다중 GPGPU의 스케쥴링을 이용한 고속 홀로그램 생성 방법

*이윤혁 **서영호 ***김동욱

*,***광운대학교 전자재료공학과, ** 광운대학교 인제니움학부

*winner9100@kw.ac.kr

Fast Hologram Generation Method Using Scheduling of Multi-GPGPUs

*Lee, Yoon-Hyuk **Seo, Young-Ho ***Kim, Dong-Wook

*,***Dept. of Electronic Materials Eng., Kwangwoon University
** Ingenium college of liberal arts, Kwangwoon University

요약

컴퓨터 생성 홀로그램(CGH)는 방대한 계산 량을 가지고 있어, 고해상도의 홀로그램을 생성하기 위하여 고속 홀로그램 생성 방법이 필요하다. 본 논문에서는 다중 GPGPU의 스케쥴링 기법을 이용하여 고속화 하는 방법을 제안한다. 첫 번째로는 커널 내에서 공유 메모리를 이용한 스케쥴링 기법을 통하여 고속화를 하고, 두 번째로는 GPGPU간의 P2P(peer-to-peer)데이터 전송을 이용한 스케쥴링을 했다. nVidia의 GTX680 2개 GPGPU를 이용하여 기존의 방법보다 약 50%의 속도 향상을 확인하였다.

1. 서론

CGH(Computer Generated Hologram)은 모든 유효한 객체 정보(밝기 값이 0이 아닌 객체 정보)를 이용하여 하나의 홀로그램 화소를 계산하는 방법이다[1]. 이러한 계산 방법은 반복하는 요소가 매우 많기 때문에 크기가 증가함에 따라 계산 량이 기하급수적으로 증가한다[1]. 이러한 문제로 고 해상도의 홀로그램을 생성하기 위해서 고속의 홀로그램 생성 방법들이 연구되었는데 크게 두 가지로 분류 할 수 있다. 첫 번째는 FPGA또는 ASIC을 이용하여 하드웨어로 구현하는 것이고, 두 번째는 GPGPU(General Purpose Graphic Processing Unit)을 이용한 병렬 프로그래밍 기법들이다.

2. 고속 홀로그램 생성 방법

2.1 커널 내에서의 스케쥴링 방법

식 1은 CGH식을 Fresnel 영역에서의 근사화한 식이다[1]. $I_\alpha(u,v)$는 홀로그램 평면의 u,v좌표에서의 홀로그램 화소의 밝기이다. N은 유효한 객체정보의 개수이고, $A_j(x,y,z)$는 x,y,z좌표에서의 객체의 밝기이다. 또한 λ는 사용된 참조파이고, p는 홀로그램의 화소의 크기이다.

$$I_\alpha(u,v) = \sum_{j=0}^{N-1} A_j(x,y,z)\cos\left[2\pi\left\{\frac{z}{\lambda}+\frac{p^2}{2\lambda z}\{(x-u)^2+(y-v)^2\}\right\}\right] = \sum_{j=0}^{N-1} A_j(x,y,x)\cos\left[2\pi\{\theta_x+\theta_y+\theta_z\}\right] \quad (1)$$

식 1에서 θ_x와 θ_y는 홀로그램 화소의 각각 u축과 v축에 대하여 계산하고 이를 같은 행 또는 열에서 공유하여 쓸 수 있는 변수들이다[1]. 따라서 GPU 커널 하나에서 32x32 쓰레드를 구성하고 그림 1과 같이 32개의 화소들에 대하여 32개의 θ_x와 θ_y를 계산하여 공유 메모리[2]에 저장할 수 있다. 그림 1은 커널 내의 스케쥴링을 도식화 하였고, 그림 2는 각 단계별 쓰레드에서 수행되는 동작을 좌표에 따라 도식화 하였다.

그림 2(a)은 글로벌 메모리[2]로부터 32개의 유효한 객체의 정보를 공유메모리로 로드시키는 것이다. 이때 모든 공유메모리[2]에 로드시킨 뒤 모든 쓰레드[2]는 동기화 과정을 거쳐야 한다. 그림 2(b)는 $\theta_x+\theta_z$를 구하는 것으로 공유메모리에 저장된 객체 정보 32개와 각 쓰레드의 u좌표에 대한 정보를 이용하여 구한 뒤 다른 공유메모리에 저장한다. 그림 2(c)는 θ_y를 구하는 것으로 공유메모리의 32개의 객체 정보와 각 쓰레드의 v좌표에 대한 정보를 이용하여 구하고 또 다른 공유메모리에 저장한다. 이과정이 끝나면 다시 한 번 쓰레드의 동기화를 거친 후 각 쓰레드에서는 그림 2(d)와 같이 $\theta_x+\theta_z$와 θ_y를 이용하여 32개의 객체에 대한 각 좌표에 해당하는 홀로그램 화소를 구한다. 이와 같이 공유되는 항들을 공유메모리를 이용하여 스케쥴링할 경우 θ_x, θ_y,θ_z을 계산하는 횟수가 1/1,024로 감소한다.

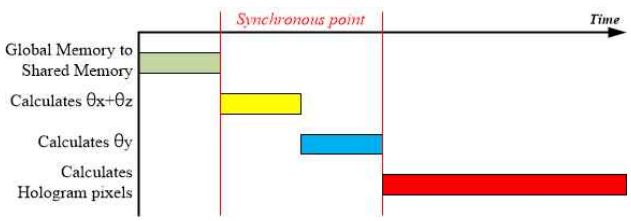

그림 1. 커널 내에서의 스케쥴링.

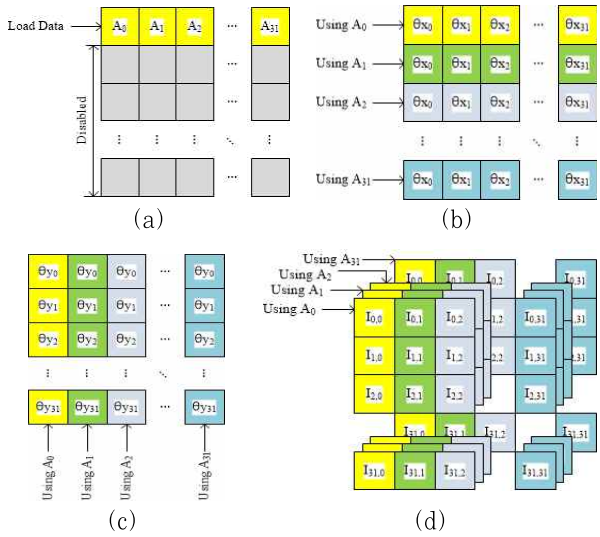

그림 2. 32x32 쓰레드에서 수행되는 동작

2.2 GPGPU간의 스케쥴링 방법

여러 개의 GPGPU를 이용하여 홀로그램을 계산을 할 경우 각 GPGPU에서 계산한 중간 홀로그램을 호스트(CPU)측[2]으로 전송한 뒤 누적덧셈을 통하여 최종 홀로그램을 생성한다[1]. 그림 3은 위와 같은 방법으로 홀로그램을 계산하는 스케쥴링을 도식화 하였다.

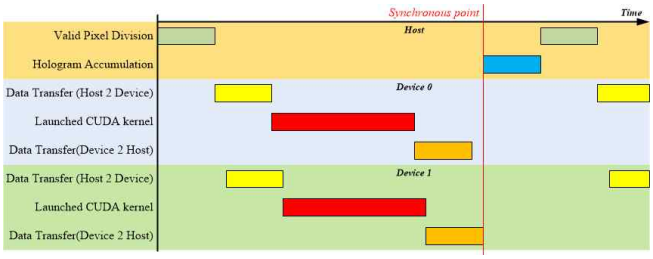

그림 3. 기존의 다중 GPGPU의 스케쥴링.

위 그림 3의 스케쥴링에서 각 GPGPU에서 호스트로의 데이터 전송을 줄이고 GPGPU간의 통신(P2P)[2]를 이용하여 그림 4와 같이 스케쥴링을 하면 중간홀로그램들의 누적과정을 없앨 수 있다. 또한 데이터의 전송횟수를 줄이는 효과가 생긴다.

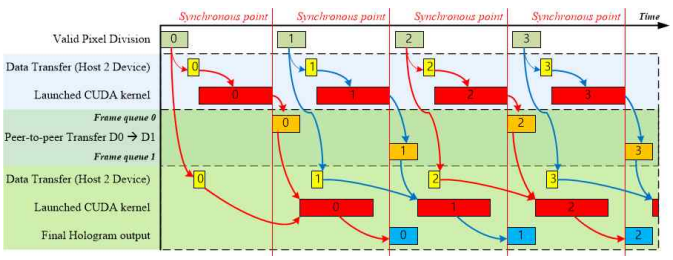

그림 4. 제안하는 다중 GPGPU의 스케쥴링.

4. 구현결과

구현한 고속 홀로그램 생성 알고르즘은 nVidia사의 GTX680 두 개의 GPGPU를 사용하였고, CUDA 툴킷을 이용하였다. 또한 사용한 객체는 표 1과 같이 총 4개의 테스트 셋을 이용했다.

표 1. 테스트에 사용된 객체 영상

영상	획득방법
Rabbit	인터넷
Sujin	수직리그 시스템[3]
Baby	Middlebury[4]
Billiard	DoF Pro[5]

표 2는 테스트 셋을 이용하여 각 단계별 평균 속도를 나타내었다. 생성한 홀로그램의 크기는 1,024 x 1,024 크기이고, 유효 객체 정보는 10k ~ 20k이다. 제안하는 알고리즘을 적용할 경우 약 50%정도의 속도 개선이 있다.

표 2. 각 단계별 평균속도

	평균 속도
기존 방법	737.1 ms
커널내 스케쥴링(2.1)	339.6 ms
GPU간 스케쥴링(2.2)	697.1 ms
2.1 + 2.2	305.7 ms

5. 결론

본 논문에서는 GPGPU의 스케쥴링을 통한 CGH의 고속화 방법을 제안하였다. 두 가지의 스케쥴링 기법을 제안 하여 50%의 개선을 하였지만 추후 여러 가지 스케쥴링 기법을 통하여 더 높은 효율의 스케쥴링을 구현할 수 있을 것으로 사료된다.

감사의글

이 논문은 2014년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임
(NRF-2014R1A2A1A11052433).

참고문헌

[1] Y.H. Lee, Y.H. Seo, J. S. Yoo, D. W. Kim, “High-Performance Computer Generated Hologram by Optimized Implementation of Parallel GPGPUs”, Journal of the Optical Society of Korea, Vol. 18, No. 6, Dec. 2014.
[2] http://docs.nvidia.com/cuda/index.html#axzz49YxtZrnZ
[3] Y.H. Seo, Y.H. Lee, J.M. Koo, W.Y. Kim, J. S. Yoo, D. W. Kim, “Digital holographic video service system for natural color scene”, Optical Engineering, SPIE, Vol.52(11), Nov. 2013.
[4] http://www.dofpro.com/cgigallery.htm/
[5] http://vision.middlebury.edu/stereo/data/