



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

병렬 연산 가속기와 여기 진폭 기법을
이용한 3차원 역시간 구조보정



2019년 8월

부경대학교 대학원

에너지자원공학과

이 종 우

공 학 석 사 학 위 논 문

병렬 연산 가속기와 여기 진폭 기법을
이용한 3차원 역시간 구조보정

지도교수 하 완 수

이 논문을 공학석사 학위논문으로 제출함

2019년 8월

부경대학교 대학원

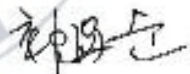
에너지자원공학과

이 종 우

이종우의 공학석사 학위논문을 인준함

2019년 8월



주 심 공학박사 최요순 

위 원 공학박사 하완수 

위 원 공학박사 손한암 

초 록

본 연구에서는 여기 진폭 기법과 병렬 컴퓨팅을 위한 고성능 가속기를 이용하여 3차원 음향파 시간 영역 역시간 구조보정 알고리즘을 개발하였다. 여기 진폭 기법은 매 시간마다 송신원 파동장과 수신기 파동장을 상호 상관하는 기존의 방법과 다르게 최대 진폭값을 갖는 파동장과 이에 대응되는 시간만을 사용한다. 따라서 여기 진폭 기법을 적용하여 전체 송신원 파동장을 저장하기 위해 필요한 메모리를 줄일 수 있다. 여기진폭기법을 이용한 역시간 구조보정 알고리즘을 가속화하기 위해 Graphical processing units(GPUs)과 제온파이 프로세서를 활용하였다. 모든 여기 진폭 및 여기 시간 자료들이 GPU 메모리와 제온 파이 프로세서의 고대역폭 메모리에 저장될 수 있기 때문에, 자료 데이터 통신을 줄이고 역시간 구조보정 프로그램의 성능을 향상시킬 수 있다. SEG/EAGE 3차원 속도 모델과 Overthrust 속도 모델을 통해 여기 진폭 기법을 이용한 역시간 구조 보정을 구현하여한 GPU 프로그램과 제온 파이 프로그램의 성능을 비교하여 GPU 프로그램의 계산 속도가 더 빠른 것을 알 수 있었다.

주요어 : 여기 진폭 기법, 역시간 구조 보정, 가속기, 병렬 연산

학 번 : 201755340

목 차

List of Figures	iii
List of Tables	v
1. 서 론	1
2. 여기 진폭 기법	4
2.1 상호상관을 이용한 역시간 구조보정	4
2.2 여기 진폭 기법을 이용한 역시간 구조보정	7
2.3 여기 진폭 기법 검증	8
3. 다중 CPU를 이용한 3차원 역시간 구조보정	15
4. 가속기를 이용한 3차원 역시간 구조보정	19
4.1 다중 GPGPU를 이용한 3차원 역시간 구조보정	19
4.1.1 GPGPU 프로세서	19
4.1.2 GPGPU 병렬 연산 시스템	21
4.1.3 수치 예제	23
4.2 다중 Xeon Phi를 이용한 3차원 역시간 구조보정	28
4.2.1 Xeon Phi 프로세서	28
4.2.2 Xeon Phi 병렬 연산 시스템	30
4.2.3 수치 예제	31
5. 성능 비교	34
6. 토의 및 결론	36
참고 문헌	38

List of Figures

Fig. 2.1 The Marmousi velocity model	10
Fig. 2.2 A shot gather from the Marmousi model	10
Fig. 2.3 A filter for muting and tapering	11
Fig. 2.4 A muted shot gather	11
Fig. 2.5 A smooth velocity model for migration	12
Fig. 2.6 A one-shot migration image using the crosscorrelation imaging condition	12
Fig. 2.7 A migration image obtained using the crosscorrelation imaging condition	12
Fig. 2.8 An excitation amplitude map	13
Fig. 2.9 A travelttime map for the excitation amplitude shown in Fig. 2.8	13
Fig. 2.10 A one-shot migration image using the excitation amplitude imaging condition	13
Fig. 2.11 A migration image obtained using the excitation amplitude imaging condition	14
Fig. 2.12 A one-shot migration image obtained using the excitation amplitude imaging condition and the data without preprocessing	14
Fig. 2.13 A migration image obtained using the excitation amplitude imaging condition and the data without preprocessing	14
Fig. 3.1 The SEG/EAGE 3d salt velocity model	16
Fig. 3.2 A smooth salt velocity model for migration	17
Fig. 3.3 Two one-shot migration images using the excitation amplitude	

imaging condition	18
Fig. 4.1 Server setting using two nodes and four GPU cards	22
Fig. 4.2 The SEG/EAGE 3d salt velocity model	25
Fig. 4.3 The migration image using the excitation amplitude imaging condition	25
Fig. 4.4 A section of the 72 nd common shot gather before and after preprocessing	26
Fig. 4.5 The 3d overthrust velocity model	27
Fig. 4.6 The migration image using the excitation amplitude imaging condition	27
Fig. 4.7 The migration image using the excitation amplitude imaging condition	33
Fig. 4.8 The migration image using the excitation amplitude imaging condition	33

List of Tables

Table 5.1 GPGPU and Xeon Phi benchmarking results	35
---	----



1. 서 론

과학 및 공학 기술 발전에 따라 인공 지능, 기계 학습, 빅데이터, 물리탐사, 입자 물리, 화학, 유체 역학 등 다양한 분야에서 대량의 데이터 처리와 그에 따른 수치 연산이 증가하고 있다.

석유 가스 탄성과 탐사 분야에서도 방대한 탐사 자료의 양과 자료 처리 알고리즘 고도화에 따라 정확한 지하 정보를 얻기 위한 계산이 갈수록 증가하고 있다. 육상과 해상에서의 3차원 및 4차원 탐사 자료 취득에 따라 탄성과 자료의 양이 크게 증가하고 있으며, 완전 파형 역산이나 역시간 구조보정과 같이 석유 가스 탐사 성공률 향상을 위한 자료 처리 기술이 확산되면서 컴퓨터 수치 계산량 또한 크게 증가하고 있다.

대량의 데이터를 처리하기 위해서는 컴퓨터 계산 자원이 상당히 많이 필요하다(Kim et al., 2016). 기존의 CPU는 코어 수가 부족하여 계산 능력 확장에 한계가 있기 때문에 현재 세계적으로 각각의 코어의 성능은 떨어지지만 CPU에 비해 훨씬 더 많은 코어를 탑재하여 병렬 연산을 할 수 있는 다양한 가속기들에 대한 연구가 활발히 진행되고 있다(Lee and Ha, 2018). 대표적으로 많이 사용되는 연산 가속기로는 범용 그래픽 처리 장치(General Purpose computing on Graphics Processing Units, GPGPU), 매니코어(Many Integrated Core, MIC) 구조의 Xeon Phi 등이 있다. GPGPU는 그래픽 프로세서의 고성능 그래픽 병렬 연산 능력을 일반 수치 연산에 사용할 수 있도록 하는 기술로, 기존의 서버에 GPGPU 프로그래밍이 가능한 그래픽 카드를 추가로 설치하여 개발 및 활용할 수 있다(Kim et al., 2016). 탄성과 탐사 분야에서도 3차원 구조보정이나 완전 파형 역산과 같이 많은 자료를 처리해야하는 기법들을 그래픽 카드의 수천 개의 코어를 이용한 병렬 연산을 통해 효율적으로 수행하는 사례가 많아지고 있다(Liu

et al., 2013; Shin et al., 2014). 매니 코어 구조의 제온 파이 프로세서는 GPU에 비해 코어 수는 적지만 개별 코어가 CPU의 코어와 유사하여 이식성이 좋은 가속기로서 탄성과 자료 처리에 많이 사용되고 있다(Arslan et al., 2014; Ryu et al., 2017; Tobin et al., 2017). 최근 개발된 제온 파이 x200 (Knight Landing) 프로세서는 이전 세대의 제온 파이 Knight Corner 보조 프로세서와 달리 시스템 부팅을 지원하고 내장 고대역폭 메모리 외에도 컴퓨터의 주 메모리에 직접 접근이 가능하다. 따라서 이전 세대의 PCIe 통신 제약을 없앴, 고성능 연산이 가능한 독립 실행형 프로세서이다.

본 연구에서는 GPGPU 기술과 MIC 기술을 이용하여 3차원 역시간 구조보정 기술을 개발하였다. 역시간 구조보정에 일반적으로 많이 사용하는 상호상관 Imaging condition을 적용하기 위해서는 매 시간마다 생성되는 파동장을 저장해야하기 때문에 대량의 메모리가 필요하다. 그렇게 될 경우엔 GPU와 CPU간의 데이터 이동이 많아져서 효율이 떨어지고 자료의 크기에 따라 오히려 CPU만 사용하는 경우보다 더 많은 시간이 걸릴 수 있다. 또한 호스트 디바이스 없이 독립 실행이 가능한 제온파이 x200 프로세서에서도 용량이 제한적인 고대역폭의 MCDRAM (Multi-Channel Dynamic Random-Access)의 활용도가 떨어져 성능 향상에 제약이 있다.

이러한 메모리 문제를 해결하기 위해 메모리 사용량을 크게 줄일 수 있는 여기 진폭 기법을 역시간 구조보정에 도입하였다(Nguyen and McMechan, 2012). 여기 진폭 기법은 매 시간마다 송신원 파동장과 수신기 파동장을 상호상관하여 이미지를 얻는 기존의 방법과 다르게 각 격자점에서 가장 진폭이 큰 값과 그 시간의 수신기 파동장을 사용한다. 따라서 여기 진폭 기법에서는 모든 송신원 파동장을 저장할 필요 없이 각 격자점마다 가장 큰 진폭값과 그 값을 가지는 샘플링 시간만을 저장하게 되어 필요한 메모리를 크게 줄일 수 있다.

본 연구에서는 먼저 여기 진폭 기법을 소개하고 2차원 수치예제를 통해 이를 검증한다. 이 후 가속기를 이용한 프로그램과의 성능 비교를 위하여 기존의 서버용 CPU를 사용한 3차원 역시간 구조보정 프로그램을 수행한다. 그리고 가속기를 효율적으로 사용하기 위해서 해당 프로세서들과 병렬화를 위한 시스템 구조에 대해 살펴보고자 한다. 다음으로 수치예제를 통해 GPU와 제온파이 프로세서에서 여기진폭기법을 적용한 3차원 역시간 구조보정 프로그램을 수행하여 CPU 프로그램 대비 성능 향상정도를 알아보고 각 가속기 프로그램간의 성능을 비교하였다.

본 연구를 통해 고성능 3차원 역시간 구조보정 계산 효율을 향상시켜 기존의 CPU 시스템 대비 생산성 향상 및 원가 절감을 기대할 수 있다. 또한 가속기를 이용한 병렬 처리 기술 연구를 통해 고성능 컴퓨팅 기술을 축적하여 다양한 탄성과 자료 처리 기법에 적용할 수 있을 것으로 기대된다.

2. 여기 진폭 기법

2.1 상호상관을 이용한 역시간 구조보정

본 연구에서는 역시간 구조보정을 수행하기 위해 필요한 송신원 파동장을 얻기 위해 음향 파동 방정식 모델링을 수행하였다. 3차원 시간 영역에 대한 음향 파동 방정식은 다음과 같다.

$$\frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + f \quad (1)$$

여기서 v 는 P파의 전파 속도, u 는 압력장, t 는 시간, f 는 송신원을 나타낸다. 식(1)의 파동 방정식을 수치 해석적으로 풀기 위해 미분항을 유한차분법으로 근사시켰으며 그 식은 다음과 같다.

$$u_{i,j,k}^{k+1} = 2u_{i,j,k}^k - u_{i,j,k}^{k-1} + v_{i,j,k}^2 (\Delta t)^2 \left(\frac{u_{i-1,j,k}^k + u_{i,j-1,k}^k + u_{i,j,k-1}^k - 6u_{i,j,k}^k + u_{i+1,j,k}^k + u_{i,j+1,k}^k + u_{i,j,k+1}^k}{h^2} + f_{i,j,k}^k \right) \quad (2)$$

여기서 첨자 k 는 시간 인덱스, Δt 는 시간 간격을 나타낸다. i, j, k 는 3차원 공간 인덱스이며 $\Delta x, \Delta y, \Delta z$ 는 공간격자 간격을 나타낸다. 따라서 과거와 현재의 파동장을 이용해 미래의 파동장을 계산하게 된다. 위 식을 이용하여 매 시간마다 루프를 돌며 모델링을 수행하여 송신원 파동장을 얻을 수 있다.

단일 송신원에 대한 역시간 구조보정 수행 시 일반적으로 영지연 상호상관 방법을 많이 사용하는데 다음과 같은 식으로 나타낼 수 있다

(Claerbout, 1971).

$$I(\vec{x}) = \sum_t^{T_{\max}} U(\vec{x}, t) D(\vec{x}, t) \quad (3)$$

여기서 D 는 송신원 파동장을 나타내는데 앞서 언급한 송신원 모델링을 통해 얻을 수 있다. U 는 수신기 파동장으로 송신원 모델링 시 수신기에 기록된 파를 역시간으로 전파하여 얻을 수 있다. 상호상관 방법을 사용할 경우 하부 영상의 진폭이 작아지기 때문에 필요에 따라 위 식을 송신원 파동장의 자기상관으로 나누어주기도 한다.

$$I(\vec{x}) = \frac{\sum_t^{T_{\max}} U(\vec{x}, t) D(\vec{x}, t)}{\sum_t^{T_{\max}} D(\vec{x}, t) D(\vec{x}, t)} \quad (4)$$

이러한 방법들의 경우 매 시간마다 관측자료를 역전파하여 얻은 수신기 파동장과 기존의 송신원 파동장의 상호상관을 계산해야 하는데 이를 위해서 모든 송신원 파동장을 저장하고 있어야 한다. 송신원 파동장의 크기는 단정도 실수를 사용할 경우 $4 \times N \times NT$ 바이트를 차지하게 되는데, 여기서 N 은 수치 모델의 전체 격자 수, NT 는 모델링을 위한 시간 샘플 개수이다. 2차원의 경우 메모리 사용량이 크지 않지만 3차원의 경우 $N = 500 \times 500 \times 500$, $NT = 5000$ 이면 메모리 사용량이 2 TB를 넘어가게 된다. 일반적인 서버 컴퓨터에서 이러한 양의 데이터를 메인 메모리상에 저장할 수 없기 때문에 하드디스크를 사용하게 되는데, 메인 메모리에 비해

속도가 크게 느린 하드디스크를 사용하게 되면 입출력 병목 현상으로 인해
역시간 구조조정 계산 시간이 느려지게 된다.



2.2 여기 진폭 기법을 이용한 역시간 구조보정

일반적인 상호상관 방법을 사용했을 때 생기는 메모리 및 데이터 입출력 문제를 해결하기 위해 송신원 파동장의 일부만 사용하는 방법, 송신원 파동장의 일부만 저장해 놓았다가 필요한 부분만 다시 계산하는 방법 등이 제안되었다(Symes, 2007; Dussaud et al., 2008; Sun and Fu, 2013). 이러한 방법들 중 본 연구에서는 여기 진폭 기법(Excitation Amplitude)을 도입하여 메모리 사용량을 크게 줄이고자 한다(Nguyen and McMechan, 2012; Nguyen and McMechan, 2014). 여기 진폭 기법의 Imaging condition은 다음과 같다.

$$I(\vec{x}) = \frac{U(\vec{x}, t_E)}{D_{\max}(\vec{x}, t_E)} \quad (5)$$

위 식에서 $D_{\max}$ 는 송신원 파동장의 모든 격자점 상에서 진폭이 최대가 되는 시간 샘플을 의미하고, t_E 는 그 때의 시간을 의미한다. 여기 진폭 기법을 이용할 경우 송신원 파동장의 최대 진폭과 해당 시간을 기록해 두었다가 역전파시 해당 시간의 수신기 파동장을 송신원 파동장의 최대 진폭으로 나누어 주게 된다. 따라서 단정도 실수를 사용할 경우 메모리 사용량은 $2 \times 4 \times N$ 바이트로 줄어들게 되는데, 최대 진폭과 시간을 기록하기 위해 전체 격자의 두 배 만큼의 메모리 공간을 사용하게 된다. 메모리 사용량은 시간 샘플 개수와 상관없이 전체 격자 수에만 비례하게 되고, $N = 500 \times 500 \times 500$ 이면 1 GB 미만의 메모리만 사용하게 된다.

2.3 여기 진폭 기법 검증

여기 진폭 기법 검증을 위해 2차원 Marmousi 속도모델을 이용하여 역시간 구조보정을 수행하였다(Fig. 2.1). 수치 모델에는 16 m 격자를 사용하였고, 속도 모델 위에 물 층을 추가하여 총 격자 수는 576×219 이다. 샘플링 간격을 1 ms로 해서 총 4초간 모델링을 수행하여 180개의 공통송신원모음 자료를 생성하였다(Fig. 2.2). 구조보정 잡음을 줄이기 위해 관측 자료를 그대로 사용하지 않고 최대 오프셋을 제한하였으며 직접파를 제거하였다(Figs. 2.3, 2.4)(Nguyen and McMechan, 2012).

역시간 구조보정을 위해서는 평활화한 속도모델을 사용하였다(Fig. 2.5). 상호상관 Imaging condition을 사용할 경우 송신원 파동장을 저장하기 위해 총 1924.8 MB의 메모리를 사용하였다. Figs. 2.6, 2.7은 상호상관 Imaging condition을 사용했을 때 하나의 송신원으로부터 얻은 영상과 전체 송신원으로부터 얻은 영상이다. 구조보정 영상이 실제 모델을 잘 반영하고 있음을 알 수 있다.

동일한 조건으로 여기 진폭 기법을 이용해 역시간 구조보정을 수행하였다. 여기 진폭 Imaging condition을 사용할 경우 송신원 파동장의 최대값과 해당 시간을 저장하기 위해 사용한 메모리는 총 1 MB가 안 되었다. Figs. 2.8, 2.9에 여기 진폭과 해당 시간을 도시하였다. 여기 진폭 Imaging condition에 따라 역전파시 수신기 파동장의 각 격자점에서 Fig. 2.9에 해당하는 시간의 파동장을 취하여 Fig. 2.8의 여기 진폭으로 나누어 주었다. 그 결과로 얻은 단일 송신원 구조보정 영상과(Fig. 2.10) 전체 송신원 구조보정 영상(Fig. 2.11)은 Marmousi 모델의 구조를 잘 나타내고 있다. 상호상관 Imaging condition을 사용한 결과(Fig. 2.7)와 여기 진폭 Imaging condition을 사용한 결과(Fig. 2.11)를 비교해보면 여기 진폭을 이용한 경우

구조보정 잡음이 좀 더 많지만 배경의 저주파 잡음은 상호상관을 이용한 경우에 더 많음을 알 수 있다.

본 수치 실험에서는 구조보정 잡음을 줄이기 위해 관측 자료에 선처리를 수행하였다(Figs. 2.3, 2.4). 비교를 위해 선처리를 수행하지 않을 경우의 여기 진폭을 이용한 구조보정 영상 또한 확인하였다. 그 경우 송신원 위치에서 먼 곳에서 여기 진폭 시간의 불연속으로 인해 단일 송신원 구조보정 영상 또한 불연속적인 면을 가지게 되고(Fig. 2.12), 전체 송신원 영상에도 불연속 면이 나타나게 된다(Fig. 2.13). 따라서 여기 진폭 기법을 이용할 경우 선처리를 통해 이러한 잡음을 미리 없애주는 과정이 필요함을 알 수 있다.



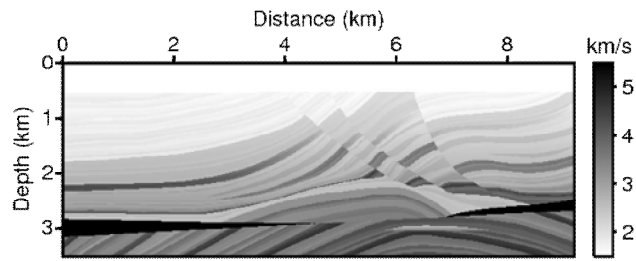


Fig. 2.1. The Marmousi velocity model.

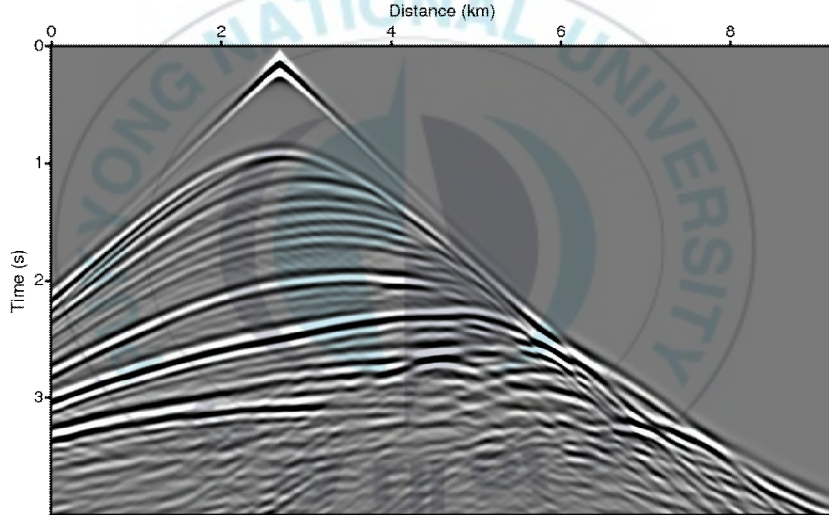


Fig. 2.2. A shot gather from the Marmousi model.

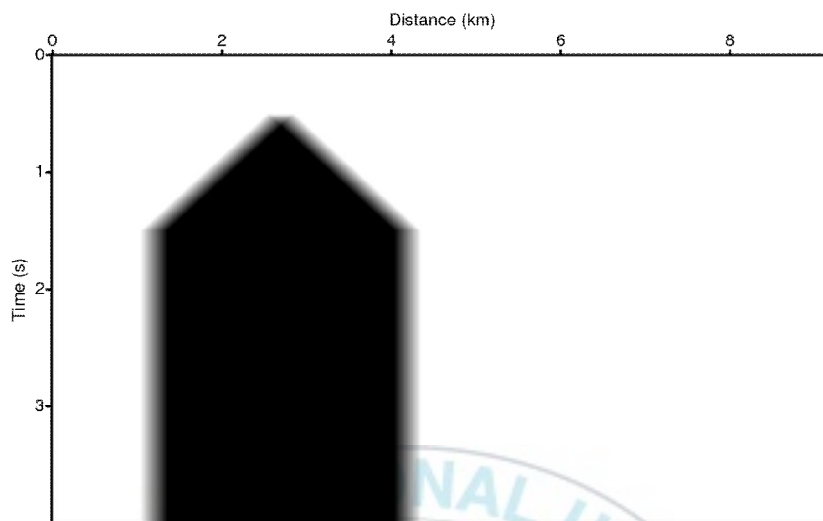


Fig. 2.3. A filter for muting and tapering.

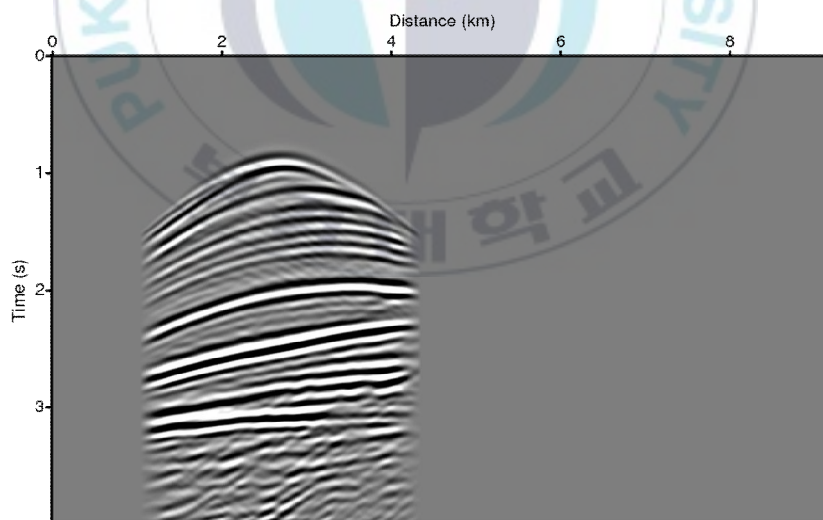


Fig. 2.4. A muted shot gather.

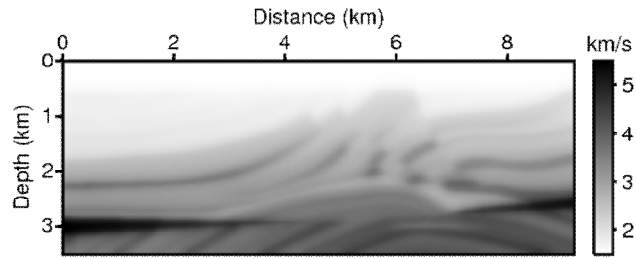


Fig. 2.5. A smooth velocity model for migration.

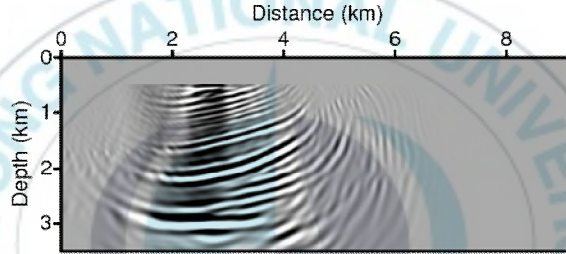


Fig. 2.6. A one-shot migration image using the crosscorrelation imaging condition.

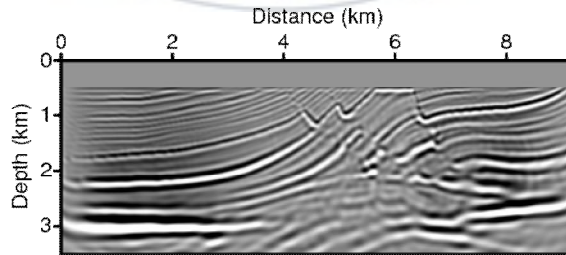


Fig. 2.7. A migration image obtained using the crosscorrelation imaging condition.

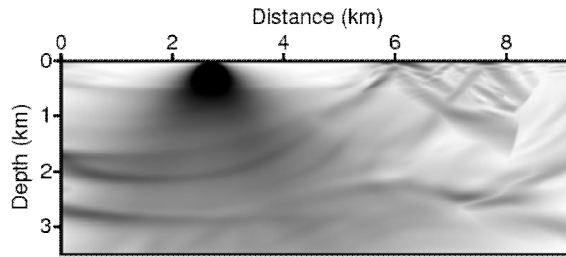


Fig. 2.8. An excitation amplitude map.



Fig. 2.9. A traveltide map for the excitation amplitude shown in Fig. 2.8.

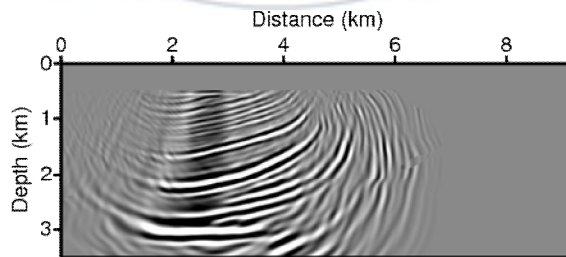


Fig. 2.10. A one-shot migration image using the excitation amplitude imaging condition.

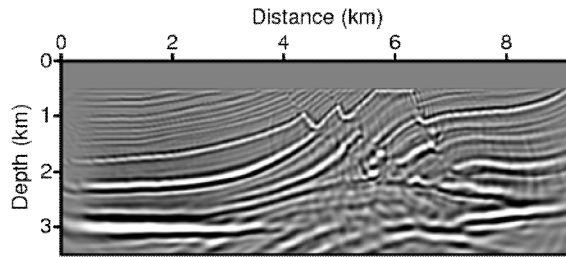


Fig. 2.11. A migration image obtained using the excitation amplitude imaging condition.

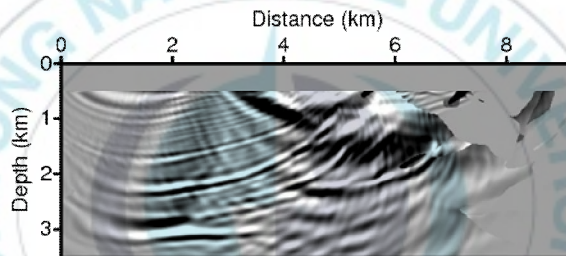


Fig. 2.12. A one-shot migration image obtained using the excitation amplitude imaging condition and the data without preprocessing.

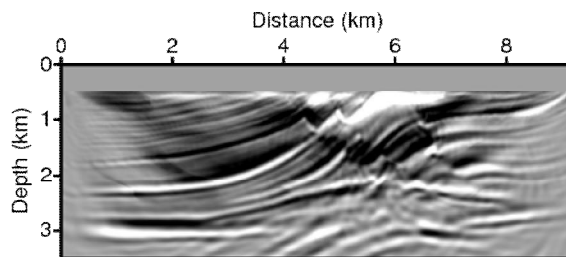


Fig. 2.13. A migration image obtained using the excitation amplitude imaging condition and the data without preprocessing.

3. 다중 CPU를 이용한 3차원 역시간 구조보정

앞에서 살펴본 여기 진폭 기법을 이용하여 다중 CPU를 사용하는 3차원 역시간 구조보정 프로그램을 작성하였다. 다중 CPU 병렬화는 Message Passing Interface (MPI)를 이용하여 구현하였다. 프로그램 확인을 위해서 3차원 SEG/EAGE 암염돔 속도모델을 사용하였다(Fig. 3.1). 속도모델의 크기는 $676 \times 676 \times 201$, 격자 크기는 20 m이다. 관측 자료는 2 ms의 샘플링 간격으로 총 5초간 기록하였다. 2차원 예제에서와 같이 전처리 후의 자료를 사용하였다. 상호상관 Imaging condition을 이용할 경우 전체 영역에서 송신 파형을 저장하려면 855 GB의 저장 공간이 필요한 반면 여기 진폭 기법을 이용하면 701 MB의 저장 공간만을 사용하게 된다.

Intel Xeon E7-4850 (2.3 GHz) CPU를 이용하여 시험해본 결과, 공간에 대한 2차 유한 차분식을 사용하더라도 단일 코어를 이용한 단일 송신원 모델링에만 1시간이 넘게 걸렸다. 역전파까지 수행하여 하나의 송신원으로부터 구조보정 영상을 생성하는 데에는 2시간 이상이 걸렸다. 서버 성능을 고려하여 CPU를 이용한 실험에서 많은 수의 송신원을 이용한 영상을 구하지는 않았다. Fig. 3.2는 역시간 구조보정에 사용한 평활화된 속도모델, Fig. 3.3의 영상들은 단일 송신원을 이용하였을 때의 구조보정 영상들이다.

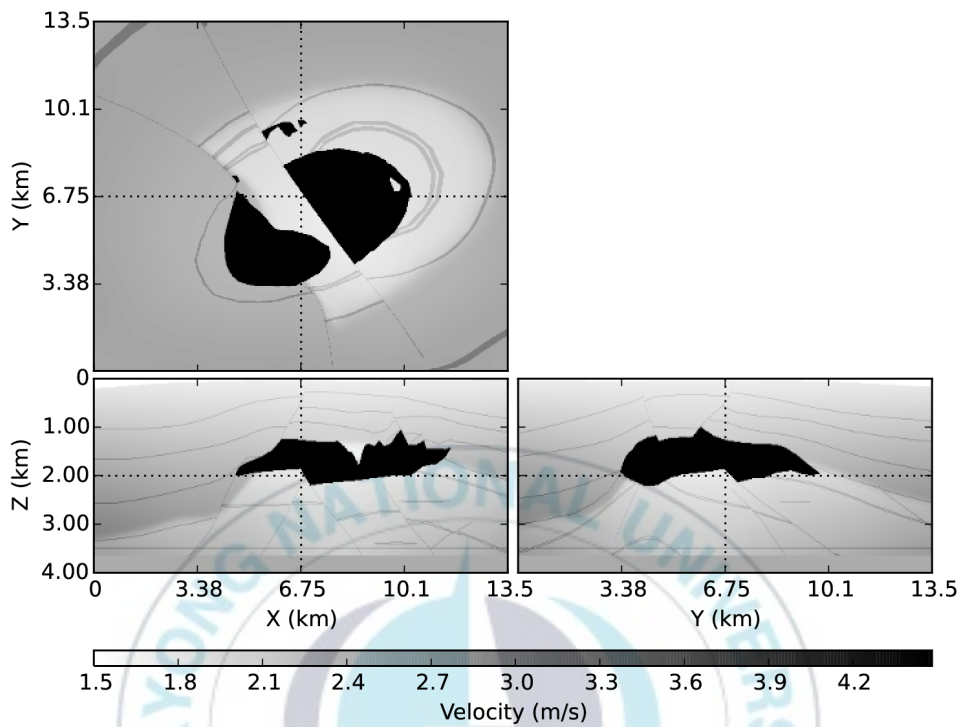


Fig. 3.1. The SEG/EAGE 3d salt velocity model.

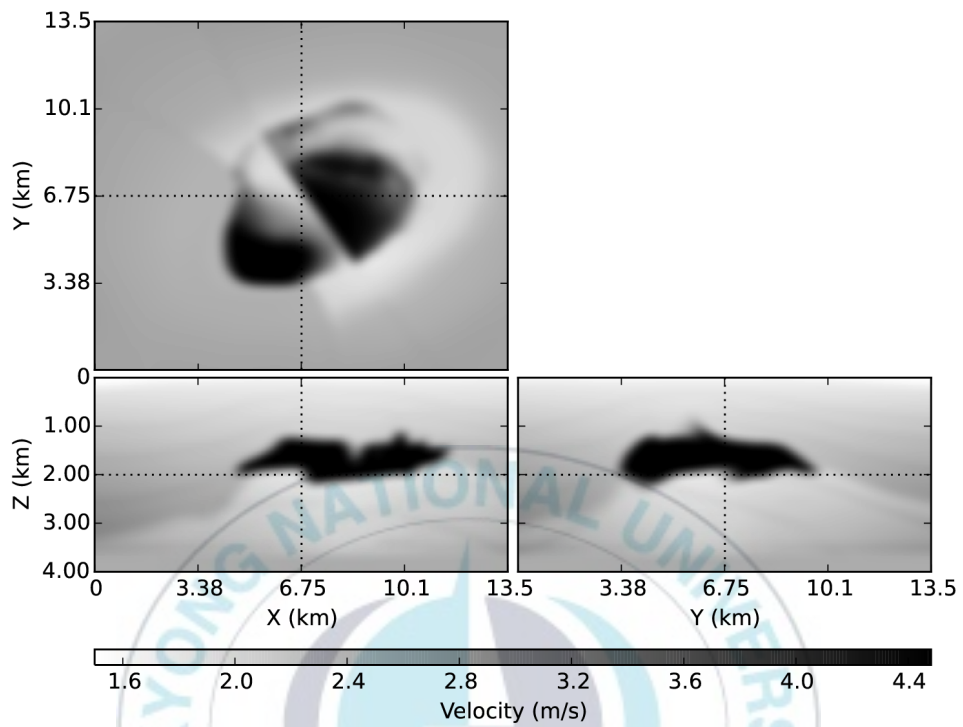


Fig. 3.2. A smooth salt velocity model for migration.

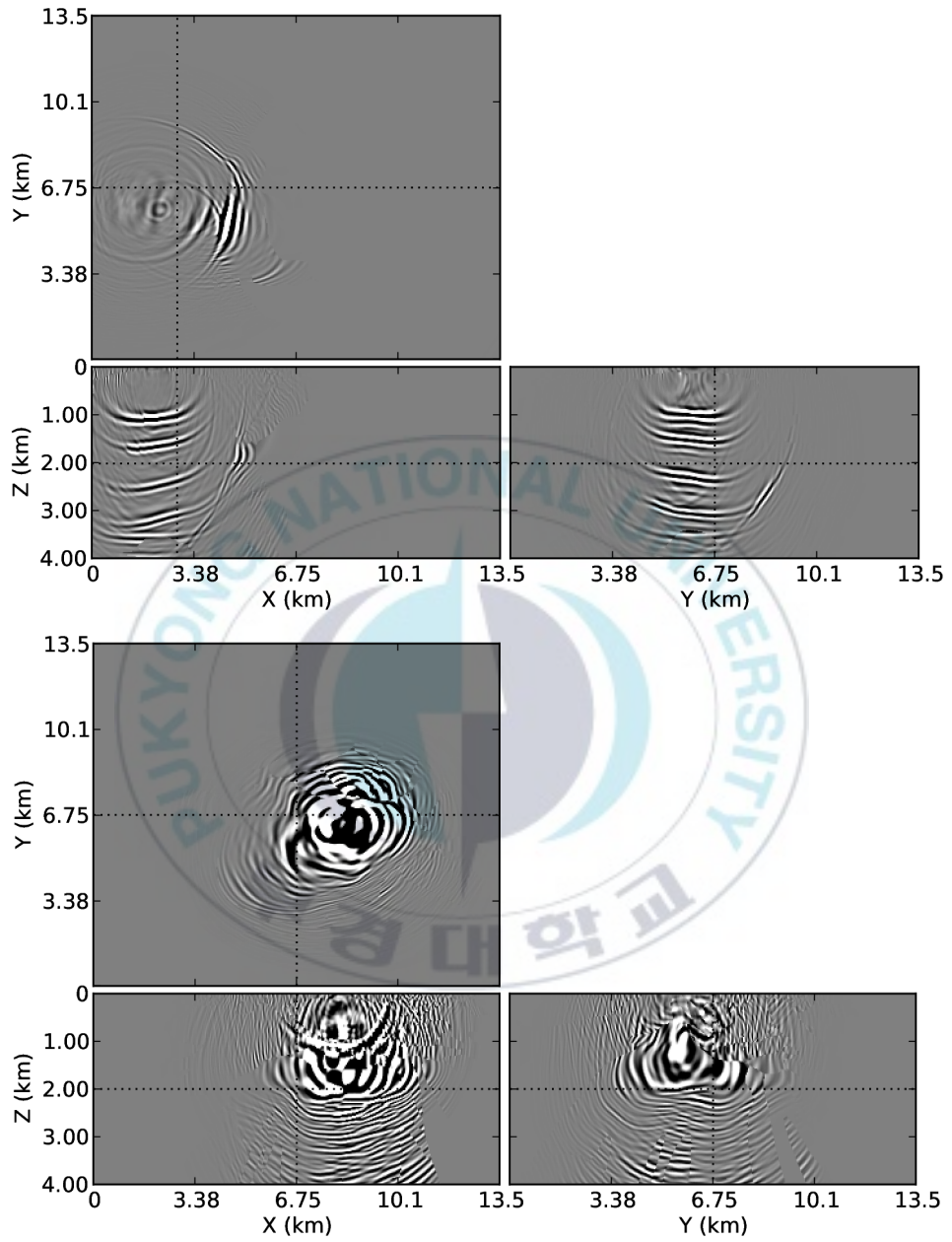


Fig. 3.3. Two one-shot migration images using the excitation amplitude imaging condition.

4. 가속기를 이용한 3차원 역시간 구조보정

4.1 다중 GPGPU를 이용한 3차원 역시간 구조보정

4.1.1 GPGPU 프로세서

GPGPU는 일반적으로 컴퓨터 그래픽스를 위한 계산에 이용되던 GPU를 일반 수치 연산에 활용하는 기술이다. 이는 기존의 서버에 GPGPU 프로그래밍 용 그래픽 카드와 컴파일러를 설치하고 CUDA, OpenCL, OpenACC 등의 GPGPU 프로그래밍 언어를 통해 사용할 수 있다(Kim et al., 2016). 기존에 수치 연산에 사용되던 중앙 처리 장치(Central Processing Unit, CPU)와 비교하여 수천 개의 코어를 활용한 빠른 계산, 성능 대비 낮은 가격, 적은 전력 소모, 작은 공간 사용 등이 가능하다는 장점이 있다. 이를 바탕으로 생명 과학, 의료 장치, 제조, 인공지능 등 다양한 분야에서 활용되고 있다(LIU et al., 2010). 특히 석유·가스 탐사 분야에서도 탄성과 자료 처리 기법의 기본이 되는 파동 전파 모델링부터 많은 계산이 필요한 역시간 구조보정, 완전파형역산 등의 고영상화 기법 연구에 활용하고 있다(Kim et al., 2016).

GPU를 활용하여 역시간 구조보정을 수행할 때 문제가 되는 것은 지하 영상을 얻기 위한 계산시간 보다 호스트 프로세서와 디바이스 프로세서 사이의 데이터 이동시간이 훨씬 길다는 것이다(Yang et al., 2014). 먼저 GPU는 스스로 메모리 공간을 할당할 수 없기 때문에 호스트 프로세서로부터 메모리를 할당하고 자료를 입·출력하는 작업이 필수적이다. 또한 일반적으로 많이 사용하는 상호상관 방법은 송신원 파동장을 모두 저장하고 있어야하는데 3차원 자료의 경우 그 양이 매우 많기 때문에 GPU의 메모

리로는 모두 저장할 수 없다. 따라서 프로그램을 실행하면서 CPU와의 데이터 통신을 하게 되는데 GPU의 메모리 대역폭에 비해 CPU와의 통신 대역폭이 현저히 낮기 때문에 실제 GPU에서 계산을 하는데 소요되는 시간보다 훨씬 더 많은 시간이 걸리게 된다. 그러므로 역시간 구조보정 수행 시 GPU를 효율적으로 활용하기 위해서는 프로그램 실행 시 필요한 메모리를 줄여 GPU의 메모리를 최대한 사용하는 것이 중요하다. 따라서 본 연구에서는 메모리 사용량을 줄일 수 있는 여기 진폭 기법을 역시간 구조 보정에 적용하여 수치 예제를 수행하였다.



4.1.2 GPGPU 병렬 연산 시스템

역시간 구조조정 수행 시 더 정확한 영상을 얻기 위해서 다중 송신원을 사용하게 된다. 다중 송신원을 이용할 경우 각각의 송신원에서의 구조보정을 수행하고 마지막에 그 결과를 중합하여 최종 영상을 얻을 수 있다. 따라서 CPU를 병렬화하여 각각의 CPU들이 각각의 송신원 구조보정을 수행할 수 있도록 하는 것이 효율적이다. 각각의 CPU들이 각각의 송신원 구조보정을 독립적으로 수행하고 최종 결과만 합치게 되면 프로세서 간 통신량을 최소화할 수 있다. GPGPU를 이용한 병렬화 또한 같은 방식으로 구현하는 것이 효율적이다. 연산 시간의 대부분을 차지하는 파동방정식 유한차분법 모델링 부분만 GPU로 구현하고 프로그램의 나머지 부분은 다중 CPU를 사용하는 경우와 동일한 구조를 유지할 수 있다. 연산에 사용한 시스템은 두 개의 노드에 걸쳐 총 네 장의 GPU 카드를 장착한 시스템으로, 송신원 병렬화를 위해 MPI를 이용하는 동시에 GPU 병렬 연산을 위해서는 Nvidia사의 Compute Unified Device Architecture (CUDA) 프로그래밍 언어를 이용하였다(Fig. 4.1).

본 연구에 사용한 GPU 카드는 Nvidia사의 Titan Xp이다. 해당 모델의 GPU 코어 수는 3840개, 내장 메모리는 총 12 GB이다. 이와 같은 방식으로 다중 GPU를 이용하는 역시간 구조조정 모듈을 개발하여 3차원 SEG/EAGE 암염동 모델과 Overthrust 모델에 적용하였다.

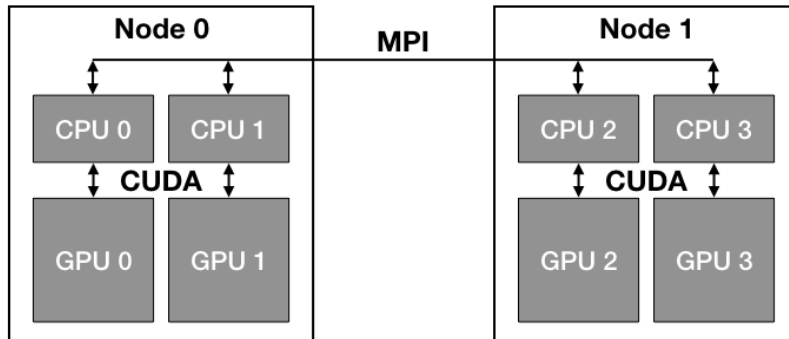


Fig. 4.1. Server setting using two nodes and four GPU cards.



4.1.3 수치 예제

앞에서와 같이 3차원 SEG/EAGE 암염동 속도모델의 크기는 $676 \times 676 \times 201$, 격자 크기는 20 m이다. 관측 자료는 2 ms의 샘플링 간격으로 총 6초간 기록하였다. 송신원은 $y = 6$ km 지점의 한 라인을 따라 100 m 간격으로 135개를 생성하였고, 역시간 구조보정에 사용하는 자료의 최대 오프셋은 2.0 km로 제한하였다. 2차원 예제에서와 같이 관측 자료에 전처리를 적용하였다. 구조보정에 들어가는 모델링 시 공간 차분식의 차수는 8차식을 사용하였다. 경계면으로 인한 반사파들의 영향을 줄이기 위해 Higdon 경계조건을 적용하였다(Higdon, 1986). Fig. 4.1의 시스템을 이용하여 연산한 결과, 135개 송신원을 이용한 역시간 구조보정에 총 43분 51초가 걸렸다. 단일 송신원 연산에 약 79초가 걸린 것으로, 더 적은 수의 샘플을 이용했던 CPU 연산에 비해 상당히 큰 속도 향상이 있음을 알 수 있다. Fig. 4.2은 역시간 구조보정에 사용한 실제 속도모델, Fig. 4.3는 한 라인의 송신원 135개를 이용한 구조보정 결과이다. 해당 라인의 구조보정 영상 단면이 잘 나오는 것을 확인할 수 있다.

이번에는 3차원 Overthrust 속도모델을 이용해 역시간 구조보정을 수행하였다. 모델 크기는 $801 \times 801 \times 187$, 격자 크기는 25 m이다. 관측 자료는 1.5 ms 샘플링 간격으로 총 6초간 기록하였다. 공간에 대한 유한 차분식은 4차식을 사용하였으며 앞서와 마찬가지로 Higdon 경계조건을 적용하였다(Higdon, 1986). 송신원은 $y = 10$ km 지점의 한 라인을 따라 200 m 간격으로 98개를 생성하였다. 구조보정을 위한 자료의 최대 오프셋은 2.0 km로 제한하였다. Fig. 4.4에 전처리 전후의 공통송신원모음을 도시하였다. 앞서와 같은 장비를 이용하여 연산한 결과 총 46분 54초가 걸렸고, 단일 송신원 연산에 평균적으로 약 112초가 걸렸다. Fig. 4.5은 역시간 구조보정에

사용한 속도모델, Fig. 4.6는 한 라인의 송신원 98개를 이용한 구조보정 결과이다. 역시 해당 라인의 영상 단면이 제대로 나오는 것을 확인할 수 있다.



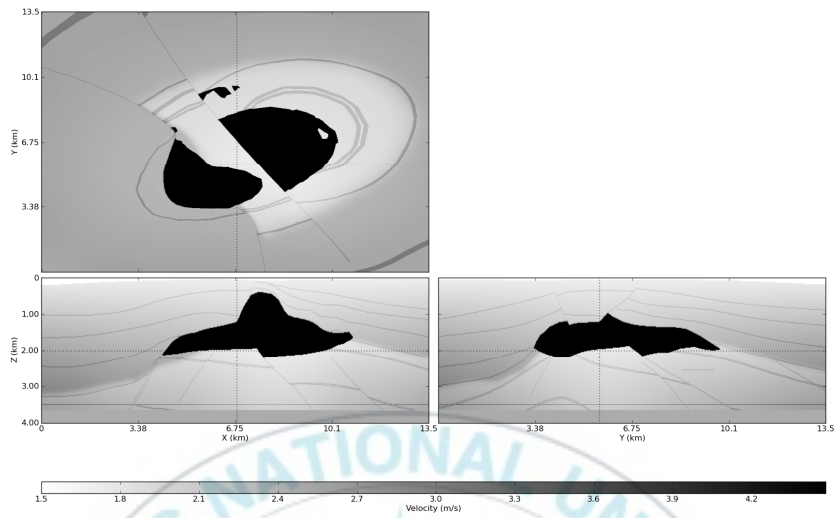


Fig. 4.2. The SEG/EAGE 3d salt velocity model.

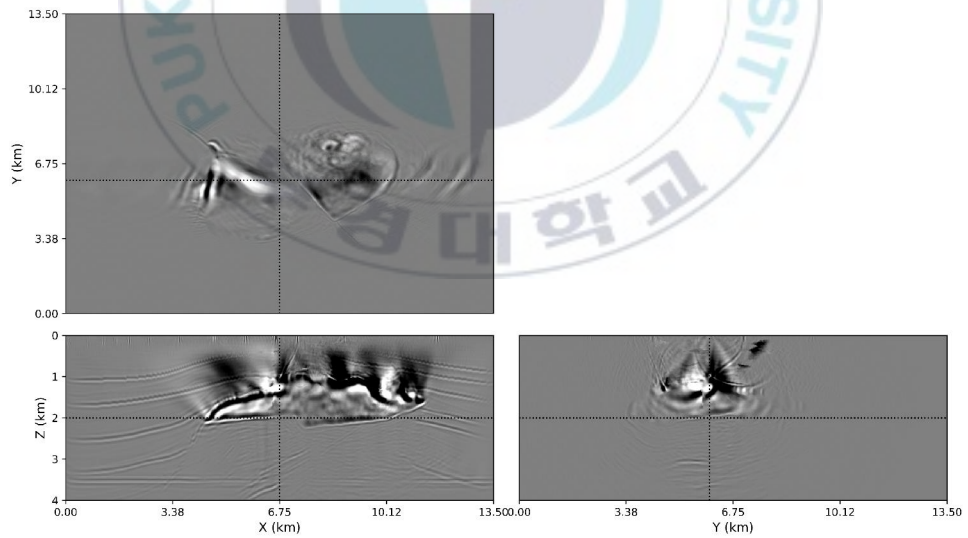


Fig. 4.3. The migration image using the excitation amplitude imaging condition.

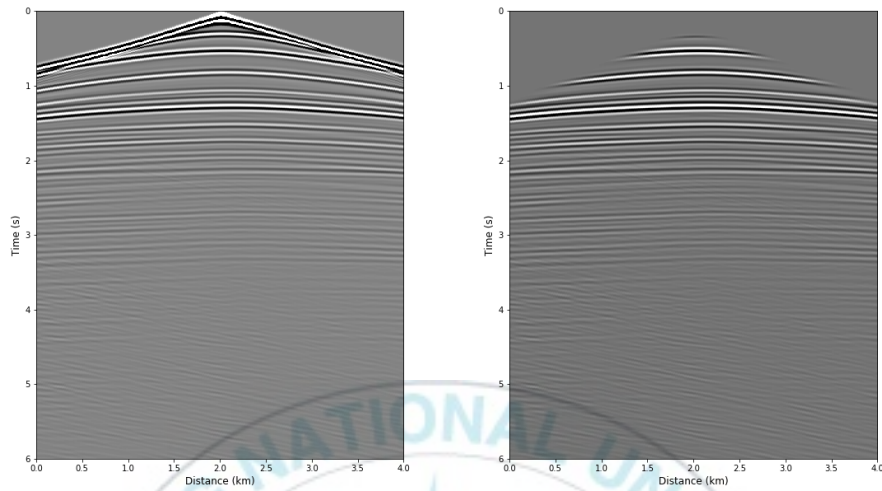


Fig. 4.4. A section of the 72nd common shot gather before and after preprocessing.

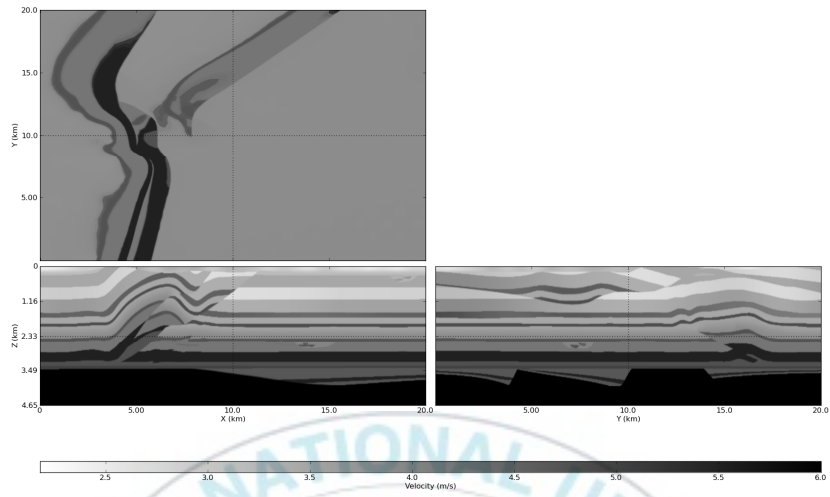


Fig. 4.5. The 3d overthrust velocity model.

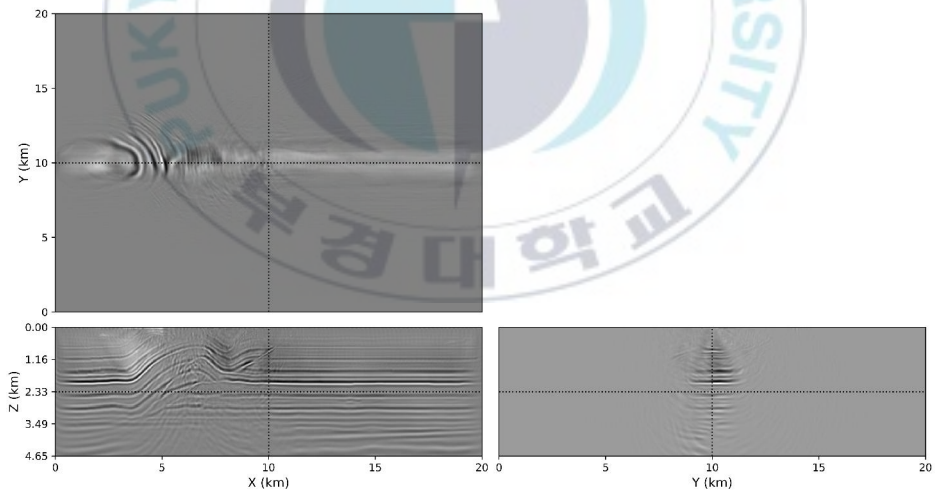


Fig. 4.6. The migration image using the excitation amplitude imaging condition.

4.2 다중 Xeon Phi를 이용한 3차원 역시간 구조보정

4.2.1 Xeon Phi 프로세서

Intel사의 Xeon Phi 프로세서는 하나의 칩에 수십 개 이상의 코어를 집적하는 매니코어(Many Integrated Core, MIC) 구조의 고성능 병렬 연산용 프로세서이다. 수치 연산만이 가능한 GPU의 코어들에 비해 수는 적지만 x86 코어들로 이루어져 각각의 코어가 높은 성능을 보일뿐 아니라 기존에 CPU에 사용하던 프로그램을 큰 수정 없이도 실행할 수 있는 장점이 있다.

본 연구에서는 Xeon Phi 2세대 프로세서인 x200 (Knights Landing) 모델을 사용하였다. 이전 세대의 Knights Corner 모델은 호스트 프로세서 없이 스스로 운영체제 부팅이 불가능한 보조 프로세서였다(Ryu et al., 2017). 이러한 불편함을 해소하고자 개발된 Knights Landing 모델은 PCIe 통신 제약 없이 스스로 운영체제에서 부팅이 가능한 독립형 프로세서이다. 필요에 따라서는 PCIe 연결 기반 보조프로세서로도 활용이 가능하다. 또한 이전의 Xeon Phi 프로세서들과 2진 호환이 있어서 이전 프로세서들에서 개발한 소프트웨어를 Knight Landing에서도 실행할 수 있다(Sodani et al., 2016).

Knights Landing 프로세서는 대용량의 DDR과 고대역폭의 MCDRAM(high bandwidth memory)의 두 가지 유형의 메모리가 있다. 이와 같은 두 가지 메모리를 이용해 Knights Landing 프로세서가 독립적으로 많은 자료를 빠르게 처리할 수 있다. DDR 메모리는 일반서버의 주 메모리에 해당되며 패키지 외부에서 최대 384 GB의 대용량 메모리를 제공한다. 하지만 최대 대역폭은 약 90 GB/s 정도로 대규모 컴퓨팅을 수행하기엔 부족하다. 이런 문제를 해결하기 위해 MCDRAM이 패키지 내부에 통

합되어 있다. 총 용량은 16 GB이며 최대 대역폭은 450 GB/s 이상으로 대규모 컴퓨팅에 적합한 굉장히 빠른 자료 전송이 가능하다(Jeffers et al., 2016). 따라서 Knights Landing 프로세서에서 역시간 구조보정을 효율적으로 수행하기 위해서 앞서 GPU를 활용한 예제처럼 필요한 메모리를 줄여 고대역폭의 MCDRAM만 이용한다면 최적화된 성능을 얻을 수 있을 것이다. 이를 위해 앞선 예제와 동일하게 여기 진폭 기법을 역시간 구조 보정에 적용하여 수치 예제를 수행하였다.



4.2.2 Xeon Phi 병렬 연산 시스템

앞서 다중 GPGPU를 사용한 예제와 같이 Xeon Phi x200 프로세서들이 각각의 송신원 구조보정을 수행할 수 있도록 시스템을 구성하였다. GPGPU와 달리 Xeon Phi 프로세서는 호스트 프로세서 없이 독립적으로 사용할 수 있기 때문에 프로세서간의 통신을 위해 MPI를 이용하여 네 개의 노드를 직접 병렬화 하였다. 또한 최적의 성능을 얻기 위해서 각 노드의 MCDRAM만을 사용하였다. 고대역폭의 MCDRAM은 노드 당 16GB의 메모리를 가지기 때문에 역시간 구조보정에 필요한 메모리를 고려하여 하나에 노드에 여러 개의 MPI 작업을 할당할 수 있다. 하지만 각 노드당 MPI 작업이 늘어나면 각 MPI 작업에 사용되는 코어 수가 줄기 때문에 이를 고려하여야 한다. 예를 들어, 앞선 수치 예제에서 사용한 3차원 암염동 모델에서 단정도 연산으로 역시간 구조보정을 수행하기 위해서는 약 4 GB의 메모리가 필요하다. 따라서 각 프로세서 당 최대 4개의 MPI 작업을 할당할 수 있다. 이 경우 총 64개의 코어를 4개의 MPI 작업에서 사용해야하기 때문에 각 작업은 16개의 코어를 사용하게 된다. MPI 작업을 나누고 할당된 코어들을 병렬화하여 쓰레드로 이용하기 위해서 OpenMP를 사용하였다. Xeon Phi x200 프로세서는 독립적으로 모든 작업을 실행하기 때문에 유한 차분법 모델링 부분뿐만 아니라 모든 과정들이 프로세서 내의 코어에서 진행된다.

본 연구에서 사용한 Xeon Phi x200 (Knights Landing) 모델의 CPU 코어 수는 64개, CPU 클럭은 1.30 Ghz, L2 캐시는 32 MB이다. 이와 같은 방식으로 다중 XeonPhi를 이용하는 역시간 구보조정 모듈을 개발하여 3차원 SEG/EAGE 암염동 모델과 Overthrust 모델에 적용하였다.

4.2.3 수치 예제

다중 Xeon Phi 프로세서와 여기 진폭 기법을 적용한 3차원 역시간 구조보정 프로그램을 개발하였다. 수치 예제에서는 MPI와 OpenMP를 동시에 사용하여 네 개의 노드에 걸쳐 병렬 연산을 수행하였다.

첫 번째 수치 예제에서는 SEG/EAGE 3차원 암염동 속도모델을 사용하였다(Fig. 4.2). 속도모델의 크기는 $676 \times 676 \times 201$, 격자 크기는 20 m이다. 관측 자료는 2 ms 샘플링 간격으로 총 6초간 기록하였다. 송신원은 $y=6$ km 지점의 라인을 따라 100 m 간격으로 135개를 생성하였다. 최대 오프셋은 2.0 km로 제한하였다. 역시간 구조보정은 공간에 대해 8차 유한차분식을 사용하였고 Keys 경계조건을 적용하여 수행하였다(Keys, 1985). 네 개의 노드에 각각 하나의 MPI 작업을 할당하고 노드별로 64개의 OpenMP 쓰레드를 사용한 경우 역시간 구조보정 시간은 총 84분 53초, 각 노드에 두 개의 MPI 작업을 할당하고 노드별로 32개의 OpenMP 쓰레드를 사용한 경우 역시간 구조보정 시간은 총 79분 4초가 걸렸다. Fig. 4.7은 한 라인의 송신원 135개를 이용한 구조보정 결과이다. 해당 라인의 구조보정 영상 단면이 잘 나오는 것을 확인할 수 있다.

두 번째 수치 예제에서는 3차원 Overthrust 모델을 사용하였다(Fig. 4.5). 모델 크기는 $801 \times 801 \times 187$, 격자 크기는 25 m이다. 송신원은 $y=10$ km 지점의 한 라인을 따라 200 m 간격으로 98개를 생성하였다. 구조보정을 위한 최대 오프셋은 2.0 km로 제한하였다. 관측 자료는 1.5 ms 샘플링 간격으로 총 6초간 기록하였다. 역시간 구조보정은 공간에 대해 4차 유한차분식을 사용하였고 Keys 경계조건을 적용하여 수행하였다(Keys, 1985). 앞 예제의 결과를 바탕으로 네 개의 노드에 각각 두 개의 MPI 작업을 할당하고 노드별로 32개의 OpenMP 쓰레드를 사용한 결과, 역시간 구조보정에

총 78분 34초가 걸렸다. Fig 4.8은 해당라인 구조보정 결과로 영상 단면이 잘 나온 것을 확인할 수 있다.



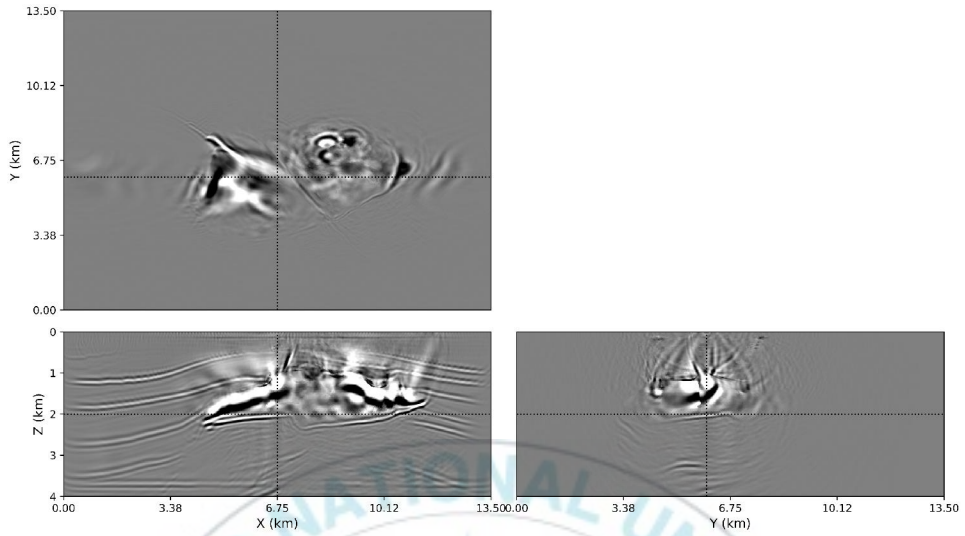


Fig. 4.7. The migration image using the excitation amplitude imaging condition.

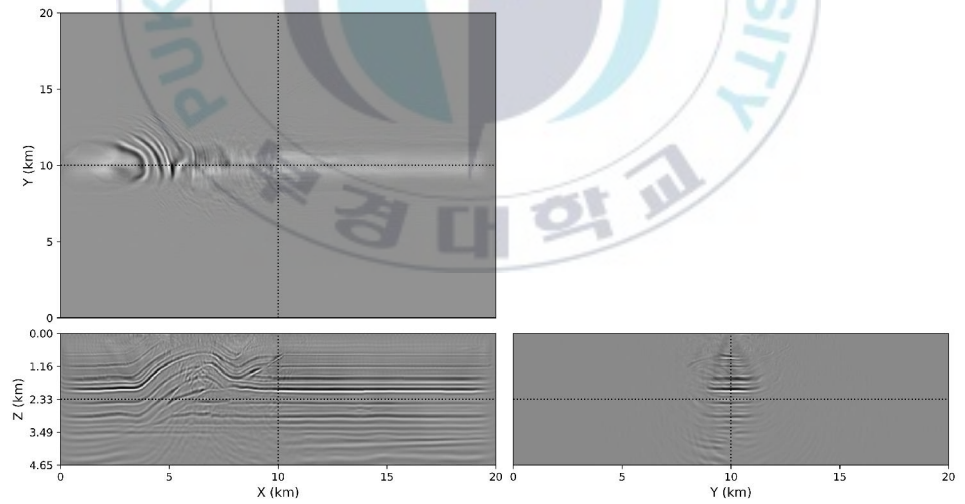


Fig. 4.8. The migration image using the excitation amplitude imaging condition.

5. 성능 비교

수치 예제를 통해 다중 GPGPU를 이용한 3차원 역시간 구조보정 프로그램과 다중 Xeon Phi 프로세서를 이용한 3차원 역시간 구조보정 프로그램의 성능을 비교하였다. 두 프로그램의 역시간 구조보정 알고리즘은 여기진폭 방법으로 동일하지만, 모델링 부분은 각 프로세서에 맞게 서로 다른 알고리즘으로 작성되어 있다. 성능 비교에 사용한 GPGPU는 Nvidia사의 Titan Xp 모델, Xeon Phi는 Intel사의 Xeon Phi x200 (Knights Landing) 7210 모델이다. 본 연구에서는 연구실에서 보유한 장비를 이용하여 성능을 비교하였으나, 성능 비교 결과는 사용하는 모델의 사양에 따라 달라질 수 있다. GPGPU 4장과 Xeon Phi 네 개의 노드를 사용해서 앞의 예제에서의 역시간 구조보정 결과를 Table 5.1에 제시하였다.

성능 비교 결과 실험에 사용한 Xeon Phi x200 7210 모델보다는 Titan Xp 모델의 성능이 우수함을 알 수 있었다. SEG/EAGE 암염돔 모델은 8차 유한 차분식, Overthrust 모델은 4차 유한 차분식을 사용한 결과로, GPGPU에 비해 Xeon Phi에서 유한 차분식 차수가 높아질수록 연산 시간이 더 오래 걸리는 것을 확인할 수 있다. 단, Titan Xp 모델의 단정도 연산 속도는 10 TFlops가 넘는 반면에 배정도 연산 속도는 0.4 TFlops 미만으로, 배정도 연산을 사용할 경우 Xeon Phi 또는 다른 GPU를 사용할 필요가 있다.

Table 5.1. GPGPU and Xeon Phi benchmarking results.

Model	GPGPU (s)	Xeon Phi (s)	Speedup
SEG/EAGE Salt	2571	4744	1.845
Overthrust	2814	4714	1.675



6. 토의 및 결론

본 연구에서는 병렬 연산 가속기와 여기 진폭 기법을 이용한 3차원 역시간 구조보정을 수행하였다. 시간 루프를 돌때마다 파동 전파 모델링과 영상을 얻기 위한 계산을 해야하기 때문에 계산량이 많은 3차원 역시간 구조보정의 성능을 높이기 위해 가속기를 이용하였다. 하지만 일반적으로 많이 사용하는 상호상관방법은 송신원 파동장을 모두 저장해야하기 때문에 큰 메모리 공간이 필요하다. 본 연구에서는 Titan XP GPU와 Xeon Phi x200 (Knights Landing) 프로세서를 가속기로 사용하였는데 Titan XP는 12 GB의 내장 메모리, Xeon Phi x200 프로세서는 16 GB의 고대역폭 메모리를 내장하고 있는데 모든 송신원을 저장하기에는 매우 부족하다. 이를 해결하기 위해 GPU는 호스트 프로세서인 CPU의 메모리, Xeon Phi 프로세서는 대용량이지만 낮은 대역폭의 DDR 메모리를 추가로 사용할 수 있다. 하지만 자료의 이동이 많아져 효율이 크게 떨어지고 오히려 기존에 CPU만 사용한 프로그램보다 성능이 나빠질 수 있다.

역시간 구조보정에 필요한 메모리를 줄여 GPU와 CPU간의 자료 이동을 줄이고 Xeon Phi의 고대역폭의 MCDRAM만을 사용한다면 최적화된 성능을 얻을 수 있을 것이다. 따라서 본 연구에서는 여기 진폭 기법을 역시간 구조보정에 적용하였다. 여기 진폭 기법은 모델의 각 격자점에서의 최대 진폭과 해당 시간만 저장하면 되기 때문에 메모리 사용량을 크게 줄일 수 있다. 먼저 여기 진폭 기법을 2차원 예제를 통해 검증하였다. 기존의 상호상관방법과 비교하여 메모리 사용량을 크게 줄일 수 있었지만 각 격자점마다 하나의 값만을 사용하기 때문에 불연속면이 생기는 것을 확인하였다. 따라서 직접파 제거, 오프셋 제한과 같은 선처리를 통해 이를 해결하였다.

또한 3차원 수치 예제를 통해 다중 GPGPU와 다중 Xeon Phi 프로세서

에서의 성능을 비교하였다. 병렬 프로그래밍 시 연산순서가 재배열되기 때문에 CPU와 GPU에서 연산 결과가 근소하게 달라지지만 배정도 연산에서는 그 값이 매우 작아 영상에서는 영향을 미치지 않을 것으로 생각된다. 본 예제에서는 단정도 연산을 하였고 GPU프로그램과 CPU프로그램이 다른 경계조건을 사용하였기 때문에 두 프로그램의 영상 단면이 정확하게 일치하지는 않지만 두 프로그램이 모두 해당 구조를 잘 나타내는 것을 확인할 수 있었다. 두 프로그램 모두 기존의 CPU 프로그램 대비 크게 성능이 향상 되었고, Xeon Phi x200 모델보다 Titan Xp GPU 프로그램의 성능이 더 뛰어난 것을 알 수 있었다. Titan XP GPU는 각 카드당 3840개의 수많은 코어로 이루어져 매우 빠른 계산이 가능하기 때문에 자료 입출력을 줄이는 것이 성능 향상에 큰 영향을 끼치게 된다. 따라서 본 연구에서 여기진폭 기법을 적용한 것이 성능을 최적화하는데 중요한 역할을 했다는 것을 알 수 있었다. 단 Titan Xp 모델의 배정도 연산 속도는 단정도 연산 속도에 비해 많이 떨어지므로, 배정도 연산의 경우 Xeon Phi 또는 다른 GPU를 사용할 필요가 있다.

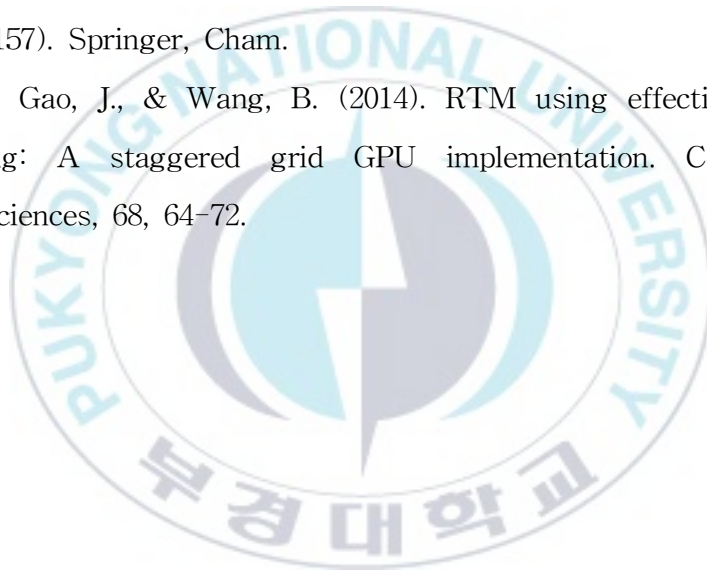
본 연구에서 개발한 효율적인 3차원 역시간 구조보정 모듈은 핵심 알고리즘을 공유하는 완전 파형 역산에도 적용하여 효율적인 결과를 얻을 수 있을 것이라 예상된다. 또한 가속기 연산 기술은 역시간 구조보정 외에도 다양한 탄성과 자료 처리 분야에 활용하여 기존의 CPU 프로그램 대비 효율 및 생산성을 높일 수 있을 것이다.

참고문헌

- Arslan, V., Blanc, J. Y., Tchiboukdjian, M., Thierry, P., & Thomas-Collignon, G. (2014, September). Design and Performance of an Intel Xeon Phi based Cluster for Reverse Time Migration. In EAGE Workshop on High Performance Computing for Upstream.
- Claerbout, J. F. (1971). Toward a unified theory of reflector mapping. *Geophysics*, 36(3), 467-481.
- Dussaud, E., Symes, W. W., Williamson, P., Lemaistre, L., Singer, P., Denel, B., & Cherrett, A. (2008). Computational strategies for reverse-time migration. In SEG Technical Program Expanded Abstracts 2008 (pp. 2267-2271). Society of Exploration Geophysicists.
- Higdon, R. L. (1986). Absorbing boundary conditions for difference approximations to the multidimensional wave equation. *Mathematics of computation*, 47(176), 437-459.
- Jeffers, J., Reinders, J., & Sodani, A. (2016). Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition. Morgan Kaufmann.
- Keys, R. G. (1985). Absorbing boundary conditions for acoustic media. *Geophysics*, 50(6), 892-902.
- Kim, A., Ryu, D., & Ha, W. (2016). Time-domain 3D Wave Propagation Modeling and Memory Management Using Graphics Processing Units. *Geophysics and Geophysical Exploration*, 19(3), 145-152.
- Lee, J., & Ha, W. (2018). Comparison of Parallel Computation

- Performances for 3D Wave Propagation Modeling using a Xeon Phi x200 Processor. *Geophysics and Geophysical Exploration*, 21(4), 213-219.
- LIU, H. W., Li, B., Liu, H., TONG, X. L., & Liu, Q. (2010). The algorithm of high order finite difference pre stack reverse time migration and GPU implementation. *Chinese Journal of Geophysics*, 53(4), 600-610.
- Liu, G., Liu, Y., Ren, L., & Meng, X. (2013). 3D seismic reverse time migration on GPGPU. *Computers & Geosciences*, 59, 17-23.
- Nguyen, B. D., & McMechan, G. A. (2012). Excitation amplitude imaging condition for prestack reverse-time migration. *Geophysics*, 78(1), S37-S46.
- Nguyen, B. D., & McMechan, G. A. (2014). Five ways to avoid storing source wavefield snapshots in 2D elastic prestack reverse time migration. *Geophysics*, 80(1), S1-S18.
- Ryu, D., Jo, S. H., & Ha, W. (2017). Parallelizing 3D Frequency-domain Acoustic Wave Propagation Modeling using a Xeon Phi Coprocessor. *Geophysics and Geophysical Exploration*, 20(3), 129-136.
- Shin, J., Ha, W., Jun, H., Min, D. J., & Shin, C. (2014). 3D Laplace-domain full waveform inversion using a single GPU card. *Computers & Geosciences*, 67, 1-13.
- Sodani, A., Gramunt, R., Corbal, J., Kim, H. S., Vinod, K., Chinthamani, S., ... & Liu, Y. C. (2016). Knights landing: Second-generation intel xeon phi product. *Ieee micro*, 36(2), 34-46.

- Sun, W., & Fu, L. Y. (2013). Two effective approaches to reduce data storage in reverse time migration. *Computers & Geosciences*, 56, 69–75.
- Symes, W. W. (2007). Reverse time migration with optimal checkpointing. *Geophysics*, 72(5), SM213–SM221.
- Tobin, J., Breuer, A., Heinecke, A., Yount, C., & Cui, Y. (2017, June). Accelerating seismic simulations using the intel xeon phi knights landing processor. In *International Supercomputing Conference* (pp. 139–157). Springer, Cham.
- Yang, P., Gao, J., & Wang, B. (2014). RTM using effective boundary saving: A staggered grid GPU implementation. *Computers & Geosciences*, 68, 64–72.



Abstract

In this study, we developed a 3D acoustic time-domain reverse-time migration algorithm using the excitation amplitude method and high-performance accelerators for parallel computing. The excitation amplitude imaging condition avoids full crosscorrelation between the source and receiver wavefields and exploits the maximum amplitude wavefield and the corresponding time. By adopting the excitation amplitude method, we could save the memory required to store the full source wavefield. We run the reverse-time migration algorithm using the excitation amplitude method on graphical processing units (GPUs) and Xeon Phi processors to accelerate the algorithm. Since the excitation amplitude and the excitation time can be saved in the GPU memory and the high bandwidth memory of Xeon Phi processor, we can reduce data communication and enhance the performance of the reverse-time migration program. We demonstrated the proposed method using the SEG/EAGE 3D salt and the overthrust velocity models. We confirmed that efficiency of accelerated programs has been improved compare to CPU program. We also confirmed that performance of GPU program is faster than that of Xeon Phi program.

Keywords : Excitation amplitude method, Reverse time migration,
Accelerator, Parallel computing

Student Number : 201755340