PARALLEL IMPLEMENTATION OF UNMIXNG ALGORITHM FOR VARIABLE-ENDMEMBER LINEAR MIXTURE MODEL

Naoki Takahashi

SGI Japan
Yebisu Garden Place Tower 31F,
4-20-3 Ebisu, Shibuya-ku,
Tokyo 150-6031, Japan

Kenta Obata, Hiroki Yoshioka

Department of Applied Information Science
and Technology, Aichi Prefectural University
1522-3 Kumabari, Nagakute,
Aichi 480-1189, Japan

1. INTRODUCTION

This paper describes parallel implementation of unmixing algorithm for a recently developed linear mixture model, variable-endmember linear mixture model (VELMM) [1]. The model has been developed to retrieve two independent biophysical parameters, namely, leaf area index (LAI) and fraction of vegetation cover (FVC) from remotely sensed surface reflectance. Since the model consists of a linear sum of two nonlinear functions, the unmixing algorithm involves a constrained optimization process which takes a considerable computation time [2, 3]. For the purpose of parameter retrieval from satellite images, computation cost becomes impractically large (nearly a week by a workstation of decent specification), since this unmixing algorithm should be applied to all the pixels one by one. In order to routinely use the algorithm, its cost must be dramatically reduced, where parallel implementation plays an important roll [4, 5, 6, 7, 8, 9, 10] as one realistic solution.

The objectives of this paper are

1. to speed up the parameter retrieval by implementing the unmixing algorithm onto a parallel machine (distributed memory multi-core system, using up to 256-core simultaneously),
2. to investigate parallel efficiency of the algorithm as a function of number of core used as well as task-parallel strategy/scheme (how to distribute tasks among different nodes, cpus, and cores over the system), and
3. to discuss applicability of our implementation strategy, in general, to pixel-based parameter retrieval from spectral reflectance data by environmental satellites.

2. VARIABLE-ENDMEMBER LINEAR MIXTURE MODEL

Unlike the standard linear mixture mode (LMM), variable-endmember LMM (VE-LMM) [2, 3, 11, 12] is to take into account variability of endmember spectra within a class. Our VE-LMM assumes two classes (vegetation and non-vegetation) with four independent parameters in the model. The reflectance spectrum from a target pixel $\hat{\boldsymbol{\rho}}$ is modeled by mixing two spectra of variable endmember, $\boldsymbol{r_v}(p_1)$ and $\boldsymbol{r_s}(p_2)$ as

$$\hat{\boldsymbol{\rho}}(\omega_1, \omega_2, p_1, p_2) = \omega_1 \boldsymbol{r_v}(p_1) + \omega_2 \boldsymbol{r_s}(p_2) \tag{1}$$

where ω_1 and ω_2, are the weights of the endmembers that represent fractions of area covered by green vegetation and non-vegetation, respectively. These two parameters are constrained into unity, $\omega_1 + \omega_2 = 1$ in this study. The other parameters, p_1 and p_2, are related directly to the endmember determination of the two classes: For vegetation endmember, leaf area index (LAI) is chosen for p_1, and for non-vegetation endmember, reflectance of red band is chosen as p_2 to represent soil brightness.

In the model, the spectrum of each class varies continuously along with a single line in a reflectance subspace (Fig. 1). A set of spectra were chosen from actual satellite images and radiative transfer models by fixing soil brightness (constant soil-brightness line). While, endmember spectrum of non-vegetation class is chosen from a group of spectra that forms a soil line (amount of vegetation is fixed at zero.) By setting such constraints, reflectance of each endmember forms a single curve in reflectance subspace (Fig. 1).

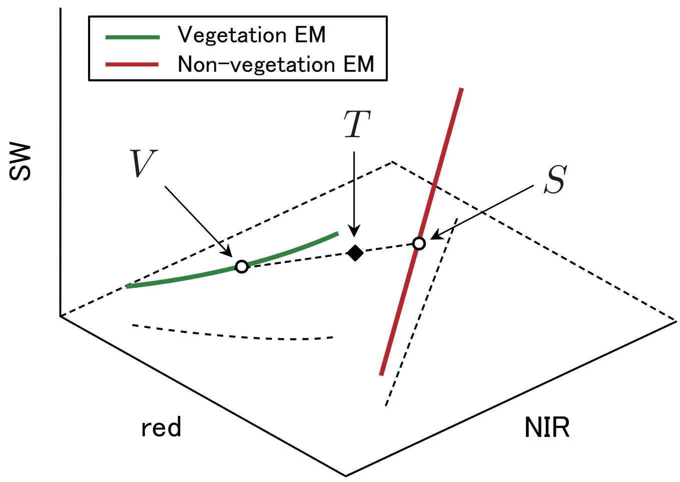

Fig. 1. Illustrtion of VE-LMM in reflctance subspace

3. UNMIXING ALGORITHM

A suitable pair of endmember spectra are chosen numerically by unmixing the model out of these infinite number of endmember spectrum, which allows us to determine corresponding amount of biomass as a continuous variable (in stead of discrete one) in addition to a fraction of vegetated area as a weight of vegetation endmember.

Using the unity constraints explicitly in the model equation, three parameters are left undetermined. The determinations of those three are done from the information of observed reflectance spectrum, represented by a vector $\boldsymbol{\rho_t}$. The following cost function is defined as a difference between $\boldsymbol{\rho_t}$ and $\hat{\boldsymbol{\rho}}$,

$$h(\omega_1, p_1, p_2) = ||\boldsymbol{\rho_t} - \hat{\boldsymbol{\rho}}(\omega_1, p_1, p_2)||_2. \tag{2}$$

The three parameters are determined by finding a set of parameters which minimizes the above cost function. For the solutions of the above optimization problem, a function of Quasi-Newton method provided in the numerical library of MATLAB was employed. One example of such a process is shown in Fig. 2. In the figure, the vector of initial guess is updated after each iteration and then finally reached to the global minimum which gives the solution to the problem. Therefore, the target of parallel implementation is mainly this part of the algorithm.

4. PARALLEL IMPLEMENTATION STRATEGY

Since the parameter retrieval of each pixel is totally independent of the other pixels, the computation for one pixel can be assigned to a single core independently. In a distributed memory parallel environment with N processing units, it is possible to perform simultaneous retrievals from N pixels. This strategy is illustrated in Fig. 3. The detailed explanation of the implementation will be explained in the conference.

5. PRELIMINARY RESULTS

A satellite image by Landsat7-ETM+ sensor was used to investigate parallel efficiency of the unmixing algorithm. Parallel computation was performed on a distributed memory system (cluster system) consisted of 32 blade computers equipped with two quad-core CPUs on each blade, hence total of 256 cores can be used in parallel. (The CPU is Xeon 5570, 2.93 GHz.) The algorithm is written using MATLAB functions and implemented in the system on STAR-P environment. To compare computation time with a stand-alone machine, we run the same code on a workstation of similar specification (Xeon 5160, 3.00 GHz).

We compared the performance with different image/data size of 1000 by 1000, 3000 by 3000, and 6000 by 6000. The results were summarized in Table 1. The cluster machine runs the code nearly 100 times faster at the full size of the image, although the computation time by the workstation is an estimation from the case of smaller size of image. Further results and discussions will be presented at the meeting.

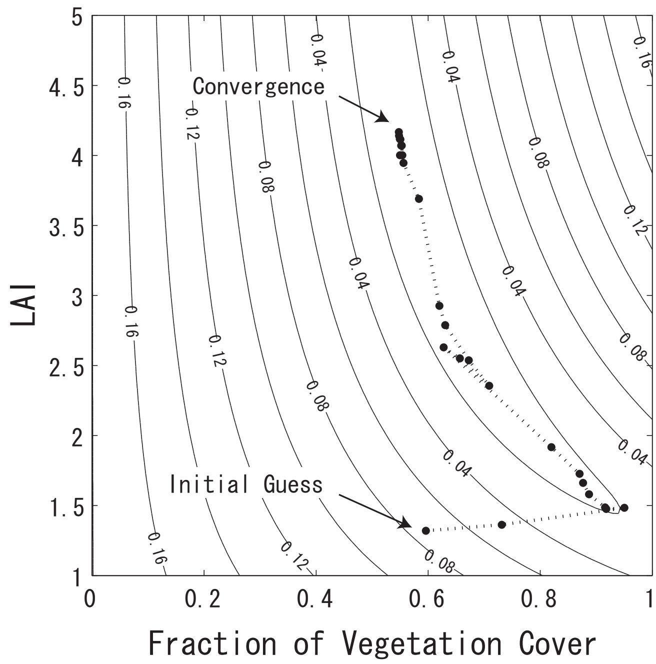

Fig. 2. Example of optimization process by unmixing algorithm

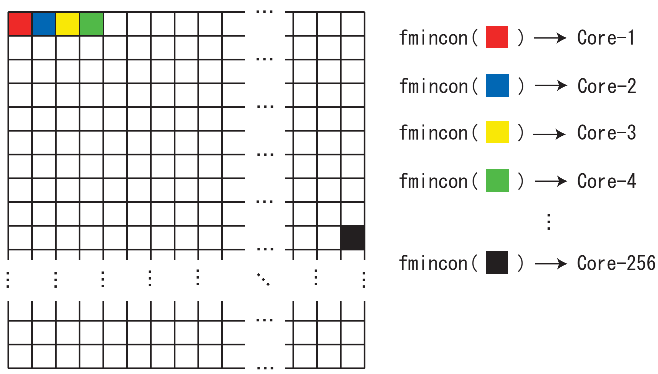

Fig. 3. Illustration of parallel implementation strategy

Table 1. Computation time and parallel efficiency of unmixing algorithm

Data size	1000×1000	3000×3000	6000×6000
Workstation with single CPU	4.5 h	38.2 h	7 days (estimated)
Parallel machine with 256 cores	0.06 h	0.45 h	1.5 h
Efficiency	×75	×85	×112

6. REFERENCES

[1] K. Obata, H. Yoshioka, and H. Yamamoto, “Unmixing a variable endmember linear mixture model for estimation of heat island mitigation by green vegetation,” in *IEEE IGARSS08*, 2008, vol. V, pp. 498–501.

[2] D. A. Roberts, M. Gardner, R. Church, S. Ustin, G. Scheer, and R. O. Green, “Mapping chaparral in the santa monica mountains using multiple endmember spectral mixture models,” *Remote Sens. Environ.*, vol. 65, pp. 267–279, 1998.

[3] C. A. Bateson, G.P. Asner, and C.A. Wessman, “Endmember bundles: a new approach to incorporating endmember variability into spectral mixture analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 38, pp. 1083–1094, 2000.

[4] A. Plaza and C.-I. Chang, “High-performance computer architectures for remote sensing data analysis: overview and case study,” in *High Performance Computing in Remote Sensing*, A. J. Plaza and C.-I. Chang, Eds., pp. 9–41. Chapman & Hall/CRC: Boca Raton, FL, 2007.

[5] D. Gillis and J. H. Bowles, “Parallel implementation of the ORASIS algorithm for remote sensing data analysis,” in *High Performance Computing in Remote Sensing*, A. J. Plaza and C.-I. Chang, Eds., pp. 69–95. Chapman & Hall/CRC: Boca Raton, FL, 2007.

[6] J. C. Tilton, “Parallel implementation of the recursive approximation of an unsupervised hierarchical segmentation algorithm,” in *High Performance Computing in Remote Sensing*, A. J. Plaza and C.-I. Chang, Eds., pp. 97–107. Chapman & Hall/CRC: Boca Raton, FL, 2007.

[7] G. P. Asner, R. S. Haxo, and D. E. Knapp, “Computing for analysis and modeling of hyperspectral imagery,” in *High Performance Computing in Remote Sensing*, A. J. Plaza and C.-I. Chang, Eds., pp. 109–130. Chapman & Hall/CRC: Boca Raton, FL, 2007.

[8] A. Plaza, D. Valencia, J. Plaza, and C.-I. Chang, “Parallel implementation of endmember extraction algorithms from hyperspectral data,” in *IEEE Geosci. Remote Sens. Lett.*, 2006, vol. 3, pp. 334–338.

[9] A. Plaza and J. Plaza, “Parallel morphological classification of hyperspectral imagery using extended opening and closing by reconstruction operations,” in *IEEE IGARSS08*, 2008, vol. 1, pp. I–580–I–61.

[10] C. Gerardino, Y. Rivera, J. Goodman, and W. Rivera, “Parallel implementation of an inversion model for hyperspectral remote sensing,” in *IEEE Mid-West Symp. Circuits Syst. (MWSCAS)*, 2006, vol. 2, pp. 606–609.

[11] C. Song, “Spectral mixture analysis for subpixel vegetation fractions in the urban environment: How to incorporate endmember variability,” *Remote Sens. Environ.*, vol. 95, pp. 248–263, 2005.

[12] J. Settle, “On the effect of variable endmember spectra in the linear mixture model,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, pp. 389–396, 2006.