Two-level parallel independent component analysis endmember extraction algorithms

LUO Wenfei1, GAO Lianru2

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 maintain 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


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 least 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 multiprocessor (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 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

\[ X = [x(1), x(2), \ldots, x(n)] \]  

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

\[ x(i) = \sum_{j=1}^{m} \xi e_j a(i)_j \]  

where \( e_j, j=1, 2, \ldots, m \) is the endmember in the image, and \( a(i)_j \) is the abundance fraction corresponding to \( e_j \) in the spectral pixel \( x(i) \). The abundance fractions in \( x(i) \) compose of an abundance vector denoted by

\[ a(i) = [a(i)_1, \ldots, a(i)_m]^T \]  

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 \( W \) by nongaussianity, satisfying

\[ \hat{a}(i) = W x(i) \]  

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

\[ Y = ZX \]  

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. \( W \) is given by

\[ W^{(k+1)} = E\{Y g(W^{(k)} Y)^T\} - E\{g'(W^{(k)} Y)^T\} W^{(k)} \]  

where \( W^{(k)} \) denotes the new matrix of \( W \) in the \( (k+1) \)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, \( W^{(k+1)} \) 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 \( Y \);

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

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

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

Step 5 After terminating the procedure of iteration, estimate the abundance \( \hat{a}_i \), \( i=1, 2, \ldots, m \) where \( w_i \) is the \( i \)-th IC;

Step 6 Select the endmembers by

\[ \epsilon_i = x_{i_0} \]  

and

\[ k_i = \text{avg} \{a(i)_j\} \]  

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, to improve the speed of calculating \( W^{(k)} \) 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 \( Y \) is decomposed into \( \{Y_1, Y_2, \ldots, Y_a\} \) by using the partitioning rule of successive I/O access. The \( j \)-th TE contains the partitioning \( Y_j, j=1, 2, \ldots, a \) and calculates Eq. (6) as follows.

\[ W(j)^{(k+1)} = E\{Y_j g(W^{(k)} Y)^T_j\} - E\{g'(W^{(k)} Y)^T_j\} W^{(k)} \]  

Obviously

\[ W^{(k)} = \frac{1}{S} \sum_{j=1}^{a} S_j W(j)^{(k)} \]  

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

Step 1 Decompose \( Y \) into \( \{Y_1, Y_2, \ldots, Y_a\} \);

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

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

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

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

Step 6 The \( i \)-th TE, \( j=1, 2, \ldots, a \), estimate the abundance as

\[ \hat{a}_i = w_i Y_j, i=1, \ldots, m \]  

Step 7 And extract a candidate endmember by

\[ \epsilon_i = x_{i_0}, \]  

and

\[ k_i = \text{avg} \{a(i)_j\} \]  

Step 8 The master wait for \( \epsilon_i, i=1, \ldots, m, j=1, \ldots, a \) from all TEs. Reduce the maximal abundance and yield \( \epsilon_i, i=1, \ldots, a \);
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 node provides a scalable inter-node parallelism. These executable 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, Y is decomposed into row blocks (Plaza, 2002) \( Y = \{ Y_1, Y_2, \cdots, Y_p \} \) in coarse grain and attach to every PE. In the second level of data partitioning, for the \( p \)-th PE, \( p=1, \cdots, b \), decompose \( Y_p \) into \( \{ Y_{1p}, Y_{2p}, \cdots, Y_{bp} \} \) 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:

1. **Step 1** Apply two levels data partitioning strategy to the hyper-spectral remote sensing image;
2. **Step 2** The master initialize \( W^{(0)} \) and broadcast it to all PEs, i.e. workers;
3. **Step 3** For the \( p \)-th PE, \( p=1, \cdots, b \), execute Step 3 and Step 4 of SMM-PICA and obtain \( W(p)^{0,1} \);
4. **Step 4** The master collect all \( W(p)^{0,1}, p=1, \cdots, b \) and reduce \( W^{0,1} \) by Eq. (9);
5. **Step 5** The master decorrelate and renormalize \( W^{0,1} \), broadcast it to all PEs and go to Step 3 until satisfy the convergence condition;
6. **Step 6** All PEs execute Step 6 and Step 8 of SMM-PICA within their own data partitioning;
7. **Step 7** The master execute the maximal reduction operation from the results collecting from all PEs and then obtain endmembers \( e_i \), \( i=1, \cdots, a \);
8. **Step 8** Return all endmembers.

In Step 4 and Step 5 of SynPICA-2L EEA, all PEs calculate \( W^{0,1} \) simultaneously in every 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.

1. **Step 1** The master randomize a group of numbers ranged from 1 to \( p \) with uniformed distribution;
2. **Step 2** Map these numbers into each pixel in the image, yields \( \text{map}(s,t) \in \{1, 2, \cdots, b\} \) where \( (s,t) \) is the coordinate of the pixel;
3. **Step 3** The master partition the image by \( Y_s=\{y(s,t)| \text{map}(s,t) \in Y_s \} \) and \( \text{map}(s,t)=i \), \( i=1, \cdots, b \) (12)
4. **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:

1. **Step 1** The master collects candidate endmembers, denoted by \( E_i \), from all PEs;
2. **Step 2** Input \( E_i \), use a fast EEA to output the final endmember set \( E \);
3. **Step 3** Return \( E_i \).

A fast EEA in Step 2 should be implemented by a low-computation EEA including VCA, ATGP or IEA, except for the spatial information incorporating EEs 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:

1. **Step 1** The master executes RSP to generate data partitioning \( \{ Y_1, Y_2, \cdots, Y_p \} \) and transform to corresponding PEs;
2. **Step 2** Every PEs execute SMM-PICA independently;
3. **Step 3** The master waits for all PEs and executes AC-ER to obtain the final endmember set \( E \);
4. **Step 4** Return \( E \).

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
Two-level parallel independent component analysis endmember extraction algorithms

Luo Wenfei, et al.

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 convergence of SMM-PICA in every PEs. Two-level grouping parallel ICA (GrpPICA-2L) algorithm avoids asynchronous data partitioning and takes advantage of the SynPICA-2L and AsyPICA-2L. GrpPICA-2L divides the PEs into several groups \( \{G_1, G_2, \ldots, G_q\} \).

Let \( i_l \) be the first serial number for PE in group \( G_i \), and \( i_r \) be the last one. Then

\[
G_i = \{PE_{j}, j=i_l, i_l+1, \ldots, i_r\}, \quad i=1, \ldots, q
\]

(13)

The intra-group PEs should be topologically neighbor in network. And make each \( i_l-i_r \) identical with maintain load balance. GrpPICA-2L is summarized as follows:

Step 1 Divide 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 \( G_i \), \( \psi=1, \ldots, q \) 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 \( E \);
Step 6 Return \( E \).

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.

![AVIRIS image in Cuprite Nevada (648 nm)](image)

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.

![Executive time of parallel ICA algorithms](image)

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.

![SAD RMSE curve of AsyPICA-2L](image)

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

REFERENCES

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

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


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，分别提出具有二级并行特性的二级同步、二级异步并行端元提取算法，并结合两者的优点，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

1 引 言


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，
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,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。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:

$$X=[x(1), x(2), \cdots, x(n)]$$  (1)

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

$$x(i) = \sum_{j=1}^{m} e_j a_j(i)$$  (2)

式中, $e_j, j=1, 2, \cdots, m$ 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 $e_j$ 的丰度含量,组成丰度向量:

$$a(i)=[a_1(i), \cdots, a_m(i)]^T$$  (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。再以非高斯性特征为准则,寻找一个最佳的分离矩阵 $W$, 使得每个像元中各端元的丰度估计:

$$\hat{a}(i) = Wz(i)$$  (4)

尽可能地接近于丰度向量 $a(i)$. 式中 $\hat{a}(i)=\hat{a}_1(i), \cdots, \hat{a}_m(i)$ 为估计的丰度向量, $Z$ 为主成分向量所组成的矩阵。为描述方便,记主成分向量为:

$$Y=ZX$$  (5)

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

$$W^{(k)} = E\{Yg(W^{(k)}Y)^T\} - E\{g(W^{(k)}Y)^T\}W^{(k)}$$  (6)

式中, $W^{(k)}$ 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 $W^{(k+1)}$ 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 $Y$;

步骤2 初始化分离矩阵 $W^{(0)}$;

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 $W^{(k)}$ 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 $\hat{a}_i, Y, i=1, \cdots, m, W$ 为 $W$ 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:

$$e_i = x_i, \text{且} \lambda_i = \arg \max_j |a_{j}(i)|$$  (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 $W^{(k)}$, 因此, 提高 $W^{(k)}$ 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Y划分为{Y_1, Y_2, ..., Y_p}，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Y_j划分，j = 1, 2, ..., a。在ICA-EEA步骤(3)中，不同TE对式(6)的计算如下:

\[ W(j)^{(k)} = E\left\{ Y_j \bar{W} W^{(k)} Y_j \right\} - E\left\{ \bar{W} W^{(k)} \right\} W^{(k)} \]

式中，s为Y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 将图像Y划分为{Y_1, Y_2, ..., Y_p}；
步骤2 初始化分离矩阵W^{(0)}；
步骤3 对每一个TE，计算式(8)；
步骤4 等待各TE计算完成后，采用式(9)对W^{(k)}
进行规约；
步骤5 对得到的W^{(k)}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:

\[ \hat{a}_j = w_{j} \forall Y_j, i=1, ..., m \]  (10)

步骤7 选取候选端元:

\[ e_{i,j} = x_i, \text{且} \quad \hat{a}_j = \max_{i=1, ..., a} \left| a_{i,j} \right| \]  (11)

步骤8 等待各TE搜索完候选端元后，e_{i,j} = 1, ..., m, j = 1, ..., a, 以对应的丰度最大绝对值进行规约，得到最终的端元e_i,

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，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Y={Y_1, Y_2, ..., Y_p}，划分数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Y={Y_{1p}, Y_{2p}, ..., Y_{pp}，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W^{(0)}，并把W^{(0)}广播到各从PE；
步骤3 对于第p个从PE，p = 1, ..., b，执行SMM-PICA的步骤3, 4，获得W^{(p)}(0)，并把结果传输给主PE；
步骤4 主PE等待W^{(p)(0)}，p = 1, ..., b 收集完毕后，通过式(9)规约W^{(p)}；
步骤5 主PE对得到的W^{(p)}进行归一化以及对称正交化，并广播到各从PE进入下一步迭代；
步骤6 迭代完成后，各从PE获得的最终分离矩阵，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e_i,

步骤8 返回所有的端元，算法结束。

SynPICA-2L算法在多次迭代中同步计算全局分离矩阵W^{(p)}, 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-EEA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。
1211

罗文斐 等:二级并行独立成分数析端元提取算法

描述如下:

步骤1 主PE执行RSP算法,产生划分\(\{Y_1, Y_2, \ldots, Y_b\}\),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\(E\);

步骤4 返回\(E\),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;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\(\{G_1, G_2, \ldots, G_q\}\)。考虑PE编号的任意性,可假设组内PE连续编号,并令\(i\)为分组内第一个PE的编号,\(i\)为最后一个编号;

\[
G_i = \{PE_{s, i}, PE_{s, i+1}, \ldots, PE_{s, q}\}, \quad i=1, \ldots, q
\]

通常让组内PE在网络拓扑结构中具有相邻性,并应使\(i, i+q\)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\(G_i\),\(i=1, \ldots, q\),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等待各分组执行完毕后,执行AC-ER算法,获得最终端元集合\(E\);

步骤6 返回\(E\),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\(q\)个分组间进行异步通信,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。

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级增大，当节点数为20时，误差达到0.13。

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级增大，当节点数为20时，误差达到0.13。
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：

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值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


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