pISSN : 1976-0620, eISSN : 2384-0633
http://dx.doi.org/10.7742/jksr.2015.9.7.487 "J. Korean Soc. Radiol., Vol. 9, No. 7, December 2015"

Improvement in Reconstruction Time Using Multi-Core Processor on Computed Tomography

Kwon Su Chon

Department of Radiological Science, Catholic University of Daegu

다중코어 프로세서를 이용한 전산화단층촬영의 재구성 시간 개선

천권수

대구가톨릭대학교 방사선학과

Abstract

The reconstruction on the computed tomography requires much time for calculation. The calculation time rapidly increases with enlarging matrix size for improving image quality. Multi-core processor, multi-core CPU, has widely used nowadays and has provided the reduction of the calculation time through multi-threads. In this study, the calculation time of the reconstruction process would improved using multi-threads based on the multi-core processor. The Pthread and the OpenMP used for multi-threads were used in convolution and back projection steps that required much time in the reconstruction. The Pthread and the OpenMP showed similar results in the speedup and the efficiency.

Keyword : Computed Tomography, Reconstruction, Mult-thread, Pthread, OpenMP

요 약

전산화단층촬영에서 재구성 과정은 상당한 시간이 요구된다. 단면 영상의 품질을 높이기 매트릭스 크기를 증가시키면 재구성 시간이 매우 빠른 속도로 증가한다. 다중코어 프로세서는 오늘날 광범위하게 사용되고 있으며, 다중코어 프로세서의 다중 스레드를 이용하여 계산 시간을 줄이는 것이 가능하다. 본 연구는 다중 스레드로 CT의 재구성 시간을 개선하였다. 다중 스레드를 위해 Pthread와 OpenMP를 이용하였고, 재구성 과정에서 많은 시간이 소비되는 컨볼루션과 역투역 과정을 자세히 조사하였다. Pthread와 OpenMP 모두 스피드업과 효율성 측면에서 비슷한 성능을 나타내었다.

중심단어 : 전산화단층촬영, 재구성, 다중 스레드, Pthread, OpenMP

Ⅰ. INTRODUCTION

전산화단층촬영(CT, Computed Tomography)[1]은 의료[2] 및 산업[3] 분야에 광범위하게 사용되고 있다. CT는 투영 영상을 이용하여 재구성함으로써 물체의 단면 영상을 얻는다[4],[5]. 2차원 단면영상을 적층하여 3차원 영상을 얻어 의료 및 산업에 활용하는 경우도 많아지고 있다[6].

광원과 검출기를 고정하고 시료를 회전시키거나 또는 광원과 검출기를 일체형으로 하고 고정된 시료 주위를 회전(갠트리 방식)하면서 물체의 투영 영상을 얻

Corresponding Author: Kwon Su Chon E-mail: kschon@cu.ac.kr Tel: +82-53-850-2521
Addr. Catholic University of Daegu, 13-13 HAYANG-RO, HAYANG-EUP, GYEONGSAN-SI, GYEONGSANGBUK-DO, 712-702, REP. OF KOREA
Received : December 01, 2015 Revised : December 15, 2015 Accepted : December 25, 2015

고 이 투영 영상을 재구성(Reconstruction)함으로써 물체의 2차원 단면영상을 얻는다[7]. 영상 재구성은 전산화단층촬영장치에 사용되는 컴퓨터의 성능이 우수하더라도 상당한 시간이 소비된다. 따라서 시간과 재구성된 단면의 영상의 품질 사이에는 등가교환(Trade-off)이 필요하다. 의료영상의 경우는 데이터의 크기 및 재구성 속도 때문에 대부분 512×512 매트릭스로 재구성하고 있다. 그러나 고해상도의 공간분해능이 요구되는 상황에서는 투영의 개수도 많아지고, 영상의 매트릭스(Matrix)도 1024×1024, 2048×2048, 4096×4096 등으로 증가[8]하기 때문에 재구성의 시간이 기하급수적으로 증가하게 된다.

최근에는 GPU(Graphic Processing Unit)를 이용하여 재구성 시간을 개선하려는 연구가 CT, SPECT, MR 분야에서 활발히 진행되고 있다[9],[10]. GPU를 이용하여 필터보정 역투영법[11],[12](Filtered Back Projection), 반복적 재구성[13],[14](Iterative Reconstruction)에서 상당한 개선을 이루었다. 그러나 상대적으로 CPU(Central Processing Unit)를 이용한 개선은 많이 이루어지고 있지 않아 폭넓은 연구가 필요한 상황이다.

2000년 중반을 기점으로 CPU 발전의 방향이 크게 달라졌다. 2005년 이전에는 CPU의 클럭 속도(Clock Speed)를 개선하는 방향으로 기술이 발전하였다. 컴퓨터의 처리 속도는 CPU의 클럭 속도에 비례하여 빨라지기 때문이었다. 즉, 단일 코어 환경에서 클럭 속도가 곧 컴퓨터의 성능을 가늠하는 척도가 되었던 것이었다. 그러나 2005년 이후에는 CPU의 발열 문제로 클럭 속도를 높이는데 한계에 도달하였다. 그래서 CPU의 클럭 속도를 높이는 방향에서 다중 CPU를 사용하는 방향으로 기술이 발전하였다.

현재 사용되고 있는 컴퓨터는 대부분 다중 코어 CPU를 사용하고 있지만, CT에서 재구성은 단일 스레드(Thread)를 사용하고 있어 CPU 자원을 비효율적으로 사용해왔다. 본 연구는 다중코어 CPU의 성능을 활용할 수 있도록 재구성 알고리즘을 병렬화하여, 다중 스레드로 재구성의 속도를 향상시키도록 하였다.

현재 광범위하게 사용하고 있는 다중명령 다중데이터 처리(MIMD)의 병렬 프로그래밍은 크게 2가지로 나눌 수 있다[15]. 공유 메모리를 사용하는 방법과 분산 메모리를 사용하는 방법이다. 분산 메모리 기법은 각 프로세서가 자신의 메모리를 가지고 있으면서 프로세서와 메모리쌍이 상호 연결된(Inter-connection) 네트워크를 통해 통신한다. 하나의 프로그램을 다수의 개인용 컴퓨터를 이용하여 계산하는 것이 그 예가 될 수 있다. MPI(Message Passing Interface)가 대표적인 병렬처리 방법이다. 공유 메모리를 사용하는 방법은 다중 코어 프로세서가 메모리를 공유하도록 하여, 단일 프로그램을 각각의 프로세서가 협력해서 구동하는 방식이다. 이 방식은 POSIX thread[16](일명 pthread)와 Open MP[17]가 대표적이다.

OpenMp는 C언어의 상위 레벨 확장인 반면, Pthread는 OpenMP에서 사용할 수 없는 몇 가지 더 낮은 수준(Low Level)의 병렬화 기법을 제공한다. 사용자 측면에서 보면 OpenMP는 병렬로 실행될 코드의 블록만 간단하게 구분하면 되고, 세밀한 제어는 컴파일러나 런타임(Run-time) 시스템이 자동으로 알아서 결정한다. 반면 Pthread는 사용자가 명시적으로 각 스레드의 프로그램 형태를 설정해야 한다. 본 논문에서는 Pthread와 OpenMP를 통한 다중 스레드로 CT의 영상재구성 시간을 개선하였다.

Ⅱ. MATERIAL AND METHODS

영상재구성의 성능 개선이 목적이기 때문에 평행빔에 대한 Shepp-Logan 모형[18]을 살펴본다. 우선 Shepp- Logan 모형에 대한 투영을 구하고, Lam-Rak 필터[19]를 이용하여 컨볼루션(Convolution)을 수행하고, 역투행을 수행하여 필터보정 역투행을 수행한다[20]. 주파수 공간에서 필터링(Filtering)을 수행할 수 있지만, 여기서는 실(Real) 공간에서 컨볼루션을 통해 필터링을 수행하도록 하였다.

영상 재구성의 매트릭스는 1024 × 1024가 되도록 했으며, 투영은 720개(0.25° 당 1개의 투영)를 사용하도록 하였다. 컨볼루션 과정에서 샘플링 이론(Sampling theory)을 적용하기 위해 필터는 2048 (1024 × 2)개의 데이터를 사용하였고, 시노그램(Sinogram)의 크기도 2048 × 720 이 되도록 Zero-padding하여, 샘플링에 의한 영상의 공간분해능 저하를 최소화 하도록 하였다.

영상재구성 프로그램은 Visual Studio 2010 (Microsof

t, USA)을 사용하여 C/C++로 작성하였으며, API(Application Program Interface)기반에서 구동하게 하였다. Shepp-Logan 모형을 구현하고 이 데이터를 비트맵(Bitmap)으로 변환하여 더블 버퍼링 기법으로 영상을 표시하였다.

성능 검증을 위해 컨볼루션, 역투영에서 시간을 측정하였다. 각 단계에 전후에 타이머에 기록된 값을 이용하여 시간을 초 단위로 표시하였다. 성능의 표시를 위해 스피드업(Speedup)[15]

$$S = \frac{T_{serial}}{T_{parall}} \quad \text{식(1)}$$

와 병렬 프로그램의 효율성(Efficiency)[15]

$$E = \frac{T_{serial}}{p\,T_{parall}} \quad \text{식(2)}$$

를 이용하였다. 여기서 Tserial은 통상적인 방법(단일 스레드)에 의한 수행 시간이고, Tparallel은 다중 스레드로 계산하여 걸린 시간이다. p는 스레드 개수를 나타낸다. 프로그램을 구동하는데 사용된 컴퓨터 사양은 Table 1에 표시하였다.

Table 1. Specifications of the computer

항목	사양	비고
CPU	i7 X990	헥사코어, 12스레드
Speed	3.47 GHz	
Memory	16 GB	
Graphic Card	GTX560 Ti	NVIDIA, RAM: 4 GB
OS	Windows 7	

Ⅲ. RESULT

1. 단일 스레드

Shepp-Logan 모형, 컨볼루션, 역투영 모두 3중 루프(loop)로 구성되어 있다. 3중 루프를 이중 루프로 바꾸면 다중 스레드로 작업하기에 매우 용이하다. 10개의 타원으로 구성된 Shepp-Logan 모형을 살펴보자. 우선 하나의 타원에 대해 가로와 세로 방향으로 1024 만큼 반복하면 영상이 만들어진다. 나머지 9개의 타원에 대해 반복하면 3중 루프가 된다. 가로와 세로 방향으로 반복되는 2중 루프를 행 우선(Row-major) 방식인 1차원 방법으로 표현하면 단일 루프로 바꿀 수 있다. Fig. 1은 의사코드(Pseudocode)로 표현한 3중 루프 단순화 방법이다.

```
for(m = 0; n < 10)                    // 타원
  for(y = 0; y <  SizeX)              // 세로
    for(x = 0; x < Sizey)             // 가로
    {
      Image[y * SizeX + x]
    }
```

(a) 3중 루프 방식

```
for(m = 0; n < 10)                    // 타원
  for(n = 0; n < SizeX * SizeY)       // 루프 통합
  {
    y = k / SizeX                     // 세로
    x = n % SizeX                     // 가로
    Image[y * SizeX + x]
  }
```

(b) 2중 루프 방식

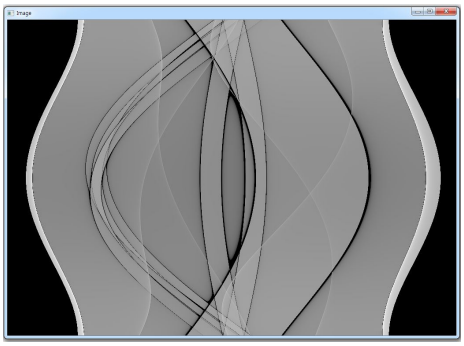
Fig. 1. Loop structures of the Shepp-Logan phantom.

Table 2. Time for each implementation step

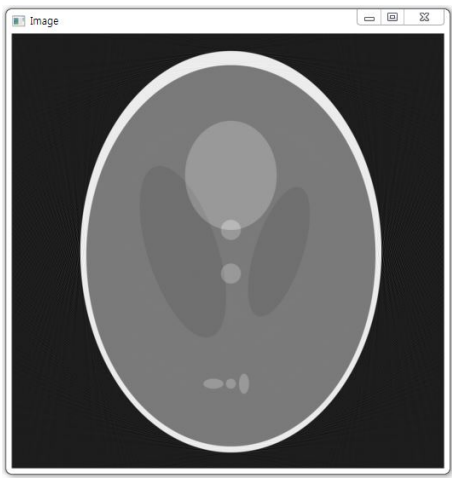
Steps		Time (sec)	Remarks
Phantom		0.733	2 loops
Convolution	Sinogram	0.093	2 loops
	Filtering	22.418	2 loops
Back projection		50.669	2 loops
Total		73.211	

3중 루프를 2중 루프로 바꾸고 Shepp-Logan 모형을 실행하면 두 경우에 대해 소요되는 시간은 0.904 sec와 0.905 sec로 거의 같다. 그러나 Fig. 1에서 타원의 개수를 반복하는 for 루프와 영상의 가로와 세로 부분에 대한 for 루프를 교환하면 2중 또는 3중 루프에서 소모되는 시간은 각각 0.733 sec와 0.734 sec가 되어 for 루

프의 위치에 따라 구현 속도가 조금의 차이가 발생하였다. Table 2는 각 단계에서 소모되는 시간을 나타낸 것이다. 역투영 과정이 가장 많은 시간이 소모된다는 것을 알 수 있다. Fig. 2는 필터보정 역투영법으로 획득된 Shepp-Logan 영상을 나타낸 것이다. 원본 영상을 매우 잘 복원한다는 것을 알 수 있다.



(a)



(b)

Fig. 2. Sinogram after convolution with Lam-Rak filter and phantom image after back projection.

2. 다중 스레드

본 논문에서는 CPU의 다중코어 환경에서 다중 스레드를 구현하기 위하여 Pthread와 OpenMP를 사용하였다. Pthread에서는 p개의 스레드가 for 루프를 p개로 분할해서 수행하도록 코드를 작성하였다. n 번째 스레드가 수행하는 for 루프의 시작과 마지막은 아래의 식과 같이 계산할 수 있다.

$$Inital: \ n \times \frac{SizeX \times SizeY}{p} \qquad \text{식(3)}$$

$$Final: \ (n+1) \times \frac{SizeX \times SizeY}{p} - 1$$

실행과정은 pthread_create( ) 함수로 스레드를 생성시키고, 스레드 별로 함수를 수행한 후, pthread_join( ) 함수를 호출함으로써 사용한 스레드를 메인 스레드로 통합시키는 절차로 되어 있다. Fig. 3은 Pthread로 Back Projcetion( ) 함수를 다중 스레드로 수행하는 코드를 나타낸 것이다.

```
for(p = 0; p <  nThread; p++)     // 스레드
{
  pthread_create(&thread_handles[p],
                 NULL,
                 BackProjection,
                 (void *) p);
}

for(p = 0; p <  nThread; p++)
{
  pthread_join(thread_handles[p], NULL);
}
```

Fig. 3. Creation and destruction of thread in Pthread.

반면 OpenMP는 전처리 명령어를 적고, 병렬로 수행될 부분을 블록({ })으로 묶어주면 되므로 매우 간단하다. 실행되는 스래드 개수를 결정하기 위해 num_thread( ) 함수를 이용했다. Pthread와 비교하기 위해 컨볼루션, 역투영 함수를 각각 병렬로 수행하도록 하였다. Table 3은 스레드 개수별로 소요된 시간을 나타낸 것이다. 단일 스레드에 비해 이득이 매우 크다는 것을 알 수 있다.

스레드의 개수가 증가할수록 계산 시간은 빠르게 감소한다. 영상재구성에 사용된 컴퓨터는 Table 1에서와 같이 12개의 물리적 스레드를 발생시킬 수 있기 때문에 12개의 스레드를 발생시키는 것이 바람직하다. 스레드를 12개 이상 발생시켜도 계산 시간의 향상은 거의 없는 것으로 나타났다. 스레드 개수를 물리적으

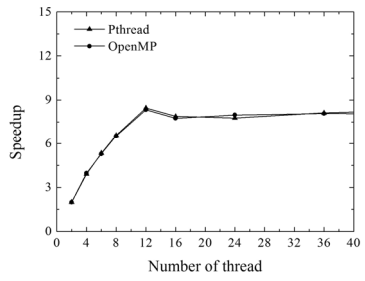
로 발생시킬 수 있는 스레드 개수보다 많이 발생시키면 오히려 성능저하가 발생한다.

Table 3 Implementation time of convolution and back projection for Pthread and OpenMP

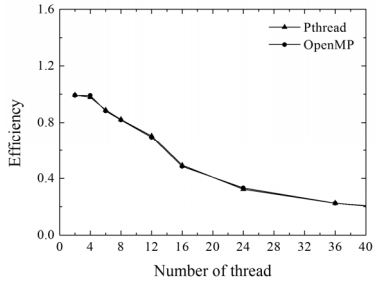
Thread number	Pthread		OpenMP	
	C*(sec)	B**(sec)	C*(sec)	B**(sec)
2	11.403	25.459	11.45	25.538
4	5.975	12.948	6.162	12.777
6	4.539	9.501	4.368	9.579
8	4.056	7.707	4.274	7.754
12	3.807	5.990	3.650	6.084
16	3.869	6.427	3.807	6.536
24	3.717	6.521	3.791	6.349
36	3.635	6.240	3.806	6.271

C*: Convolution

B**: Back projection



(a)



(b)

Fig. 4. Speedup (a) and efficiency (b) for Pthread and OpenMP.

다중 스레드를 사용해서 얼마나 이득이 있었는지 조사하였다. Fig. 4는 Pthread와 OpenMP에 대한 스피드업(S)과 효율성(E)을 나타낸 것이다. 효율성(Fig. 4(b))이 감소하는 것은 다중 스레드에서 나타나는 일반적인 현상이다[15]. Pthread와 OpenMP는 스피드업 및 효율성에서 모두 비슷한 성능을 나타내었다.

단일 스레드를 사용할 때의 CPU 사용은 대략 8%이었으나, 다중 스레드의 경우에는 스레드 발생 개수에 따라 CPU의 사용량이 달랐다. Pthread 및 OpenMP에서 공히 2, 4, 6, 8, 12 스레드 발생에 대해 16%, 32%, 50%, 66%, 100%의 사용을 보였다. 단일 스레드에 비해 CPU를 효율적으로 사용한다는 것을 알 수 있다.

Ⅳ. DISCUSSION

1024 × 1024 크기로 재구성할 때 소요되는 시간을 1로 가정하면 512 × 512와 2048 × 2048 크기는 각각 0.247배(12.526 sec)와 4.093배(207.371 sec) 걸린다. 매트릭스 개수가 증가하면 for 루프의 횟수가 증가하여 역투영 과정에 시간이 크게 소모된다는 것을 알 수 있다. 임상에서 매크릭스의 크기를 크게 증가시키지 못하는 이유가 여기에 있다.

대부분의 컴퓨터 CPU는 다중코어를 가지는 프로세서를 채용하고 있으나 사용하는 프로그램은 주로 단일 스레드를 사용하는 경우가 많다. 다중코어 프로세서에서 코어 개수(또는 물리적으로 발생시킬 수 있는) 만큼 스레드를 만들어 작업을 분할하면 시간 또는 스레드당 작업량을 줄일 수 있다. 필터보정 역투영 방법을 사용하는 CT 영상재구성에서 컨볼루션과 역투영 과정은 매우 많은 시간을 요구하므로 단면 영상의 매트릭스 개수를 증가시키는데 한계가 있었으나, 다중 스레드를 사용하면 상당한 시간 절약이 가능하다.

스레드로 분할하여 for 루프를 수행하기 위해 3중 루프를 2중 루프로 변환하였다. 루프를 줄이더라도 실질적인 시간의 차이가 없기 때문에 2중 루프로 변환하여 작업하는 것이 편리하였다. Fig. 1의 (b)방식에서 타원과 영상에 대한 for루프의 순서를 바꾸면 대략 0.17초의 차이가 발생하는데 이는 타원의 개수가 10개로 캐싱(caching)의 비효율성으로 인해 손실이 발생하기 때문이다.

본 연구에서 for 루프를 계산하는 과정만 다중 스레

드로 분할하여 작업을 수행한 결과 Pthread와 OpenMP는 비슷한 성능(Speedup, Efficiency)을 보였다. CPU가 12개의 스레드를 지원하기 가지고 있기 때문에 12 스레드에서 최적의 값을 보였다. 물리적으로 발생시킬 수 있는 스레드 보다 더 많은 스레드로 작업을 할 수 있지만, 스피드업에서는 오히려 조금씩 감소한다는 것을 알 수 있다. 그리고 효율은 12 스레드 이후는 스피드업의 변화가 없기 때문에 스레드의 역수로 감소한다는 것을 알 수 있다.

최근에는 GPU(Graphic Process Unit)를 이용하여 계산 속도를 향상시키는 연구들이 많이 진행되고 있다[9-14]. GPU에서는 다중코어의 스레드 개수와는 차원이 다른 엄청난 개수의 스레드를 발생시켜 동시에 계산하도록 한다. 예를 들어 640개의 투영 및 1024 × 1024 크기에 대해 1024 × 1024개의 스레드를 이용하여 역투영을 계산하면 Fig. 5와 같이 1초미만으로 계산 속도를 크게 향상 시킬 수 있다.

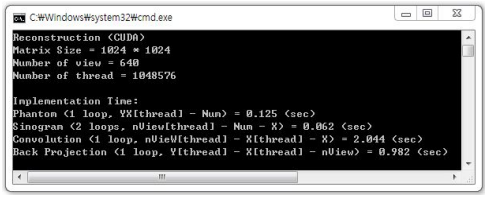


Fig. 5. Reconstruction using GPU(CUDA).

GPU를 이용하기 위해서는 메인 메모리에 있는 데이터를 GPU로 이동 시켜야 하는 등 여러 가지 추가 작업이 필요하다. 따라서 GPU로 최적의 성능을 얻기 위해서는 여러 가지 요소(예, 데이터 전송 속도, 공유 메모리 등)들이 고려되어야 한다.

의료 및 산업에서 사용되는 투영의 파일 크기가 20 MB 이상인 경우가 빈번하고, 투영의 개수가 1,000개 이상인 경우도 많다. 대용량 투영 데이터를 이용하는 영상 재구성에서 소요 시간을 줄이기 위해 CPU와 GPU의 다중 스레드를 모두 이용하는 하이브리드 재구성에 대한 연구도 진행되고 있다[21],[22]. 그러나 Z방향으로 단면 영상이 많이 필요 없는 경우는 CPU의 다중 스레드만을 이용하더라도 재구성 시간에 대한 효과를 볼 수 있을 것이다.

Ⅴ. CONCULSION

CPU의 다중 코어 프로세서를 효과적으로 이용하기 위해 Pthread와 OpenMP로 다중 스레드를 발생시켜, 720개의 투영에서 1024 × 1024 크기의 Shepp-Logan 모형을 재구성하였다. 역투영에서 단일 스레드를 이용하는 보통의 방법에 비해 Pthread와 OpenMP는 12 스레드에서 각각 8.46배 및 8.33배의 시간 향상을 보였다. Pthread와 OpenMP는 스피드업 및 효율도 비슷한 성능을 보였다. 물리적으로 발생시킬 수 있는 스레드를 모두 사용할 경우 CPU의 사용 효율은 최대가 되었다.

대용량 파일을 효과적으로 다루기 위해 CPU와 GPU의 스레드를 동시에 이용하는 하이브리드 방법에 대한 연구도 활발히 이루어질 것이다.

Acknowledgement

This work was supported by research grants from Catholic University of Daegu in 2015.

Reference

[1] G.N. Hounsfield, “Computerized transverse axial scanning (tomography): I. Description of system,” Br. J. Radiol. Vol. 46, pp.1016-1022, 1973.

[2] W.A. Kalender, “X-ray computed tomography,” Phy. Med. Biol. Vol. 51, pp.R29-R43, 2006.

[3] B.S.B. Sun W, R.K. Leach, “An overview of industrial X-ray computed tomography,” NPL Report ENG 32, 2012.

[4] K.S. Chon, "Removal of Ring Artifact in Computed Tomography," J. Kor. Soc. Radiol., Vol. 9, pp.403-408, 2015.

[5] K.S. Chon, "Noise Properties for Filtered Back Projection in CT Reconstruction," J. Kor. Soc. Radiol., Vol. 8, pp.357-364, 2015.

[6] F. Rengier, A. Mehndiratta, H. vpm Tengg-Kobligk, C.M. Zechmann, R. Unterhinninghofen, H.-U. Kauczor, F.L. Giesel, “3D printing based on imaging data: review of medical applications,” Int. J. Comput. Assist. Radiol. Surg., Vol. 5, pp.335-341, 2010.

[7] J. Hsieh, Computed Tomography: Principles, Design, Artifacts, and Recent Advances, SPIE Press, Washington, 2009.

[8] J.I. Agulleiro, J.J. Fernandez, "Fast tomographic reconstruction on multicore computers," BIOINFORMATICS, Vol. 27,

pp.582-583, 2011.

[9] G. Yan, J. Tian, S. Zhu, Y. Dai, C. Qin, "Fast cone-beam CT image reconstruction using GPU hardware," J. X-Ray Sci. Tech., Vol. 16, pp.225-234, 2008.

[10] F. Xu, K. Mueller, "Real-time 3D computed tomographic reconstruction using commodity graphics hardware,", Phys. Med. Biol. Vol. 52, pp.3405-3419, 2007.

[11] F. Xe, K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," IEEE Trans. on Nucl. Sci., Vol. 52, pp.654-663, 2005.

[12] P. Noel, A. Walczak, J. Xu, J. Corso, K. Hoffmann, S. Schafer, "GPU-based cone beam computed tomography," Comp. Med. Prog. Bio., Vol. 98, pp.271-277, 2010.

[13] K. Muller, R. Yagel, "Rapid 3-D Cone-Beam Reconstruction with the Simultaneous Algebraic Reconstruction Technique (SART) using 2-D Texture Mapping Hardware," IEEE Trans. on Med. Imag., Vol. 19, pp.1227-1237, 2000.

[14] M. Beister, D. Kolditz, W. Kalender, "Interative reconstruction methods in X-ray CT," Vol. 28, pp.94-108, 2012.

[15] P. Pacheco, An Introduction to Parallel Programming, Elsevier Inc, New York, 2011.

[16] http://opengroup.org.

[17] http://openmp.org.

[18] L.A. Shepp, B.F. Logan, "The Fourier Reconstruction of a Head Section," IEEE Trans. Nucl. Sci., Vol. 21, pp.21-43, 1974.

[19] G.N. Ramachandran, A.V. Lakhshminarayanan, "Three-dimensional reconstruction from radiographs and electron micrographs: Application of convolution instead of Fourier transforms," Proc. Natl. Sci. Acad. USA, Vol. 68, pp.2236–2240, 1971.

[20] A.C. Kak, M. Slaney, Principles of Computerized Tomographic Imaging, IEEE Press, New York, 1988.

[21] C.T. Yang, C.L. Huang, C.F. Lin, "Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters," Computer Physics Communications, Vol. 182, pp.266-269, 2009.

[22] M.D. Jones, R.Y. Yao, C.P. Bhole, "Hybrid MPI-OpenMP programing for Parallel OSEM PET Reconstruction," IEEE Trans. on Nucl. Sci., Vol. 53, pp.2752-2758, 2006.