Two-level parallel independent component analysis endmember extraction algorithms

LUO Wenfei[1], GAO Lianru[2]

1. *School of Geographical Science*, *South China Normal University*, *Guangdong Guangzhou*, 510631, *China*;
2. *Center for Earth Observation and Digital Earth*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*

Abstract: This paper analyzes parallel ICA algorithm in symmetrical multi-processing (SMP) cluster architecture. Based on our proposed single-level share memory model parallel ICA algorithm, two-level synchronous and asynchronous parallel ICA algorithms are presented, respectively, in the manner of synchronous and asynchronous parallel iteration for computing fixed-point function. By the use of these algorithms, two-level grouping parallel ICA algorithm is also proposed. In experiment with real hyperspectral remote sensing image, synchronous and grouping parallel ICA algorithms maintenance the endmember extraction accuracy with respect to the original algorithm. Meanwhile, they also demonstrate high parallel performance and greatly improve the performance of ICA endmember extraction. Asynchronous parallel ICA algorithm is suitable for the case of small number of nodes in cluster.

Key words: hyperspectral remote sensing, independent component analysis, endmember extraction, symmetrical multi-processing, cluster, parallel computing

CLC number: TP751.1 **Document code:** A

Citation format: Luo W F and Gao L R. 2011. Two-level parallel independent component analysis endmember extraction algorithms. *Journal of Remote Sensing*, **15**(6): 1202–1214

1 INTRODUCTION

Spectral unmixing is the key technique in hyperspectral remote sensing analysis. Endmember extraction (EE) is the most important and complicated procedure in spectral unmixing (Chang, 2007). Over the last decade, many endmember extraction algorithms (EEAs) have been developed, which can be categorized into three types, *i.e*. geometry based EEAs, statistics based EEAs and spatial information incorporating EEAs. Geometry based EEAs made a fast development in the early periods. Classic algorithms include the pixel purity index (PPI) (Boardman, 1995), the N-FINDR algorithm, unsupervised target generation process (ATGP) (Chang, 2003), vertex component analysis (VCA) (Nascimento, 2005). These algorithms are convenient for computation and have been widespread today. Iterative error analysis (IEA) (Neville, 1999) is a statistics based EEA that extracts endmembers by lease squares error. Recently, the techniques of blind signal separation (BSS) and projection pursuit (PP) are introduced into EE and enriched statistics based EEAs. Independent component analysis (ICA) is a classic spectral unmixing BSS algorithm (Bayliss, 1997; Chang, 2002; Nascimento, 2005), which is also used in EE (Wang, 2006). Spatial information incorporating EEAs, aiming at including the spatial information in the process of endmember extraction, have been widely developed, such as automated morphological endmember extraction (AMEE) (Plaza, 2002), spatial-spectral endmember extraction (SSEE) (Rogge, 2006) and spatial preprocessing (SPP) method (Zortea & Plaza, 2009).

However, most of EEAs consume a large amount of time when are applied to the large volume hyperspectral remote sensing images. It is a data intensive and computation intensive task which becomes a serious bottle-neck in many applications with time-critical constrains. High performance computing (HPC) technique has necessarily become a requirement in hyperspectral remote sensing image analysis. It has been a hot-spot research field in remote sensing image processing (Plaza, 2009). Recently, many state-of-the-art parallel EEAs have been developed, including parallel nonnegative matrix factorization (NMF) algorithms (Stefan, 2006; Dong, 2010), parallel endmember extraction algorithms and parallel automated morphological endmember extraction (P-AMEE) (Plaza, 2006). Du (2006) developed a parallel ICA algorithm in single process multiple data (SPMD), which estimated different independent components in each partitioning and then used internal decorrelation and external decorrelation to generate independent components. And its hardware implementation was also discussed (Du, 2004).

Received: 2010-09-28; **Accepted:** 2011-01-10

Foundation: National Natural Science Foundation of China (No.40901232/D010702; No. 40901225/D010702); National High Technology Research and Development Program of China(863 Program)(No.2008AA12Z113); National Basic Research Program of China (973 program)(No.2009CB723902)

First author biography: LUO Wenfei (1979—), male, received the Ph.D. degree in Institute of Remote Sensing Applications Chinese Academy of Sciences. His research interests are remote sensing image processing and hyperspectral remote sensing. E-mail: luowenfei@irsa.ac.cn

Since the performance of personal computer (PC) and computer network has been greatly improved, cluster of workstation (COW) has been a widespread parallel architecture (Chen, *et al*., 2002). Nowadays, in the multi-core CPU era, symmetrical multiprocessing (SMP) cluster becomes one of the most popular parallel architecture (Chai, *et al*., 2002).

By taking advantage of share memory and distribution memory parallel model provided by SMP cluster, this paper considers two-level parallel implementations for ICA EEA in SMP cluster environment. We first develop synchronous parallel ICA EEA in the share memory parallel model. And then two-level synchronous parallel ICA EEA is developed in SMP cluster environment. Based our proposed spatial domain random sub-image partitioning algorithm for data partitioning and our accelerated cascading algorithm for endmember reduction, two-level asynchronous parallel ICA EEA is further developed. Based on the performance of two-level synchronous and asynchronous parallel ICA EEA, two-level grouping parallel ICA EEA is presented. It achieves better scalability and provides higher computing performance for endmember extraction. Finally, we show an experiment for evaluating and comparing the performance of our proposed algorithms by using real hyperspectral remote sensing image.

2 BASIC ALGORITHM

Let hyperspectral image after having subtracted its mean be

$$\boldsymbol{X}=[\boldsymbol{x}(1), \boldsymbol{x}(2),\cdots, \boldsymbol{x}(n)] \tag{1}$$

where $\boldsymbol{x}(i)$, i=1, 2,⋯, n is the vector of a spectral pixel and n is the total number of the image. According to the inear spectral mixture model, all spectra $\boldsymbol{x}(i)$, i=1, 2, ⋯, n can be expressed by

$$\boldsymbol{x}(i)=\sum_{j=1}^{m}\boldsymbol{e}_j\boldsymbol{a}_j(i) \tag{2}$$

where $\mathbf{e}_j$, j=1, 2,⋯, m is the endmember in the image, and $\boldsymbol{a}_j(i)$ is the abundance fraction corresponding to $\boldsymbol{e}_j$ in the spectral pixel $\boldsymbol{x}(i)$. The abundance fractions in $\boldsymbol{x}(i)$ compose of an abundance vector denoted by

$$\boldsymbol{a}(i)=[\boldsymbol{a}_1(i),\cdots,\boldsymbol{a}_m(i)]^{\mathrm{T}} \tag{3}$$

To simplify the procedure of BSS, ICA first applies principal components analysis (PCA) to the observed variables, *i.e*. the spectral pixels in the image, so that the components are uncorrelated and their variances acquire equal unity. Then ICA finds a separation matrix $\boldsymbol{W}$ by nongaussianity, satisfying

$$\hat{\boldsymbol{a}}(i)=\boldsymbol{W}\boldsymbol{Z}\boldsymbol{x}_i \tag{4}$$

where $\hat{\mathbf{a}}(i)=[\hat{\mathbf{a}}_1(i), \cdots, \hat{\mathbf{a}}_m(i)]^{\mathrm{T}}$ is the estimation of $\boldsymbol{a}(i)$, and $\boldsymbol{Z}$ is the transformation matrix defined by PCA. Denote

$$\boldsymbol{Y}=\boldsymbol{Z}\boldsymbol{X} \tag{5}$$

FastICA algorithm (Hyvärinen, 1999) is based on a fixed-point iteration scheme to find the maximum of the nongaussianity measured by the high order statistics and the approximation of negentropy, which the convergence is fast and reliable. $\boldsymbol{W}$ is given by

$$\boldsymbol{W}^{(k+1)}:=E\left\{\boldsymbol{Y}\mathrm{g}\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}\right)\right\}-E\left\{\mathrm{g}'\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}\right)\right\}\boldsymbol{W}^{(k)} \tag{6}$$

where $\boldsymbol{W}^{(k+1)}$ denotes the new matrix of $\boldsymbol{W}^{(k)}$ in the $(k+1)^{\mathrm{th}}$ iteration. The definitions of g and g′ can be referred to the previous work of Hyvärinen (1999). To prevent different vectors from converging to the same maxima, $\boldsymbol{W}^{(k+1)}$ should be decorrelated and renormalized after each iteration.

In each of the selected p FastICA-generated independent component(IC) images, endmembers can be extracted by finding a pixel with the maximum absolute value (Wang, 2006). The ICA-based endmember extraction algorithm (ICA-EEA) is summarized as follows.

Step 1 Use PCA to uncorrelated the image and then obtain a new image $\boldsymbol{Y}$;

Step 2 Initialize $\boldsymbol{W}^{(0)}$;

Step 3 Calculate $\boldsymbol{W}^{(k+1)}$ by Eq. (6);

Step 4 To prevent from converging to the same maxima, decorrelate and renormalize $\boldsymbol{W}^{(k+1)}$ and go to step 3 until satisfy FastICA convergence condition;

Step 5 After terminating the procedure of iteration, estimate the abundance $\hat{\boldsymbol{a}}_i=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{Y}$, i=1,⋯, m where $\boldsymbol{w}_i$ is the i-th IC;

Step 6 Select the endmembers by

$$\mathbf{e}_i:=\boldsymbol{x}_{ki} \text{ and } k_i=\underset{1\le j\le n}{avg\max}(\left|\boldsymbol{a}_i(j)\right|) \tag{7}$$

Step 7 Return all endmembers.

3 TWO-LEVEL SYNCHRONOUS PARALLEL ICA ALGORITHM

Step 3 in ICA-EEA occupies the majority time of the serial program. Therefore, how to improve the speed of calculating $\boldsymbol{W}^{(k+1)}$ is the key of designing a parallel algorithm.

3.1 Share memory model parallel ICA

SMP processors share the memory in the parallel model. The efficiency can be improved by attaching threads, called thread elements (TEs), to each processor (Quinn, 2002).

Let the number of TEs be a, the image $\boldsymbol{Y}$ is decomposed into $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_a\}$ by using the partitioning rule of successive I/O access. The j-th TE contains the partitioning $\boldsymbol{Y}_j$, j=1, 2, ⋯, a and calculates Eq. (6) as follows.

$$\boldsymbol{W}(j)^{(k+1)}=E\left\{\boldsymbol{Y}_j\mathrm{g}\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}_j\right)\right\}-E\left\{\mathrm{g}'\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}_j\right)\right\}\boldsymbol{W}^{(k)} \tag{8}$$

Obviously

$$\boldsymbol{W}^{(k+1)}:=\frac{1}{a\sum_{j=1}^{a}s_j}\sum_{j=1}^{a}s_j\boldsymbol{W}(j)^{(k+1)} \tag{9}$$

where $\boldsymbol{S}_j$ denotes the total number of samples in.$\boldsymbol{Y}_j$ Therefore, share memory model parallel ICA (SMM-PICA) algorithm is summarized as follows.

Step 1 Decompose $\boldsymbol{Y}$ into $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_a\}$;

Step 2 Initialize $\boldsymbol{W}^{(0)}$;

Step 3 For each TE, calculate Eq. (8);

Step 4 Wait for all TEs and reduce $\boldsymbol{W}^{(k+1)}$ by Eq. (9);

Step 5 Decorrelate and renormalize $\boldsymbol{W}^{(k+1)}$ and go to step 3 until satisfy the FastICA convergence condition;

Step 6 The i-th TE, j=1, ⋯, a, estimate the abundance as

$$\hat{\boldsymbol{a}}_{i,j}=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{Y}_j,\ i=1,\cdots,m \tag{10}$$

Step 7 And extract a candidate endmember by

$$\boldsymbol{e}_{i,j}:=\boldsymbol{x}_{k_i,j} \text{ and } k_i=\underset{1\le l\le n}{avg\max}(\left|\boldsymbol{a}_{i,j}(l)\right|) \tag{11}$$

Step 8 The master wait for $\boldsymbol{e}_{i,j}$ i=1, ⋯, m, j=1, ⋯, a from all TEs. Reduce the maximal abundance and yield $\boldsymbol{e}_i$, i=1, ⋯, a;

Step 9 Return all endmembers.

Step 3 and **step 6** are parallel segments of the algorithm, in which the majority computation are processed by all TEs concurrently. **Step 4**, **step 5** and **step 8** contain barrier and reduction operations, where **step 4** and **step 5** execute once in each iteration and **step 8** executes only once during the whole procedure in the algorithm. The computation of the reduction operations in these steps is much less than calculation in **step 3** and **step 6**.

3.2 Two-level synchronous parallel ICA EEA

The parallel capability can be scaled to the SMP nodes in SMP cluster, where the processes run by each SMP nodes provide a scalable inter-node parallelism. These executive inter-node processes are called process elements (PEs). Within a PE, there are TEs which provide intra-node parallelism. A commonly used method in SMP cluster is that coarse-grained parallelism is achieved among PEs, while fine-grained parallelism is obtained by TEs (Nakajima, 2007).

Two levels data partitioning strategy is presented as follows. Suppose there are b PEs in the SMP cluster and every PE contains a TEs. In the first level of data partitioning, $\boldsymbol{Y}$ is decomposed into row blocks (Plaza, 2002)$\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_b\}$ in coarse grain and attach to every PE. In the second level of data partitioning, for the p-th PE, p=1,…b, decompose $\boldsymbol{Y}_p$ into $\{\boldsymbol{Y}_{p,1}, \boldsymbol{Y}_{p,2}, \cdots, \boldsymbol{Y}_{p,a}\}$ by step 1 of SMM-PICA in fine grain and attach to each TEs in the p-th PE.

According to two levels data partitioning, two-level synchronous parallel ICA (SynPICA-2L) EEA is given in the following:

Step 1 Apply two levels data partitioning strategy to the hyperspectral remote sensing image;

Step 2 The master initialize $\boldsymbol{W}^{(0)}$ and broadcast it to all PEs, *i.e.* workers;

Step 3 For the p-th PE, $\forall p$=1, ⋯, b, execute **Step 3** and **Step 4** of SMM-PICA and obtain $\boldsymbol{W}(p)^{(k+1)}$;

Step 4 The master collect all $\boldsymbol{W}(p)^{(k+1)}$, p=1,⋯, b and reduce $\boldsymbol{W}^{(k+1)}$ by Eq. (9);

Step 5 The master decorrelate and renormalize $\boldsymbol{W}^{(k+1)}$, broadcast it to all PEs and go to **Step 3** until satisfy the convergence condition;

Step 6 All PEs execute **Step 6** and **Step 8** of SMM-PICA within their own data partitioning;

Step 7 The master execute the maximal reduction operation from the results collecting from all PEs and then obtain endmembers $\boldsymbol{e}_i$, i=1, ⋯, a;

Step 8 Return all endmembers.

In **Step 4** and **Step 5** of SynPICA-2L EEA, all PEs calculate $\boldsymbol{W}^{(k+1)}$ simultaneously in every synchronous iteration. And a maximal reduction operation in **Step 7** executes only once, while other operations are optimized for parallel execution by SMM-PICA and obtain two levels of parallelism, which make full use of the computation resource of SMP cluster.

The PCA transformation in **Step 1** of ICA EEA can be parallelism in the similar manner as FastICA.

4 TWO-LEVEL ASYNCHRONOUS PARALLEL ICA ALGORITHM

In asynchronous parallel ICA algorithm, all PEs execute SMM-PICA independently. However, data partitioning and endmember reduction should be considered.

4.1 Random subimage partitioning algorithm

Since the nature substance is usually continuous and homogeneous in local area, the data partitioning in a PE would only contain a small number of substance types in local area when using the data partitioning strategy in SynPICA-2L. Therefore, the samples attached to each PE cannot present the global features of the image. To avoid this embarrassment, random subimage partitioning (RSP) algorithm is proposed as follows.

Suppose there are b PEs.

Step 1 The master randomize a group of numbers ranged from 1 to p with uniformed distribution;

Step 2 Map these numbers into each pixel in the image, yeilds map$(s,t) \in \{1, 2, \cdots, b\}$ where (s,t) is the coordinate of the pixel ;

Step 3 The master partition the image by

$$\boldsymbol{Y}_i=\{\boldsymbol{y}(s,t)|\forall \boldsymbol{y}(s,t) \in \boldsymbol{Y}\ \textbf{and}\ map(s,t)=i\},\ i=1, \cdots, b \qquad (12)$$

Step 4 The master transfer the data partitioning to the corresponding PE.

Since we use the uniformed distribution in **Step 1**, RSP can obtain good load balance only when performed in a homogeneous parallel environment where the hardware performance of every SMP nodes is identical.

4.2 Accelerated cascading-endmember reduction algorithm

In the reduction step, the master collects candidate endmembers from all PEs and finally determines the endmembers. In this case, the growing number of the candidate endmembers, which is proportionate to the PEs, degrades the performance of parallel algorithm. According to the idea of the accelerated cascading algorithm (Chen, 2009), a fast EEA can be used in the reduction step. Accelerated cascading-endmember reduction (AC-ER) algorithm is given in the following:

Step 1 The master collects candidate endmembers, denoted by $\boldsymbol{E}_c$, from all PEs;

Step 2 Input $\boldsymbol{E}_c$, use a fast EEA to output the final endmember set $\boldsymbol{E}$;

Step 3 Return $\boldsymbol{E}_c$.

A fast EEA in **Step 2** should be implenmentd by a low-computation EEA including VCA, ATGP or IEA, except for the spatial information incorporating EEAs such as AMEE.

4.3 Two-level asynchronous parallel ICA algorithm

Incorporating with RSP and AC-ER, two-level asynchronous parallel ICA (AsyPICA-2L) algorithm is summarized as follows.

Step 1 The master executes RSP to generate data partitioning $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_b\}$ and transform to corresponding PEs;

Step 2 Every PEs execute SMM-PICA independently;

Step 3 The master waits for all PEs and executes AC-ER to obtain the final endmember set $\boldsymbol{E}$;

Step 4 Return $\boldsymbol{E}$.

AsyPICA-2L avoids the synchronous communication in each iteration. There is only one collective communication in the AC-ER. But there is additional serial computation in RSP and AC-ER.

5 TWO-LEVEL GROUPING PARALLEL ICA ALGORITHM

AsyPICA-2L should wait for the slowest PE before endmember

reduction, resulting in idle time of other PEs, which degrades the efficiency substantially. Even RSP maintains the load balance for AsyPICA-2L because too much data partitioning would decrease the number of samples attaching to PEs and degrade the the convergence of SMM-PICA in every PEs. Two-level grouping parallel ICA (GrpPICA-2L) algorithm avoids asynchronous data partitioning and takes adavancetage of the SynPICA-2L and AsyPICA-2L.

GrpPICA-2L devides the PEs into several groups $\{\boldsymbol{G}_1, \boldsymbol{G}_2, \cdots, \boldsymbol{G}_q\}$.

Let i_l be the first serial number for PE in group $\boldsymbol{G}_i$, and i_r be the last one. Then

$$\boldsymbol{G}_i=\{\mathrm{PE}_j, j=i_l, i_{l+1}\cdots, i_r\}, \forall i=1, \cdots, q \tag{13}$$

The intra-group PEs should be topologically neighbor in network. And make each $i_r–i_l$, $\forall i=1, \cdots, q$ identical with maintain load balance. GrpPICA-2L is summarized as follows.

Step 1 Devide PEs into q groups by Eq. (13) ;

Step 2 The master executes RSP to generate data partitioning for every groups ;

Step 3 For every group, decompose their data partitioning uniformly for every PEs ;

Step 4 For the i-th group $\boldsymbol{G}_i$, $\forall i=1,\cdots, q$, execute intra-group SynPICA-2L to generate candidate endmembers;

Step 5 The master waits for all groups and execute AC-ER to obtain the final endmember set $\boldsymbol{E}$;

Step 6 Return $\boldsymbol{E}$.

GrpPICA-2L balances the convergence and efficiency by the parameter q. Asynchronous parallel computation resides in q intergroups, while synchronous communication can be resisted on every intragroup. Therefore the efficiency is improved. Especially, it is equivalent to SynPICA-2L when $q=1$ and to AsyPICA-2L when $q=b$.

6 EXPERIMENT

We use 614×512 pixels real hyperspectral remote sensing image of the June 19, 1997 Cuprite Nevada (Fig. 1), collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. After removing the atmospheric absorption bands, 184 bands are used. The total size of the sub-image is 113 MB. The virtual dimension of this image is 20 (Kruse, *et al*., 2003). The number of endmembers to be extracted is set to 20 since ICA can not determined all endmembers in hyperspectral remote sensing image (Nascimento, 2005). VCA is used in AC-ER.

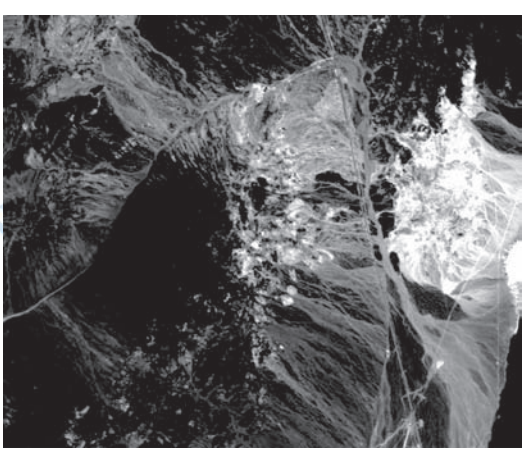

Fig. 1 AVIRIS image in Cuprite Nevada (648 nm)

The testing platform consists of 20 nodes connected by Gigabit Ethernet. Each node has 1.6 GHz Intel Core 2 Duo CPU, a 1 GB memory and an 80 GB hard drive. The system runs on Windows XP SP2 operating system and uses MPICH2 message passing library and Visual C++ 2008 OpenMP supported compiler for SMP cluster environment.

The executive time of the original serial algorithm is 1701485 ms (about 30 mins). And the executive time of SMM-PICA is 1307953 ms, which is 1.3 times faster than original algorithm.

SynPICA-2L, AsyPICA-2L and GrpPICA-2L are applied in the image respectively, where the parameter q of GrpPICA-2L is set to 2. The number of the nodes of the cluster ranges from 2 to 20. The comparison of executive time among these algorithms and a PE-based single level synchronous parallel algorithm in coarse grain is shown in Fig. 2. Obviously, SynPICA-2L obtains better performance than PE-based single level synchronous parallel algorithm. As the number of nodes increasing, the executive time decreases. When it reaches 20, the executive time of SynPICA-2L and GrpPICA-2L are less than 78000 ms.

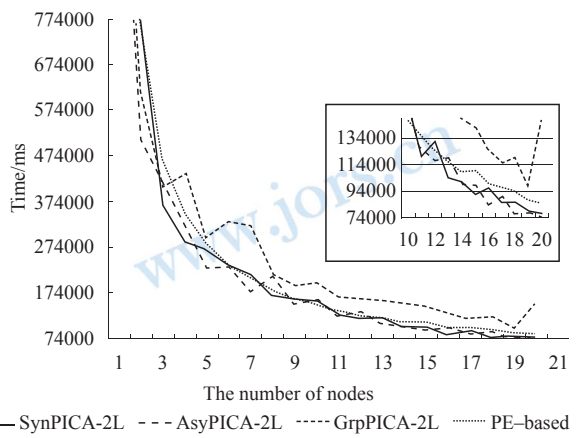

Fig. 2 Executive time of parallel ICA algorithms

The endmember extraction error of these parallel algorithms is compared to the original algorithm by the root mean square error (RMSE) of spectral angle distance (SAD) (Chang, 2003). The results demonstrate that the endmember extracted by SynPICA-2L are consistent with the original algorithm. The SAD RMSE of GrpPICA-2L is about 0.04, while that of AsyPICA-2L greatly increases as the number of nodes increasing (See Fig. 3) and it reaches 0.13 when the number of nodes is 20.

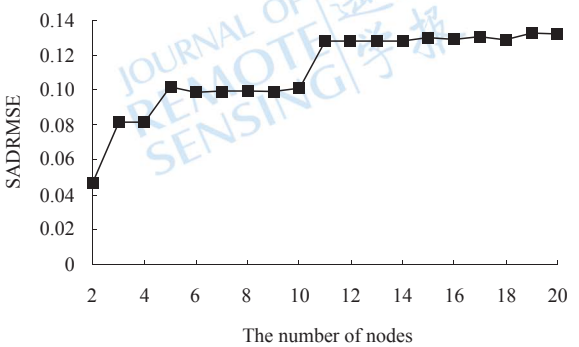

Fig. 3 The SAD RMSE curve of AsyPICA-2L

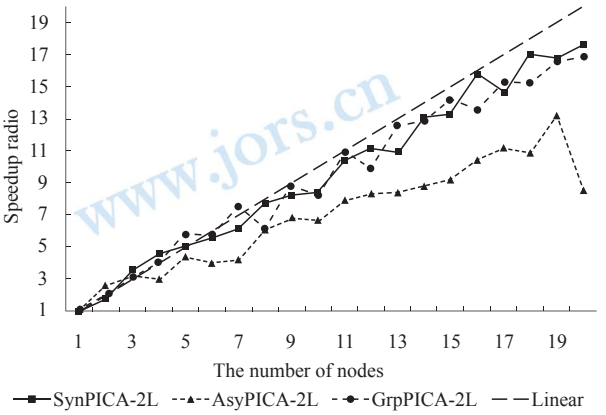

Fig. 4 The speedup radio curves of parallel ICA EEAs

Fig. 4 compares the speedup radio curves of our proposed algorithms. The result indicates that the speedup radio of AsyPICA-2L is higher than SynPICA-2L and GrpPICA-2L. When the number of nodes is larger than 8, the performance of AsyPICA-2L degrades and presents to be unstable. SynPICA-2L and GrpPICA-2L approximate to linear speedup radio. The performance of GrpPICA-2L is better than SynPICA-2L when the number of nodes is odd. In other cases, similar performance are observed for both GrpPICA-2L and SynPICA-2L.

7 CONCLUSION

This paper presented three parallel ICA EEAs, namely SynPICA-2L, AsyPICA-2L and GrpPICA-2L in SMP cluster environment. The conclusion is summarized as follows.

(1) SynPICA-2L and GrpPICA-2L maintain the accuracy of endmember extraction, while AsyPICA-2L degrades the accuracy as the number of nodes increasing; (2) SynPICA-2L approximates to the performance of linear speedup radio and has highly scalability; (3) The performance of AsyPICA-2L is higher than SynPICA-2L when there is a small number of nodes in the cluster, while with low scalability; (4) GrpPICA-2L has the best performance among these two-level parallel ICA EEAs.

As future work, we will improve our algorithm by (1) optimizing the parallel code in SMM-PICA since the speedup radio is only 1.3 in dual-core CPU; (2) optimizing the load balance of GrpPICA-2L; (3) further evaluating on the parameter q to obtain the best value for GrpPICA-2L.

Acknowledgements: The author acknowledges Jet Propulsion Laboratory (JPL) for providing AVIRIS data.

REFERENCES

Bayliss J, Gualtieri J A and Cromp R F. 1997. Analysing hyperspectral data with independent component analysis. 26th AIPR Workshop: Exploiting New Image Sources and Sensors: SPIE Press, **3240**: 133–143 DOI: 10.1117/12.300050

Boardman J W, Kruse F A and Green R O. 1995. Mapping target signatures via partial unmixing of AVIRIS data. Fifth JPL Airborne Earth Science Workshop. Pasadena CA: JPL Press: 23–26

Chai L, Gao Q and Panda D K. 2007. Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System. Proc. Seventh IEEE International Symposium on Cluster Computing and the Grid 2007. IEEE Computer Society Press: 471–478

Chang C I. 2003. Hyperspectral Imaging: Techniques for Spectral Detection and Classification. New York: Kluwer Academic/Plenum Publishers: 73–88

Chang C I. 2007. Hyperspectral Data Exploitation: Theory and Applications. New Jersey: Wiley-Interscience Press

Chen G L, Wu J M, Zhang F and Zhang L B. 2002. Parallel Computer Architectures. Beijing: Higher Education Press

Chen G L. 2009. Design and Analysis of Parallel Algorithms. 3rd ed. Beijing: Higher Education Press

Dong C, Zhao H J and Wang W. 2010. Parallel nonnegative matrix factorization algorithm on the distributed memory platform. *International Journal of Parallel Programming*, **38**(2): 117–137 DOI: 10.1007/s10766-009-0116-7

Du H T, Qi H R and Peterson G D. 2004. Parallel ICA and its hardware implementation in hyperspectral image analysis. Independent Component Analyses, Wavelets, Unsupervised Smart Sensors, and Neural Networks II. SPIE Press, **5439**: 74–83 DOI: 10.1117/12.543962

Du H T, Qi H R and Wang X L. 2006. A parallel independent component analysis algorithm. Proceedings of the 12th International Conference on Parallel and Distributed Systems. Minnesota: IEEE Computer Society Press, **1**: 151–160 DOI: 10.1109/ICPADS.2006.17

Hyvärinen A. 1999. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, **10**(3), 626–634 DOI: 10.1109/72.761722

Kruse F A, Boardman J W and Huntington J F. 2003. Comparison of airborne hyperspectral data and EO-1 Hyperion for mineral mapping. *IEEE Transactions on Geoscience and Remote Sensing*, **41**(6): 1388–1400 DOI: 10.1109/TGRS.2003.812908

Nakajima K. 2007.Parallel Multistage Preconditioners Based on a Hierarchical Graph Decomposition for SMP Cluster Architectures with a Hybrid Parallel Programming Model. High Performance Computing and Communications Third International Conference. NewYork: Springer Berlin Heidelberg: 384-395 DOI:10.1007/978-3-540-75444-2_39

Nascimento J M P and Dias J M B. 2005. Does independent component analysis play a role in unmixing hyperspectral data? *IEEE Transactions on Geoscience and Remote Sensing*, **43**(1): 175–187

Neville R A, Staenz K, Szeredi T, Lefebvre J and Hauff P. 1999. Automatic endmember extraction from hyperspectral data for mineral exploration. In Proc. 21st Can. Symp. Remote Sensing, Ottawa: 21–24

Plaza A, Benediktsson J A, Boardman J W, Brazile J, Bruzzone L, Camps-Valls G, Chanussot J, Fauvel M, Gamba P, Gualtieri A, Marconcini M, Tilton J C and Triani G. 2009. Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, **113**(1): S110–S122 DOI: 10.1016/j.rse.2007.07.028

Plaza A, Martinez P, Perez R and Plaza J. 2002. Spatial/Spectral Endmember Extraction by Multidimensional Morphological Operations. *IEEE Transactions on Geoscience and Remote Sensing*, **40**(9): 2024–2041

Plaza A, Valencia D, Plaza J and Chang C I. 2006. Parallel implementation of endmember extraction algorithms from hyperspectral data. *IEEE Geoscience and Remote Sensing letters*, **3**(3): 334-338 DOI: 10.1109/LGRS.2006.871749

Plaza A, Valencia D, Plaza J and Martinez P. 2006. Commodity cluster-based parallel processing of hyperspectral imagery. *Journal of Parallel and Distributed Computing*, **66**(3): 345–358 DOI: 10.1016/j.jpdc.2005.10.001

Quinn M J. 2003. Parallel Programming in C with MPI and OpenMP. New York: McGraw-Hill Science/Engineering/Math

Robila S A and Maciak L G. 2006. A parallel unmixing algorithm for hyperspectral images. Intelligent Robots and Computer Vision XXIV. *SPIE Press*, **6384**: 63840F.1-63840F.11 DOI: 10.1117/12.685655

Rogge D M, Rivard B, Zhang J K and Feng J L. 2006. Iterative spectral unmixing for optimizing per-pixel endmember sets. *IEEE Transactions on Geoscience and Remote Sensing*, **44**(12): 3725–3736 DOI: 10.1109/TGRS.2006.881123

Wang J and Chang C I. 2006. Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, **44**(9): 196–2895 DOI: 10.1109/TGRS.2006.874135

Winter M E. 1999. N-FINDR: an Algorithm for Fast Autonomous Spectral Endmember Determination in Hyperspectral Data. SPIE Imaging Spectrometry V. San Diego CA: SPIE Press: 266-275

Zortea M and Plaza A. 2009. Spatial preprocessing for endmember extraction. *IEEE Transactions on Geoscience and Remote Sensing*, **47**(8): 2679–2693 DOI: 10.1109/TGRS.2009.2014945

二级并行独立成分分析端元提取算法

罗文斐[1]，高连如[2]

1. 华南师范大学 地理科学学院，广东 广州 510631;
2. 中国科学院对地观测与数字地球科学中心，北京 100190

摘 要：在多对称处理器集群体系结构下进行独立成分分析并行算法研究，在共享内存模型一级并行算法基础上，通过同步、异步迭代两种方式并行计算固定点函数，分别提出具有两级并行特性的二级同步、二级异步并行端元提取算法，并结合两者的优势，进一步提出二级分组并行算法。实验评价表明，二级同步、分组并行算法在保持原算法精度的同时，大大提高了原算法的效率，体现出良好的并行计算性能，而二级异步并行算法可在节点数较少的情况下适用。

关键词：高光谱遥感，独立成分分析，端元提取，多对称处理器，集群，并行计算

中图分类号：TP751.1 **文献标志码**：A

引用格式：罗文斐，高连如. 2011. 二级并行独立成分分析端元提取算法. 遥感学报，**15**(6): 1202–1214
Luo W F and Gao L R. 2011. Two-level parallel independent component analysis endmember extraction algorithms. *Journal of Remote Sensing*, **15**(6): 1202–1214

1 引 言

混合像元分解(也称“光谱解混”)是高光谱遥感图像分析的关键(Chang，2007)，其中端元提取是光谱解混过程中最复杂、最重要的一个步骤。近十多年来发展了大量的算法，主要归纳为3大类：利用几何特征的端元提取算法、采用统计特征的端元提取算法以及结合空间维信息的端元提取算法。其中，利用几何特征的端元提取算法是早期发展最快的算法，出现的经典算法如纯像元指数(Pixel Purity Index，PPI)(Boardman 等，1995)、N-FINDR算法(Winter，1999)、非监督目标生成过程(Unsupervised Target Generation Process，ATGP)(Chang，2003)以及顶点成分分析算法(Vertex Component Analysis，VCA)(Nascimento和Dias，2005)等，这类算法通常具有计算简便的特点，因此得到了广泛的应用。基于统计特征的算法如早期的迭代误差分析(Iterative Error Analysis，IEA)(Neville，1999)通过统计最小二乘误差来提取端元。盲信号分离技术与投影寻踪技术的引入，使得统计特征的端元提取算法得到了全面的发展，独立成分分析(Independent component analysis，ICA)是一种典型的光谱解混算法(Bayliss，1997；Chang，2003；Nascimento和Dias，2005)，同时也实现了端元提取(Wang和Chang，2006)。此外，结合空间维信息的端元提取算法在近几年迅速发展，提出了自动形态学端元提取(Automated Morphological Endmember Extraction，AMEE)方法(Plaza，2002)、空间/光谱集成的端元提取方法(Spatial-Spectral Endmember Extraction，SSEE)(Rogge 等，2006)、空间预处理(Spatial Preprocessing，SPP)方法(Zortea和Plaza，2009)。

然而，针对数据量巨大的高光谱遥感图像，不少算法需要大量的计算时间，这种同时兼有数据密集型和计算密集型的工作，成为了高光谱遥感应用的严重瓶颈，尤其是那些具有时间限制的关键任务。高性能并行计算在近些年得到了飞速发展，为进一步提高高光谱遥感图像计算的效率提供了很好的契机，并行高光谱遥感图像分析算法成为了近些年研究的热点(Plaza 等，2009)。近几年提出了不少并行算法，

收稿日期：2010-09-28；**修订日期**：2011-01-10

基金项目：国家自然科学基金项目(编号: 40901232/D010702 , 40901225/D010702)；国家高技术研究发展计划(863计划) (编号: 2008AA12Z113)；国家重点基础研究发展计划(973计划) (编号: 2009CB723902)

第一作者简介：罗文斐(1979—)，男，副教授，2008年博士毕业于中国科学院遥感应用研究所，目前主要从事高性能遥感图像处理、高光谱遥感方面的研究工作。E-mail: luowenfei@irsa.ac.cn。

如并行非负矩阵分解算法(Robila和Maciak，2006；Dong等，2010)、并行端元提取算法(Plaza 等，2006a)、并行AMEE算法(Plaza 等，2006b)等。Du等人(2006)提出了一种在单指令多数据(Single Process Multiple Data，SPMD)系统下的并行ICA算法，该算法在不同的图像划分中提取各自的独立成分，通过局部与全局去相关的方法得到最终的结果，并设计了相关的硬件实现(Du 等，2004)。

目前个人微机和网络传输的性能已得到大幅度提高，工作站集群(Cluster of Workstation，COW)已在并行计算机系统中得到了广泛的应用(陈国良 等，2002)。同时，随着CPU进入多核时代，对称多处理器(Symmetrical Multi-Processing，SMP)集群成为了目前最为广泛使用的并行计算机系统之一(Chai 等，2007)。

SMP集群的并行体系结构，同时兼有了共享内存以及分布式内存两种并行计算机模型的优势。利用这一优势，本文考虑在SMP集群环境中进行并行算法研究。在ICA计算过程中，通过并行计算固定点迭代函数，首先提出在共享内存模型下的一级同步ICA并行算法，进一步在SMP集群中提出二级同步ICA并行算法。在图像划分与端元归并问题上，分别提出空间域随机子图像划分与加速级联端元归并算法，在此基础上实现二级异步ICA并行算法。通过结合同步与异步算法的优势，提出二级分组ICA并行算法，在保证算法效率的同时可获得较好的可扩展性。最后，通过真实图像检验3种并行算法的性能。

2 基础算法

令一幅高光谱遥感图像(假设图像均值已平移到原点)为：

$$\boldsymbol{X}=[\boldsymbol{x}(1),\boldsymbol{x}(2),\cdots,\boldsymbol{x}(n)] \tag{1}$$

式中，$\boldsymbol{x}(i)$, i=1, 2, …, n 为像元光谱向量，n为像元总数。每一个像元光谱 $\boldsymbol{x}(i)$, i=1, 2, …, n均由端元线性混合而成：

$$\boldsymbol{x}(i)=\sum_{j=1}^{m}\boldsymbol{e}_j\boldsymbol{a}_j(i) \tag{2}$$

式中，$\boldsymbol{e}_j$, j=1, 2, …, m 为图像端元向量，m 为端元总数，$a_j(i)$ 表示第 i 个像元中，包含端元 $\boldsymbol{e}_j$ 的丰度含量，组成丰度向量：

$$\boldsymbol{a}(i)=[\boldsymbol{a}_1(i),\cdots,\boldsymbol{a}_m(i)]^{\mathrm{T}} \tag{3}$$

ICA算法首先通过主成分变换(Principal Components Analysis, PCA)去除图像观测信号之间的相关性来简化后续的提取过程。再以非高斯特征为准则，寻找一个最佳的分离矩阵$\boldsymbol{W}$，使得每个像元中各端元的丰度估计：

$$\hat{\boldsymbol{a}}(i)=\boldsymbol{W}\boldsymbol{Z}\boldsymbol{x}_i \tag{4}$$

尽可能地接近丰度含量$\boldsymbol{a}(i)$，式中$\hat{\boldsymbol{a}}(i)=[\hat{\boldsymbol{a}}_1(i),\cdots,\hat{\boldsymbol{a}}_m(i)]^{\mathrm{T}}$为估计的丰度向量，$\boldsymbol{Z}$为主成分向量所组成的矩阵。为描述方便，记主成分图像为：

$$\boldsymbol{Y}=\boldsymbol{Z}\boldsymbol{X} \tag{5}$$

快速ICA(FastICA)算法(Hyvärinen，1999)具有较快的收敛速度，非高斯特征可通过二阶以上的统计量、负熵来描述。分离矩阵$\boldsymbol{W}$由式(6)得到：

$$\boldsymbol{W}^{(k+1)}:=E\left\{\boldsymbol{Y}\mathrm{g}\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}\right)\right\}-E\left\{\mathrm{g}'\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}\right)\right\}\boldsymbol{W}^{(k)} \tag{6}$$

式中，$\boldsymbol{W}^{(k)}$为第k次迭代的分离矩阵，g和g′函数的定义详见Hyvärinen(1999)的工作。为了避免不同的成分收敛于同一个方向，新的$\boldsymbol{W}^{(k+1)}$需要经过归一化以及对称正交化，使其成为正交方向上的单位向量，进入下一步的迭代。

在ICA计算结果的各分量中，可根据丰度估计的绝对值大小提取对应分量的端元光谱向量(Wang和Chang，2006)。总结ICA端元提取算法(ICA-EEA)步骤如下：

步骤1 利用PCA变换对图像进行去相关处理，得到新图像$\boldsymbol{Y}$；

步骤2 初始化分离矩阵$\boldsymbol{W}^{(0)}$;

步骤3 利用式(6)对进行迭代；

步骤4 为避免收敛于同一个方向，对得到的$\boldsymbol{W}^{(k+1)}$进行归一化以及对称正交化；

步骤5 迭代完成后，计算各成分的丰度估计$\hat{\boldsymbol{a}}_i=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{Y}$, i=1, …, m, $\boldsymbol{w}_i$为$\boldsymbol{W}$的第i个独立分量；

步骤6 根据每一个成分的丰度估计，选取端元：

$$\boldsymbol{e}_i:=\boldsymbol{x}_{k_i}\ \text{且}\ k_i=\underset{1\le j\le n}{avg\max}(|\boldsymbol{a}_i(j)|) \tag{7}$$

步骤7 返回所有的端元，算法结束。

3 二级同步ICA并行算法

上述算法，主要的计算量集中在**步骤3**，即每一步迭代计算新的$\boldsymbol{W}^{(k+1)}$，因此，提高$\boldsymbol{W}^{(k+1)}$的计算速度是并行算法考虑的关键。

3.1 共享内存模型下的并行算法

SMP处理器之间共享内存，在这一模型中，每个并行处理单元采用线程可以提高实际执行的效率(Quinn，2003)。因此，算法中把每个处理单元称为线程单元(Thread Element，TE)。

假设具有a个TE，把图像$\boldsymbol{Y}$划分为$\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_a\}$，划分原则只需保持连续地址访问即可提高效率，第$j$个TE负责$\boldsymbol{Y}_j$划分，$j$=1, 2, …, a。在ICA-EEA步骤(3)中，第j个TE对式(6)的计算如下：

$$\boldsymbol{W}(j)^{(k+1)} = E\left\{\boldsymbol{Y}_j \mathrm{g}\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}_j\right)\right\} - E\left\{\mathrm{g}'\left(\boldsymbol{W}^{(k)\mathrm{T}}\boldsymbol{Y}_j\right)\right\}\boldsymbol{W}^{(k)} \quad (8)$$

显然

$$\boldsymbol{W}^{(k+1)} := \frac{1}{a\sum_{j=1}^{a} s_j}\sum_{j=1}^{a} s_j \boldsymbol{W}(j)^{(k+1)} \quad (9)$$

式中，s_j为$\boldsymbol{Y}_j$的样本总数。

由此可得共享内存模型并行ICA算法(Share Memory Model Parallel ICA，SMM-PICA)：

步骤1 对图像$\boldsymbol{Y}$划分为$\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_a\}$；

步骤2 初始化分离矩阵$\boldsymbol{W}^{(0)}$；

步骤3 对于每一个TE，计算式(8)；

步骤4 等待各TE计算完成后，采用式(9)对$\boldsymbol{W}^{(k+1)}$进行规约；

步骤5 对得到$\boldsymbol{W}^{(k+1)}$进行归一化以及对称正交化，并进入下一步迭代；

步骤6 迭代完成后，第j个TE，j=1,…, a进行丰度估计，得：

$$\hat{\boldsymbol{a}}_{i,j} = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{Y}_j,\ i=1, \ldots, m \quad (10)$$

步骤7 选取候选端元：

$$\boldsymbol{e}_{i,j} := \boldsymbol{x}_{k_i,j} \text{ 且 } k_i = \underset{1\le l\le n}{avg\max}\left(\left|\boldsymbol{a}_{i,j}(l)\right|\right) \quad (11)$$

步骤8 等待各TE搜索完候选端元后，$\boldsymbol{e}_{i,j}$ i=1, …, m; j=1, …, a，以对应的丰度最大绝对值进行规约，得到最终$\boldsymbol{e}_i$, i=1, …, a；

步骤9 返回所有的端元，算法结束。

其中，**步骤3**、6为算法的并行部分，把计算量最为集中的部分分给多个TE并行执行，**步骤4**、5和8为计算量较小的同步规约操作，**步骤4**、5每次迭代执行一次，**步骤8**在算法中只执行一遍。

3.2 二级同步并行算法

在SMP集群中，其并行能力可扩展到各计算节点之间，每个计算节点产生进程进行节点间的并行计算，其执行实体称为进程单元(Process Element，PE)。在每个PE内部，采用多个TE进行并行计算。因此，粗粒度并行化可通过PE获得，细粒度并行化则通过TE获得(Nakajima, 2007)。

在图像划分中，采用二级划分策略。假设集群中共有b个同构的计算节点，即b个PE，每个计算节点均具有a个TE。首先为每个PE分配图像粗划分$\boldsymbol{Y}=\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_b\}$，划分原则可采用按空间域的行方向进行划分(Plaza 等，2002)，对于第p个PE，p=1, …, b，进一步进行细划分$\boldsymbol{Y}_p=\{\boldsymbol{Y}_{p,1}, \boldsymbol{Y}_{p,2}, \cdots, \boldsymbol{Y}_{p,a}\}$，把划分结果分配给各TE。在此基础上可得以下的二级同步ICA并行算法(2-level Synchronous Parallel ICA, SynPICA-2L)，采用主－从模式设计：

步骤1 主PE对图像进行粗划分，并把划分子图像传输到各从PE，从PE根据获得的子图像进行细划分；

步骤2 主PE初始化分离矩阵$\boldsymbol{W}^{(0)}$，并把$\boldsymbol{W}^{(0)}$广播到各从PE；

步骤3 对于第p个从PE，$\forall p$=1, …, b，执行SMM-PICA的**步骤3**、4，获得$\boldsymbol{W}(p)^{(k+1)}$，并把结果传输给主PE；

步骤4 主PE等待$\boldsymbol{W}(p)^{(k+1)}$, P=1, …, b 收集完毕后，通过式(9)规约$\boldsymbol{W}^{(k+1)}$；

步骤5 主PE对得到$\boldsymbol{W}^{(k+1)}$进行归一化以及对称正交化，并广播到各从PE进行下一步迭代；

步骤6 迭代完成后，各从PE获得最终的分离矩阵，并执行SMM-PICA算法的**步骤6—8**；

步骤7 主PE根据各从PE丰度最大绝对值进行最大值规约，并收集所对应的端元，得到最终的端元$\boldsymbol{e}_i$, i=1, …, a；

步骤8 返回所有的端元，算法结束。

SynPICA-2L算法在每次迭代中同步计算全局分离矩阵$\boldsymbol{W}^{(k+1)}$，见**步骤4**、5，在**步骤7**进行了一次最大值规约，但其余部分通过SMM-PICA算法进行二级并行计算，从而充分利用了计算资源。

另外，原ICA-EE**步骤1**的去相关预处理步骤，可采用类似的方法实现，篇幅关系，本文不再累述。

4 二级异步ICA并行算法

异步并行算法采用各PE独立执行SMM-PICA进行并行化，需要重新考虑数据划分与端元规约问题。

4.1 随机子图像划分算法

在高光谱遥感图像中，由于自然地物的分布通常具有一定的连续性和局部区域性，如果采用SynPICA-2L中的划分方法，容易导致PE被分配的子图像仅包含局部地区少数地物的样本，其样本空间不具有全局代表性。因此采用以下的随机子图像划分(Random Subimage Partitioning，RSP)算法，假设共有b个PE：

步骤1 主PE产生一组与像元数目相同的随机数，该随机数采用均匀分布随机数产生器生成，变化范围从1到p的整数；

步骤2 把这组随机数映射到每一个像元，得$\mathrm{map}(s,t)\in\{1,2,\cdots,b\}$，其中，$(s,t)$为图像像元坐标；

步骤3 主PE产生划分

$$\boldsymbol{Y}_i=\{\boldsymbol{y}(s,t)\,|\,\forall\boldsymbol{y}(s,t)\in\boldsymbol{Y}\text{且}map(s,t)=i\} \tag{12}$$

式中，$i=1,\cdots,b$；

步骤4 主PE把划分传输到各从PE。

RSP算法只有在同构并行环境，即各PE运算能力基本相同的集群中，才可获得较好的负载均衡，这主要是由于随机数采用均匀分布随机数产生器所致。

4.2 加速级联–端元规约算法

在进行规约时，主PE从各PE收集候选端元，并确定最终的端元集合。在规约操作中，候选端元的数目与PE数目成正比，意味着规约的性能会随之下降。因此，根据加速级联算法(陈国良，2009)的思想，在规约操作中采用可替代的快速端元提取算法能够有效地解决端元确定问题。加速级联-端元规约算法(Accelerated Cascading-Endmember Reduction，AC-ER)描述如下：

步骤1 主PE接收各从PE提取端元后，形成候选端元集合$\boldsymbol{E}_c$；

步骤2 以$\boldsymbol{E}_c$作为输入信号，采用一种快速端元提取算法，生成最终的端元集合$\boldsymbol{E}$；

步骤3 返回$\boldsymbol{E}$，算法结束。

其中，**步骤2**中的快速端元提取算法，可采用计算量较少的端元提取算法来实现，例如VCA、ATGP或IEA算法，但不能是结合空间维信息的端元提取算法，如AMEE算法。

4.3 二级异步并行算法

结合RSP与AC-ER算法，二级异步ICA并行(2-level Asynchronous Parallel ICA, AsyPICA-2L)算法描述如下：

步骤1 主PE执行RSP算法，产生划分$\{\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_b\}$，并把划分传输给对应的从PE；

步骤2 对各个从PE独立执行SMM-PICA算法；

步骤3 等待各PE执行完毕后，主PE执行AC-ER算法，获得最终端元集合$\boldsymbol{E}$；

步骤4 返回$\boldsymbol{E}$，算法结束。

与SynPICA-2L算法相比，AsyPICA-2L算法避免了迭代过程中产生的同步开销，仅AC-ER算法需要一次同步通信，但增加了RSP和AC-ER算法中的额外串行计算部分。

5 二级分组ICA并行算法

AsyPICA-2L中各PE对所得的图像划分进行独立计算，但最后必须等待最后一个PE完成后才可进行端元规约。尽管采用了RSP算法维持负载均衡，但过多的图像划分仍然会大大减少PE被分配的样本数量，从而影响各PE的收敛速度，严重降低了异步算法的性能。二级分组ICA并行算法(2-level Grouping Parallel ICA，GrpPICA-2L)则结合了同步与异步算法的特点，利用分组同步来避免过多的异步计算。

该算法把PE划分为若干个分组$\{\boldsymbol{G}_1,\boldsymbol{G}_2,\cdots,\boldsymbol{G}_q\}$。考虑PE编号的任意性，可假设组内PE连续编号，并令i_l为$\boldsymbol{G}_i$组内第一个PE的编号，i_r为最后一个编号：

$$\boldsymbol{G}_i=\left\{\mathrm{PE}_j,j=i_l,i_{l+1}\cdots,i_r\right\},\forall i=1,\cdots,q \tag{13}$$

通常让组内PE在网络拓扑结构中具有相邻性，并应使$i_r-i_l,\forall i=1,\cdots,q$相等或尽可能接近，以保持负载均衡。GrpPICA-2L算法描述如下：

步骤1 输入参数q，并按式(13)对PE进行分组；

步骤2 主PE使用RSP算法产生每个分组的划分；

步骤3 各分组为每个PE平均分配组内划分的样本；

步骤4 对于第i个分组$\boldsymbol{G}_i,\forall i=1,\cdots,q$，对组内各PE间执行SynPICA-2L算法，产生各分组的端元集合；

步骤5 主PE同步等待各分组执行完毕后，执行AC-ER算法，获得最终端元集合$\boldsymbol{E}$；

步骤6 返回$\boldsymbol{E}$，算法结束。

GrpPICA-2L通过q参数来调节算法的稳定性和执行效率，在q个分组间进行异步计算，同步通信仅在组内发生，因此提高了同步通信的效率。特别的，当$q=1$时，简化为SynPICA-2L算法；当$q=b$时，简化为AsyPICA-2L算法。

6 实验分析

真实图像采用了Cuprite Nevada地区的AVIRIS (Airborne Visible/Infrared Imaging Spectrometer)高光谱数据(614×512像元，1997年6月19日飞行)，如图1所示。实验中选择了429-2470 nm的184个波段(去除了大气吸收波段)，图像大小为113 MB，图像中端元所在子空间虚拟维度约为20(Kruse 等，2003)，由于ICA算法无法提取所有的端元(Nascimento和Dias，2005)，本实验端元提取数目取20。AC-ER采用了VCA算法。

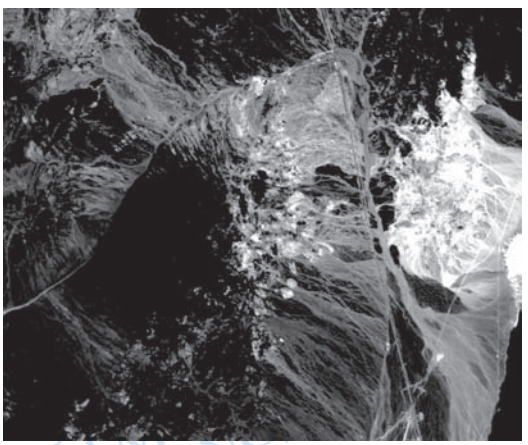

图1 Cuprite Nevada地区AVIRIS图像(648 nm波段)

测试平台包括了20个计算节点，每个计算节点是配置为1.6 GHz Intel Core 2 Duo CPU、1 G内存以及80 G硬盘的微机，计算节点由千兆以太网互连。采用MPICH2以及Visual C++2008 OpenMP进行并行程序设计，数学库使用BLAS/LAPACK。

在单个计算节点上，多次运行原串行版本算法的平均时间为1701485 ms(约30 min)，SMM – PICA算法的时间为1307953 ms，加速比为1.3。

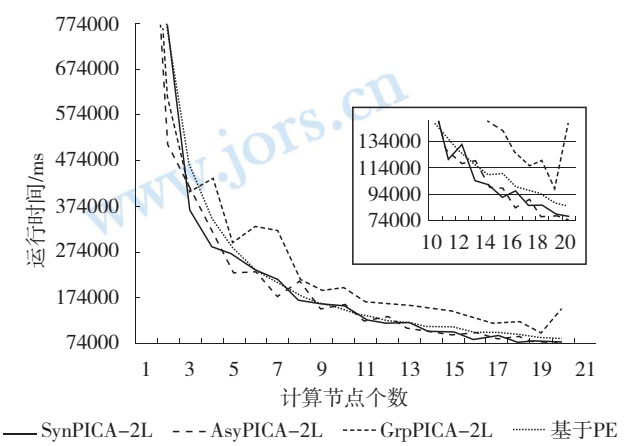

图2 ICA并行算法运行时间

分别采用3种二级并行算法对图像进行端元提取，其中GrpPICA-2L的q取2。计算节点个数从2递增至20，各算法的运行时间如图2所示。此外，图中还比较了仅采用粗粒度并行化的单级同步并行算法，从图中可见，二级同步并行算法能获得更高的效率。同时，随着计算节点的增加，各算法的运行时间逐步减少，当到达20个计算节点时，SynPICA-2L和GrpPICA-2L的执行时间均小于78000 ms。

在精度上，把3者所提取的端元结果与原算法的结果进行比较，精度评价采用光谱角度距离(Spectral Angle Distance，SAD)(Chang，2003)的总体均方误差进行度量，SynPICA-2L与原算法提取结果一致，GrpPICA-2L总体误差均为0.04。AsyPICA-2L误差如图3所示，随着计算节点的增多误差逐渐增大，当节点数为20时，误差达到0.13。

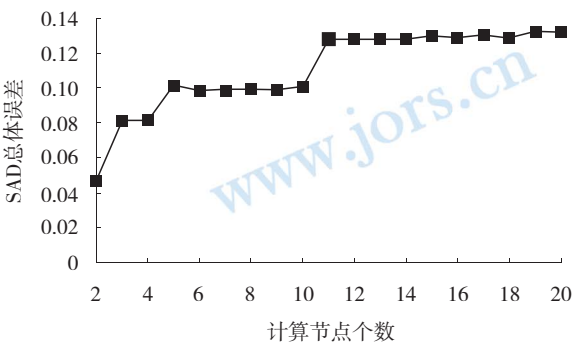

图3 AsyPICA-2L的SAD总体误差比较

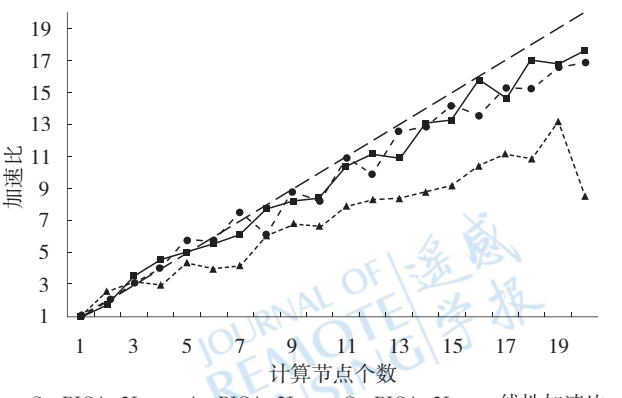

图4 ICA并行算法的加速比

图4比较了3种算法的加速比。可以发现，随着节点个数的增加，AsyPICA-2L的加速比不如SynPICA-2L和GrpPICA-2L，当节点数超过8之后，AsyPICA-2L加速比上升缓慢，并略显不稳定。比较SynPICA-

2L和GrpPICA-2L，两者较为接近线性加速比。当计算节点为奇数时，GrpPICA-2L略优于SynPICA-2L，其余情况两者性能相当。

7 结 论

本文在SMP集群环境中分别提出了SynPICA-2L、AsyPICA-2L以及GrpPICA-2L 3种ICA并行算法。可得以下结论：

(1)精度方面，SynPICA-2L与GrpPICA-2L能够很好地保持原算法的精度，当计算节点增多时，AsyPICA-2L精度逐渐下降；(2)SynPICA-2L接近于线性加速比，具有较好的可扩展性；(3)AsyPICA-2L在计算节点少的情况下其性能优于SynPICA-2L，但可扩展性较差；(4)GrpPICA-2L兼有了AsyPICA-2L较高的性能和SynPICA-2L较好的可扩展性，在3者中体现出最好的并行性能。

在今后工作中，以下几个方面仍需改进：(1)在单节点的SMM-PICA算法中，计算资源利用率仍然不高，双核CPU的加速比只为1.3，需要在代码上作进一步优化；(2)虽然GrpPICA-2L具有较好的并行性能，分组之间的负载均衡仍有待改进；(3)GrpPICA-2L中的q在实验中仅检验了 取值为2的情况，但q通常取值为多少才可使算法达到最佳，需要结合更多的数据进行检验。

志 谢 感谢Jet Propulsion Laboratory(JPL)提供的AVIRIS实验数据。

REFERENCES

Bayliss J, Gualtieri J A and Cromp R F. 1997. Analysing hyperspectral data with independent component analysis. 26th AIPR Workshop: Exploiting New Image Sources and Sensors: SPIE Press, **3240**: 133–143 DOI: 10.1117/12.300050

Boardman J W, Kruse F A and Green R O. 1995. Mapping target signatures via partial unmixing of AVIRIS data. Fifth JPL Airborne Earth Science Workshop. Pasadena CA: JPL Press: 23–26

Chai L, Gao Q and Panda D K. 2007. Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System. Proc. Seventh IEEE International Symposium on Cluster Computing and the Grid 2007. IEEE Computer Society Press: 471–478

Chang C I. 2003. Hyperspectral Imaging: Techniques for Spectral Detection and Classification. New York: Kluwer Academic/Plenum Publishers: 73–88

Chang C I. 2007. Hyperspectral Data Exploitation: Theory and Applications. New Jersey: Wiley-Interscience Press

Chen G L, Wu J M, Zhang F and Zhang L B. 2002. Parallel Computer Architectures. Beijing: Higher Education Press

Chen G L. 2009. Design and Analysis of Parallel Algorithms. 3rd ed. Beijing: Higher Education Press

Dong C, Zhao H J and Wang W. 2010. Parallel nonnegative matrix factorization algorithm on the distributed memory platform. *International Journal of Parallel Programming*, **38**(2): 117–137 DOI: 10.1007/s10766-009-0116-7

Du H T, Qi H R and Peterson G D. 2004. Parallel ICA and its hardware implementation in hyperspectral image analysis. Independent Component Analyses, Wavelets, Unsupervised Smart Sensors, and Neural Networks II. SPIE Press, **5439**: 74–83 DOI: 10.1117/12.543962

Du H T, Qi H R and Wang X L. 2006. A parallel independent component analysis algorithm. Proceedings of the 12th International Conference on Parallel and Distributed Systems. Minnesota: IEEE Computer Society Press, **1**: 151–160 DOI: 10.1109/ICPADS.2006.17

Hyvärinen A. 1999. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, **10**(3), 626–634 DOI: 10.1109/72.761722

Kruse F A, Boardman J W and Huntington J F. 2003. Comparison of airborne hyperspectral data and EO-1 Hyperion for mineral mapping. *IEEE Transactions on Geoscience and Remote Sensing*, **41**(6): 1388–1400 DOI: 10.1109/TGRS.2003.812908

Nakajima K. 2007.Parallel Multistage Preconditioners Based on a Hierarchical Graph Decomposition for SMP Cluster Architectures with a Hybrid Parallel Programming Model. High Performance Computing and Communications Third International Conference. NewYork: Springer Berlin Heidelberg: 384-395 DOI:10.1007/978-3-540-75444-2_39

Nascimento J M P and Dias J M B. 2005. Does independent component analysis play a role in unmixing hyperspectral data? *IEEE Transactions on Geoscience and Remote Sensing*, **43**(1): 175–187

Neville R A, Staenz K, Szeredi T, Lefebvre J and Hauff P. 1999. Automatic endmember extraction from hyperspectral data for mineral exploration. In Proc. 21st Can. Symp. Remote Sensing, Ottawa: 21–24

Plaza A, Benediktsson J A, Boardman J W, Brazile J, Bruzzone L, Camps-Valls G, Chanussot J, Fauvel M, Gamba P, Gualtieri A, Marconcini M, Tilton J C and Triani G. 2009. Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, **113**(1): S110–S122 DOI: 10.1016/j.rse.2007.07.028

Plaza A, Martinez P, Perez R and Plaza J. 2002. Spatial/Spectral Endmember Extraction by Multidimensional Morphological Operations. *IEEE Transactions on Geoscience and Remote Sensing*, **40**(9): 2024–2041

Plaza A, Valencia D, Plaza J and Chang C I. 2006. Parallel implementation of endmember extraction algorithms from hyperspectral data. *IEEE Geoscience and Remote Sensing letters*, **3**(3): 334-338 DOI: 10.1109/LGRS.2006.871749

Plaza A, Valencia D, Plaza J and Martinez P. 2006. Commodity cluster-based parallel processing of hyperspectral imagery. *Journal of Parallel and Distributed Computing*, **66**(3): 345–358 DOI: 10.1016/j.jpdc.2005.10.001

Quinn M J. 2003. Parallel Programming in C with MPI and OpenMP. New York: McGraw-Hill Science/Engineering/Math

Robila S A and Maciak L G. 2006. A parallel unmixing algorithm for hyperspectral images. Intelligent Robots and Computer Vision XXIV. *SPIE Press*, **6384**: 63840F.1-63840F.11 DOI: 10.1117/12.685655

Rogge D M, Rivard B, Zhang J K and Feng J L. 2006. Iterative spectral unmixing for optimizing per-pixel endmember sets. *IEEE Transactions on Geoscience and Remote Sensing*, **44**(12): 3725–3736 DOI: 10.1109/TGRS.2006.881123

Wang J and Chang C I. 2006. Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, **44**(9): 196–2895 DOI: 10.1109/TGRS.2006.874135

Winter M E. 1999. N-FINDR: an Algorithm for Fast Autonomous Spectral Endmember Determination in Hyperspectral Data. SPIE Imaging Spectrometry V. San Diego CA: SPIE Press: 266-275

Zortea M and Plaza A. 2009. Spatial preprocessing for endmember extraction. *IEEE Transactions on Geoscience and Remote Sensing*, **47**(8): 2679–2693 DOI: 10.1109/TGRS.2009.2014945

附中文参考文献

陈国良，吴俊敏，章锋，章隆兵. 2002. 并行计算机体系结构. 北京:高等教育出版社

陈国良. 2009. 并行算法的设计与分析 (第三版). 北京：高等教育出版社