-01	7.2e-01	1.5e-01	2.2e-01	6.8e-01	7.9e-01	8.6e-01	1.1e+00	8.1e-01	8.8e-01	8.0e-01	8.4e-01
simulation 11	1.7e-01	2.1e-01	3.3e-02	4.9e-02	1.9e-01	2.2e-01	2.5e-01	3.3e-01	3.7e-01	3.5e-01	1.6e-01	1.6e-01
simulation 12	3.6e-02	4.4e-02	6.2e-03	9.1e-03	4.4e-02	5.1e-02	6.4e-02	9.6e-02	3.4e-01	3.2e-01	1.4e-02	1.6e-02

The smallest value from each row has been bolded without considering the “ground truth” column.

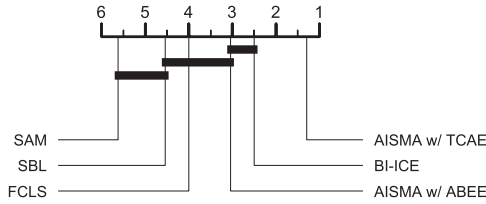


Fig. 9. Critical difference diagram based on the F_1 -scores shown in Table III.

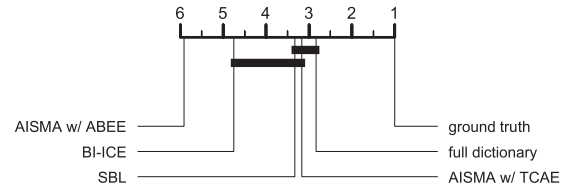


Fig. 10. Critical difference diagram calculated from the RL2E values shown in Table IV.

in Table III. We excluded NCLS from the diagram because FCLS has a larger F_1 -score in most simulations. The horizontal bars in Fig. 9 indicate cliques in which the performance differences of the algorithms are not significant according to the one-sided Wilcoxon signed-rank test with Holm correction on significance level 0.05. AISMA with TCAE performed the best in endmember identification by a significant margin.

In all subsequent critical difference diagrams, the horizontal bars have the same meaning as in Fig. 9. That is, a horizontal bar indicates a clique in which the performance differences of the algorithms are not significant according to the one-sided Wilcoxon signed-rank test with Holm correction on significance level 0.05.

The average RL2E of the estimated fractional abundance vectors is shown in Table IV. Specifically, the shown values are

averages over the 1 000 000 RL2Es obtained in each simulation. The column “full dictionary” reports the FCLS and NCLS results obtained with the whole dictionary. In contrast, the results in the column “ground truth” were obtained by running FCLS and NCLS with the ground truth endmembers. In the columns titled “SBL,” “AISMA w/ ABEE,” and “AISMA w/ TCAE,” the endmembers selected by the respective algorithms were used for FCLS and NCLS unmixing.

The ASC included in FCLS is not satisfied in all real-world unmixing scenarios [7], [8]. Nevertheless, SBL natively uses FCLS for abundance estimation, whereas AISMA uses NCLS. Thus, SBL and AISMA could have different relative L^2 errors even if they would have selected the same endmember subsets. Therefore, we ran both FCLS and NCLS with the endmember

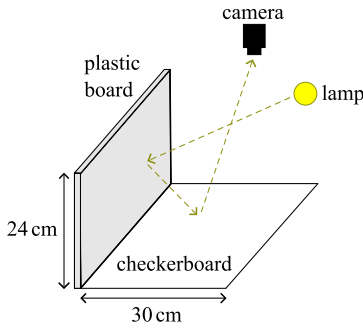


Fig. 11. Schematic illustration of the Scene III setup. Figure adapted from [48].

subsets selected by SBL and AISMA. To further facilitate fair comparison, we always ran BI-ICE with and without an embedded ASC.

The critical difference diagram of Fig. 10 was calculated from the RL2Es shown in Table IV. Specifically, each column contains two RL2Es per simulation, and we always took into account the smaller one.

Unmixing with ground truth endmembers yielded the best abundance estimates by a significant margin, while AISMA with ABEE performed the worst by a significant margin. Furthermore, full dictionary ranked better than BI-ICE, SBL, and AISMA with TCAE. Thus, the endmember subsets selected by the three aforementioned sparse unmixing algorithms did not lead to better abundance estimates than the full dictionary.

VI. EXPERIMENTS WITH CLOSE-RANGE HYPERSPECTRAL IMAGES

The unmixing dataset introduced by Zhao et al. [48] has been specifically designed for unmixing experiments. It consists of close-range hyperspectral images measured in a laboratory and abundance ground truth is available for all pixels.

The images in the Zhao's dataset have been divided into four groups named Scene I, Scene II, Scene II RGB, and Scene III. Scene I samples are printed checkerboards, whereby the linear mixing model should work well. In contrast, the Scene II samples are homogeneous mixtures of colored quartz sands. Since Scene II samples may be regarded intimate mixtures, multiple scattering is expected to reduce the validity of the linear mixing model in Scene II. Scene II RGB samples are nonhomogeneous but otherwise similar to Scene II samples.

As illustrated in Fig. 11, Scene III images were taken from Scene I checkerboards while a plastic board was standing next to the checkerboard. Because of the Scene III setup, double scattering is prominent. That is, most incident photons scatter both from the checkerboard and the plastic board before being detected.

In all three scenes, there are mixtures that contain all endmembers from the scene. To retain focus on sparse unmixing, we opted to use a single dictionary Φ that contains the endmember spectra from all the three scenes. Specifically, we formed each endmember spectrum by averaging over a hyperspectral image that was taken from a sample consisting of the respective

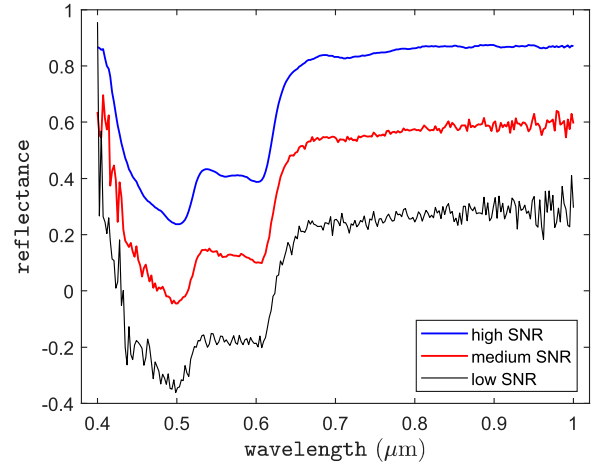


Fig. 12. Illustration of the three different SNR levels with the help of Scene 1 mixture 1. For clarity, the low- and medium-SNR curves have been offset in the vertical direction by -0.6 and -0.3 , respectively.

endmember only. The eight endmember spectra included in Φ have been plotted in [48, Figs. 3, 5, and 8].

We used low-, medium-, and high-SNR mixture spectra in our experiments. The low-SNR spectra correspond to individual pixels, whereas the high-SNR spectra are averages over all the pixels in the image. The medium-SNR spectra were generated by applying a $3 \times 3 \times 1$ box filter on the raw mixture images. Thus, the filter width was three pixels in both spatial dimensions, whereas no averaging was done along the wavelength axis. Fig. 12 illustrates the differences between the three SNR levels by using the Scene I mixture 1 as an example. The low-SNR spectrum corresponds to pixel (30, 30) in the raw image, whereas the medium-SNR spectrum has been taken from the pixel (30, 30) after the $3 \times 3 \times 1$ box filter was applied on the image. The high-SNR spectrum is the average over all the pixels. Visually, the high-SNR spectrum is the smoothest, whereas the low-SNR spectrum appears to contain the largest amount of noise.

In the unmixing experiments with Scene I and Scene III spectra, we did not use any spectral preprocessing. The reason is that the preprocessing methods we tried, namely, linear detrending and logarithm transformation, did not appreciably improve the results. In Scene II and Scene II RGB experiments, we used the generalized logarithm transformation

$$A = -\log_{10} \left(\frac{R + \sqrt{R^2 + c}}{2} \right) \quad (21)$$

with $c = 10^{-6}$ to convert the measured reflectance spectra to absorbance spectra [24]. For more information about the generalized logarithm transformation (21) used in Scene II and Scene II RGB experiments, see Appendix A.

A. Unmixing Results

The endmember identification metrics for Zhao's dataset [48] are shown in Table V. The values shown on each row are averages over all mixture spectra belonging to the indicated scene and SNR level. The first three algorithms are not sparse unmixing algorithms, and therefore, we focused on the last three

TABLE V
AVERAGE ENDMEMBER IDENTIFICATION METRICS FOR THE ZHAO'S DATASET [48]

	SAM classifier			FCLS			NCLS			SBL			BI-ICE			AISMA with ABEE			AISMA with TCAE			
	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	REC	PRE	F ₁	
Scene I																						
low SNR	0.30	0.71	0.42	0.93	0.55	0.68	0.97	0.51	0.67	0.96	0.63	0.75	0.94	0.70	0.79	0.57	0.98	0.70	0.81	0.96	0.86	
med SNR	0.30	0.70	0.42	0.94	0.55	0.68	0.99	0.52	0.68	0.99	0.61	0.74	0.98	0.62	0.75	0.58	1.00	0.71	0.95	0.96	0.95	
high SNR	0.30	0.70	0.42	0.95	0.57	0.70	1.00	0.53	0.68	1.00	0.56	0.71	1.00	0.57	0.72	0.57	1.00	0.70	1.00	1.00	1.00	
Scene II																						
low SNR	0.27	0.62	0.38	0.99	0.59	0.73	0.93	0.52	0.66	0.91	0.70	0.78	0.89	0.72	0.78	0.69	1.00	0.80	0.85	0.99	0.90	
med SNR	0.26	0.60	0.36	1.00	0.58	0.73	0.94	0.53	0.67	0.93	0.64	0.74	0.92	0.64	0.74	0.70	1.00	0.80	0.90	0.96	0.92	
high SNR	0.24	0.57	0.34	1.00	0.58	0.72	0.98	0.54	0.69	0.96	0.59	0.72	0.95	0.58	0.70	0.69	1.00	0.80	0.92	0.86	0.87	
Scene II RGB																						
low SNR	0.33	0.74	0.46	0.98	0.63	0.75	0.96	0.50	0.65	0.96	0.74	0.82	0.93	0.77	0.83	0.68	0.95	0.77	0.84	0.93	0.87	
med SNR	0.33	0.74	0.45	0.99	0.64	0.76	0.98	0.54	0.69	0.98	0.71	0.81	0.97	0.70	0.80	0.70	0.99	0.80	0.93	0.96	0.94	
high SNR	0.25	0.50	0.33	1.00	0.62	0.75	1.00	0.55	0.71	1.00	0.75	0.85	1.00	0.67	0.80	0.71	1.00	0.82	1.00	1.00	1.00	
Scene III																						
low SNR	0.37	0.99	0.53	0.61	0.58	0.58	0.83	0.71	0.76	0.87	0.76	0.80	0.77	0.76	0.76	0.42	0.92	0.55	0.51	0.93	0.63	
med SNR	0.37	1.00	0.54	0.61	0.58	0.58	0.83	0.72	0.76	0.93	0.74	0.81	0.78	0.73	0.75	0.43	0.97	0.57	0.77	0.95	0.83	
high SNR	0.38	1.00	0.54	0.58	0.52	0.54	0.90	0.75	0.81	1.00	0.73	0.83	0.85	0.76	0.80	0.42	1.00	0.57	0.93	0.97	0.94	

To make a comparison between the sparse unmixing algorithms, the largest value of each metric has been bolded from every row without considering the first three algorithms.

TABLE VI
AVERAGE RELATIVE L^2 ERROR FOR THE ZHAO'S DATASET [48]

	full dictionary		ground truth		SBL		BI-ICE		AISMA w/ ABEE		AISMA w/ TCAE	
	FCLS	NCLS	FCLS	NCLS	FCLS	NCLS	ASC	no ASC	FCLS	NCLS	FCLS	NCLS
low SNR	2.7e-01	2.7e-01	6.7e-02	6.9e-02	2.7e-01	2.2e-01	3.4e-01	2.8e-01	5.1e-01	4.8e-01	1.8e-01	1.8e-01
med SNR	2.6e-01	2.3e-01	6.3e-02	6.4e-02	2.6e-01	1.7e-01	3.4e-01	2.8e-01	5.0e-01	4.7e-01	9.3e-02	9.4e-02
high SNR	1.9e-01	1.8e-01	5.3e-02	4.8e-02	2.0e-01	1.2e-01	2.5e-01	2.3e-01	5.2e-01	4.8e-01	5.3e-02	4.8e-02
Scene II												
low SNR	2.6e-01	3.5e-01	2.0e-01	2.4e-01	3.5e-01	3.3e-01	2.7e-01	3.5e-01	5.9e-01	6.4e-01	3.9e-01	3.5e-01
med SNR	2.1e-01	2.8e-01	1.7e-01	2.0e-01	2.9e-01	2.8e-01	2.2e-01	2.9e-01	5.7e-01	5.9e-01	3.1e-01	2.7e-01
high SNR	1.8e-01	2.4e-01	1.5e-01	1.7e-01	2.3e-01	2.4e-01	1.9e-01	2.5e-01	5.7e-01	6.4e-01	2.9e-01	2.5e-01
Scene II RGB												
low SNR	4.3e-01	5.0e-01	4.0e-01	4.1e-01	4.4e-01	4.6e-01	4.9e-01	6.1e-01	5.6e-01	7.0e-01	4.8e-01	5.0e-01
med SNR	3.3e-01	3.7e-01	3.2e-01	3.1e-01	3.3e-01	3.4e-01	3.6e-01	4.5e-01	4.9e-01	5.3e-01	3.4e-01	3.5e-01
high SNR	1.5e-01	1.5e-01	1.4e-01	1.1e-01	1.4e-01	1.1e-01	1.5e-01	1.5e-01	5.2e-01	5.0e-01	1.4e-01	1.1e-01
Scene III												
low SNR	1.1e+00	6.8e-01	3.0e-01	1.3e-01	7.5e-01	5.1e-01	1.2e+00	6.7e-01	6.9e-01	6.8e-01	6.5e-01	6.4e-01
med SNR	1.1e+00	7.1e-01	2.7e-01	1.1e-01	6.8e-01	4.5e-01	1.2e+00	7.1e-01	6.1e-01	6.1e-01	4.1e-01	3.7e-01
high SNR	1.1e+00	7.2e-01	2.3e-01	9.9e-02	5.7e-01	3.6e-01	1.3e+00	7.2e-01	5.2e-01	5.2e-01	4.1e-01	1.6e-01

The smallest RL2E value outside the ground truth column has been bolded for each SNR level.

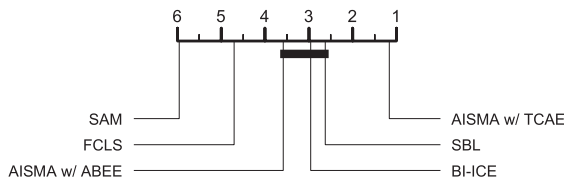


Fig. 13. Critical difference diagram calculated from the F_1 -scores shown in Table V.

algorithms when boldfacing the largest value of each metric from every row.

A critical difference diagram calculated from the F_1 -scores is shown in Fig. 13. AISMA with TCAE had the best F_1 -scores by a significant margin.

The RL2Es for Zhao's dataset [48] are shown in Table VI. The values shown on each row are averages over all mixture spectra belonging to the indicated scene and SNR level. To compare the unmixing algorithms with each other, we bolded the smallest

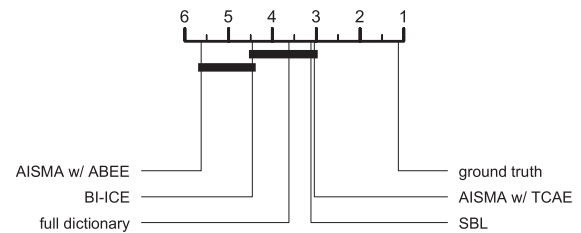


Fig. 14. Critical difference diagram calculated from the RL2E values shown in Table VI.

value from each row without considering the "ground truth" column.

A critical difference diagram calculated from the RL2Es of Table VI is shown in Fig. 14. Specifically, each column in Table VI contains two RL2Es per SNR level, and we always took into account the smaller one. Unmixing with ground truth

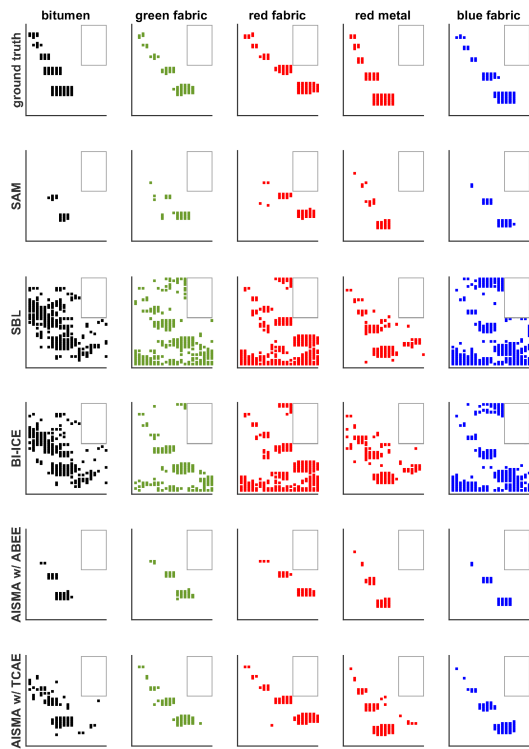


Fig. 15. First row shows the ground truth composition of the pixels, whereas the last five rows show the predictions of the algorithms. From left to right, the columns correspond to bitumen, green fabric, red fabric, red metal, and blue fabric. The upper right corner has been masked in each subplot, because the corresponding ground area contained objects whose spectra were not provided within the HySU dataset.

endmembers yielded the best abundance estimates by a significant margin. Furthermore, AISMA with TCAE performed significantly better than AISMA with ABEE. However, the endmember subsets selected by BI-ICE, SBL, and AISMA with TCAE did not lead to significantly better abundance estimates than the full dictionary.

The hyperspectral image of Scene II RGB sample 4 (viz. spatial pattern D) is represented as a false color image in the left panel of Fig. 16 in Appendix B. The estimated and ground truth abundance maps for the Scene II RGB sample 4 are shown in Fig. 18 in Appendix C.

VII. EXPERIMENT WITH A REMOTE-SENSING IMAGE

Remote sensing is the traditional application area of hyperspectral imaging. Therefore, we carried out an experiment with an airborne hyperspectral image, which was specifically acquired for the purposes of testing unmixing algorithms [49]. Specifically, the image is from a grass-covered football field, where targets of different size were deployed. Each target was made of either bitumen, green fabric, red fabric, red metal, or blue fabric.

Corresponding to each material, there were five targets having the sizes of $3\text{ m} \times 3\text{ m}$, $2\text{ m} \times 2\text{ m}$, $1\text{ m} \times 1\text{ m}$, $0.5\text{ m} \times 0.5\text{ m}$, and $0.25\text{ m} \times 0.25\text{ m}$. In comparison, the reported pixel size was $0.7\text{ m} \times 0.7\text{ m}$. Thereby, the $0.5\text{ m} \times 0.5\text{ m}$ and $0.25\text{ m} \times 0.25\text{ m}$ targets could not give rise to any pure pixels, but appeared entirely as subpixel objects. The arrangement of the targets has been graphically illustrated in [49, Fig. 1].

Since there are five target materials lying on grass, the total number of the endmembers is six. The HySU dataset [49] provides two versions of these endmember spectra, namely, spectra extracted from the hyperspectral image and spectra measured with a field spectrometer. The dictionary Φ we used for unmixing comprises the endmember spectra extracted from the hyperspectral image. The endmember spectra included in Φ are shown as solid lines in [49, Fig. 5].

In our experiment, we used only a part of the original hyperspectral image. The part that we used contains all the targets and is approximately the same as the one shown in [49, Fig. 4b]. The right panel of Fig. 16 in Appendix B is a false color representation of the hyperspectral image that we used.

The first row of Fig. 15 shows the endmembers that are present in each pixel. From left to right, the columns correspond to bitumen, green fabric, red fabric, red metal, and blue fabric. As in [49, Fig. 4b], the upper right corner has been masked because it contains endmembers whose spectra were not provided within the HySU dataset.

The second row of Fig. 15 shows the results of SAM classifier. The third, fourth, fifth, and sixth rows show the unmixing results obtained with SBL, BI-ICE, AISMA with ABEE, and AISMA with TCAE, respectively.

The SAM classifier and AISMA with ABEE perform well in avoiding false positives but have trouble detecting the smallest and the second smallest targets. In contrast, SBL and BI-ICE yield many false positives for all the five target materials, resulting in the third and fourth row being qualitatively rather different from the first row. The compositional maps estimated by AISMA with TCAE are in relatively good agreement with the ground truth for green fabric, red fabric, red metal, and blue fabric. For bitumen, the match with the ground truth is poorer, but even then the majority of estimated bitumen pixels are located close to the ground truth of bitumen pixels. Overall, AISMA with TCAE managed to detect even some of the smallest targets, while the number of false positives remained relatively low.

VIII. EXPERIMENTS WITH JASPER RIDGE AND URBAN DATASETS

The remote sensing images provided in the Jasper Ridge and Urban datasets have been used in many unmixing studies. The remote sensing image included in the Jasper Ridge dataset has been extracted from a larger image [50]. The Jasper Ridge and Urban datasets can be downloaded, e.g., from [51].

The Jasper Ridge dataset contains two versions of the hyperspectral image with the difference being in the number of wavelength bands. The first version has 224 wavelength bands extending from 380 to 2500 nm, while the number of wavelength bands in the second version is 198 following the removal of the bands that are affected by atmospheric absorption.

We used the Jasper Ridge image that has 198 wavelength bands, and a false color representation of the used image is shown in the left panel of Fig. 17 in Appendix B. The spatial plane of the hyperspectral image is divided into 100×100 pixels with each pixel corresponding to a $20\text{ m} \times 20\text{ m}$ region in the scene [50]. The abundance map provided in the Jasper Ridge dataset has four endmembers, namely, tree, water, dirt, and road.

The Urban dataset contains two versions of the hyperspectral image with the difference being in the number of wavelength bands. The first version has 210 wavelength bands extending from 400 to 2500 nm, while the number of wavelength bands in the second version is 162 following the removal of the bands that are affected by atmospheric absorption.

We used the Urban image that has 162 wavelength bands, and a false color representation of the used image is shown in the right panel of Fig. 17 in Appendix B. The spatial plane of the hyperspectral image is divided into 307×307 pixels with each pixel corresponding to a $2 \text{ m} \times 2 \text{ m}$ region in the scene [50]. Three different reference maps indicating the pixelwise fractional abundances are provided in the Urban dataset. We used the reference map that has six endmembers, namely, asphalt road, grass, tree, roof, metal, and dirt [50], [52].

The abundance maps for the Jasper Ridge and Urban images are shown, respectively, in Figs. 19 and 20 in Appendix C. Specifically, the reference maps provided in the datasets are shown on the first rows of Figs. 19 and 20, while the abundance maps obtained with the unmixing algorithms are shown on the subsequent rows. The abundance maps obtained with the SAM classifier and AISMA with ABEE are not as smooth as the reference maps. The abundance maps given by the other unmixing algorithms are in good agreement with the reference maps.

However, it should be noted that the reference maps included in the Jasper Ridge and Urban datasets are not ground truth maps, as they have been calculated with certain algorithms instead of being determined through *in situ* measurements [49], [53], [54]. At the same time, we share the view of many scholars that comparison of estimated maps with ground truth maps is the most objective way to assess the performance of unmixing algorithms (see, for instance, [48], [49], [55], [56], and [57]).

To further examine the plausibility of reference maps that have been calculated with algorithms, we consider the first row of Fig. 19. We note that the water area has pixels in which the fractional abundance of the road endmember does not vanish but ranges from tiny positive values to about 0.2. However, these pixels are scattered in such a way that we consider it unlikely that they would correspond to a road that is partly or thinly covered by water. Further contextual information can be gathered from the larger Jasper Ridge image [50] and physical maps of the Jasper Ridge area [58], and our view is that they do not suggest the presence of a road in the water area of the Jasper Ridge image. As a result, we feel that the reference maps provided in the Jasper Ridge dataset suffer from plausibility issues, which would not be present in ground truth maps determined through *in situ* measurements.

Owing to the lack of ground truth in Jasper Ridge and Urban datasets, we consider the related experiments less conclusive than the ones carried out in Sections VI and VII. Overall, we believe that the results obtained with the Jasper Ridge and Urban datasets qualitatively indicate the applicability of AISMA with TCAE to the unmixing of remote sensing images.

IX. ABLATION AND TIMING EXPERIMENTS

In Section IX-A, we examine the role of TCAE in the performance of AISMA by performing an ablation experiment.

Algorithm 4: Termination Condition Oracle.

Input 1: Ground truth fractional abundance vector \mathbf{x}

Input 2: Fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$

Output: Critical iteration critIT

- 1: Set $\text{TP}_i = \|\mathbf{x} \circ \mathbf{x}^{(i)}\|_0$. The number of true positives in $\mathbf{x}^{(i)}$.
 - 2: Set $\text{FP}_i = \|\mathbf{x}^{(i)}\|_0 - \text{TP}_i$. The number of false positives in $\mathbf{x}^{(i)}$.
 - 3: Define $\text{FP}_{\min} = \min_i \text{FP}_i$.
 - 4: Set $\text{critIT} = \text{argmax}_i \text{TP}_i$ s.t. $\text{FP}_i = \text{FP}_{\min}$.
 - 5: Return critIT as the output.
-

Specifically, we will repeat the simulations of Table II with TCAE replaced by a termination condition that determines the critical iteration with the assistance of the ground truth fractional abundance vector. Thus, we get to compare the performance of AISMA with TCAE against an upper limit that AISMA cannot exceed with any conceivable termination condition.

In Section IX-B, we state the computational complexity of AISMA with TCAE and conduct a runtime comparison between the algorithms.

A. Role of TCAE in the Performance of AISMA

To examine to the role of TCAE in the performance of AISMA, we define a new termination condition called termination condition oracle (TCO). TCO determines the critical iteration with the assistance of the ground truth fractional abundance vector \mathbf{x} , and it is described in Algorithm 4.

In addition to the ground truth fractional abundance vector \mathbf{x} , TCO receives the fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ generated by ISMA as inputs. With the assistance of the ground truth fractional abundance vector \mathbf{x} , TCO calculates the number of true positives and false positives for each of the estimated fractional abundance vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$.

Step 4 of TCO chooses the critical iteration critIT to maximize the number of true positives, which are subject to the constraint of the minimization of false positives. In practice, the number of false positives will be zero if the endmember remaining at the last iteration is a true positive (viz. belongs to the mixture). Otherwise, the number of false positives will be one. We stress that the performance of AISMA with TCO sets an upper limit to the performance that AISMA can attain with any conceivable termination condition, because TCO uses the ground truth fractional abundance vector to determine the critical iteration.

We repeated the simulations from Table II with TCAE replaced by TCO, and the endmember identification results are shown in Table VII. In the low-SNR simulations with smaller dictionaries, TCAE led to considerably smaller recall than TCO. Otherwise, we feel that the results obtained with TCAE are close to the best possible performance obtained with TCO.

In the medium- and high-SNR simulations, the difference between the F_1 -scores obtained with TCAE and TCO does not appreciably depend on the number of endmembers in the dictionary. In the low-SNR simulations, the gap between the

TABLE VII
PERFORMANCE OF AISMA WITH TCAE AND TCO

	AISMA with TCAE			AISMA with TCO		
	REC	PRE	F ₁	REC	PRE	F ₁
simulation 1	0.74	1.00	0.82	0.98	1.00	0.98
simulation 2	0.94	1.00	0.96	1.00	1.00	1.00
simulation 3	0.99	1.00	0.99	1.00	1.00	1.00
simulation 4	0.61	0.89	0.69	0.70	0.97	0.78
simulation 5	0.88	0.97	0.91	0.91	1.00	0.94
simulation 6	0.97	0.99	0.98	0.98	1.00	0.99
simulation 7	0.44	0.71	0.51	0.50	0.81	0.58
simulation 8	0.79	0.93	0.84	0.82	0.99	0.88
simulation 9	0.95	0.99	0.96	0.96	1.00	0.98
simulation 10	0.42	0.64	0.48	0.45	0.75	0.53
simulation 11	0.79	0.91	0.83	0.80	0.99	0.86
simulation 12	0.95	0.98	0.96	0.96	1.00	0.97

TABLE VIII
RUNTIMES OF THE ALGORITHMS

dictionary	runtime (s)					
	SAM	FCLS	NCLS	SBL	BI-ICE	AISMA w/ TCAE
Φ_2	0.06	15	12	115	203	30
Φ_3	0.05	18	14	277	593	38
Φ_4	0.07	22	16	713	1409	64

F₁-scores obtained with TCAE and TCO decreases with increasing dictionary size. Thus, the fixed value $R = 3$ set in Step 8 of TCAE does not limit the generalizability of TCAE to dictionaries having different numbers of endmembers.

B. Computational Complexity and Runtime Analysis

The main computational complexity in AISMA with TCAE results from the least-squares fit, which is performed in Step 4 of ISMA. Specifically, the asymptotic computational complexity of AISMA with TCAE is $O(n^3m)$, with n and m being the number of columns and rows, respectively, in the dictionary Φ (see Appendix D).

We examined the runtimes of the unmixing algorithms by running simulations in a MATLAB environment with a laptop having a 1.9-GHz CPU and 8 GB of memory. To illustrate how the number of endmembers affects runtime, we used the dictionaries Φ_2 , Φ_3 , and Φ_4 in the simulations. Each runtime reported in Table VIII is the average time it takes to unmix 10 000 mixture spectra. For further details on the timing experiment, see Appendix E.

AISMA with TCAE was about two and three times slower than FCLS in the simulations with Φ_2 and Φ_4 , respectively. In comparison to SBL, AISMA with TCAE was about four and 11 times faster in the simulations with Φ_2 and Φ_4 , respectively. BI-ICE was roughly two times slower than SBL in all the three simulations.

We conclude that SBL and BI-ICE have a larger computational complexity than AISMA with TCAE with respect to the

number of endmembers in the dictionary. Since the number of endmembers in a real-world unmixing scenario is typically less than 20, the results with the dictionary Φ_3 suggest that AISMA with TCAE could be up to seven times faster than SBL and up to 16 times faster than BI-ICE in real-life unmixing applications.

X. DISCUSSION

The main contribution of our research, namely TCAE, is an extension to ISMA that automatically selects the best solution from the sequence of fractional abundance vectors generated by ISMA. Although we developed TCAE as an extension to ISMA, TCAE can be used with any unmixing algorithm that yields a sequence of estimated fractional abundance vectors with an increasing sparsity level. To elaborate, let us assume that $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ is a sequence of estimated fractional abundance vectors with an increasing sparsity level $\|\mathbf{x}^{(i)}\|_0 = i$. By changing the labeling, one can form a sequence $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$, in which $\|\mathbf{x}^{(i)}\|_0 = n - i + 1$. Subsequently, the Δ_i values can be calculated by using the formulas from Steps 5, 10, and 11 of ISMA.

For example, this means that TCAE can be used to select the best fractional abundance vector from the set of fractional abundance vectors generated by a matching pursuit algorithm. Similar logic also dictates that TCAE can be used to select the best solution from the set of Pareto-optimal solutions to the biobjective minimization problem (3). Thus, TCAE can be used in multiobjective sparse unmixing to select the best solution from the set of Pareto-optimal solutions.

TCAE selects the best solution from the set of Pareto-optimal solutions with the help of elbow estimation. Thus, TCAE belongs to the same family of methods as ABEE, the weighted sum of objective values method, and the distance to the extreme line method [15]. An important novelty in our work is that we seek the elbow from the points $(\|\mathbf{x}^{(i)}\|_0, \tilde{\Delta}_i)$ instead of the points $(\|\mathbf{x}^{(i)}\|_0, \|\mathbf{y} - \Phi\mathbf{x}^{(i)}\|^2)$, in which $i \in \{1, 2, \dots, n\}$ and n is the number of solutions. In comparison to the distance to the extreme line method, TCAE has the advantage that it can adaptively choose the end points of the line segment that is used to determine the location of the elbow.

A. Future Directions

Abundance estimates tend to be the most accurate when the endmembers participating in the given pixel have been correctly identified [10]. Although the critical difference diagrams 9 and 13 show that AISMA with TCAE had the best F₁-scores by a significant margin, the critical difference diagrams 10 and 14 indicate that the abundance estimates yielded by AISMA with TCAE were not significantly better than the ones obtained with SBL, BI-ICE, and full dictionary. Thus, the results suggest that F₁-score is not the best possible performance metric for endmember identification if the goal of endmember selection is to improve the abundance estimates. A possible future direction would be to study the correlation between various endmember identification metrics and the relative L^2 error.

Since AISMA with TCAE is based on ISMA, it does not have any specific mechanism for mitigating problems that arise from endmember variance. Accordingly, we carried out simulations and real-data experiments with such data that do not exhibit endmember variance in any appreciable degree.

Although we did not consider endmember variance, it is common in real-world data. Therefore, extending the combination of ISMA and TCAE to handle endmember variance would increase its applicability to real-world problems. We surmise that grouping the spectra representing the same mineral as in SMESMM [13] would lead to better results than simply adding variations of the same mineral to the dictionary Φ . With the grouping, the iterative elimination of the endmember spectra could be carried out one group at a time instead of a single endmember spectrum at a time.

XI. CONCLUSION

In this article, we studied the problem of automatic identification of materials from mixed pixels. We developed a new elbow estimation method TCAE that enables a rapid and reliable identification, providing better performance than a state-of-the-art elbow detection method and two Bayesian unmixing algorithms. Specifically, Figs. 9, 10, 13, and 14 show that 1) AISMA performed endmember identification and quantitative abundance estimation significantly better with TCAE than ABEE, and 2) AISMA with TCAE outperformed SBL and BI-ICE in endmember identification.

Since AISMA does not have any hyperparameters, we used a simple version of SAM classifier that does not have user-specified thresholds. Specifically, the used version of SAM classifier considers a pixel to consist of the endmember for which the spectral angle between the pixel spectrum and the endmember spectrum is the smallest. Figs. 9 and 13 show that AISMA with TCAE outperformed the SAM classifier in endmember identification.

Timing experiments indicated that AISMA with TCAE could be up to seven times faster than SBL and up to 16 times faster than BI-ICE in real-life unmixing applications. Thus, AISMA with TCAE is better suited for applications that require real-time or near real-time performance. Overall, our conclusion is that AISMA with TCAE facilitates automatic, reliable, and rapid identification of endmembers from mixed pixels.

APPENDIX A

SPECTRAL PREPROCESSING IN SCENE II AND SCENE II RGB EXPERIMENTS

The reason for using the generalized logarithm transformation (21) instead of $A = -\log_{10} R$ in Scene II and Scene II RGB experiments is that in some rare instances, noise may cause a negative reflectance value R . We believe that the chosen value $c = 10^{-6}$ in (21) may be considered small, because it is several orders of magnitude smaller than typical reflectance values such as the ones exhibited by the high-SNR curve in Fig. 12. The

advantage of choosing a small value for c is that the generalized logarithm transformation is then practically equivalent to $A = -\log_{10} R$, but avoids the problem posed by slightly negative reflectance values arising from noise.

[48, Table VI] shows the RMSEs of various unmixing algorithms for Scene II mixtures. The shown values were calculated with reflectance spectra, while the used dictionary contained only the Scene II endmember spectra. The mean RMSEs of FCLS, K-Hype, and Hapke were 0.1630, 0.1221, and 0.0940, respectively. In comparison, we have calculated that the mean RMSE of FCLS with absorbance spectra is 0.0872.

We conclude that conversion from reflectance to absorbance substantially improves the applicability of the linear mixing model (1) for Scene II spectra. Since Scene II RGB samples are similar to Scene II samples, we assume that the generalized logarithm transformation (21) improves the validity of the linear mixing model (1) also for Scene II RGB spectra. Therefore, we feel that our choice to preprocess spectra with (21) in Scene II and Scene II RGB experiments is justified.

APPENDIX B FALSE COLOR IMAGES

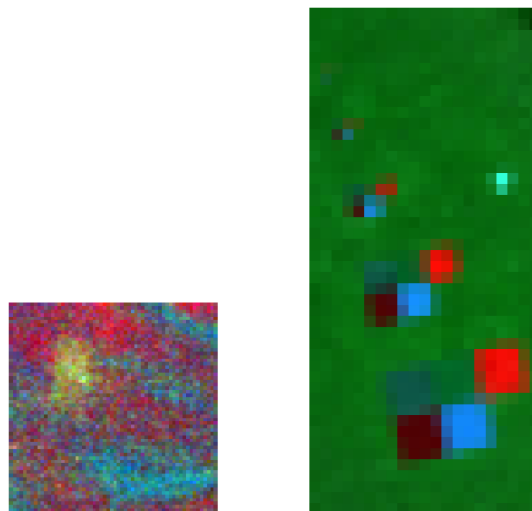


Fig. 16. Left panel shows the hyperspectral image of Scene II RGB sample 4 (spatial pattern D) as a false color image. The right panel is a false color representation of the remote sensing image used in Section VII.

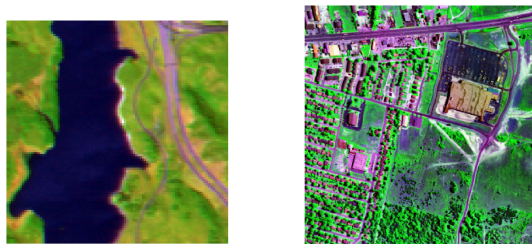


Fig. 17. Left and right panel show the false color representations of the Jasper Ridge and Urban hyperspectral images, respectively.

APPENDIX C
ABUNDANCE MAPS

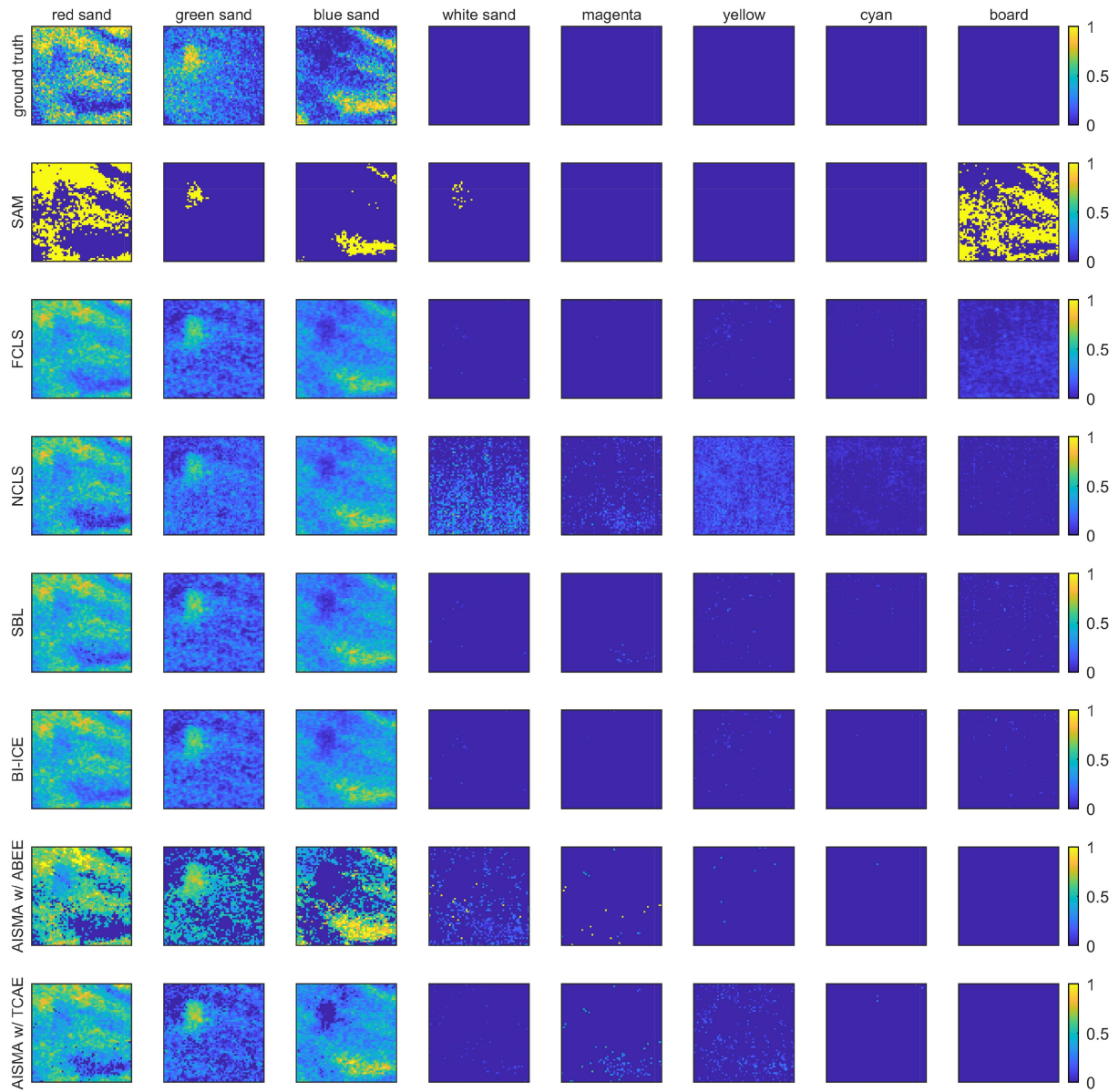


Fig. 18. Abundance maps for the Scene II RGB sample 4 (spatial pattern D). The first three columns correspond to the endmembers constituting the sample, while the last five columns correspond to the other endmembers included in the dictionary. The first row contains the ground truth abundance maps, while the following rows show the estimated abundance maps. Comparison of the FCLS and NCLS results shows that ASC has a substantial effect on the estimated abundance maps. The SBL maps were calculated by running FCLS with the endmembers selected by the RVM-based endmember selection. Similarly, embedded ASC was used in BI-ICE, while the AISMA maps were calculated by executing FCLS with the endmembers selected by AISMA.

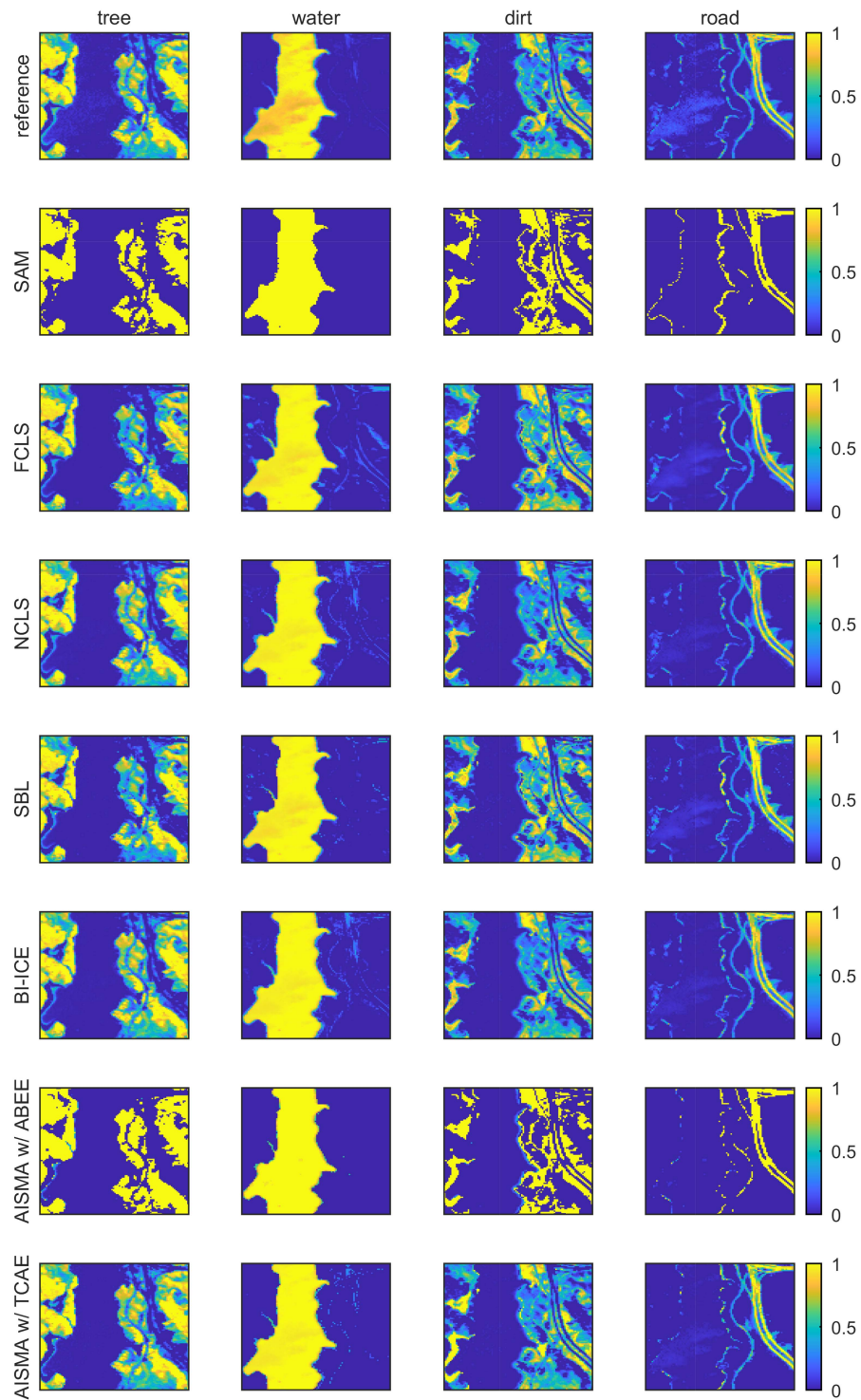


Fig. 19. Abundance maps for the Jasper Ridge image. The first row shows the reference maps included in the Jasper Ridge dataset, while the following rows show the estimated abundance maps. The reference maps satisfy ASC, and we scaled the fractional abundance vectors in the estimated maps so that each estimated abundance map also satisfies ASC.

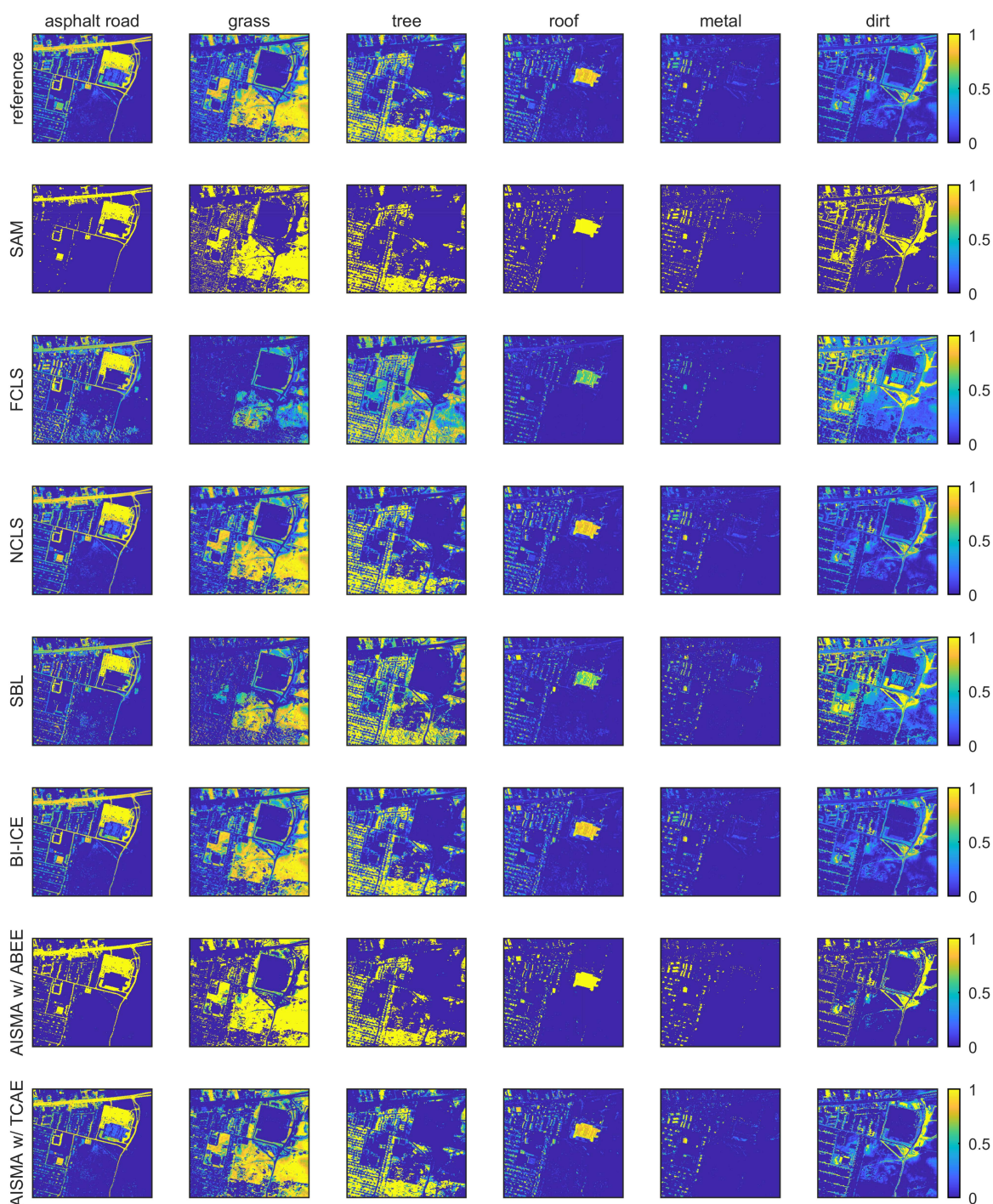


Fig. 20. Abundance maps for the Urban image. Reference maps based on four, five, and six endmembers are provided in the Urban dataset, and we opted to use the ones based on six endmembers. The first row shows the reference maps, while the following rows show the estimated abundance maps. The reference maps satisfy ASC, and we scaled the fractional abundance vectors in the estimated maps so that each estimated abundance map also satisfies ASC.

APPENDIX D

COMPUTATIONAL COMPLEXITY OF AISMA WITH TCAE

The main computational complexity in AISMA with TCAE results from the least-squares fit, which is performed in Step 4 of ISMA. The least-squares fit is based on matrix multiplication and matrix inversion. The multiplication of an $s \times t$ matrix by a $t \times u$ matrix has the complexity of $O(stu)$ [59], whereas inversion of an $n \times n$ matrix with Gauss–Jordan elimination has the complexity of $O(n^3)$ [60]. Relatedly, we recall that the mixture spectrum $\mathbf{y} \in \mathbb{R}^m$ and the dictionary $\Phi \in \mathbb{R}^{m \times n}$.

The least-squares fit $\arg\min_{\mathbf{x}} \|\mathbf{y} - \Phi\mathbf{x}\|$ begins with the matrix multiplications $\Phi^T\mathbf{y}$ and $\Phi^T\Phi$, which have the complexities of $O(mn)$ and $O(n^2m)$, respectively. Subsequently, the inversion of $\Phi^T\Phi \in \mathbb{R}^{n \times n}$ has the complexity of $O(n^3)$. Since $\Phi^T\mathbf{y} \in \mathbb{R}^{n \times 1}$ has already been calculated, the final matrix multiplication $(\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}$ has the complexity of $O(n^2)$. Thereby, the complexity of the least-squares fit is $O(n^3 + n^2m)$.

Step 4 of ISMA is repeated n times, i.e., as many times as there are endmembers in the dictionary Φ . Thus, the overall computational complexity resulting from Step 4 is $n \cdot O(n^3 + n^2m) = O(n^4 + n^3m)$. Since the dictionary Φ must be undercomplete in ISMA (i.e., $n < m$), the asymptotic computational complexity of AISMA with TCAE is $O(n^3m)$.

APPENDIX E

RUNTIME EXPERIMENT

By using (18), we constructed ten sets of mixture spectra for each of the dictionaries Φ_2 , Φ_3 , and Φ_4 . Each set contained 10 000 mixture spectra and each mixture spectrum was corrupted with correlated noise. The SNR of each mixture spectrum was randomly chosen from the interval 20 to 50 dB (cf. Table II). The number of components in each mixture was randomly drawn from $\{1, 2, \dots, 5\}$.

Each unmixing algorithm was timed as it processed the ten sets. For further averaging, the process of constructing the ten sets and timing the algorithms was repeated ten times. Each runtime reported in Table VIII is the mean of the so-obtained 100 durations. Thus, each runtime reported in Table VIII is the average time it takes to unmix 10 000 mixture spectra.

ACKNOWLEDGMENT

The authors would like to thank Dr. F. Chen for illuminating discussions and providing them with the MATLAB codes related to his article [32]. The authors would also like to thank Dr. Matti Möttönen and Prof. Marko Höyhty for valuable comments in the final stages of manuscript preparation.

REFERENCES

- [1] M. B. Stuart, A. J. S. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, 2019, Art. no. 3071.
- [2] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.
- [3] H. Aasen, E. Honkavaara, A. Lucieer, and P. J. Zarco-Tejada, "Quantitative remote sensing at ultra-high resolution with UAV spectroscopy: A review of sensor technology, measurement procedures, and data correction workflows," *Remote Sens.*, vol. 10, no. 7, 2018, Art. no. 1091.
- [4] T. Kääriäinen and T. Dönsberg, "Active hyperspectral imager using a tunable supercontinuum light source based on a MEMS Fabry–Perot interferometer," *Opt. Lett.*, vol. 46, no. 22, pp. 5533–5536, 2021.
- [5] E. Maggiori, A. Plaza, and Y. Tarabalka, *Models for Hyperspectral Image Analysis: From Unmixing to Object-Based Classification*. Cham, Switzerland: Springer, 2018, pp. 37–80.
- [6] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.
- [7] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.
- [8] J. M. Bioucas-Dias et al., "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.
- [9] J.-J. Wei and X. Wang, "An overview on linear unmixing of hyperspectral data," *Math. Probl. Eng.*, vol. 2020, pp. 1–12, 2020.
- [10] D. M. Rogge, B. Rivard, J. Zhang, and J. Feng, "Iterative spectral unmixing for optimizing per-pixel endmember sets," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3725–3736, Dec. 2006.
- [11] E. Liu, Z. Wu, and H. Zhang, "A multiobjective method leveraging spatial–spectral relationship for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5539416.
- [12] M. Gong, H. Li, E. Luo, J. Liu, and J. Liu, "A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 234–248, Apr. 2017.
- [13] F. Chen, K. Wang, and T. F. Tang, "Spectral unmixing using a sparse multiple-endmember spectral mixture model," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5846–5861, Oct. 2016.
- [14] L. Li, X. Yao, R. Stolkin, M. Gong, and S. He, "An evolutionary multi-objective approach to sparse reconstruction," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 827–845, Dec. 2014.
- [15] J. J. Liang, X. P. Zhu, C. T. Yue, Z. Li, and B. Qu, "Performance analysis on knee point selection methods for multi-objective sparse optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2018, pp. 1–8.
- [16] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Parallel Problem Solving from Nature VIII*, X. Yao et al., Eds. Berlin, Germany: Springer, 2004, pp. 722–731.
- [17] X. Jiang, M. Gong, T. Zhan, K. Sheng, and M. Xu, "Efficient two-phase multiobjective sparse unmixing approach for hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2418–2431, 2021.
- [18] Y. Wan, Y. Zhong, A. Ma, and L. Zhang, "Multi-objective sparse subspace clustering for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2290–2307, Apr. 2020.
- [19] M. Song, Y. Zhong, A. Ma, X. Xu, and L. Zhang, "Multiobjective subpixel mapping with multiple shifted hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8176–8191, Nov. 2020.
- [20] X. Jiang, M. Gong, H. Li, M. Zhang, and J. Li, "A two-phase multiobjective sparse unmixing approach for hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 508–523, Jan. 2018.
- [21] L. Zhuang, C. Lin, M. A. T. Figueiredo, and J. M. Bioucas-Dias, "Regularization parameter selection in minimum volume hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9858–9877, Dec. 2019.
- [22] F. Chen, S. Lu, P. Zhao, T. van de Voorde, and W. Xu, "Urban land cover mapping from airborne hyperspectral imagery using a fast jointly sparse spectral mixture analysis method," *Can. J. Remote Sens.*, vol. 46, no. 3, pp. 330–343, 2020.
- [23] Z. Wu, J. Sun, Y. Zhang, Z. Wei, and J. Chanussot, "Recent developments in parallel and distributed computing for remotely sensed big data processing," *Proc. IEEE*, vol. 109, no. 8, pp. 1282–1305, Aug. 2021.
- [24] H. Zhao and X. Zhao, "Nonlinear unmixing of minerals based on the log and continuum removal model," *Eur. J. Remote Sens.*, vol. 52, no. 1, pp. 277–293, 2019.
- [25] B. Somers, G. P. Asner, L. Tits, and P. R. Coppin, "Endmember variability in spectral mixture analysis: A review," *Remote Sens. Environ.*, vol. 115, no. 7, pp. 1603–1616, 2011.

- [26] D. A. Roberts, M. E. Gardner, R. L. Church, S. L. Ustin, G. J. Scheer, and R. O. Green, "Mapping chaparral in the Santa Monica mountains using multiple endmember spectral mixture models," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 267–279, 1998.
- [27] F. Chen and Y. Zhang, "Sparse hyperspectral unmixing based on constrained $l_p - l_2$ optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1142–1146, Sep. 2013.
- [28] N. Agrawal, H. Govil, S. Chatterjee, G. Mishra, and S. Mukherjee, "Evaluation of machine learning techniques with AVIRIS-NG dataset in the identification and mapping of minerals," *Adv. Space Res.*, 2022.
- [29] C. Lawson and R. Hanson, *Solving Least-Squares Problems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1974, ch. 23.
- [30] K. H. Haskell and R. J. Hanson, "An algorithm for linear least squares problems with equality and nonnegativity constraints," *Math. Program.*, vol. 21, pp. 98–118, 1981.
- [31] D. Heinz and Chein-I-Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.
- [32] F. Chen, K. Wang, and T. Tang, "Hyperspectral image unmixing using a sparse Bayesian model," *Remote Sens. Lett.*, vol. 5, no. 7, pp. 642–651, 2014.
- [33] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.
- [34] Apr. 24, 2023. [Online]. Available: http://www.vectoranomaly.com/downloads/SB2_Release_200.zip
- [35] K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "A novel hierarchical Bayesian approach for sparse semisupervised hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 585–599, Feb. 2012.
- [36] F. Pukelsheim, "The three sigma rule," *Amer. Statist.*, vol. 48, pp. 88–91, 1994.
- [37] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *Proc. 31st Int. Conf. Distrib. Comput. Syst. Workshops*, 2011, pp. 166–171.
- [38] M. Antunes, J. Ribeiro, D. Gomes, and R. L. Aguiar, "Knee/elbow point estimation through thresholding," in *Proc. IEEE 6th Int. Conf. Future Internet Things Cloud*, 2018, pp. 413–419.
- [39] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *Proc. IEEE 16th Int. Conf. Tools Artif. Intell.*, 2004, pp. 576–584.
- [40] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.
- [41] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.
- [42] S. K. Meerdink, S. J. Hook, D. A. Roberts, and E. A. Abbott, "The ECOSTRESS spectral library version 1.0," *Remote Sens. Environ.*, vol. 230, 2019, Art. no. 111196.
- [43] A. M. Baldrige, S. J. Hook, C. I. Grove, and G. Rivera, "The ASTER spectral library version 2.0," *Remote Sens. Environ.*, vol. 113, no. 4, pp. 711–715, 2009.
- [44] R. F. Kokaly et al., "USGS spectral library version 7," 2017. [Online]. Available: <https://doi.org/10.3133/ds1035>
- [45] N. Francos, G. Notesco, and E. Ben-Dor, "Estimation of the relative abundance of quartz to clay minerals using the visible-near-infrared-shortwave-infrared spectral region," *Appl. Spectrosc.*, vol. 75, no. 7, pp. 882–892, 2021.
- [46] W.-K. Ma et al., "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan. 2014.
- [47] Å. Rinnan, F. van den Berg, and S. Engelsen, "Review of the most common pre-processing techniques for near-infrared spectra," *TrAC Trends Anal. Chem.*, vol. 28, no. 10, pp. 1201–1222, 2009.
- [48] M. Zhao, J. Chen, and Z. He, "A laboratory-created dataset with ground truth for hyperspectral unmixing evaluation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2170–2183, Jul. 2019.
- [49] D. Cerra et al., "DLR HySU-A benchmark dataset for spectral unmixing," *Remote Sens.*, vol. 13, no. 13, 2021, Art. no. 2559.
- [50] F. Zhu, "Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey," 2017, *arXiv:1708.05125*.
- [51] Apr. 24, 2023. [Online]. Available: <https://rslab.ut.ac.ir/data>
- [52] X. Liu, W. Xia, B. Wang, and L. Zhang, "An approach based on constrained nonnegative matrix factorization to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 757–772, Feb. 2011.
- [53] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5412–5427, Dec. 2014.
- [54] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 88, pp. 101–118, 2014.
- [55] C. Cui, Y. Zhong, X. Wang, and L. Zhang, "Realistic mixing miniature scene hyperspectral unmixing: From benchmark datasets to autonomous unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5502515.
- [56] C. V. S. S. M. Kumar, S. S. Jha, R. R. Nidamanuri, and V. K. Dadhwal, "Benchmark studies on pixel-level spectral unmixing of multi-resolution hyperspectral imagery," *Int. J. Remote Sens.*, vol. 43, no. 4, pp. 1451–1484, 2022.
- [57] R. Heylen, M. Parente, and P. Gader, "A review of nonlinear hyperspectral unmixing methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, Jun. 2014.
- [58] "Envi tutorial: Vegetation hyperspectral analysis," Apr. 24, 2023. [Online]. Available: http://www.spweather.com/enviuser/tutorial/hyperspectral_tools/Vegetation_Analysis.pdf
- [59] S. S. Skiena, *Algorithm Analysis*. London, U.K.: Springer, 2008, pp. 31–64.
- [60] G. Sharma, A. Agarwala, and B. Bhattacharya, "Technical note: A fast parallel Gauss Jordan algorithm for matrix inversion using CUDA," *Comput. Struct.*, vol. 128, pp. 31–37, 2013.



Risto Sarjonen received the Doctor of Science (Technology) degree in theoretical physics from Aalto University, Espoo, Finland, in 2015.

He is currently a Research Scientist with the VTT Technical Research Centre of Finland, Espoo. His current and past research interests include theoretical physics, hyperspectral imaging, machine learning, and signal processing.



Tomi Rätty (Member, IEEE) received the Ph.D. degree in information processing science from the University of Oulu, Oulu, Finland.

He is currently with Microsoft, Redmond, WA, USA. He is also an Associate Professor with the Faculty of Information Technology and Electrical Engineering, University of Oulu. He is the author or coauthor of more than 40 papers published in various conferences and journals. His current research interests include machine learning technologies, data science, and software platforms.