

## TCP 네트워크 통신을 이용한 디지털 홀로그램 생성 시스템의 구현

김창섭 송중석 박종일\*

한양대학교

cskim@mr.hanyang.ac.kr, jssong@mr.hanyang.ac.kr, jipark@hanyang.ac.kr\*

## Implementation of computer-generated hologram using TCP network communication

Kim, Changseob, Song, Joongseok Park, Jong-II\*

Hanyang University

## 요약

컴퓨터 생성 홀로그램(CGH: computer generated hologram) 기법은 기존의 홀로그램의 광학적 장치의 단점을 보완하여 범용 컴퓨터에서 홀로그램을 생성할 수 있도록 하는 기술이다. CGH는 입력으로 주어지는 물체의 3차원 정보와 출력으로 나오는 디지털 홀로그램의 해상도에 따라 그 연산량이 결정 된다. CGH는 단순하고 반복적인 수학적 계산을 통하여 디지털 홀로그램을 생성하게 되는데, 기존의 연구들에서는 GPU(graphic processing unit)를 이용하여 알고리즘들을 병렬적으로 처리한다. 본 논문에서는 기존연구에서 쓰인 GPU를 이용한 CGH를 개선하여 GPU가 장착되지 않은 상용 컴퓨터에서 GPU가 장착된 다른 컴퓨터들의 연산 자원을 활용하여 CGH를 수행 할 수 있는 프로그램의 개발 방법을 제안 한다. 본 시스템은 GPU가 요구되지 않는 한 개의 서버 컴퓨터와 GPU가 장착된 다수의 클라이언트들로 구성되어 있다. 서버 측에서 물체의 3차원 정보를 입력 받아 각각의 클라이언트들에게 적절한 연산량을 분배하고, 각 클라이언트들은 이미 알려진 GPU 기반 CGH를 통하여 연산을 수행 한 뒤, 그 결과를 서버로 다시 전송하게 된다. 서버는 수신한 각 결과들을 누적하여 입력 받은 물체에 대한 하나의 온전한 홀로그램을 생성할 수 있게 된다.

## 1. 서론

홀로그램은 입력으로 주어지는 물체의 3차원 정보와 광학적 장치를 통하여 간섭 패턴을 필름에 기록하고 다시 그 물체의 3차원 정보를 재현해내는 기술이다. 하지만 광학적 장치는 매우 높은 구축비용과 광학에 대한 전문 지식이 요구 된다. 또한 주변 광원의 차단 및 광학적 장치에 대한 무진동 환경을 구축해야 한다. 이러한 단점들을 극복하기 위하여 Brown과 Lohmann은 기존의 방법에서 쓰이던 광학적 장치의 수학적 모델링을 통하여, 범용 컴퓨터를 사용하여 홀로그램을 생성할 수 있도록 하는 컴퓨터 생성 홀로그램(CGH: computer generated hologram)을 개발하였다[1]. 하지만 CGH는 그 알고리즘의 특성상 연산량이 매우 방대하다. 하나의 연산은 비교적 간단한 수학적 연산이지만, 주어진 물체에 대한 각 광원과 생성되는 홀로그램의 각 픽셀별로 독립적인 연산을 모두 수행해야 하기 때문에 총 연산량은 매우 커지게 된다. 따라서 기존의 CGH 연구에서는 FPGA(field programmable gate array)를 이용하여 하드웨어(H/W: hardware)적으로 연산 자체를 고속화 하거나 많은 연산 코어를 가지고 있는 GPU(graphic processing unit)를 이용하여 전체 연산을 병렬적으로 수행 한다.

GPU를 이용한 CGH의 경우에 FPGA의 경우에 비해 상대적으로 구현 비용이 낮지만, 실험실 환경에서 모든 컴퓨터에 충분한 사양의 GPU를 장착하는 데에는 많은 어려움이 있다. 본 논문에서는 TCP 네트워크 통신을 이용한 3D 물체의 홀로그램 고속 생성 시스템의 구현

방법을 제안한다. 제안하는 방법은 GPU가 장착되지 않은 한 개의 서버 컴퓨터와 다수의 클라이언트 컴퓨터들로 구성된 TCP 네트워크 통신 기반의 병렬 연산 CGH 기법의 구현에 대한 내용이며, GPU가 장착된 다수의 클라이언트 PC의 연산 능력을 서버가 효율적으로 활용할 수 있도록 구현이 되어있다. 제안하는 방법을 통해, GPU가 없는 서버 측 환경에서 이미 구성되어 있는 네트워크를 통하여 GPU가 장착된 클라이언트 측 자원을 효율적으로 사용함으로써 보다 빠른 시간 안에 CGH 연산을 완료할 수 있도록 하는 시스템을 쉽게 구현 할 수 있게 된다.

2장에서는 컴퓨터 생성 홀로그램(CGH)에 대해 설명하고, 3장에서는 다중 GPU 기반의 CGH 연산 기법에 대해 설명한다. 3장에서는 제안하는 방법에 대해 설명하고, 4장에서는 제안하는 방법에 대한 실험 결과를 나타내며, 5장에서는 본 논문의 결론에 대해서 설명하고 있다.

## 2. 컴퓨터 생성 홀로그램(CGH)

CGH 이전의 홀로그램 연구에서는 매우 복잡한 광학적 장치와 엄격히 제한된 실험환경 하에 두 빛(물체를 거친 빛과 직접적으로 감광 필름에 조사되는 빛)의 간섭 패턴을 감광 필름에 기록하는 방식을 택하고 있었다. CGH는 이전의 광학적 실험을 수학적으로 모델링하여 감광 필름에 기록되는 간섭패턴을 범용 컴퓨터를 이용하여 계산해 낼 수

\*교신저자

있도록 하는 기법이다. 다음 수식 (1)이 CGH의 주요 수식이다.

$$I_{\alpha} = \sum_j^N A_j \cos(k \sqrt{(px_{\alpha} - px_j)^2 + (py_{\alpha} - py_j)^2 + z_j^2}) \quad (1)$$

수식 (1)에서  $I_{\alpha}$ 는 디지털 홀로그램,  $A_j$ 는 CGH의 입력에 해당하는 3D 객체,  $x_{\alpha}$ 와  $y_{\alpha}$ 는 디지털 홀로그램의 좌표,  $x_j, y_j$ 는 3D 객체의 좌표,  $z_j$ 는 3D 객체의 깊이 정보를 나타내고,  $p$ 는 홀로그램의 화소크기(pixel pitch)를 나타낸다[3].

수식에서 알 수 있듯이 3D 객체의 각 좌표에 대하여 독립적으로 수행가능하기 때문에 GPGPU(general-purpose GPU)를 이용하여 연산을 수행하는 연구가 많이 진행되어 있고, 그 연구를 3장에서 설명한다.

### 3. 다중 GPU 기반의 CGH 연산 기법

본 시스템의 클라이언트 측에서 사용된 CGH 연산 기법은 다음 그림 1에 나타난다. 이 기법은 다중 GPU를 기반으로 CGH연산을 효과적으로 수행할 수 있다[3]. 서버에 연결된 각 클라이언트들은 서버에서 송신한 물체의 부분적인 3차원 정보를 통하여 CGH를 통해 홀로그램을 생성, 그 결과를 다시 서버 쪽으로 전송하게 된다.

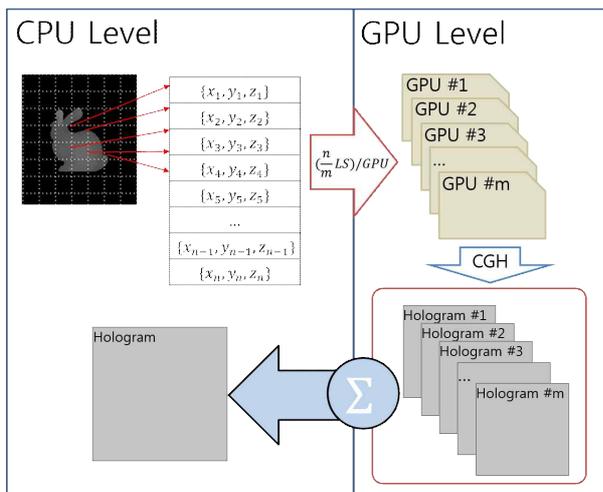


그림 1. 다중 GPU 기반 디지털 홀로그램 생성.

## 4. TCP 네트워크 통신을 이용한 디지털 홀로그램 생성 시스템

### 4.1. 서버와 각 클라이언트의 동적 연결

다음 그림 2는 TCP 네트워크 통신을 이용한 홀로그램 고속 생성 시스템의 대략적인 구조를 나타낸다. 본 시스템은 한 개의 서버 PC와 다수의 클라이언트 PC들로 구성되어 있다. 서버는 우선 그림 2에 나와 있는 'listen' 쓰레드를 통하여 설정된 포트를 개방하고 각 클라이언트들은 우선적으로 'listen' 쓰레드로 접속하게 된다. 새로운 클라이언트가 접속 할 때마다 'listen' 쓰레드에서는 새로운 쓰레드를 생성하고 생

성된 쓰레드에 접속한 클라이언트와의 통신에 필요한 정보를 넘겨준다. 이를 통해 서버는 각 클라이언트와 개별적인 쓰레드를 두고 TCP 소켓을 이용하여 통신을 한다. 또한 서버에서는 메인 쓰레드를 통하여 각 쓰레드에 명령을 내릴 수 있다.

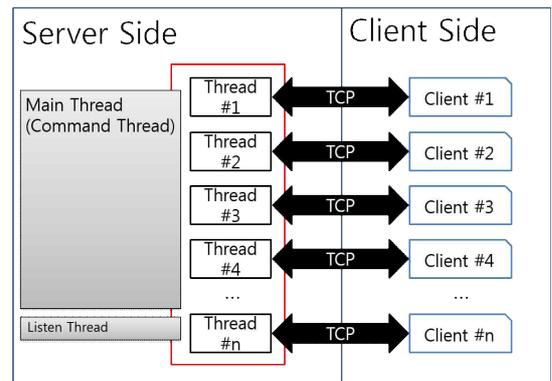


그림 2. 서버 / 클라이언트 연결도.

### 4.2. 서버와 클라이언트 간 광원 데이터 전송

다음 그림 3은 TCP 네트워크 통신을 이용한 홀로그램 고속 생성 시스템의 통신 흐름도를 나타낸다. 각 클라이언트와의 통신은 미리 약속된 흐름에 따라 진행된다. 클라이언트가 서버에 접속하면 클라이언트는 자신의 GPU 정보를 전송한다. 그 후, 메인 쓰레드에서 명령이 오면 명령에 맞는 일련의 통신을 수행한다. 그림 3에 나와 있는 'Test Command' 경우에 비교적 적고, 동일한 광원을 각 클라이언트들에게 전송하고 연산시간 만을 받아온다. 이를 토대로 그림 3에 나와 있는 'Target Command'에서는 연산하려는 3D 물체의 광원을 클라이언트의 연산 능력에 따라 분산하여 각 클라이언트들에게 송신한다. 'Target Command'는 'Test Command'과는 다르게 연산 결과를 받아 오게 된다.

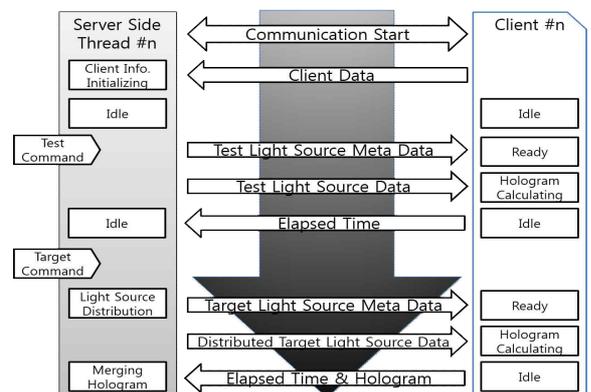


그림 3. 통신 흐름도.

### 4.3. 클라이언트 PC 연산 능력에 따른 적용적 광원 정보 분할

다음 수식 (2)와 (3)을 통하여 'Test Command'에서 측정된 각 클라이언트의 연산 시간을 통하여 각 클라이언트에 송신될 광원의 수를

계산할 수 있다. 이를 통하여 각 클라이언트에서 CGH 연산 시간은 거의 동일하게 측정 될 수 있다.

$$p_i = \frac{1}{t_i} \quad (2)$$

$$L_i = L \frac{p_i}{\sum p_k} \quad (3)$$

수식 (2)에서  $t_i$ 는 미리 측정 된 각 클라이언트의 같은 개수의 광원에 대한 CGH 연산시간을 나타낸다.  $p_i$ 는  $t_i$ 의 역수로 각 클라이언트의 CGH 연산 능력을 의미한다. 수식 (3)에서  $L$ 은 연산하려는 3D 물체의 총 광원의 수를 의미하고,  $L_i$ 는 각 클라이언트에 할당 될 광원의 수를 의미 한다.

#### 4.4 각 클라이언트들로부터 수신한 결과 누적

각 쓰레드는 앞서 설명한 바와 같이 메인 쓰레드의 명령에 따라 일련의 통신을 수행한다. ‘Target Command’에서는 받은 각 CGH 연산 결과를 최종 결과 변수에 수신이 완료된 차례대로 누적한다. 이때 최종 결과 변수는 모든 쓰레드가 공유하는 자원이기 때문에 각 클라이언트의 결과를 누적하는 시점은 크리티컬 섹션(CS: critical section)이 된다[4]. 따라서 각 쓰레드의 크리티컬 섹션에 대한 접근을 제어해야 할 필요가 있다. 이는 각 쓰레드에서 병목현상을 유발시킬 수 있다. 하지만 본 논문에서 구현한 시스템의 경우에 CGH 수행 시간은 비등하게 측정되는 반면, 통신의 특성상 결과 수신시간이 각 클라이언트마다 다르게 측정 되므로 크리티컬 섹션이 존재하더라도 총 수행시간에 큰 영향을 미칠 정도의 병목현상은 나타나지 않는다.

### 4. 실험 및 결과

다음 표 1은 각 클라이언트 별로 34093개의 광원에 대한 및 광원 데이터 송신 시간 및 CGH 연산 수행 시간을 나타낸다. 서버에서는 이미지로부터 광원 정보를 얻기 위해 OpenCV 라이브러리를 사용하였다. 각 클라이언트의 GPU는 1번 클라이언트의 경우 GeForce GTX TITAN 2 대, 2번 클라이언트의 경우 GeForce GTX 580, 3번 클라이언트의 경우 GeForce GTX TITAN, 마지막으로 4번 클라이언트의 경우 GeForce GTX 750 Ti가 장착되어 있다. 그리고 각 클라이언트들은 CGH 연산을 위해 Nvidia사에서 제공하는 GPGPU기술인 CUDA를 사용하였다[5].

표 1. 동일한 광원에 대한 데이터 송신 시간 및 CGH 연산 수행 시간

	광원 송신	CGH연산
1	1ms	351ms
2	315ms	850ms
3	1ms	591ms
4	2ms	1550ms

표 1의 각 클라이언트 별 CGH 연산 수행 시간과 수식 (1)과 (2)를 이용하여, 각 클라이언트의 CGH 연산 능력을 분석하고, 이에 따라 생성하려는 3D 물체의 광원 정보를 적응적으로 분배하여 각 클라이언트들에 송신한다. 각 클라이언트들은 수신한 광원을 이용하여 CGH 연산을 수행한 후 서버로 1024\*1024 해상도의 홀로그램을 송신하였다. 다음 표 2는 이 실험에 대한 결과이다.

표 2. TCP 네트워크 통신을 이용한 디지털 홀로그램 생성 실험 결과

	광원	광원 송신	홀로그램 수신	CGH연산
1	62554개	2ms	646ms	145ms
2	25831개	4ms	1210ms	160ms
3	37151개	1ms	1724ms	152ms
4	14119개	1ms	1703ms	179ms

### 5. 결론

본 논문에서는 GPU가 장착되지 않은 컴퓨터에서 다른 컴퓨터의 GPU의 연산 자원을 이용하여 CGH 연산을 수행할 수 있는 시스템의 구현 방법을 제안 한다. 본 시스템을 통하여 서버측은 GPU를 필요로 하지 않고, 각 클라이언트의 GPU 자원을 이용하여 부분적으로 생성된 홀로그램들을 TCP 소켓 통신을 기반으로 수신, 누적하여 온전한 홀로그램을 생성할 수 있다. 다만, 실험 결과에서 볼 수 있듯이 광원 정보를 서버에서 클라이언트로 송신하는 시간이나 각 클라이언트의 CGH 연산 시간에 비하여 연산결과를 수신하는 시간이 상대적으로 매우 높게 측정되는 것을 볼 수 있다. 이는 각 클라이언트에서 생성하는 결과 영상의 해상도가 1024\*1024의 경우 4MB(mega byte)의 용량을 갖기 때문인데, 이는 통신 데이터의 양으로는 대용량에 속한다. 따라서 이를 개선하기 위한 방법으로, 각 클라이언트에서 결과 영상을 압축한 뒤 서버로 전송하는 방법이 있다. 또한, 표 1에서 볼 수 있듯이, 광원 송신 시간이 상대적으로 높게 측정되는 경우가 발생하는데 이는 성능이 낮은 라우팅 장비를 사용하는 경우에 해당한다. 따라서, 고성능 라우팅 장비를 사용한다면, 이런 문제는 해결될 수 있다.

### 감사의 글

"본 연구는 미래창조과학부 및 정보통신기술진흥센터의 ICT융합고급연구과정지원사업의 연구결과로 수행되었음" (IITP-2015-H8601-15-1005)

### 참고문헌

- [1] B. R. Brown and A. W. Lohmann, "Complex spatial filtering with binary masks", Applied Optics, volume 5, pp. 967-969, June 1966.
- [2] H. J. Choi and Y. H. Seo, "Fast computation algorithm of Fresnel holograms using recursive addition method", Applied The Journal of Korea Information and Communications Society, volume 33, May 2008.
- [3] Song, Joongseok, et al. "Real-time generation of high-definition resolution digital holograms by using multiple graphic processing units." Optical Engineering 52.1, pp. 015803-015803, 2013.
- [4] E. Dijkstra. "Solution of a Problem in Concurrent Programming Control", Communications of the ACM, volume 8, pp. 569, 1965.
- [5] <http://docs.nvidia.com/cuda>