Fast and Reliable Noise Estimation for Hyperspectral Subspace Identification

Peter Benner, Vedran Novaković, Antonio Plaza, *Fellow, IEEE*, Enrique S. Quintana-Ortí, and Alfredo Remón

***Abstract*—In this letter, we introduce an efficient algorithm to estimate the noise correlation matrix in the initial stage of the *hyperspectral signal identification by minimum error* (HySime) method, commonly used for signal subspace identification in remotely sensed hyperspectral images. Compared with the current implementations of this stage, the new algorithm for noise estimation relies on the reliable QR factorization, producing correct results even when operating with single-precision arithmetic. Additionally, our algorithm exhibits a lower computational cost, and it is highly parallel. The experiments on a multicore server, using two real hyperspectral scenes, expose that these theoretical advantages carry over to the practical results.**

***Index Terms*—Hyperspectral imaging, least squares problems, multicore processors, noise estimation, subspace identification.**

I. Introduction

CURRENT hyperspectral sensors collect considerable amounts of data at a high pace, and future hyperspectral missions will feature even higher sampling rates and/or higher spectral resolution [1]. For example, Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)[1] produces more than 18 MB/s, whereas the spaceborne HYPERION[2] gathers 1.33 MB/s of hyperspectral information [2], in both cases corresponding to the solar radiation reflected by the Earth's surface from the visible to infrared regions of the spectrum.

Under the conditions of the linear mixing model (see, e.g., [3]) the spectral vectors lie in a low-dimensional subspace, being a linear combination of a few *endmembers*, often much smaller than the number of spectral bands. A key initial step for hyperspectral applications is *subspace identification*, which aims to determine the minimum dimension of the subspace that enables an accurate and economic representation of the spectral vectors, and reduces the cost (time) and storage requirements for subsequent hyperspectral operations such as target detection, classification, and unmixing.

HySime [4] is an efficient hyperspectral dimensionality reduction algorithm for subspace identification. The interest of HySime is demonstrated by the number of recent high-performance implementations of this algorithm on a variety of parallel platforms, from field programmable gate arrays (FPGAs) to graphics processors (GPUs), digital signal processors (DSPs), and general-purpose multicore architectures [5], [6]. Due to the computational complexity of the algorithm, exploiting the hardware concurrency of these architectures is crucial to deliver a (near) real-time response in a number of scenarios, including biological threat detection, monitoring of chemical contamination, wildfire tracking, etc.

HySime consists of two stages: an initial estimation of the noise present in the original hyperspectral image and the computation of the *noise correlation matrix* is subsequently followed by the signal subspace identification and the computation of the *signal correlation matrix*. The algorithm selects the subset of eigenvectors that best represents the signal subspace in the minimum mean square error sense, applying multiple regression and yielding a global unsupervised procedure. The noise estimation step in HySime has been adopted by several other algorithms for noise characterization in remotely sensed hyperspectral imaging [3], [7].

In this paper, we target the reliable and efficient computation of the noise correlation matrix in HySime. Existing implementations [4]–[6] of the original algorithm for this initial stage rely on the *normal equations method* to address the solution of a collection of "coupled" *linear least squares* (LLS) problems [8]. These implementations exploit the properties of the coefficient matrices of these systems to formulate a reduced-cost solver. Nevertheless, the normal equations-based approach is often inferior from the numerical point of view to the computationally more expensive QR factorization [8], which can potentially result in the former producing an inconsistent identification of the number of endmembers.

The main contribution of this letter is the introduction of a numerically reliable *structure-aware* noise estimation algorithm, based on the QR factorization, that exploits the connection between the coefficient matrices of the LLS problems to yield a theoretical cost lower than that of the original normal equations algorithm. Our experimental results, using

Manuscript received October 8, 2014; revised November 27, 2014; accepted December 26, 2014.

P. Benner and A. Remón are with the Group on Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany (e-mails: benner@mpi-magdeburg.mpg.de; remon@mpi-magdeburg.mpg.de).

V. Novaković is with the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 10000 Zagreb, Croatia (e-mail: venovako@math.hr).

A. Plaza is with the Hyperspectral Computing Laboratory (HYPERCOMP), Department Technology of Computers and Communications, University of Extemadura, 10071 Cáceres, Spain (e-mail: aplaza@unex.es).

E. S. Quintana-Ortí is with the Department of Engineering and Computer Science, Universidad Jaume I de Castellón, 12071 Castellón de la Plana, Spain (e-mail: quintana@icc.uji.es).

Digital Object Identifier 10.1109/LGRS.2014.2388133

[1] http://aviris.jpl.nasa.gov

[2] http://eo1.usgs.gov

multithreaded implementations of these algorithms on a platform equipped with an up-to-date Intel Xeon processor, illustrate the practical superiority of the structure-aware noise estimation algorithm for two real hyperspectral scenes. In particular, the new algorithm only requires the use of single-precision (SP) arithmetic to yield accurate detection of the subspace dimension for the two testbeds. On the other hand, to obtain the correct result, the original algorithm requires more expensive and, thus, slower double-precision (DP) arithmetic.

This letter is organized as follows. In Section II, we review the original algorithm for noise estimation based on the normal equations. In Section III, we outline the QR-based alternative and then propose a structure-aware variant of this method. In Section IV, we discuss the parallelization of the proposed method using the OpenMP interface, LAPACK [9] and BLAS [10]; and in Section V, we evaluate the performance of the multithreaded implementation of these algorithms. Finally, a few remarks close this letter in Section VI.

II. Noise Estimation via Normal Equations

We will employ the following notation hereafter. Given a matrix $A \in \mathbb{R}^{p\times q}$, partitioned column-wise as $A = (a_1, a_2, \ldots, a_q)$, then $A_{\partial_j} = (a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_q)$, $1 \le j \le q$. Furthermore, $A_{\partial_j,\partial_j}$ denotes the matrix that results from deleting the jth column and row vectors of A, and $A_{\partial_j,j}$ is the jth column vector of A_{∂_j}. Given a vector $v \in \mathbb{R}^r$, $\|v\|_2$ denotes the vector 2-norm; see, e.g., [8].

Consider that matrix $Z \in \mathbb{R}^{n\times l}$ contains the n spectral vectors (or image pixels) each consisting of l bands. The success of multiple regression theory in hyperspectral applications in general, and the HySime algorithm for subspace identification in particular, is rooted in the high correlation between neighboring spectral bands of hyperspectral images [4].

Let us denote the jth column vector of Z as $z_j \in \mathbb{R}^n$, which "comprises" the jth band of the image. The algorithm for noise estimation underlying HySime then assumes that z_j is a linear combination of the remaining $l-1$ bands, i.e.,

$$z_j = Z_{\partial_j}\beta_j + \xi_j \tag{1}$$

where $\beta_j \in \mathbb{R}^{l-1}$ is the *regression vector*, and $\xi_j \in \mathbb{R}^n$ is the *modeling error vector*. Least squares estimators for β_j and ξ_j are given, respectively, by the solution of the LLS problem:

$$\tilde{\beta}_j = \min_{x\in\mathbb{R}^{l-1}} \left\| Z_{\partial_j} x - z_j \right\|_2 \tag{2}$$

and the residual

$$\tilde{\xi}_j = z_j - Z_{\partial_j}\tilde{\beta}_j. \tag{3}$$

Finally, the *noise correlation matrix* can be approximated as

$$\tilde{R}_n = \frac{1}{n}\left(\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_l\right)^T \left(\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_l\right) \in \mathbb{R}^{l\times l}. \tag{4}$$

Existing implementations of HySime tackle (2) via the normal equations method, which simply sets

$$\tilde{\beta}_j = \left(Z_{\partial_j}^T Z_{\partial_j}\right)^{-1} Z_{\partial_j}^T z_j. \tag{5}$$

Input: $Z \in \mathbb{R}^{n\times l}$ containing the spectral vectors
Output: $\tilde{\xi}_j \in \mathbb{R}^n$, $j = 1, 2, \ldots, l$, noise estimations
Cost: $3nl^2 + 4l^3$ flops

$M = Z^T Z$
$\hat{M} = M^{-1}$
for $j = 1, 2, \ldots, l$
  $\tilde{\beta}_j = \left(M_{\partial_j,\partial_j} - \frac{1}{m_{j,j}} M_{\partial_j,j} M_{\partial_j,j}^T\right) \hat{M}_{\partial_j,j}$
  $\tilde{\xi}_j = z_j - Z_{\partial_j}\tilde{\beta}_j$
end

Fig. 1. HySimeNE: Noise estimation based on the normal equations method. Here, $m_{j,j}$ denotes the entry in position (j, j) of M.

In principle, this approach requires the solution of a symmetric positive definite (s.p.d.) linear system per band. However, as described in [4], this can be avoided by exploiting the properties of the coefficient matrices $Z_j = Z_{\partial_j}^T Z_{\partial_j}$ of the linear systems and the fact that Z_{∂_i} and Z_{∂_k}, with $i \ne k$, only differ in two rows/columns. The result is the algorithm for noise estimation based on the normal equations, HySimeNE, in Fig. 1. Exploiting the symmetric structure of some of the (partial) results yields a cost of $3nl^2 + 4l^3$ floating-point arithmetic operations (flops) for the algorithm. (Hereafter, we neglect lower order terms in the cost expressions.)

From the numerical perspective, this approach has two major flaws. First, since it is based on the normal equations method, the error in the solution is proportional to the square of the *condition number* of Z [8], denoted as $\kappa(Z)$, which in hyperspectral scenarios can be high. Second, the algorithm explicitly constructs the inverse of M, which can be a source of additional rounding errors during the calculations.

In the next section, we present an alternative procedure for noise estimation that avoids these two numerical pitfalls while reducing the cost of the original algorithm.

III. Noise Estimation via the QR Factorization

A known alternative to the use of the normal equations for the solution of LLS problems is to rely on the *QR factorization* [8]. In the particular case of noise estimation, this method first decomposes Z_{∂_j} as

$$Z_{\partial_j} = Q_j R_j \tag{6}$$

where $Q_j \in \mathbb{R}^{n\times n}$ is orthogonal, and $R_j \in \mathbb{R}^{n\times(l-1)}$ is upper triangular. If we define

$$d_j = Q_j^T z_j = \begin{pmatrix} d_j^t \\ d_j^b \end{pmatrix}, \begin{array}{l} \}\ l-1 \\ \}\ n-l+1 \end{array} \tag{7}$$

the sought-after estimator for the jth model error vector is then given by the residual of the LLS problem:

$$\tilde{\xi}_j = Q_j \begin{pmatrix} 0 \\ d_j^b \end{pmatrix}. \begin{array}{l} \}\ l-1 \\ \}\ n-l+1. \end{array} \tag{8}$$

Provided that a numerically stable approach is employed for the computation of the QR factorization above, this method ensures that the error in the residual is proportional to $\kappa(Z)$ [8]. Furthermore, the method does not involve any matrix

Input: $Z \in \mathbb{R}^{n\times l}$ containing the spectral vectors
Output: $\tilde{\xi}_j \in \mathbb{R}^n,\ j = 1, 2, \ldots, l$, noise estimations
Cost: $2nl^3 - 2l^4/3$ flops

for $j = 1, 2, \ldots, l$
Factorize $Z_{\partial_j} = Q_j R_j$

$$d_j = Q_j^T z_j = \begin{pmatrix} d_j^t \\ d_j^b \end{pmatrix},\ \tilde{\xi}_j = Q_j \begin{pmatrix} 0 \\ d_j^b \end{pmatrix},$$

with $d_j^t \in \mathbb{R}^{l-1}, d_j^b \in \mathbb{R}^{n-l+1}$
end

Fig. 2. HYSIMEQR: Noise estimation based on the QR factorization.

inversion. However, the computation of the QR factorizations is expensive, yielding a total cost (i.e., considering all bands) of $2nl^3 - 2l^4/3$ flops. The HYSIMEQR algorithm for noise estimation based on the QR factorization is given in Fig. 2.

We next describe how to reorganize the QR-based procedure into a structure-aware variant that exploits the special relationship between the matrices Z_{∂_j} involved in the QR factorizations in order to reduce the cost of this computation.

Let us assume we have obtained the "first" QR factorization

$$Z_{\partial_1} = Q_1 R_1 = Q_1(r_2, r_3, \ldots, r_l) \tag{9}$$

where $(r_2, r_3, \ldots, r_l)$ denotes a column-wise partitioning of R_1, as well as the model error vector $\tilde{\xi}_1$ from (7) and (8).

The remaining $l-1$ model error vectors, $\tilde{\xi}_j, j = 2, 3, \ldots, l$, can then be inexpensively obtained as follows. Let us define

$$Y_j = Q_1^T \left(Z_{\partial_j}\Pi\right) \tag{10}$$

where $\Pi \in \mathbb{R}^{(l-1)\times(l-1)}$ simply permutes the first column vector of Z_{∂_j} to the last column vector. Then

$$\begin{aligned} Y_j &= Q_1^T \left(Z_{\partial_j}\Pi\right) && (11)\\ &= Q_1^T(z_2, \ldots, z_{j-1}, z_{j+1}, \ldots, z_l, z_1) && (12)\\ &= (r_2, \ldots, r_{j-1}, r_{j+1}, \ldots, r_l, d_1). && (13) \end{aligned}$$

The key to this procedure lies in recognizing and exploiting the structure of Y_j in (13). Concretely,

$$Y_j = \begin{pmatrix} U_{11} & U_{12} & u_{13} \\ 0 & H_{22} & y_{23} \\ \underbrace{0}_{j-2} & \underbrace{0}_{l-j} & \underbrace{y_{33}}_{1} \end{pmatrix}, \begin{array}{l} \}\ j-2 \\ \}\ l-j+1 \\ \}\ n-l+1 \end{array} \tag{14}$$

with U_{11} upper triangular and H_{22} upper Hessenberg (i.e., all its entries below the first subdiagonal equal zero). Therefore, the QR factorization of Y_j can be obtained via, e.g., a sequence of $l-j$ Givens rotations [8], $G_1, G_2, \ldots, G_{l-j}$, that annihilate the entries in the first subdiagonal of H_{22}, i.e.,

$$G_{l-j} \cdots G_2 G_1 Y_j = \begin{pmatrix} U_{11} & U_{12} & u_{13} \\ 0 & U_{22} & h_{23} \\ 0 & 0 & h_{33} \end{pmatrix}, \tag{15}$$

where U_{22} is upper triangular, and $h_{23} = G_{l-j} \cdots G_2 G_1 y_{23}$. This can then be followed by a single Householder reflector

Input: $Z \in \mathbb{R}^{n\times l}$ containing the spectral vectors
Output: $\tilde{\xi}_j \in \mathbb{R}^n,\ j = 1, 2, \ldots, l$, noise estimations
Cost: $2nl^2 - l^3/6$ flops

Factorize $Z_{\partial_1} = Q_1 R_1$

$$d_1 = Q_1^T z_1 = \begin{pmatrix} d_1^t \\ d_1^b \end{pmatrix},\ \bar{\xi}_1 = \begin{pmatrix} 0 \\ d_1^b \end{pmatrix},$$

with $d_1^t \in \mathbb{R}^{l-1}, d_1^b \in \mathbb{R}^{n-l+1}$
for $j = 2, 3, \ldots, l$
Assemble $Y_j = (r_2, \ldots, r_{j-1}, r_{j+1}, \ldots, r_l, d_1)$
Factorize $Y_j = \bar{Q}_j \bar{R}_j$, with $\bar{R}_j$ upper triangular and $\bar{Q}_j^T = H_j G_{l-j} \cdots G_2 G_1$

$$d_j = \bar{Q}_j^T z_j = \begin{pmatrix} d_j^t \\ d_j^b \end{pmatrix},\ \bar{\xi}_j = \bar{Q}_j \begin{pmatrix} 0 \\ d_j^b \end{pmatrix},$$

with $d_j^t \in \mathbb{R}^{l-1}, d_j^b \in \mathbb{R}^{n-l+1}$
end

$$\left(\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_l\right) = Q_1\left(\bar{\xi}_1, \bar{\xi}_2, \ldots, \bar{\xi}_l\right)$$

Fig. 3. HYSIMESA: Noise estimation based on the structure-aware QR factorization.

[8], say H_j, which annihilates the entries of h_{33}, so that

$$\begin{aligned} H_j G_{l-j} \cdots G_2 G_1 Y_j &= \bar{Q}_j^T Y_j && (16)\\ &= \begin{pmatrix} U_{11} & U_{12} & u_{13} \\ 0 & U_{22} & u_{23} \\ 0 & 0 & 0 \end{pmatrix} = \bar{R}_j && (17) \end{aligned}$$

is upper triangular. In other words, $Y_j = \bar{Q}_j \bar{R}_j$ is a QR factorization of Y_j, and therefore

$$d_j = \bar{Q}_j^T Q_1^T z_j = \bar{Q}_j^T r_j \tag{18}$$

partitioned as in (7), yields the required approximation of the noise vector

$$\tilde{\xi}_j = Q_1 \bar{Q}_j \begin{pmatrix} 0 \\ d_j^b \end{pmatrix}. \tag{19}$$

The algorithm based on this "structure-aware" variant of the QR factorization is given in Fig. 3. The initial QR factorization (before the loop body) requires $2l^2(n - l/3)$ flops, and the computation of d_1 and $\tilde{\xi}_1$ do not add a significant cost to this. Taking into account that the triangular matrices $\bar{R}_j$ do not need to be explicitly constructed, the Givens rotations (and Householder reflector) that compose $\bar{Q}_j$ can be computed and applied with a cost of only $l^2/2$ flops per band/factorization. Again, the subsequent computations of d_j in (18) and $\tilde{\xi}_j$ in (19) add negligible costs to this. Therefore, this structure-aware procedure performs to $2nl^2 - l^3/6$ flops only.

IV. PARALLELIZATION ON MULTICORE PROCESSORS

All three noise estimation variants of HYSIME have two structural properties in common, making them amenable to straightforward parallelization in a multithreaded architecture. From Figs. 1–3, it follows that no data dependencies exist between the computations performed in different for-loop iterations, which allows the iterations to be executed

```
Factorize Z_∂1 = Q_1 R_1                                   [xGEQRF]_p
d_1 = Q_1^T z_1                                            [xORMQR]_p
Assemble ξ̄_1
#pragma omp parallel for num_threads(p)
for j = 2, 3, ..., l
    Assemble Y_j = (r_2, ..., r_{j-1}, r_{j+1}, ..., r_l, d_1)
    for k = 1, 2, ..., l - j
        Construct G_k                                      [xROTG]_1
        Apply G_k to build Y_j = Q̄_j R̄_j                   [xROT]_1
    end
    Generate H_j                                           [xLARFG]_1
    d_j = z_j
    for k = 1, 2, ..., l - j
        d_j = G_k d_j                                      [xROT]_1
    end
    d_j = H_j d_j = ( d_1^T ), ξ̄_j = H_j^T ( 0 )          [xLARF]_1×2
                    ( d_1^b )               ( d_1^b )
    for k = l - j, l - j - 1, ..., 1
        ξ̄_j = G_k^T ξ̄_j                                    [xROT]_1
    end
end
(ξ̃_1, ξ̃_2, ..., ξ̃_l) = Q_1 (ξ̄_1, ξ̄_2, ..., ξ̄_l)          [xORMQR]_p
```

Fig. 4. OpenMP implementation of HYSIMESA. The names within brackets specify the LAPACK/BLAS routine that performs the operation. In the routines' names, x = S or D for SP or DP arithmetic.

concurrently with respect to each other. Moreover, the computations surrounding the for-loops in Figs. 1 and 3 are rich in blocked BLAS-3 and BLAS-2 operations and can be efficiently computed in parallel by invoking highly tuned multithreaded implementations of the appropriate kernels available, e.g., in Intel's Math Kernel Library (MKL).[3]

In Fig. 4, we outline the multithreaded parallelization of HYSIMESA using the OpenMP interface, BLAS and LAPACK, for a multicore platform with p cores. The parallelization of HYSIMENE and HYSIMEQR follows the same pattern.

The initial factorization and vectors d_1 and $(\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_l)$ in Fig. 4 are, respectively computed by calling routines `xGEQRF` and `xORMQR` from LAPACK, which exploit the hardware concurrency of the target architecture by relying on multithreaded implementations of the BLAS. The OpenMP `parallel for` compiler directive distributes the for-loop iterations among p threads, with each thread being assigned a portion of the iteration range. For each for-loop iteration, a private matrix Y_j, as well as the storage for the parameters of H_j and $G_1, \ldots, G_{l-j}$, have to be provided. Data transfer routines and auxiliary storage are omitted for brevity.

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the numerical reliability and parallel performance of three algorithms for subspace identification of hyperspectral data which differ in the method for noise estimation.

- HYSIMENE employs the normal equations (see Fig. 1).
- HYSIMEQR is based on the QR factorization (see Fig. 2).
- HYSIMESA leverages the relationship among the matrices that need to be factorized via a structure-aware QR factorization (see Fig. 3).

All three algorithms employ the same code for the signal subspace estimation phase that follows the noise estimation in HYSIME. For each algorithm, we have developed two implementations that perform all floating-point arithmetic either in single or double precision.

TABLE I
NUMBER OF ENDMEMBERS DETECTED WITH THE DIFFERENT VERSIONS OF HYSIME USING SIX THREADS AND CORES

	CUPRITE		WTC	
	SP	DP	SP	DP
HYSIMENE	22	16	30	23
HYSIMEQR	16	16	23	23
HYSIMESA	16	16	23	23

The tests in this section were performed on a server equipped with an Intel Xeon E5645 processors (six cores at 2.40 GHz) and 24 GB of RAM. The codes were compiled with Intel's `icc` 11.1 and linked to Intel MKL 9.293.

We employed two standard hyperspectral benchmarks:

- CUPRITE corresponds to the online[4] AVIRIS CUPRITE scenario. This is an $n = 350 \times 350$-pixel subset of sector `f970619t01p02_02_sc03.a.rfl` with $l = 188$ spectral bands of wavelength between 0.4 and 2.5 μm.
- WTC was collected over the World Trade Center after the attacks on September 11, 2011. The image consists of $n = 512 \times 614$ pixels and $l = 224$ bands with wavelengths between 0.4 and 2.5 μm, which corresponds to the standard data cube size recorded by AVIRIS.

Table I reports the number of endmembers identified by the different algorithms and arithmetic implementations of HYSIME using the six threads/cores. All six cases (three algorithms with two implementations using different precision each) obtain the same results except for HYSIMENE+SP, which reports a much larger number of endmembers for both images. The reason for this is that HYSIMENE relies on the normal equations, which introduces an error in the results that is proportional to $\kappa(Z)^2$. In particular, $\kappa(Z) = 9.32e + 3$ for CUPRITE and $\kappa(Z) = 1.41e + 4$ for WTC, which determines the erroneous results when HYSIMENE is executed using six threads and SP arithmetic. Indeed, due to the conditioning of the problem, the results of HYSIMENE+SP for CUPRITE depended on the number of threads that were employed in its execution. In particular, with one, two, and four threads, this routine simply failed during the Cholesky factorization of the s.p.d. matrix $Z^T Z$ for CUPRITE. For comparison, Table II offers the number of endmembers detected with the virtual dimensionality (VD) method [11], an alternative approach that relies on a proper calibration of an input parameter, exposing that, for the considered scenes, HYSIME provides similar estimates to those computed with VD with the advantage of not requiring any input parameters when conducting such estimation.

Table III shows the minimum execution time over five to ten executions of the initial noise estimation stage computed with the different implementations of HYSIME. The following observations are due.

[3] https://software.intel.com/en-us/intel-mkl

[4] http://aviris.jpl.nasa.gov/freedata.

TABLE II
NUMBER OF ENDMEMBERS DETECTED BY THE VD METHOD DEPENDING ON THE FALSE ALARM PROBABILITY (P_F)

P_F	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-7}	10^{-7}	10^{-8}
CUPRITE	28	25	20	19	19	17	16
WTC	44	33	31	30	27	23	21

TABLE III
EXECUTION TIME (IN SECONDS) FOR THE NOISE ESTIMATION WITH DIFFERENT VERSIONS OF HYSIME. THE LABEL "INC." INDICATES THAT THE NUMBER OF ENDMEMBERS WAS NOT CORRECTLY ESTIMATED; AND "FAIL" THAT THE CHOLESKY FACTORIZATION FAILED

	#threads= #cores	CUPRITE SP	CUPRITE DP	WTC SP	WTC DP
HYSIMENE	1	INC.	4.97	FAIL	19.93
	2	INC.	2.57	FAIL	10.67
	4	INC.	1.74	FAIL	8.38
	6	INC.	1.87	INC.	8.09
HYSIMEQR	1	185.20	377.15	850.91	1,785.08
	2	93.06	192.15	435.12	921.37
	4	54.14	128.84	276.29	599.78
	6	50.03	117.74	243.78	539.63
HYSIMESA	1	2.28	4.49	8.24	16.65
	2	1.44	2.79	4.93	10.01
	4	1.05	2.10	3.54	7.40
	6	0.99	2.00	3.43	7.16

- The signal correlation stage (second stage, not included in the results of the table) contributes a smaller cost to the total execution time, which, e.g., ranges from 1.70 s (sequential) and 0.47 s (six threads) for CUPRITE+SP to 5.60 s (sequential) and 1.46 s (six threads) for WTC+SP.
- In general, the use of DP arithmetic roughly doubles the execution time. This is to be expected as a significant fraction of the algorithms' computations are vectorized to exploit the SIMD floating-point units of the processor.
- HYSIMEQR delivers much higher execution times than those of HYSIMENE and HYSIMESA, which is well explained by its considerably higher theoretical cost (as $n \gg l$, HYSIMEQR requires $O(nl^3)$ flops versus $O(nl^2)$ flops for the two other alternatives). The difference, though, is not as large as one could expect (factors of $l = 188\times$ for CUPRITE and $l = 224\times$ for WTC), mostly due to the efficiency of the operations in HYSIMEQR.
- HYSIMENE+DP is slightly faster than HYSIMESA+DP for CUPRITE in a multithreaded execution (1.74 s using four threads for the former versus 2.00 s with six threads for the later, which corresponds to a factor of $2.00/1.74 = 1.14\times$), thought it is also slower in a sequential execution $(4.49/4.97 = 0.90\times)$. With the larger image, i.e., WTC, HYSIMESA+DP outperforms HYSIMENE+DP both in a sequential and a multithreaded execution (factors of $1.19\times$ and $1.12\times$, respectively). In both scenarios, the differences are due to the higher theoretical cost of HYSIME, NE but also to its mildly higher degree of parallelism. Furthermore, in both scenarios we can expect little improvements in HYSIMENE and HYSIMESA from increasing the number of threads beyond 6.
- An interesting observation from the results in Table I is that it is possible to employ the SP version of HYSIMESA for the reliable identification of endmembers, but the same goal, when using HYSIMENE, requires DP arithmetic. A practical consequence is that the time differences between these two algorithms are then much higher, with execution times for HYSIMESA+SP of 2.28 s (sequential) and 0.99 s (six threads) for CUPRITE, and 8.24 s (sequential) and 3.43 s (six threads) for WTC. Compared with this, HYSIMENE+DP required 4.97 s (sequential) and 1.74 s (4 threads) for CUPRITE, and 19.93 s (sequential) and 8.09 s (6 threads) for WTC. Thus, the actual acceleration factors for HYSIMESA(+SP) with respect to HYSIMENE(+DP) are $2.17\times$ (sequential) and $1.75\times$ (multithreaded) for CUPRITE and $2.41\times$ (sequential) and $2.35\times$ (multithreaded) for WTC.

VI. CONCLUDING REMARKS

We have presented a new algorithm for accelerating the noise estimation stage of the HYSIME subspace identification method, based on the QR factorization for the solution of the associated LLS problems, which offers reliable noise estimations even if executed in single-precision arithmetic. Compared with our newly proposed algorithm, the existing implementations of this stage rely on the normal equations method, which then requires the use of more expensive double-precision arithmetic to compensate for the numerical difficulties of this approach. Our experiments with two real hyperspectral images, using multithreaded implementations of these algorithms tuned for a recent multicore processor from Intel, expose the advantages of the new algorithm, which outperforms existing algorithms for noise estimation in reliability, theoretical cost, and practical performance.

REFERENCES

[1] K. Staenz, A. Mueller, A. Held, and U. Heiden, "Technical committees corner: International Spaceborne Imaging Spectroscopy (ISIS)," *IEEE Geosci. Remote Sens. Newslett.*, no. 165, pp. 38–42, 2012.
[2] J. Kerekes and J. Baum, "Spectral imaging system analytical model for subpixel object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 5, pp. 1088–1101, May 2002.
[3] J. 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.
[4] J. Bioucas-Dias and J. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.
[5] S. Sánchez, G. León, A. Plaza, and E. Quintana-Ortí, "Assessing the performance–energy balance of graphics processors for spectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2305–2316, Jun. 2014.
[6] E. Torti, M. Acquistapace, G. Danese, F. Leporati, and A. Plaza, "Real-time identification of hyperspectral subspaces," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2680–2687, Jun. 2014.
[7] J. Bioucas-Dias *et al.*, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
[8] A. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1996.
[9] E. Anderson *et al.*, *LAPACK Users' Guide*, 3rd ed. Philadelphia, PA, USA: SIAM, 1999.
[10] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. Duff, "A set of level 3 basic linear algebra subprograms," *ACM Trans. Math. Softw.*, vol. 16, no. 1, pp. 1–17, Mar. 1990.
[11] C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 608–619, Mar. 2004.