A parallel unmixing algorithm for hyperspectral images

Stefan A. Robila*, Lukasz G. Maciak,
Center for Imaging and Optics, Dept. of Computer Science, RI 301, Montclair State University, Montclair, NJ USA 07043

ABSTRACT

We present a new algorithm for feature extraction in hyperspectral images based on source separation and parallel computing. In source separation, given a linear mixture of sources, the goal is to recover the components by producing an unmixing matrix. In hyperspectral imagery, the mixing transform and the separated components can be associated with endmembers and their abundances. Source separation based methods have been employed for target detection and classification of hyperspectral images. However, these methods usually involve restrictive conditions on the nature of the results such as orthogonality (in Principal Component Analysis – PCA and Orthogonal Subspace Projection - OSP) of the endmembers or statistical independence (in Independent Component Analysis - ICA) of the abundances nor do they fully satisfy all the conditions included in the Linear Mixing Model. Compared to this, our approach is based on the Nonnegative Matrix Factorization (NMF), a less constraining unmixing method. NMF has the advantage of producing positively defined data, and, with several modifications that we introduce also ensures addition to one. The endmember vectors and the abundances are obtained through a gradient based optimization approach.

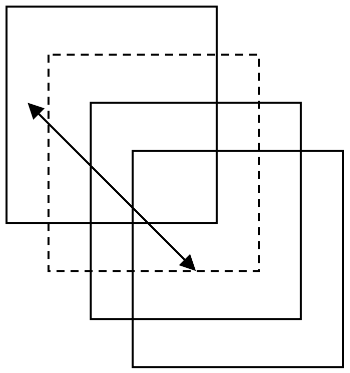
The algorithm is further modified to run in a parallel environment. The parallel NMF (P-NMF) significantly reduces the time complexity and is shown to also easily port to a distributed environment. Experiments with in-house and Hydice data suggest that NMF outperforms ICA, PCA and OSP for unsupervised endmember extraction. Coupled with its parallel implementation, the new method provides an efficient way for unsupervised unmixing further supporting our efforts in the development of a real time hyperspectral sensing environment with applications to industry and life sciences.

Keywords: hyperspectral images, blind source separation, Nonnegative Matrix Factorization, Linear Mixing Model, parallel processing, image processing

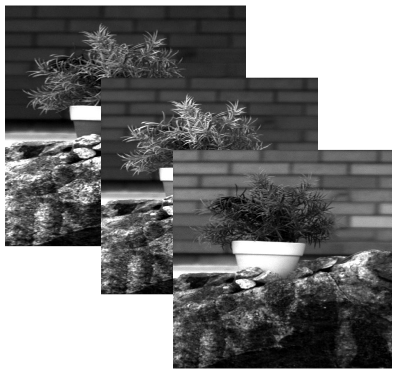
1. INTRODUCTION

Hyperspectral data are formed as collections of tens or hundreds of images of the same scene with each image corresponding to a narrow interval of energy wavelength [1]. Such images usually cover the visible to infrared ranges [2], although hyperspectral data for other spectrum parts such as Terahertz were also produced [3]. The richness of this information information becomes clear when we realize that any two materials expose differences in reflectance in one or more of the collected bands [4]. Plotting the reflectance values we obtain detailed graphs (often called spectra or pixel vectors) that characterize the materials. Figs. 1 and 2 are examples of the use of hyperspectral technology in analyzing objects. A plant arrangement that includes both artificial and natural vegetation is placed in a ceramic container on a rock formation outside a campus building in natural sunlight. A hyperspectral camera able to collect 120 images within the 400 to 900 nanometers (visible to near infrared range) is used to collect the data. A color display is obtained (Fig 1c) by associating with the fundamental colors grayscale intensity images collected within the blue, red and green wavelength intervals. In addition, plotting vectors of the reflectance values for a specific location, we get the spectra for the material present at that location. In Fig. 2 we have three such spectra obtained for different materials present in the scene with the exact location of the data collection being given by the white arrows in Fig.1c: ceramic pot (Fig.2a - lower arrow), artificial plant (Fig.2b – top right arrow) and natural vegetation (Fig.2c – top left arrow). Note that while the artificial and natural vegetation are difficult to distinguish in the color image, their spectra are significantly different. The natural vegetation displays a graph with two peaks associated to the green and near infrared wavelengths while the artificial vegetation has only one peak in the green range. We also note the significantly different shape of the ceramic pot, emphasizing its red appearance.

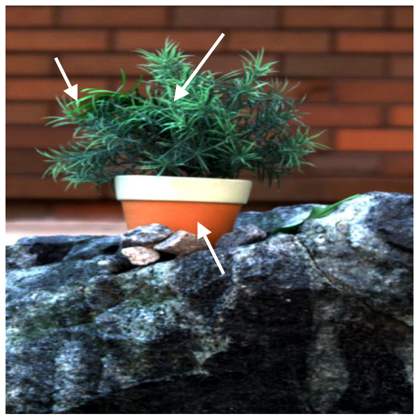
*robilas@mail.montclair.edu; phone 1 973 655-4230; fax 1 973 655-4164; http://www.csam.montclair.edu/~robila/RSL



(a)

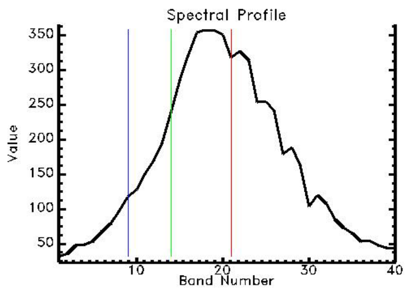


(b)

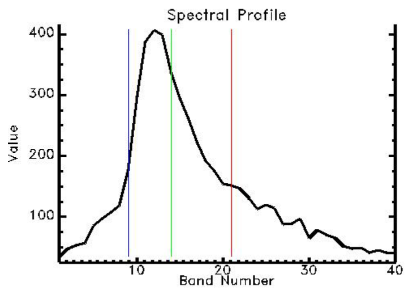


(c)

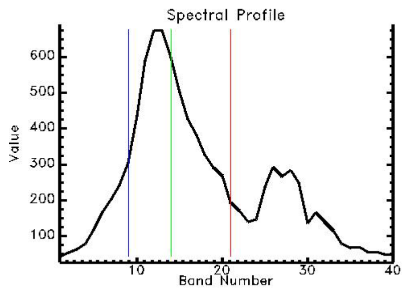
Fig. 1. a) Model for a hyperspectral data set as a stack of grayscale images b) Example of grayscale images from a hyperspectral data set corresponding to blue, green and red visible intervals (wavelengths centered at 495nm, 556nm, and 639 nm respectively) c) Color image formed by combining the three grayscale images in b according to their color of origin.



(a)



(b)



(c)

Fig. 2. Spectra of various materials a) ceramic container, b) artificial plant, c) real plant. The vertical lines correspond to bands in the blue, green and red areas (wavelengths centered at 495nm, 556nm, and 639 nm respectively) shown in Fig. 2b. The locations of the spectra collection are shown in white arrows on Fig. 1c.

While this example is illustrative of the richness of hyperspectral data, in reality, several other factors impede on the direct usage for object detection and identification. First, although spectral information is distinct among different materials, studies have shown that the variability within a single material is also often quite large [5]. For, example, every vegetation type displays considerable spectral variability both at different growth and humidity stages as well as within the same plants. Air density and humidity, light intensity and angle of incidence, as well as shadows, prevent the collection of correct spectra and lead to inaccurate results. In addition, the relatively coarse spatial resolution of the data (not a significant factor in Fig 1) for aerial and satellite images means that often a collected spectra is the combined result of several materials found in the area covered by the corresponding pixel [6]. As such, the spectral information is generally thought to be a mixture of basic spectra. Correctly modeling this mixture, as either linear or nonlinear, and then accurately providing an unmixing solution is still an open problem [2].

A second important aspect of hyperspectral data relates to the size and its redundancy [4]. Unlike regular imaging, in hyperspectral imagery we often work with hundreds of bands that combined reach sizes of tens of Megabytes. While computer technology continues to advance, supporting the speedup in processing, algorithmic techniques that efficiently use the computing environment are often needed to provide up-to-date and relevant results. Current efforts are mainly focused on data reduction (feature extraction [7]), and distributed and parallel processing [8].

Our approach contributes to a solution for both correct spectra identification and fast processing. The choice for unmixing algorithm, Nonnegative Matrix Factorization (NMF), is significantly more flexible that other approaches, such as Principal Component Analysis (PCA) [9], Independent Component Analysis (ICA) ([10, 11]) or Orthogonal Subspace

Projection (OSP) [12] by requiring relatively few restrictions on the original data or results. Previous studies support this view and suggested it as a viable unmixing method [13].

Unfortunately, as with many other approaches, NMFs processing time is strongly dependent to its iterative nature and has an update step with high computing complexity. A single run requires several tens of iterations to converge and can require hours to compute on common desktop architectures. Our approach is to use parallel processing, as provided by multiprocessor systems, now more readily available even as regular desktop. The parallel computation of the update step provides a significant speedup to the method.

The paper is organized as follows. In section 2 we provide a brief review of the spectral unmixing problem, focusing on the linear mixture model. In section 3 we present the general NMF problem and its application to unmixing. Section 4 introduces the parallel NMF algorithm we developed. Section 5 describes the experimental results obtained on hyperspectral data. The paper ends with Conclusions, Acknowledgement and References.

2. SPECTRAL UNMIXING

2.1 General Unmixing Problem

Due to the spatial resolution of hyperspectral images, a pixel usually covers an area occupied by more than one material. While this may be obvious when thinking of border or edge areas in the scene, the same thing applies even in homogenous area. A pixel from a vegetation scene, while covered mostly by the plants is often a combination of leaves, stems and exposed ground, to which we add the influence of shadow and light incidence angle [6]. Studying how all these factors contribute to forming a single reflectance value that is recorded by the hyperspectral camera as a pixel is a critical area of remote sensing research. The general unmixing problem assumes that a scene surveyed through a hyperspectral data is in fact formed as a mixture of a limited number of materials. Such materials are often called endmembers. In the unmixing problem we try to recover these endmembers and find how they contribute to each of the pixels in the image.

Previous efforts have suggested that the best model for such mixing is nonlinear [6]. Given the models for scattering of light, the random interleaving of materials, the repeated reflection of light (or energy) from one material to another, overall reflectance value obtained can be seen as a nonlinear combination of individual material reflectance together with additional components that account for angle of incidence, light energy, etc [6]. This approach, while theoretically sound, is unlikely to provide accurate solutions for correct endember identification due to its complexity. Instead, a simplified linear model was proposed.

2.2 Linear Mixing Model (LMM)

LMM is described as follow. Any *n*-dimensional observed spectra **x** from the image can be described as a linear combination of the same *m* *n*-dimensional spectra (*endmembers*) $\mathbf{s_1}$, ..$\mathbf{s_m}$, and noise [14]:

$$\mathbf{x} = \sum_{i=1}^{m} a_i s_i + \mathbf{w} = \mathbf{Sa} + \mathbf{w} \tag{1}$$

where **a** is an *m*-dimensional vector describing the fractional abundances of the endmembers in the mixture (abundance vector). The values of **a** must be positive and add up to 1:

$$a_i \geq 0, i = 1, ..., m \tag{2}$$

$$\sum_{i=1}^{m} a_i = 1 \tag{3}$$

If the materials are physically placed adjacent to each other and the scattering of light radiation is dominated at any point by a single material, then the observed values can be assumed to follow LMM [6]. However, if the materials in a pixel are grouped more closely (like mixture of various sand grain types, mineral mixtures – see the rock in the lower part of Fig. 1c, etc.) the light interacts with more than one material and the reflected value is closer to a nonlinear combination of the reflectance of the individual composing materials. In this case LMM will not work properly.

LMM can be solved through a two stage method performing identification of the endmembers followed by the computation of their abundances. When the endmembers are determined, inversion algorithms restricted by the Eqns 2 and 3 are used to obtain the abundance vectors. applied to compute the abundances. The endmembers can be identified based on general knowledge of the scene (and picked from laboratory spectral libraries) or extracted directly from the image by human observers or through selection algorithms (selection of the centroids as endmembers, orthogonal subspace projection with the projection vectors considered to be endmembers, or geometric determination with endmembers considered to be at the extremities of the spectra) [6]. An alternative approach is to compute the endmembers and abundances simultaneously. In this case, the endmembers and abundances are assumed deterministic and unknown and a maximum likelihood estimation approach is employed to determine them [15]. Simultaneous determination of both endmembers and abundances is more efficient than the two stage process, especially when no prior information on the image is available. Suggested approaches have included decorellation of the data through PCA or OSP, or the more general separation of independent features through ICA **[2]**. Recently, NMF was suggested as a better approach [13].

3. NONNEGATIVE MATRIX FACTORIZATION (NMF)

3.1 Sequential NMF

Given the observed *m* x *n* – dimensional data **x**, the goal of NMF is to find *k* x *n* dimensional **s** and a linear *m* x *p* mixing transform **W** both positively defined such that [16]:

$$\mathbf{x} = \mathbf{W}\mathbf{s} \tag{4}$$

In other words, the goal is to factorize the data matrix such that the new data matrix and the transform are positive. From this remark we reach a possible solution to NMF by minimizing any non-increasing function *f*(**W**,**s**) that takes the value zero only when Eqn.4 is satisfied and :

$$\mathbf{W} \geq 0, \mathbf{s} \geq 0 \tag{5}$$

Given such a function, a gradient descent approach can be envisioned with **W** and **s** randomly initialized and then updated by:

$$\mathbf{W} = \mathbf{W} - \frac{\partial f(\mathbf{W},\mathbf{s})}{\partial \mathbf{W}} \tag{6}$$

$$\mathbf{s} = \mathbf{s} - \frac{\partial f(\mathbf{W},\mathbf{s})}{\partial \mathbf{s}} \tag{7}$$

An elegant solution that ensures positivity is presented in [16]. The update steps are:

$$\mathbf{W} = \mathbf{W}\frac{\left(\mathbf{x}\mathbf{s}^{\mathrm{T}}\right)}{\left(\mathbf{W}\mathbf{s}\mathbf{s}^{\mathrm{T}}\right)+\varepsilon} \tag{8}$$

$$\mathbf{s} = \mathbf{s}\frac{\left(\mathbf{W}^{\mathrm{T}}\mathbf{x}\right)}{\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{s}\right)+\varepsilon} \tag{9}$$

And the optimization function is the Frobenius (or Euclidean) norm:

$$f(\mathbf{W},\mathbf{s}) = \left\|\mathbf{x} - \mathbf{W}\mathbf{s}\right\|_F^2 \tag{10}$$

At each step we also ensure that **W** and **s** are positive and s is normalized. The full algorithm is presented in Fig. 3.

```
1. Randomly initialize $\mathbf{W}$ and $\mathbf{s}$ to positive values
2. Scale the columns of $\mathbf{s}$ to sum up to one
3. Repeat:
        4. $s_{ij} = s_{ij} \frac{(\mathbf{W}^{\mathrm{T}}\mathbf{x})_{ij}}{(\mathbf{W}^{\mathrm{T}}\mathbf{W}\mathbf{s})_{ij} + \varepsilon}$
        5. $W_{ij} = W_{ij} \frac{(\mathbf{x}\mathbf{s}^{\mathrm{T}})_{ij}}{(\mathbf{W}\mathbf{s}\mathbf{s}^{\mathrm{T}})_{ij} + \varepsilon}$
        6. Scale the columns of $\mathbf{s}$ to sum up to one while non convergence
```

Fig. 3. Nonnegative Matrix Factorization Algorithm

We note that steps 4-6 will be repeated until a convergence criterion is satisfied. In our case, the convergence was based on a stability measure for $f(\, . \, , \, . \,)$ (the algorithm stops when f stabilizes). In our experiments, addition of simulated annealing that increased or decreased the update factor did not affect the convergence speed. Steps 4 and 5 are applied to each component of **s** and **W** respectively. The value of ε is relatively small and is mainly used to limit the effect of the local optima. We also note that compared with the original algorithm, the **s** data are scaled to add up to 1 columnwise.

3.2 NMF as solution to linear spectral unmixing

When solving LMM and trying to determine the endmembers and their abundances at the same time, the relationship among the endmembers becomes a major problem. In PCA and ICA, the endmembers correspond to rows in the linear transform and are orthogonal. This may be a condition too strict for endmember extraction where the base materials may be only slightly different from each other [1]. Additionally, a bigger hurdle comes when we analyze the abundance images. Traditional methods do not verify Eqns 2. and 3. Instead, least square approximations are computed after the endmembers are produced [14]. Compared to this, NMF can be seen as a natural solution to LMM. The resulting endmembers are not restricted by orthogonality and the abundance vectors are already positive. Our addition for the abundances to add up to one ensures that both LMM conditions are satisfied.

The above observation explains why NMF has become a popular research direction in linear unmixing. The algorithm in Fig. 3 while originally presented for general unmixing in [16] was also implemented for hyperspectral data unmixing [13]. Nevertheless, the tests were mainly done on a limited number of spectra and NMF was not applied on a full hyperspectral data set. In a previous paper [17], we investigated the appropriateness of the algorithm for full data and concluded that the results suggest results suggest that NMF reaches optimal solutions that clearly separate endmember information for the data. Since the two matrices were initialized to positive values and since the update step maintains the positivity, Eqn. 2 will hold for the results and Eqn. 3 is validated by step 6 in the algorithm. We also note that the current version of the algorithm needs to have the number of features predetermined apriori. In our case, we considered the number of features to be equal to the number of original spectral bands.

A major problem, however, was the computational complexity of the algorithm. Since NMF converged after approximately 100 iterations, the time required to perform the unmixing was quite large. The next section provides an algorithm that significantly reduces the execution time.

4. PARALLEL NONNEGATIVE MATRIX FACTORIZATION (P-NMF)

Parallel processing is often used as means to speed up compute intensive algorithms. In the case of NMF, the repetitive nature of the method, as well the ability to distribute the data in relatively distinct subsets supports our design for a parallel NMF. P-NMF is presented in Fig. 4. After the data are read, a sequential application creates and starts a number of parallel worker threads (Thread$_1$, Thread$_2$, …). In a programming application threads constitute separate 'threads of execution' that share data and code, the advantage being that they can be scheduled independently (parallel or not) for execution. After each thread updates the assigned portions, the master process can test for convergence. The grey-shaded areas in the figure correspond to sequential (or common) portions of the algorithm while the non-shaded area corresponds to parallel portions.

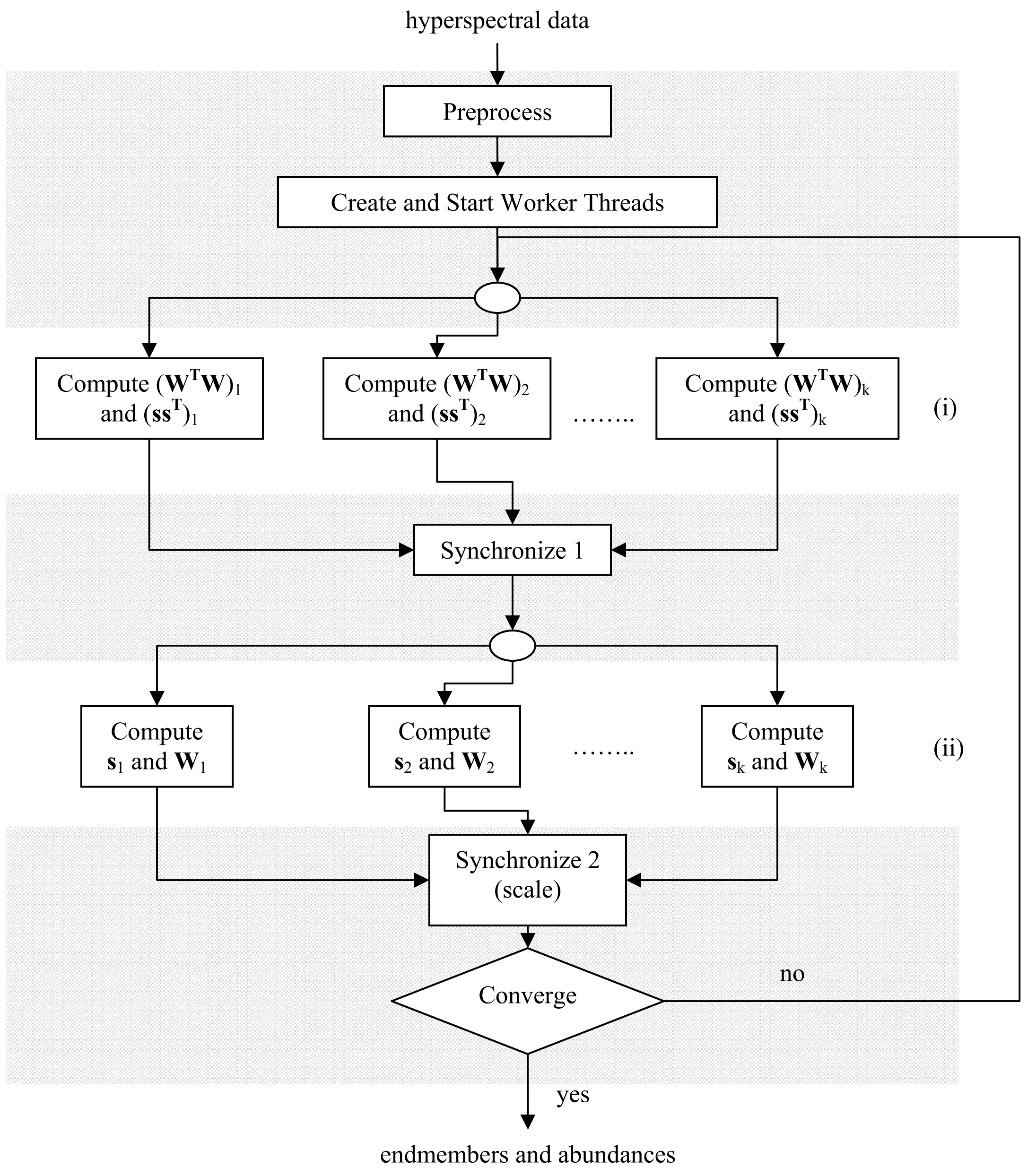

hyperspectral data
Preprocess
Create and Start Worker Threads
Compute $(\mathbf{W}^\mathbf{T}\mathbf{W})_1$ and $(\mathbf{s}\mathbf{s}^\mathbf{T})_1$
Compute $(\mathbf{W}^\mathbf{T}\mathbf{W})_2$ and $(\mathbf{s}\mathbf{s}^\mathbf{T})_2$
……..
Compute $(\mathbf{W}^\mathbf{T}\mathbf{W})_k$ and $(\mathbf{s}\mathbf{s}^\mathbf{T})_k$
(i)
Synchronize 1
Compute $\mathbf{s}_1$ and $\mathbf{W}_1$
Compute $\mathbf{s}_2$ and $\mathbf{W}_2$
……..
Compute $\mathbf{s}_k$ and $\mathbf{W}_k$
(ii)
Synchronize 2 (scale)
Converge
no
yes
endmembers and abundances

Fig. 4. Diagram of the distributed algorithm

Each of the threads will work on a separate part of **W** and **s**. Fig. 5 and Fig. 6 show how **W** and **s** are distributed among the threads. In the case of **s**, each thread will receive a relatively equal number of rows (or bands) and in case of **W** each thread will update a relatively equal number of columns (endmembers). The formula for assigning the data is as follows:

$$Thread_i : Update \quad \mathbf{s_i} = \mathbf{s}_{[i*\frac{p}{k}\ldots(i+1)*\frac{p}{k},\ldots]} \qquad \mathbf{W_i} = \mathbf{W}_{[\ldots.,i*\frac{p}{k}\ldots(i+1)*\frac{p}{k}]} \tag{11}$$

The original NMF algorithm was further modified for increased speedup by pre-computing $\mathbf{W}^\mathbf{T}\mathbf{W}$ and $\mathbf{s}\mathbf{s}^\mathbf{T}$ (that were used in steps 4 and 5 in the sequential algorithm described in Fig. 3). The resulting k x k matrices were computed in parts by the threads in a manner similar to W (see Fig. 6 and Eqn. 11). Two synchronized points were introduced to ensure that all the threads have access to the same data.

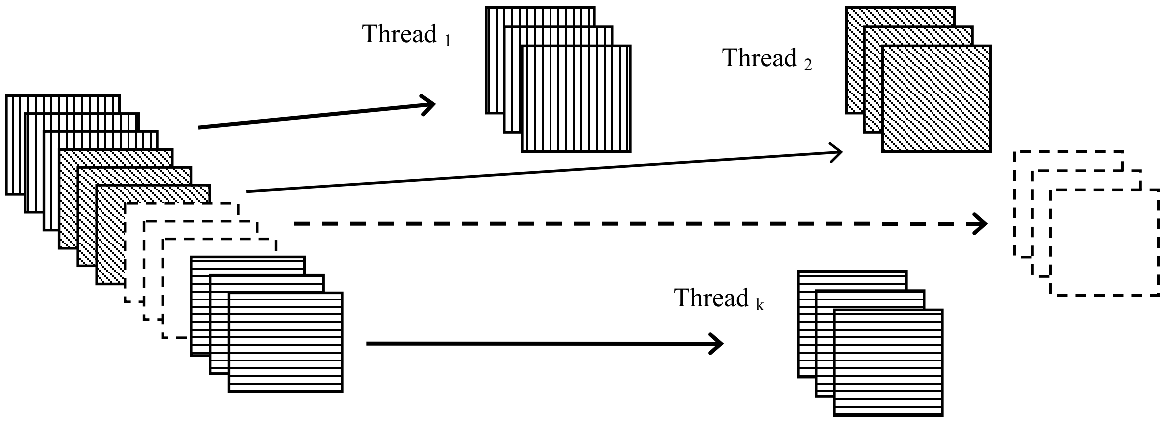


Fig. 5. Distribution of the s data for updating by threads. Each thread processes several of rows (i.e., an equal number of abundance bands).

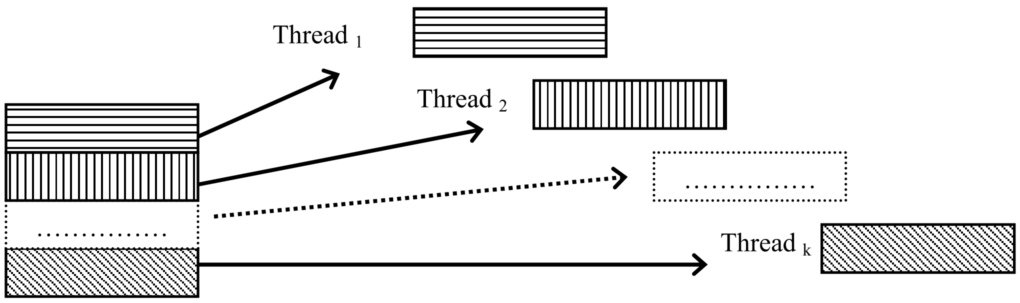


Fig. 6. Distribution of **W**, $\mathbf{W}^\mathbf{T}\mathbf{W}$ and $\mathbf{ss}^\mathbf{T}$ for updating by threads. Each thread processes several columns.

The complexity of the algorithm depends on the size of **x** (*m* x *n*), **W** (*m* x *p*), **s** (*p* x *n*) and the number of threads *k*. In the case of the computation of the common data (section (i) in Fig. 6), the complexity is:

$$O(\frac{p}{k} * p * m * n) = O(\frac{1}{k} p^2 mn) \tag{12}$$

The complexity for the second parallel section ((ii) in Fig. 6) can be computed in a similar manner as:

$$O(\frac{p}{k} * (n * m + n * p + m * n + m * p)) = O(\frac{1}{k}(pm(n+p) + pn(m+p))) \tag{13}$$

We note that the above formulas suggest that the parallel speedup is proportional with the number of threads. This is possible because the two sequential steps are computationally inexpensive (when compared with the parallel steps).

When redesigning an algorithm it is often necessary to ensure that the results will remain the same. In our case, we note that the modifications are semantically equivalent to steps in the original algorithm and do not modify its outcome. A semantic proof of this fact, while possible, was not among the goals of this paper. This equivalence is however employed when presenting the results.

5. EXPERIMENTAL RESULTS

We implemented P-MNF using Java [18]. A single application with multiple threads was designed, allowing for the use of *synchronized* method feature as an elegant way to ensure the synchronization of the parallel parts. After compilation, the code was run on a SunFire v880 machine. The system has 4 UltraSparc9 processors running at 750Mhz and 8GB of RAM running Solaris.

Two data sets, were used for the experiments and their images are shown in Fig. 7. First, we processed a hyperspectral image from the Hyperspectral Digital Imagery Collection Experiment (HYDICE) (Fig. 7a) [19]. It corresponds to a foliage scene taken from with a spatial resolution of 1.5m at wavelengths between 400nm and 2.5 micron. The data set uses sub-scenes provided by the Spectral Information Technology Application Center and has 85x185 pixels and 40 bands.

The second data set (Fig. 7b) was produced using a SOC 700 hyperspectral sensor currently available in our lab. The image was a 160x160 pixel image 120 bands equally spaced within the 400nm and 900nm (i.e. visible to near-infrared range). Forty bands uniformly extracted from the image cube were used for the experiment. The set-up is formed of an artificial plant arranged in a light brown ceramic pot. Several real leaves (shown in enhanced green in the picture) were placed between the artificial leaves (left side, top and lower right side of the green area). To benefit from full spectrum illumination, the arrangement was placed outside on a large rock formation. The background is formed of a brick wall.

We note that the same images were used in [17] and shown to produce promising results in endmember extraction. For brevity, we will focus the discussion of the results on the speedup obtained through parallel processing, referring the reader to [17] for a discussion on the accuracy of sequential NMF.

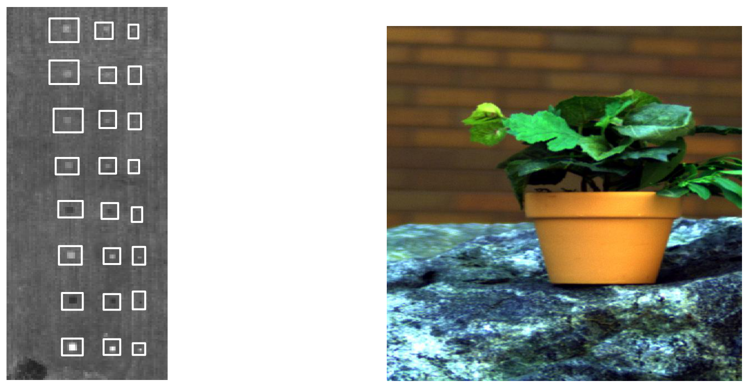
For each of the data sets, we executed five different P-NMF runs with 25 iterations each and the number of threads varying from 1 (sequential) to 8 for a total of 40 P-NMF runs. Fig. 8 displays the maximum variability for each of the data sets. For each situation (number of threads ranging from 1 to 8) we computed the maximum and minimum iteration execution time. The graph displays the differences between the maximum and minimum as percentages of the average execution time. We note that the variation is reasonably small, the maximum values reaching 14% for the Hydice data. Fig. 9 presents the average execution time for the two data sets (solid lines). For comparison, we also plotted a dotted line representing the ideal execution times as estimated from the complexity analysis done in the previous section. We note that our prediction is met by the experimental results up to four threads followed by a stability or increase of execution time starting with the case of five threads. The slight difference in execution times is explained by the overhead introduced by synchronization as well as by multiple accesses to the same memory area (for computing the matrix operations by the threads). The significant performance decrease however, noticed starting with five threads is due to the fact that our system was using up to four processors. Had the system been equipped with more CPU's, the performance improvement would have been continued. We also note an unusual performance increase for eight threads in the case of the SOC 700 data. To better understand this behavior, further test runs are planned. Current investigation of the code or the experimental data did not reveal anything relevant to explain it.

Finally, Fig. 10 displays the average speedup. For this, we divided the sequential execution time by the execution time for each thread number case. This allows us to compare both data set runs with the same measure and investigate the consistency of the results. For illustrative purposes, the ideal speedup (computed as the number of threads) is also presented. The graph confirms our preliminary analysis in suggesting the P-MNF outperforms sequential MNF consistently up to the number of processors.

6. CONCLUSIONS

We introduced a parallel spectral unmixing algorithm for hyperspectral images. P-NMF originates from the classical Nonnegative Matrix Factorization solution modified to ensure additivity to one of the abundances. While previous work suggested NMF as a viable tool for unmixing, it was also clear that the time complexity of the method will strongly affect the overall usability. P-MNF is the parallel NMF algorithm, further improved by modification of the order the data is computed. Overall, the P-NMF algorithm performed robustly and significantly increased the time performance of the endmember extraction. Of particular interest is the fact that the newly introduced method is semantically equivalent to its sequential counterpart.

While the algorithm was designed mainly for hyperspectral data, is can be applied to any situation where NMF is employed. Its elegance and simplicity allow for speedups directly proportional to the number of processors available in the parallel machine. The implementation in a high level language (Java) ensures ease of portability and further development. In addition, further changes in the sequential NMF algorithm can be easily ported in the parallel counterpart.



(a) (b)

Fig. 7. Experimental Data a) Hydice data set with panels of various materials. B) SOC 700 data set with real and artificial vegetation.

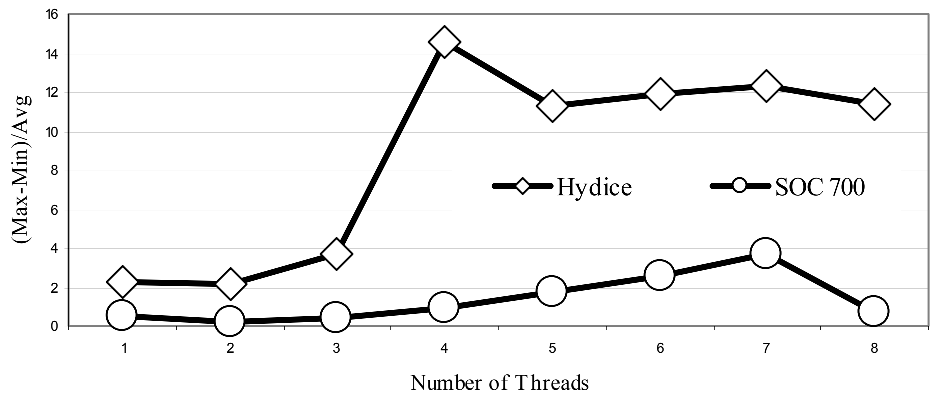
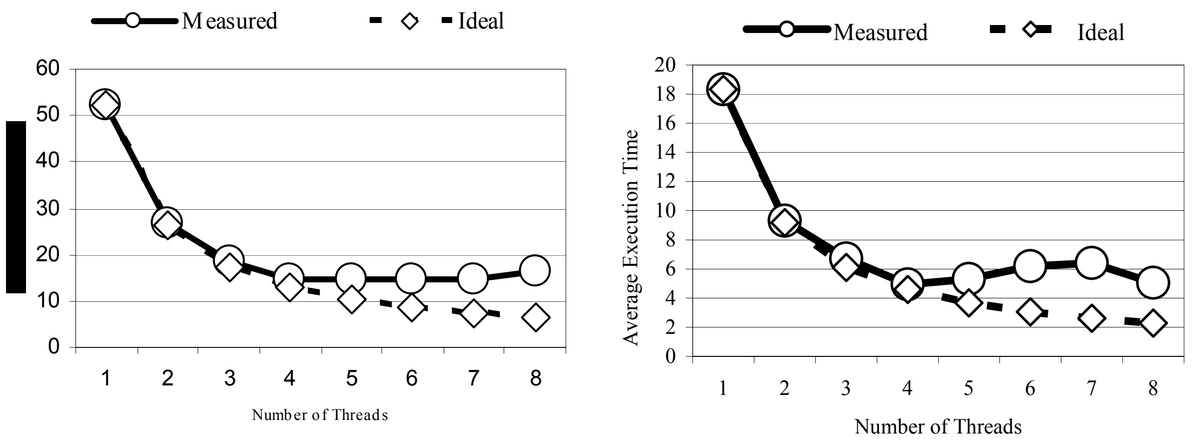


Fig. 8. Variability among the P-MNF runs. Maximum difference in execution time for an iteration as percentage of the average execution time.



(a) (b)

Fig. 9. Execution time of each P-MNF iteration for multithreaded run. a) Hydice data, b) SOC 700 data

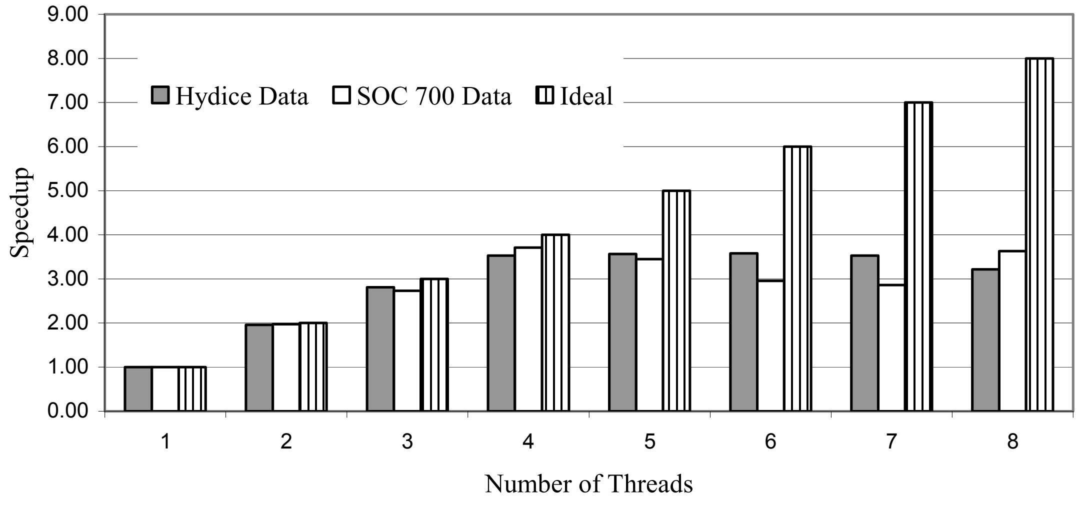


Fig. 10. Comparative P-MNF speedup obtained for the two data sets and the ideal speedup.

Given an increased use of cluster and distributed computing, it is of interest to see how easily would P-NMF be ported to become 'D-NMF' and how would the performance be affected. Recent developments in Java grid technology that allow portability of threads from one system to another **[Ref]** suggests that P-NMF may be in fact already able to run as a distributed application. Barring this possibility, a distributed version will have to take in consideration the overhead introduced by communicating data among the system, factor significantly reduced in a parallel system Nevertheless, this may be countered by improved computational performance available through individual systems as well as increased cost effectiveness.

7. ACKNOWLEDGMENT

The work was made possible by a 2005-2006 Sun Microsystems Academic Excellence Grant, as well as 2006-2008 Montclair State University FSP program

REFERENCES

1. D. Landgrebe, Signal Theory Methods in Multispectral Remote Sensing, Wiley – Interscience, 2003.
2. C.-I. Chang, Hyperspectral Imaging: Techniques for Spectral Detection and Classification, Kluwer Academic, 2003.
3. Luukanen, A.; A.J. Miller, and E.N. Grossman, "Passive hyperspectral terahertz imagery for security screening using a cryogenic microbolometer", SPIE Radar Sensor Technology IX, vol. 5789, pp. 127-134 2005.
4. J. A. Richards, and X. Jia, Remote Sensing Digital Image Analysis, Springer, 1999.
5. P.H. Crown, and D.L. Klita, "Spectral separability of conservation cover classes in the Parkland region of northeast Alberta", Bringing Conservation Technology to the Farm, PARI Handbook, paridss.usask.ca/factbook/titlepge.html, 1997.
6. N. Keshava, "A Survey of Spectral Unmixing Algorithms", Lincoln Laboratory Journal, vol. 14, No. 1, pp. 55-68, 2003.
7. S.A. Robila, "Distributed source separation algorithms for hyperspectral image processing", SPIE Enabling Technologies for Simulation Science VIII, vol. 5425, pp. 628-635, 2004
8. A. Plaza, D. Valencia, J. Plaza and P. Martinez, "Commodity Cluster-Based Parallel Processing of Hyperspectral Imagery", Journal of Parallel and Distributed Computing, in press, 2005.

9. P.J. Ready, and P.A. Wintz, “Information extraction, SNR improvement, and data compression in multispectral imagery”, IEEE Transactions on Communications, Vol. Com-21, pp. 1123-1130, 1973
10. Hyvärinen, A., J. Karhunen, and E. Oja, Independent Component Analysis, John Wiley & Sons, New York, 2001
11. S. A. Robila, “Independent Component Analysis (ICA)”, in P.K. Varshney, M.K. Arora editors. Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data, Springer, New York, 2004, pp. 109 - 132.
12. J.C. Harsanyi, and C.-I. Chang, “Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection”, IEEE Transactions on Geoscience and Remote Sensing, 32, 779-785, 1994.
13. P. Pauca, J. Piper, and R. Plemmons, "Nonnegative Matrix Factorization for Spectral Data Analysis", to appear in Linear Algebra and Applications, 2006
14. A. Plaza, P. Martinez, R. Perez and J. Plaza. \A Quantitative and Comparative Analysis of Endmember Extraction Algorithms from Hyperspectral Data", IEEE Transactions on Geoscience and Remote Sensing, Vol. 42, No. 3, pp. 650-663, 2004
15. N. Keshava, and J.F. Mustard, “Spectral unmixing”, IEEE Signal Processing Magazine, 19, no. 1, 44-57, 2002.
16. D. Lee and H. Seung. “Algorithms for Non-Negative Matrix Factorization", Advances in Neural Processing, 2000
17. S. A. Robila, L. Maciak, “Novel Approaches for Feature Extraction in Hyperspectral Images”, Proceedings IEEE LISAT, 7 pgs. on CD, 2006
18. Sun Microsystems, Java Programming Environment. http://java.sun.com/ , 2006
19. Hyperspectral Digital Imagery Collection Experiment Documentation, August, 1995.