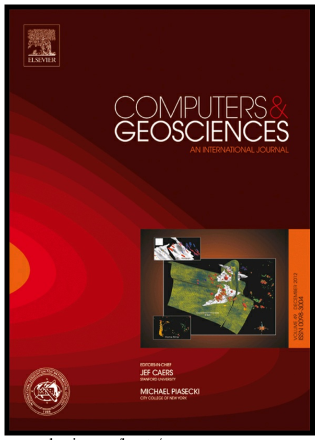


www.elsevier.com/locate/cageo

Matching pursuit parallel decomposition of seismic data

Chuanhui Li, Fanchang Zhang

PII: S0098-3004(16)30527-1
DOI: http://dx.doi.org/10.1016/j.cageo.2017.04.005
Reference: CAGEO3943

To appear in: *Computers and Geosciences*

Received date: 12 October 2016
Revised date: 21 March 2017
Accepted date: 10 April 2017

Matching pursuit parallel decomposition of seismic data

Chuanhui Li[1*], Fanchang Zhang[2]

[1]School of Geophysics and Information Technology, China University of Geosciences (Beijing), Beijing 100083, China

[2]School of Geosciences, China University of Petroleum (East China), Qingdao, 266580, China

[*]Corresponding author: lichh1987@163.com

Abstract

In order to improve the computation speed of matching pursuit decomposition of seismic data, a matching pursuit parallel algorithm is designed in this paper. We pick a fixed number of envelope peaks from the current signal in every iteration according to the number of compute nodes and assign them to the compute nodes on average to search the optimal Morlet wavelets in parallel. With the help of parallel computer systems and Message Passing Interface, the parallel algorithm gives full play to the advantages of parallel computing to significantly improve the computation speed of the matching pursuit decomposition and also has good expandability. Besides, searching only one optimal Morlet wavelet by every compute node in every iteration is the most efficient implementation.

Keywords:

Matching pursuit; Computation speed; Parallel algorithm; Seismic data

1. Introduction

Matching pursuit is an algorithm for sparse representation of signals. It adaptively decomposes a signal into a series of time-frequency atoms that match the time-frequency characteristics of the signal (Mallat and Zhang, 1993). The matching pursuit decomposition of seismic signals has been widely used in seismic data processing and interpretation, such as migration (Wang and Pann, 1996; Li et al, 1998), filtering (Nguyen and Castagna, 2000), interpolation (Øzbek, 2009), inversion (Yang et al, 2011; Zhou et al, 2013), and spectral analysis for hydrocarbon recognition (Castagna et al, 2003; Wang, 2007) and channel detection (Liu and Marfurt, 2007).

The Gabor wavelet is used as the time-frequency atoms in a conventional matching pursuit algorithm. Considering the characteristics of seismic signals, Liu et al (2004) applies the Ricker wavelet as the time-frequency atoms to decompose seismic data. Liu and Marfurt (2005), and Wang (2007) implement matching pursuit decomposition of seismic signals based on the Morlet wavelet. A conventional matching pursuit algorithm costs a huge amount of computation, because it searches the optimal time-frequency atoms from an abundant dictionary (Mallat and Zhang, 1993). Liu and Marfurt (2005) introduce complex-trace analysis into the algorithm to avoid the blind search of time-frequency atoms, greatly improving its computation speed. Based on this, Wang (2007) summarizes “three-stage procedure” for matching pursuit decomposition of seismic signals where the decomposition is more precise by executing local search around the complex-trace attributes. In order to improve the spatial continuity of the decomposition, Wang (2010) further proposes multichannel matching pursuit.

Although matching pursuit decomposition of seismic signals has been greatly improved, the local search of time-frequency atoms and iterative implementation of the algorithm still cost lots of time. To further enhance its execution speed, in addition to the algorithm itself, with the help of high-performance computers to accelerate the decomposition could also be considered. Liu and Marfurt (2007) perform matching pursuit to seismic data by searching a suite of optimal time-frequency atoms at one iteration rather than one at a time as in a conventional algorithm. Because the search of optimal time-frequency atoms has the same implementation and is independent of each other, Liu and Marfurt (2007) inspire us to execute matching pursuit decomposition in parallel by parallel computer systems. Hence, in this paper, we design a parallel decomposition algorithm of matching pursuit to effectively improve the computation speed of the matching pursuit decomposition of seismic signals. It gives full play to the advantages of parallel computing and has good expandability. We believe that the realization of matching pursuit parallel decomposition for seismic data will benefit the application of matching pursuit in the processing and interpretation of large-scale 3D seismic data.

2. Matching pursuit decomposition of seismic signals

The Morlet wavelet (Morlet et al., 1982a, 1982b) is preferred to be used as time-frequency atoms in the matching pursuit decomposition of seismic signals. The energy normalized Morlet wavelet $w(t)$ is given by:

$$w_{\gamma}(t) = \left(\frac{2\ln 2}{\beta \cdot \pi}\right)^{1/4} \sqrt{\xi} e^{-\frac{\ln 2 \cdot \xi^2 (t-u)^2}{\beta}} e^{i[2\pi\xi(t-u)+\varphi]} \quad (1)$$

where t refers to the time, ξ is the mean frequency, u is the time delay, φ is the phase shift, and β is the scale that controls the width of wavelet in time and frequency domain. Every Morlet wavelet can be characterized by a set of these four parameters: $\gamma = (\xi, u, \varphi, \beta)$. The spectrum of $w_{\gamma}(t)$ is given as follows:

$$\tilde{w}_{\gamma}(f) = \left(\frac{\beta \cdot 2\pi}{\ln 2}\right)^{1/4} \frac{1}{\sqrt{\xi}} e^{-\frac{\beta \cdot \pi^2 (f-\xi)^2}{\ln 2 \cdot \xi^2}} e^{-i[2\pi(f-\xi)u-\varphi]} \quad (2)$$

where f is the frequency. By matching pursuit decomposition, a seismic signal $s(t)$ can be represented by a linear combination of the Morlet wavelets:

$$s(t) = \sum_j a_j \cdot w_{\gamma_j}(t) + noise \qquad (3)$$

where a_j is the amplitude of the *j*th Morlet wavelet w_{γ_j}. In complex domain, Eq.(3) becomes (Liu and Marfurt, 2007):

$$S(t) = \sum_j A_j \cdot W_{\tau_j}(t) + noise \qquad (4)$$

where $S(t)$ is the analytic complex seismic signal formed by Hilbert transform, A_j is the complex amplitude consisting of the amplitude and phase of the *j*th analytic complex Morlet wavelet $W_{\tau_j}(t)$ that is also formed by Hilbert transform. Since the phase of the Morlet wavelet is included in the complex amplitude, the controlling parameters of the Morlet wavelet are changed into $\tau = (\xi, u, \beta)$.

Matching pursuit decomposition is implemented iteratively. According to Liu and Marfurt (2007), in one iteration of the decomposition, a suite of envelope peaks that fall above a user-specified percentage of the largest peak in the current (residual) trace are picked. The optimal Morlet wavelets are matched at these envelope peaks. At each envelope peak, the time delay u_j of the optimal Morlet wavelet is the time position of the envelope peak and the mean frequency ξ_j is the instantaneous frequency of the current trace at the envelope peak by complex-trace analysis. The scale β_j can be searched over a group of preselected, uniformly distributed β_j values by maximizing the inner product of the current trace and the Morlet wavelet with fixed u_j and ξ_j values (Wang, 2007). For each matched Morlet wavelet, once we obtain the preliminary estimates of $\tau_j = (\xi_j, u_j, \beta_j)$, we optimize these three parameters by local search in a small range around the preliminary estimates. The complex amplitude A_j of all the matched Morlet wavelets in one iteration can be calculated using a least-squares algorithm by minimizing the energy of the difference between the complex seismic trace and the complex matched wavelets.

$$E(t)=\left[S(t)-\sum_{j}^{J}A_j\cdot W_{\tau_j}\left(t\right)\right]^2 \tag{5}$$

where J refers to the number of the matched wavelets in one iteration. The matched Morlet wavelets are subtracted from the seismic trace at every iteration. The iteration continues until the energy of residues is below a user defined threshold value.

Figure 1 is a synthetic signal designed to show the implementation of the above matching pursuit algorithm. The synthetic signal (left-most in Figure 1) is composed of 7 Morlet wavelets with different controlling parameters as shown in Table 1.

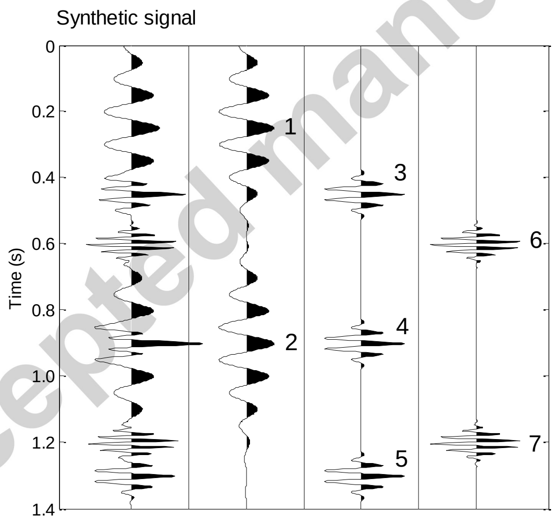
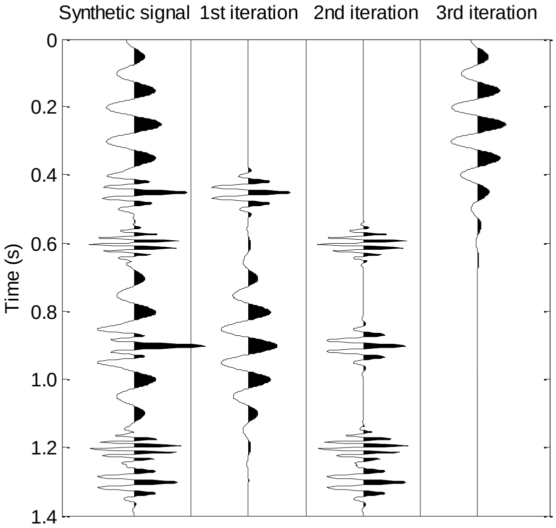


Figure 1 A synthetic signal

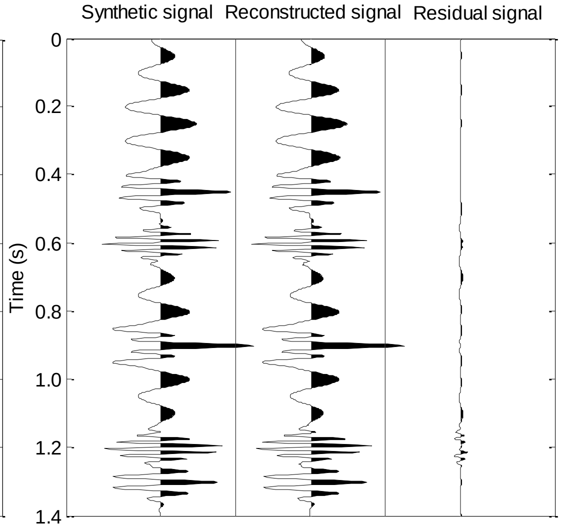
Table 1 Controlling parameters of the Morlet wavelets

No.	ξ (Hz)	u (s)	φ (°)	β	a
1	10	0.25	0	3	1.5
2	10	0.9	0	3	1.5
3	30	0.45	0	1	1
4	30	0.9	0	1	1
5	30	1.3	0	1	1
6	50	0.6	135	2	1
7	50	1.2	135	2	1

In each iteration of the decomposition, we design to pick the envelope peaks that fall above 70% of the largest peak in the current trace to search the optimal Morlet wavelets. At the first iteration, 2 envelope peaks are picked and the corresponding optimal Morlet wavelets are matched as shown in Figure 2a. In the second and third iteration, 4 and 1 optimal Morlet wavelets are matched, respectively. As we can see, these 7 matched Morlet wavelets are in good agreement with the consisting wavelets in Table 1. When we reconstruct the synthetic signal with these 7 matched wavelets, the residues between the input and reconstructed synthetic signal are quite small, as shown in Figure 2b.



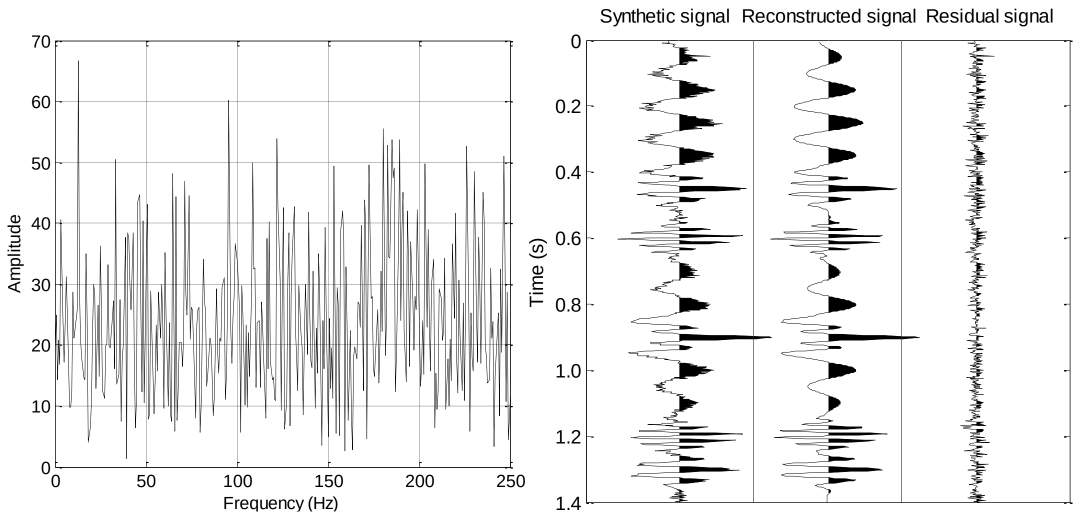
(a)



(b)

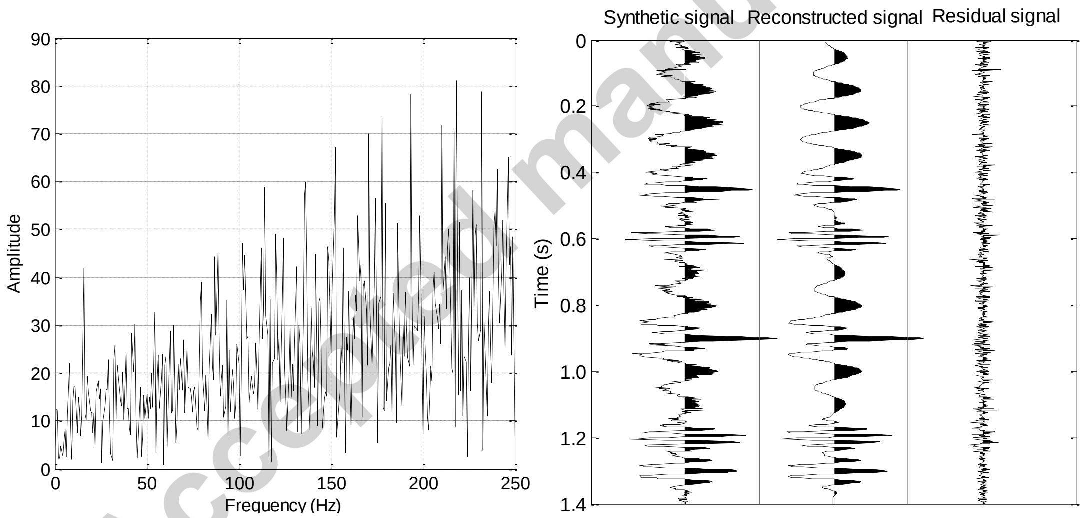
Figure 2 Matching pursuit decomposition of the synthetic signal

To further validate the noise tolerance of the algorithm, white noise (Figure 3a) with 20% of effective signal energy is added to the above synthetic signal (left-most in Figure 3b). Then we repeat the above decomposition test. After 3 iterations, 7 matched wavelets are obtained and then used to reconstruct the synthetic signal. The residues between the input and reconstructed synthetic signal are shown in Figure 3b. As we can see, the effective signal is well reconstructed and the residues are almost noise. This well indicates that the matching pursuit algorithm has a good noise tolerance. Another decomposition test to the synthetic signal with colored noise are also conducted, as shown in Figure 4. The same conclusion can be drawn from the results.



(a) **(b)**

Figure 3(a) Amplitude spectrum of white noise; (b) Matching pursuit decomposition of the synthetic signal with white noise



(a) **(b)**

Figure 4(a) Amplitude spectrum of colored noise; (b) Matching pursuit decomposition of the synthetic signal with colored noise

3. Matching pursuit parallel decomposition

3.1 Algorithm

In Liu and Marfurt's algorithm, a suite of envelope peaks are picked from the current signal where the optimal Morlet wavelets are matched in one iteration of the decomposition. Because the search of the optimal wavelets has the same implementation and is independent of each other, they can be assigned to several compute nodes to execute in parallel. Nevertheless, picking envelope peaks by a user-specified percentage of the largest peak in the current trace probably differs the number of the picked envelope peaks at every iteration (just like our example in the above section), leading to uneven jobs for a fixed number of compute nodes. This easily causes idling for parts of the compute nodes during the parallel computing, decreasing the overall efficiency of the parallel decomposition.

To give full play to the advantages of parallel computing, we fix the number of the envelope peaks where the optimal Morlet wavelets are matched in every iteration according to the number of compute nodes. Assume that the number of compute nodes is N, in every iteration the number of the envelope peaks we pick is set as M that is equal to or an integer multiple of N. These M envelope peaks are the top M largest peaks in the current trace. They are assigned to the compute nodes on average by the host process. The optimal Morlet wavelets are searched by every compute node at the received envelope peaks and returned to the host process. This procedure is implemented in parallel. The complex amplitudes of all the M matched wavelets are calculated by the host process. After the matched wavelets are subtracted from the current trace, the parallel decomposition for one iteration is completed. The whole procedure is described more intuitively as shown in the flow chart in Figure 5.

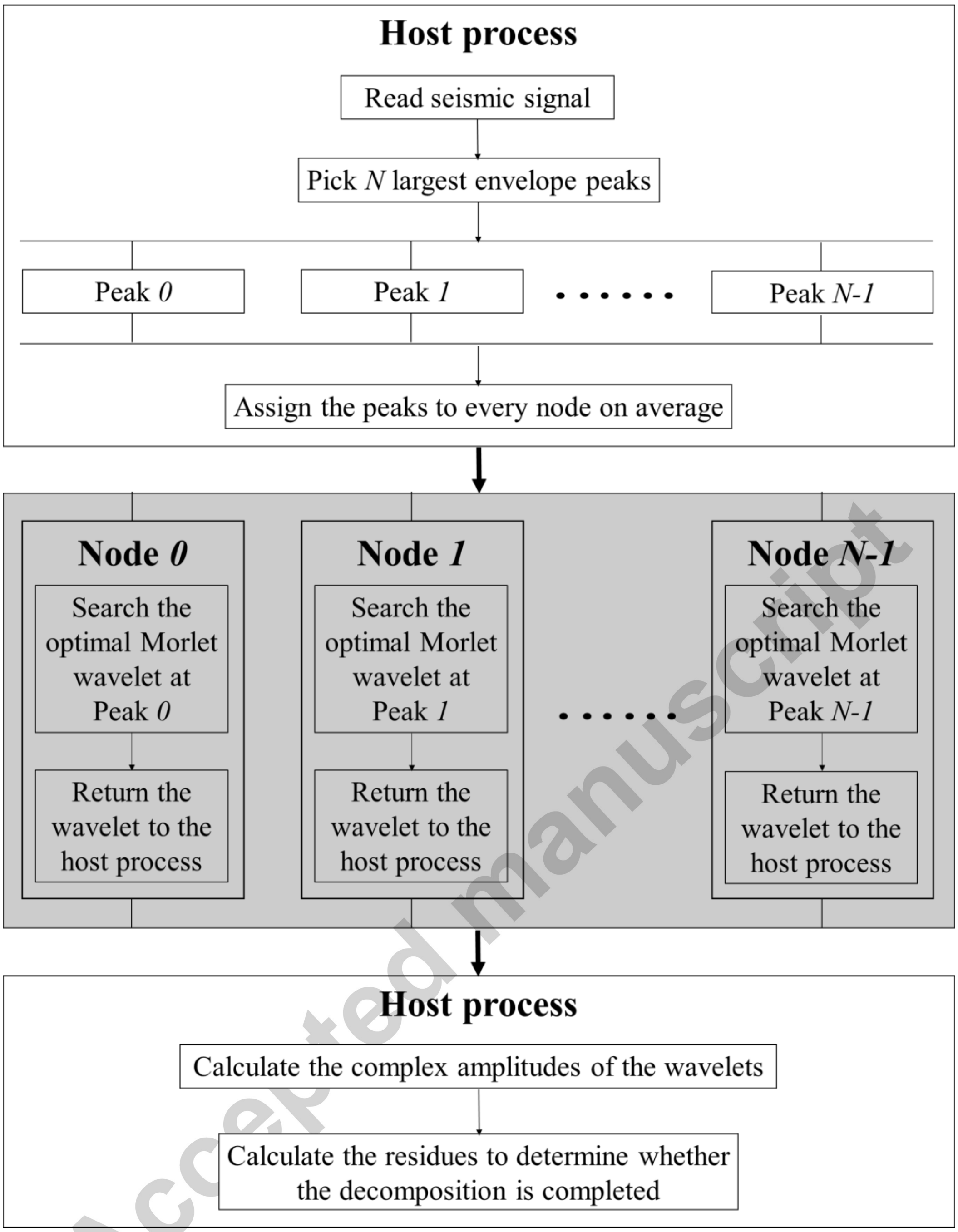


Figure 5 Matching pursuit parallel decomposition for one iteration

The gray shaded area in Figure 5 is the parallel computing part we design for matching pursuit decomposition. It is the core part of the parallel algorithm. By executing the search of optimal Morlet wavelets in parallel at every iteration with the help of multiple compute nodes, the computational speed of matching pursuit decomposition is significantly improved. The whole procedure involves the communication between the host process and every compute node. Hence, its program can be achieved by means of Message Passing Interface (MPI).

3.2 Validity analysis

Figure 6 is a real seismic signal we use to verify the above parallel algorithm of matching pursuit decomposition. The operating environment of the MPI parallel program is an Intel multiprocessor computer whose main frequency of processors is 2.40GHz.

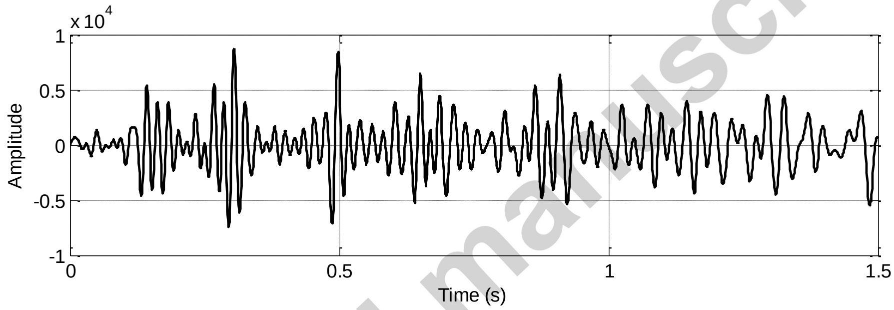
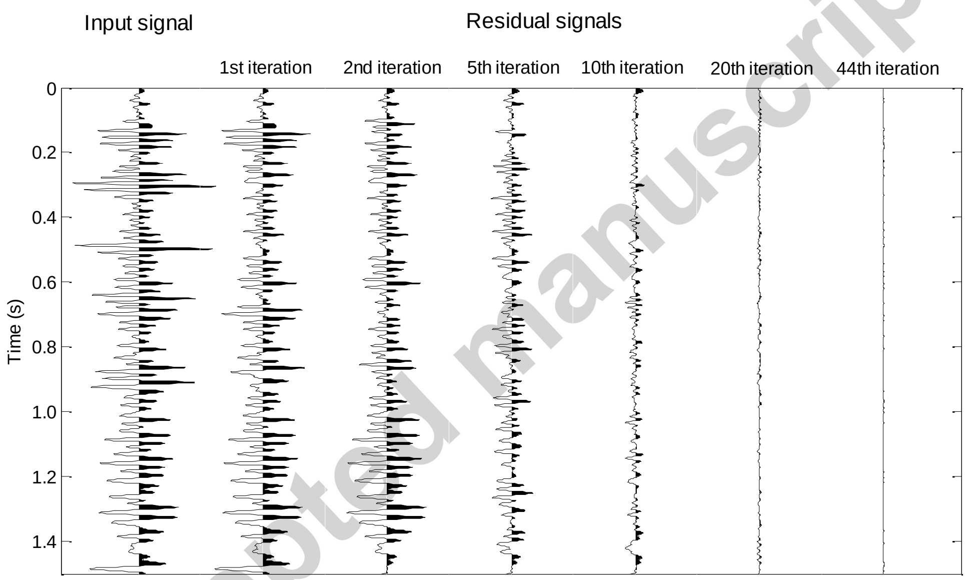
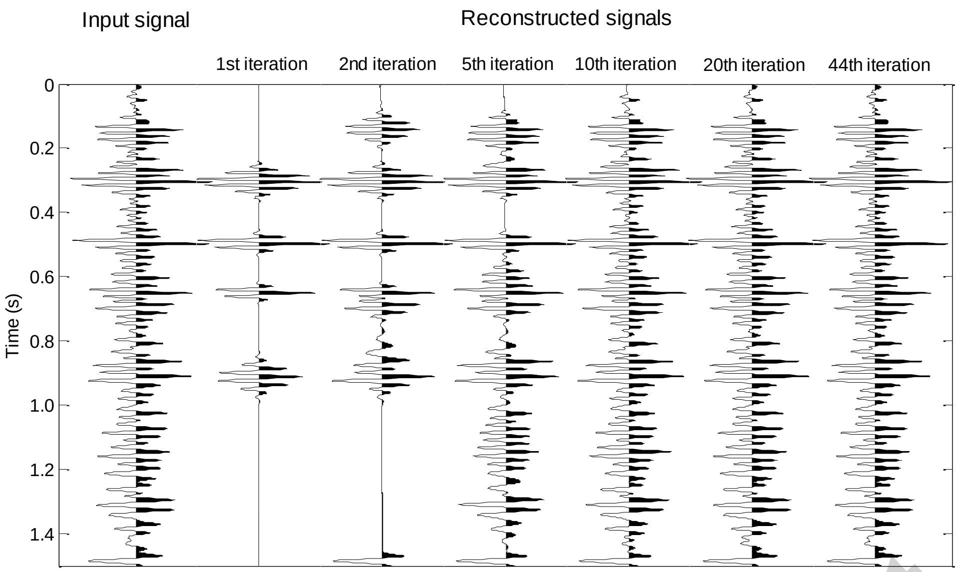


Figure 6 A real seismic signal

We first enter the environment of MPI, executing all the initializations and generating communication domain. We set 4 processors for parallel computing. Then we design to pick 4 envelope peaks from the current signal to search the optimal Morlet wavelet in every iteration of our parallel program, so the number of the processes in the communication domain is specified as 4. After 44 iterations, the program ends when the energy of the residues falls below a threshold value we defined. Hence, we obtain a total of 176 optimal Morlet wavelets. Figure 7 illustrates some residual and reconstructed signals at different iterations during the parallel decomposition. The reconstructed signals are modeled by all the matched Morlet wavelets we have obtained after a certain iteration. As we can see from the figure, the signal is decomposed gradually along with the increasing iterations and the reconstructed signal is getting close to the input signal. This indicates that the matching pursuit decomposition of seismic signals can still be successfully implemented by picking a fixed number of envelope peaks from the current signal to search the optimal wavelets in every iteration.



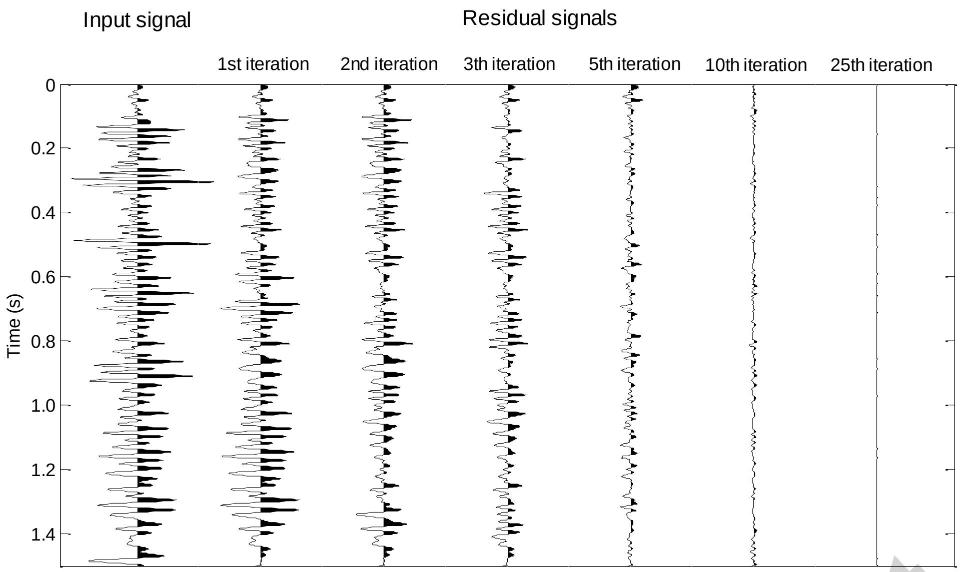
(a) Residual signals



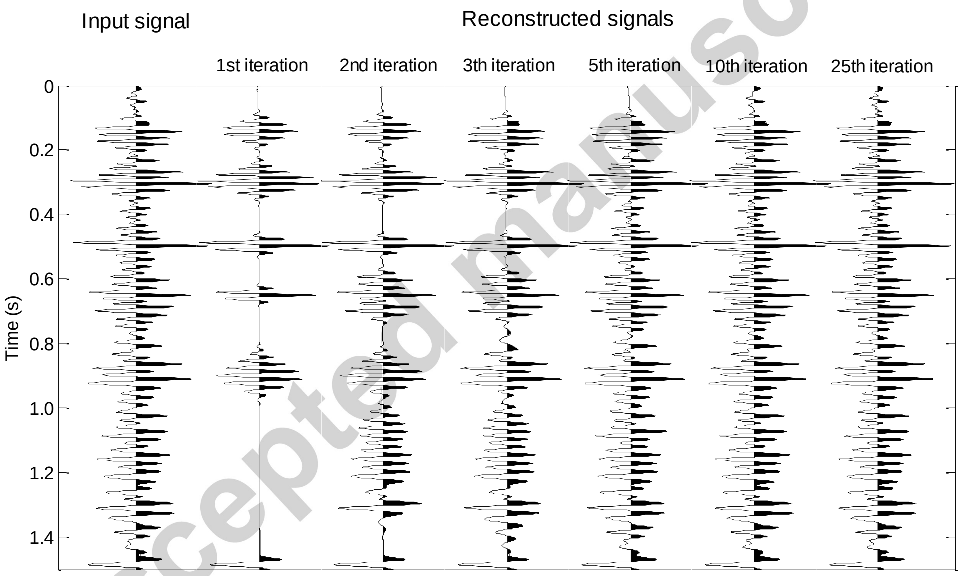
(b) Reconstructed signals

Figure 7 Residual and reconstructed signals at different iterations

Figure 8 is another test for the MPI parallel program by the same seismic signal. In this test, we still set 4 processors for parallel computing but we design to pick 8 envelope peaks in every iteration of the program. Thus, in every iteration two envelope peaks are assigned to every processor to search the optimal Morlet wavelets. As we can see from the figure, after 25 iterations, the seismic signal is decomposed thoroughly and the residual signal can be considered as at the random noise level.



(a) Residual signals



(b) Reconstructed signals

Figure 8 Residual and reconstructed signals at different iterations

3.3 Speedup and efficiency analysis

The above two tests well prove the validity of our parallel algorithm of matching pursuit decomposition. Next we further analyze its computation speed that we are more concerned about. Assume that to search one optimal Morlet wavelet at a certain peak by compute node is defined as one process. Hence, the number of processes refers to the number of the matched wavelets in one iteration. To analyze the speedup and the computational efficiency of the matching pursuit parallel decomposition, in the MPI parallel program we set different numbers of compute nodes and processes to decompose the seismic signal in Figure 6, respectively. The detailed implementations are shown in Table 2. The operating environment of the MPI parallel program is still an Intel multiprocessor computer whose main frequency of processors is 2.40GHz.

Table 2 Test for the speedup and efficiency

Trial	No. of processors	No. of processes	Iteration	No. of matched wavelets	Computing time （s）	Speedup	Overall efficiency
①	1	1	80	80	5.361713	/	/
②	2	2	40	80	2.737699	1.9585	0.97925
③	2	4	20	80	2.740146	1.9567	0.97823
④	2	8	10	80	2.744404	1.9537	0.97685
⑤	2	16	5	80	2.750807	1.9491	0.97455

⑥	4	4	20	80	1.397733	3.836	0.959
⑦	4	8	10	80	1.423375	3.7669	0.9417
⑧	4	16	5	80	1.459963	3.6725	0.9181
⑨	8	8	10	80	0.753981	7.1112	0.8889
⑩	8	16	5	80	0.789904	6.7878	0.8485

As shown in Table 2, there are 10 trials of matching pursuit parallel decomposition. Although the numbers of processors and processes are different in every trial, the number of the matched Morlet wavelets we obtain is the same by controlling the iteration to ensure the fairness of the comparisons. In Trial ①, we set 1 processor and 1 process, so this is a conventional matching pursuit implementation used as a reference. From Trial ②, the numbers of processors and processes are increasing for parallel computing.

Compared Trial ①, Trial ②, Trial ⑥ and Trial ⑨, when the number of processes is equal to the number of processors, which means only one process is assigned to every processor in every iteration, the computation speed of the matching pursuit parallel decomposition is improved significantly with the increasing number of processors. When the number of processors is set as 8, the speedup of the parallel computing is greater than 7. This also indicates good expandability of the parallel algorithm. Nevertheless, the overall efficiency of the parallel decomposition that is calculated by dividing the speedup by the number of processors is decreased with the increasing number of the processors, because the absolute traffic among the processors is increased.

In Trial ②, Trial ③, Trial ④, and Trial ⑤, the number of processors is set as 2, while the number of processes is different. As we can see, with the increasing number of processes, the speedup and overall efficiency is decreased. That's because when the number of processes is greater than 2, the number of processes assigned to every processor is more than 1. The communication among the processes in every processor decreases the overall efficiency of parallel decomposition. Hence, searching only one optimal Morlet wavelet by every processor in every iteration is the most efficient implementation for the matching pursuit parallel decomposition of seismic signals. The same conclusion can be draw from the comparison of Trial ⑥, Trial ⑦ and Trial ⑧, also from the comparison of Trial ⑨ and Trial ⑩.

4. Comparisons

The matching pursuit decomposition of seismic data is performed trace by trace. Actually, for a seismic profile or a 3D seismic data volume, the traces can be directly assigned to the compute nodes of parallel computer systems on average to implement decompositions in parallel. We use a seismic profile as shown in Figure 9 to compare this coarse-grained parallel strategy and our parallel algorithm of matching pursuit. The operating environment of these two parallel programs is still an Intel multiprocessor computer whose main frequency of processors is 2.40GHz.

First, those 800 traces are assigned to 1, 2, 4, 8, 16 and 32 processors on average to decompose in parallel, respectively. Then we implement our parallel algorithm to the traces by setting 1, 2, 4, 8, 16, 32 processors, respectively, and searching only one optimal wavelet by every processor. In these two parallel methods, we define the same threshold value to end the decompositions for fair comparisons.

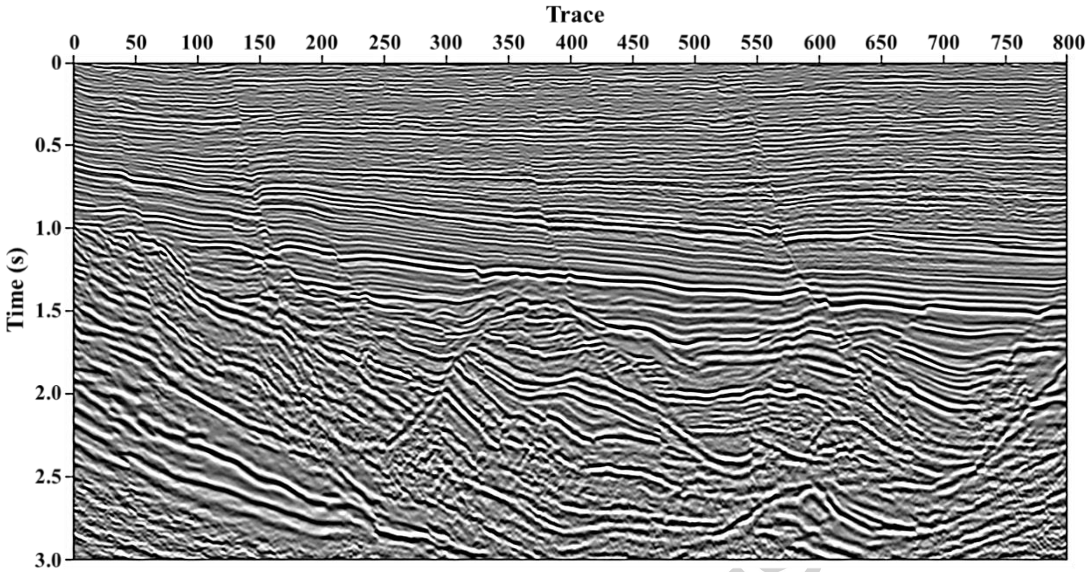
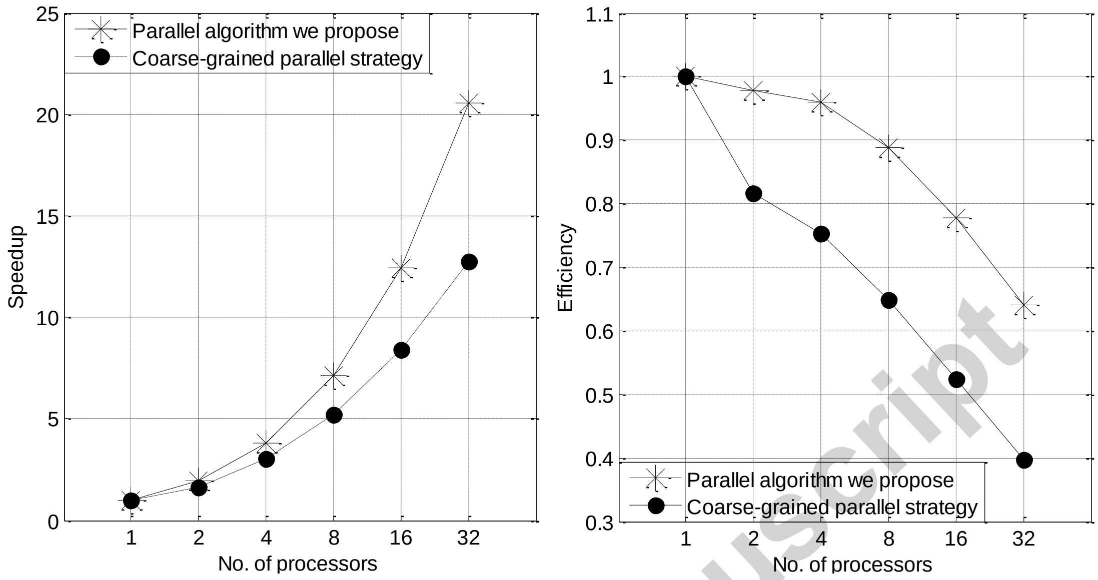


Figure 9 Seismic profile

After all the implementations are completed, we calculate the speedup and efficiency of these two methods for different numbers of processors, respectively, as shown in Figure 10. The speedup of the coarse-grained parallel strategy is apparently lower than that of our parallel algorithm of matching pursuit, and the difference between these two is increased with the increasing number of processors. Meanwhile, the overall efficiency of the coarse-grained parallel strategy is also much lower than that of our parallel algorithm. That’s because the number of iterations and optimal Morlet wavelets of seismic traces are unknown before the decomposition according to the matching pursuit algorithm. The coarse-grained parallel strategy easily leads to uneven jobs for compute nodes even though we assign the same number of traces to every compute node, affecting the overall efficiency of the parallel computing. By contract, our parallel algorithm of matching pursuit ensures even jobs for every processor, so it can give full play to the advantages of parallel computing.



(a) Speedup **(b) Efficiency**

Figure 10 Comparisons of two parallel methods

Conclusions

With the help of parallel computer systems and Message Passing Interface, the matching pursuit decomposition of seismic signals can be achieved in parallel by picking a fixed number of envelope peaks form the current signal in every iteration according to the number of compute nodes and assigning them to every compute node on average to search the optimal Morlet wavelets in parallel. The parallel algorithm gives full play to the advantages of parallel computing to significantly improve the computation speed of the matching pursuit decomposition and also has good expandability. Besides, searching only one optimal Morlet wavelet by every processor in every iteration is the most efficient implementation. We believe that the realization of matching pursuit parallel decomposition for seismic data will benefit the application of matching pursuit in the processing and interpretation of large-scale 3D seismic data.

Acknowledgments

This research is funded by the National Natural Science Foundation of China (No. 41604108).

References

Castagna, J. P., S. Sun, and R. W. Siegfried, 2003, Instantaneous spectral analysis: Detection of low-frequency shadows associated with hydrocarbons: The Leading Edge, 22, 120-127

Li, X. G., B. Wang, K. Pann, J. Anderson, and L. Deng, 1998, Fast migration using a matching pursuit algorithm: 68th Annual Meeting, SEG, Expanded Abstract, 1732-1735

Liu, J., and K. J. Marfurt, 2005, Matching pursuit decomposition using Morlet wavelets: 75th Annual Meeting, SEG, Expanded Abstract, 786-789

Liu, J., and K. J. Marfurt, 2007, Instantaneous spectral attributes to detect channels: Geophysics, 72, no.2, P23-P31

Liu, J., Y. Wu, D. Han, and X. Li, 2004, Time-frequency decomposition based on Ricker wavelet: 74th Annual Meeting, SEG, Expanded Abstract, 1937-1940

Mallat, S. and Z. Zhang, 1993, Matching pursuit with time-frequency dictionaries: IEEE transactions on signal processing, 41, 3397-3415

Morlet, J., G. Arens, E. Fourgeau, and D. Glard, 1982, Wave propagation and sampling theory-Part I: Complex signal and scattering in multilayered media: Geophysics, 1982, 47, no.2, 203-221

Morlet, J., G. Arens, E. Fourgeau, and D. Glard, 1982, Wave propagation and sampling theory-Part II: sampling theory and complex waves: Geophysics, 47, no.2, 222-236

Nguyen, T., and J. Castagna, 2000, Matching Pursuit of Two Dimensional Seismic Data and Its Filtering Application: 70th Annual Meeting, SEG, Expanded Abstract, 2067-2069

Øzbek, A., A. K. Øzdemir, and M, Vassallo, 2009, Interpolation by matching pursuit: 79th Annual Meeting, SEG, Expanded Abstract, 3254-3258

Wang, B., and K. Pann, 1996, Kirchhoff migration of seismic data compressed by matching pursuit decomposition: 66th Annual Meeting, SEG, Expanded Abstract, 1642-1645

Wang, Y. 2007, Seismic time-frequency spectral decomposition by matching pursuit: Geophysics, 72, no.1, V13-V20

Wang, Y. 2010, Multichannel matching pursuit for seismic trace decomposition: Geophysics, 74, no.5, V61-V66

Yang, H., X. Zheng, S. Ma, and J. Li, 2011, Thin-bed reflectivity inversion based on matching pursuit, 81th Annual Meeting, SEG, Expanded Abstract, 2586-2590

Zhou, D. Y., X. T. Wen, Z. H. He, and D. J. Huang, 2013, Application of Matching Pursuits in Seismic Inversion: Near Surface Geophysics Asia Pacific Conference, SEG, Expanded Abstract, 672-674

Highlights

- We design a parallel decomposition algorithm of matching pursuit
- The method effectively improve the computation speed of the matching pursuit decomposition of seismic signals
- The method gives full play to the advantages of parallel computing and has good expandability.