일반논문 (Regular Paper)
방송공학회논문지 제22권 제4호, 2017년 7월 (JBE Vol. 22, No. 4, July 2017)
https://doi.org/10.5909/JBE.2017.22.4.509
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

다수의 PC를 이용한 디지털 비디오 홀로그램의 고속 생성

박 한 훈[a], 김 창 섭[b], 박 종 일[b]‡

Fast Generation of Digital Video Holograms Using Multiple PCs

Hanhoon Park[a], Changseob Kim[b], and Jong-Il Park[b]‡

요 약

GPU를 탑재한 여러 대의 PC를 클러스터(서버-클라이언트 구조)로 구성함으로써 고해상도 디지털 홀로그램을 고속으로 생성할 수 있다. 그러나, 비디오 홀로그램의 경우, PC 사이의 데이터 전송 시간이 프레임 수에 비례하여 선형적으로 증가하기 때문에 비디오 홀로그램을 고속 생성하는 데 있어 큰 걸림돌이 된다. 본 논문에서는 이러한 데이터 전송 시간의 증가를 해결하기 위해 멀티쓰레드를 활용하는 방법을 제안한다. 기본적으로 각 클라이언트 PC에서의 홀로그램 생성은 서버로부터 광원 정보 획득, GPU를 이용한 CGH 연산, 서버로의 결과 전송의 과정으로 이루어지는데, 각 과정을 순차적으로 하지 않고 멀티쓰레딩을 통해 병렬로 수행함으로써 전체 홀로그램 생성 시간에서 데이터 전송 시간의 비율을 크게 줄일 수 있다. 실험을 통해, 150 프레임을 가지는 고해상도 비디오 홀로그램을 생성하는 시간을 약 30% 줄일 수 있음을 확인하였다.

Abstract

High-resolution digital holograms can be quickly generated by using a PC cluster that is based on server-client architecture and is composed of several GPU-equipped PCs. However, the data transmission time between PCs becomes a large obstacle for fast generation of video holograms because it linearly increases in proportion to the number of frames. To resolve the problem with the increase of data transmission time, this paper proposes a multi-threading-based method. Hologram generation in each client PC basically consists of three processes: acquisition of light sources, CGH operation using GPUs, and transmission of the result to the server PC. Unlike the previous method that sequentially executes the processes, the proposed method executes in parallel them by multi-threading and thus can significantly reduce the proportion of the data transmission time to the total hologram generation time. Through experiments, it was confirmed that the total generation time of a high-resolution video hologram with 150 frames can be reduced by about 30%.

Keyword : CGH, digital video hologram, PC cluster, multi-threading, CUDA

a) 부경대학교 전자공학과(Department of Electronic Engineering, Pukyong National University)
b) 한양대학교 컴퓨터소프트웨어학과(Department of Computer Software, Hanyang University)
‡ Corresponding Author : 박종일(Jong-Il Park)
E-mail: jipark@hanyang.ac.kr
Tel: +82-2-2220-0368
ORCID: http://orcid.org/0000-0003-1000-4067
※ 이 논문은 부경대학교 자율창의학술연구비(2016년)에 의하여 연구되었음.
· Manuscript received June 2, 2017; Revised July 5, 2017; Accepted July 5, 2017.

Ⅰ. 서 론

3차원 TV 및 영화가 보편화되고 보다 향상된 입체감, 몰입감을 가진 미디어에 대한 수요가 늘어나면서, 입체 영상을 표현하는 궁극적인 기술인 홀로그래피(holography)에 대한 관심이 다시 커지고 있다. 홀로그래피는 실 공간에 디스플레이된 3차원 영상을 아무런 부가 장치 없이 육안으로 관찰이 가능한 영상 기술이다. 이때, 디스플레이된 3차원 영상을 홀로그램(hologram)이라고 하는데, 홀로그램은 물체의 3차원 정보와 광학(optical) 장치를 이용하여 간섭(interference) 패턴을 필름에 기록하고 이를 다시 광학 장치를 이용하여 재현하여 얻어진다. 그러나, 광학 장치를 이용한 홀로그램 생성은 비싼 구축 비용, 주변 광원의 차단, 무진동 환경 구축 등 여러 구현상의 제약을 가진다[1]. 이에 광학 장치의 홀로그램 생성 과정을 식 (1)과 같이 수학적으로 모델링하여 범용 컴퓨터로 홀로그램을 생성하는 CGH (computer-generated holography) 방법이 제안되었다(그림 1 참조)[2].

식 (1)에서 I는 홀로그램의 밝기, A는 3차원 물체(광원)의 밝기, λ는 참조 파의 파장, p는 픽셀 피치, L은 3차원 물체(광원)의 수, W_h와 H_h는 홀로그램의 해상도(가로, 세로 길이)를 의미한다. 그러나, 식 (1)에서 알 수 있듯이 CGH 방법은 계산 시간이 홀로그램의 해상도와 광원의 수에 비례하여 급속하게 증가하기 때문에 높은 계산 복잡도를 가진다는 단점이 있다. 예를 들어, L = 10K이고, 홀로그램의 해상도가 1920 × 1080일 때, 식 (1)의 시그마 내부 연산(곱셈, 나눗셈, cos함수, 제곱 등)은 약 207억 번 반복된다.

CGH 방법의 높은 계산 복잡도를 해결하기 위해 크게 두 가지 방법론이 제안되었다. 우선, CGH 계산 결과들을 미리 룩업테이블에 저장해두거나[3], 특정 픽셀들의 CGH 결과로부터 나머지 픽셀들의 CGH 결과를 반복적으로 추정하거나[4], 코사인 함수를 근사화하는[5] 등의 소프트웨어적으로 개선하는 방법론이 있다. 그러나, 이러한 방법들은 많은 양의 메모리를 필요로 하거나, 고해상도를 가진 홀로그램을 고속으로 생성할 수 있을 정도로 CGH 방법의 계산 복잡도를 향상시킬 수는 없었다. 이러한 소프트웨어 기반 방법론의 한계를 극복하기 위한 방법론으로, FPGA(field programmable gate array)를 이용하여 CGH 계산 자체를 하드웨어적으로 고속화하거나[4], 식 (1)의 연산이 홀로그램의 각 픽셀에 대해 독립적으로 수행된다는 점을 고려하여, 다수의 GPU(graphic processing unit)[6,7], PC 클러스터[8,9] 등

$$I(x_h,y_h)=\sum_{l}^{L}A_l(x_l,y_l,z_l)\cos\left(\frac{2\pi z_l}{\lambda}+\frac{\pi p^2\{(x_h-x_l)^2+(y_h-y_l)^2\}}{\lambda z_l}\right),\quad x_h=0,1,\ldots,W_h-1,\ y_h=0,1,\ldots,H_h-1. \tag{1}$$

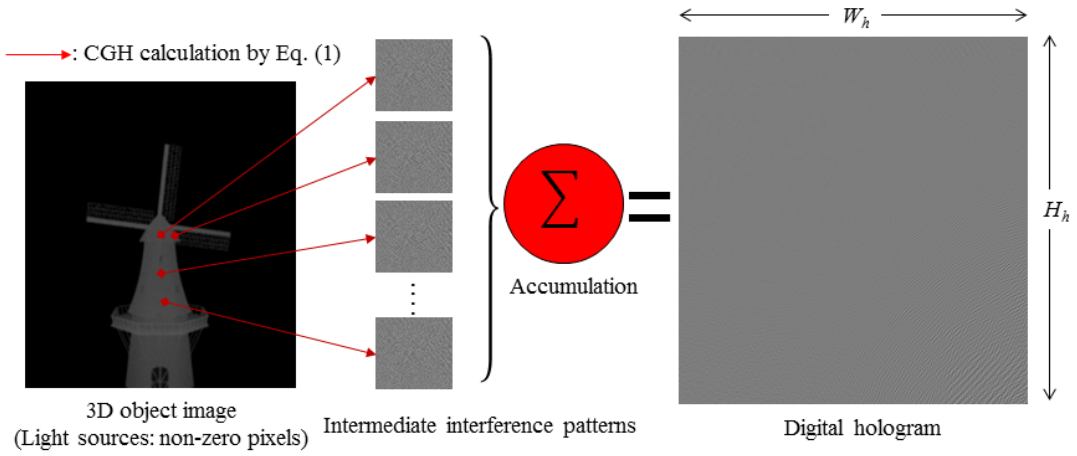

그림 1. CGH에 의한 홀로그램 생성 과정
Fig. 1. Process of generating a hologram by CGH

의 하드웨어를 이용하여 병렬 처리하는 방법들이 제안되었다. 특히, [9]의 경우 성능이 서로 다른 다수의 GPU를 가진 PC를 서버-클라이언트 형태의 클러스터로 구성하고, 서버는 각 클라이언트 PC의 성능을 주기적으로 확인하여 식 (1)의 연산에 필요한 계산량을 배분함으로써, 각 클라이언트 PC에서 GPU 기반 병렬처리를 통해 홀로그램 연산을 가속화할 뿐 아니라, 클러스터의 성능을 최적화할 수 있었다. 그러므로, 기존 방법들과 달리 수백만 개의 초다광원을 갖는 물체의 HD급 홀로그램 고속 생성이 가능했다. 또한, 서버-클라이언트 형태의 특성 상, 클라이언트 PC를 추가/제거함으로써 클러스터의 확장/축소가 용이한 장점을 가졌다. 그러나, 각 클라이언트 PC에서 계산된 식 (1)의 부분적인 연산 결과들은 서버 PC로 전송되어 통합되는데, 이러한 서버-클라이언트 간의 데이터 전송 시간이 전체 홀로그램 생성 시간 내에서 많은 비중을 차지했다[9]. 홀로그램의 해상도가 크거나 클라이언트 PC 수가 많을 경우 전송 시간에 의한 홀로그램 생성 속도 저하는 더욱 뚜렷했다.

비디오 홀로그램은 각 프레임이 홀로그램으로 구성된 비디오를 말하며, 각 프레임은 독립적으로 생성된다. 따라서, 각 프레임에 기존 홀로그램 생성 방법들을 적용하여 다수의 프레임을 가지는 비디오 홀로그램을 생성할 수 있다. 특히, [9]의 방법을 사용함으로써 초다광원을 가지는 물체에 대해 고해상도의 비디오 홀로그램을 고속으로 생성할 수 있다. 그러나, 앞서 설명한 것처럼, [9]의 방법은 클라이언트 PC로부터 서버 PC로의 홀로그램 전송 시간과 관련된 문제를 가지고 있는데, 비디오 홀로그램을 생성할 경우 더욱 치명적일 수 있다. 즉, 전송 시간은 프레임 수에 비례해서 증가하므로 많은 프레임을 가진 비디오 홀로그램의 생성은 생성과는 관계없는 과정에서 많은 시간을 소비하게 된다. 또한, 비디오 홀로그램을 실시간으로 생성할 필요가 있을 경우, 각 프레임을 생성하기 위한 전송 시간은 큰 걸림돌이 된다. 그러므로, 본 논문에서는 이러한 각 클라이언트 PC에 의한 서버 PC로의 홀로그램 전송 시간을 줄이기 위한 방안을 제시한다.

본 논문은 다음과 같이 구성된다. II 장에서는 서버-클라이언트 형태의 PC 클러스터를 이용한 기존 홀로그램 고속 생성 방법[9]에 대해 설명하고, III 장에서는 서버-클라이언트 간의 데이터 전송 시간의 비중을 줄이기 위한 제안 방법을 설명한다. IV 장에서는 실험 및 분석을 통해 제안 방법의 성능을 검증하고, V 장에서는 결론 및 향후 과제를 제시한다.

II. PC 클러스터를 이용한 홀로그램 고속 생성

다수의 GPU를 탑재한 여러 대의 PC를 클러스터로 구성함으로써 고해상도의 홀로그램을 고속으로 생성할 수 있다[9]. 구체적으로 설명하면, 원격의, 성능이 서로 다른 PC(한 대의 서버 PC와 다수의 클라이언트 PC)를 TCP/IP 네트워크로 연결하여 클러스터로 구성한다. 서버 PC에서는 홀로그램을 생성하기 위한 물체의 광원 정보(깊이 영상 형태로 제공됨)를 읽은 후, 각 클라이언트 PC의 계산 능력을 측정한다. 이는 미리 동일한 양의 광원 정보를 모든 클라이언트 PC에 전송하고, 각 클라이언트 PC에서 측정된 CGH 연산 시간(t_i)을 사용한다. 이후, 그림 2에서 보는 것처럼 서버 PC는 각 클라이언트 PC의 계산 능력에 비례하도록 식 (2)와 같이 물체의 광원 정보를 배분한다.

$$L_i = L\frac{1/t_i}{\sum_{k=1}^{N_{clients}}(1/t_k)}, (i=1,2,...,N_{dients}). \qquad (2)$$

각 클라이언트 PC는 할당받은 광원 정보로부터 GPU를 이용하여 병렬 처리를 통해 부분 홀로그램(그림 1의 간섭 패턴)을 생성한다. 각 PC에서의 부분 홀로그램 생성 역시 병렬로 진행된다. 계산 능력에 적응적으로 광원을 분배했기 때문에, 각 클라이언트 PC에서의 부분 홀로그램 생성은 거의 같은 시간에 종료된다. 각 클라이언트 PC에서 생성된 부분 홀로그램은 다시 서버 PC로 전송되며, 서버 PC는 이를 누적하여 최종적인 홀로그램을 완성한다.

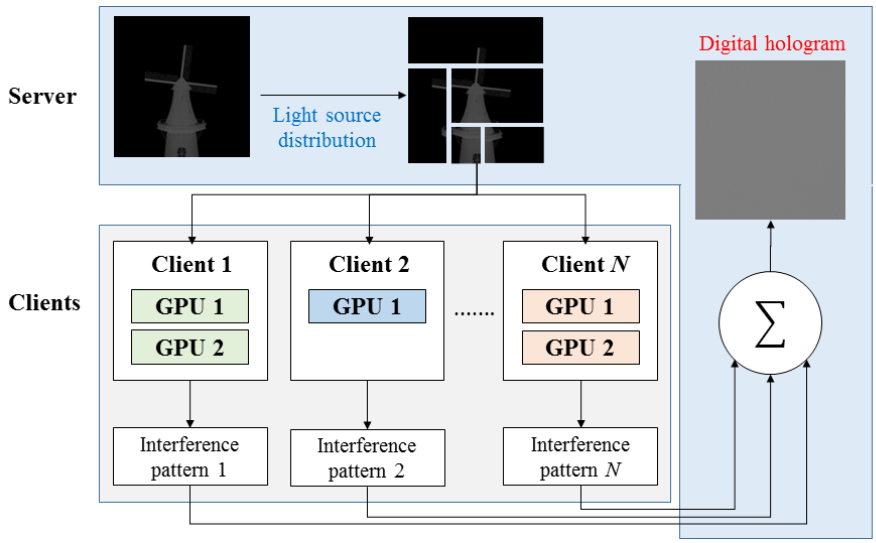

그림 2. PC 클러스터를 이용한 CGH 연산의 병렬 분산 처리
Fig. 2. Parallel and distributed processing of the CGH operations using PC cluster

Ⅲ. PC 클러스터를 이용한 비디오 홀로그램 고속 생성

Ⅱ장에서 설명한 것처럼, PC 클러스터를 이용한 기존 방법[9]에서는 각 클라이언트 PC에서 홀로그램을 생성한 후 홀로그램을 서버 PC로 전송한다. 비디오 홀로그램의 경우 각 프레임에 대해 이러한 생성, 전송이 반복되는데, 두 과정은 순차적으로 진행되기 때문에 전송 시간이 프레임 수에 비례적으로 증가하며 비디오 홀로그램을 고속 생성하는 데 있어 전송 시간이 큰 걸림돌이 되게 된다. 이러한 전송 시간의 비중을 줄이기 위해, 네트워크의 HW/SW를 개선하여 전송 과정 자체의 수행 속도를 증가시키는 것이 일차적인 해결 방안이 될 수 있다. 그러나, 네트워크 환경을 개선하는 데는 물리적인 한계가 있기 때문에 클라이언트 PC의 수나 홀로그램 해상도가 일정 수준 이상이 되면 전송 시간을 줄이기 위한 근본적인 해결 방안이 필요하다. 본 논문에서는 생성, 전송 과정을 순차적으로 반복하는 것을 병렬로 진행되도록 변형하여 전송 시간을 줄이는 방법을 제안한다. 즉, 각 클라이언트 PC에서 n번째 프레임의 부분 홀로그램을 생성하고 나면, 결과를 서버 PC로 전송하면서 동시에 n+1번째 프레임의 부분 홀로그램 생성을 병렬로 수행하도록 한다. 결국, 각 프레임의 홀로그램 생성은 각 클라이언트 PC에서의 부분 홀로그램 생성 시간 t_g와 서버 PC로의 전송 시간 t_t 중에 더 큰 시간에 의해 결정되며, 그 만큼 생성 시간은 단축될 수 있다. 예를 들어, 일반적으로는 t_t보다 t_g가 더 크기 때문에, 전체 홀로그램을 생성하는 시간은 t_g와 거의 같으며 t_t 만큼의 시간이 단축될 수 있다. 다시 말해, 전송 시간은 홀로그램 생성에 거의 영향을 주지 않게 된다.

생성과 전송을 동시에 수행하기 위한 방법은 매우 간단하다. 그림 3에서 보는 것처럼, 클라이언트 PC와 서버 PC의 주요 프로세스들을 쓰레드(thread) 함수로 구현하여 각 프로세스가 병렬로, 독립적으로 수행되도록 한다. 서버 PC에서 Control 쓰레드는 각 프레임에서의 광원 정보를 각 클라이언트 PC로 어떻게 분배할 지를 결정하고, Collect 쓰레드는 각 클라이언트 PC에서 계산된 부분 홀로그램 결과를 누적하여 최종적인 홀로그램을 완성한다. 클라이언트 PC에서 Calculation 쓰레드는 자신에게 할당된 광원 정보를 이용하여 부분 홀로그램을 생성한다. 서버 PC와 클라이언트 PC의 Send와 Receive 쓰레드는 광원 정보나 홀로그램 결과와 같은 데이터를 주고받는다. 각 쓰레드들은 메시지 패싱(message passing) 방식[10]으로 서로 통신하면서 프로세스를 효과적으로 병렬 수행한다. 각 클라이언트 내부 쓰레드 함수 중에 홀로그램을 생성하는 쓰레드(Calculation)와 생성된 결과를 서버 PC로 보내는 쓰레드(Send)가 병렬

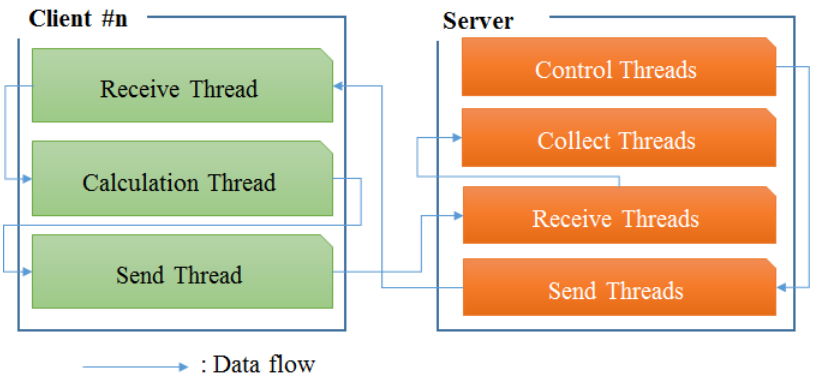

그림 3. 제안 방법의 프로그램 구조
Fig. 3. Program structure of the proposed method

로 수행되기 때문에 생성과 전송은 동시에 수행될 수 있다(광원 정보를 수신하는 Receive 쓰레드도 동시에 수행되지만, 다른 쓰레드에 비해 소요 시간이 매우 작기 때문에 동시 수행에 따른 효과는 크지 않다). 클라이언트 PC에서의 각 쓰레드 내 혹은 사이의 프로세스 흐름을 좀 더 자세히 설명하면 다음과 같다(그림 4 참조). Receive 쓰레드는 서버 PC로부터 현재 프레임의 광원 정보를 얻고, Calculation 쓰레드로 메시지를 보내서 홀로그램 생성을 수행하도록 하고 다음 프레임의 홀로그램 생성을 위한 광원 정보를 기다린다. Calculation 쓰레드는 홀로그램 생성이 끝나면 Send 쓰레드로 메시지를 보내서 서버 PC로 결과를 전송하도록 하고 다음 프레임의 홀로그램 생성을 위해 Receive 쓰레드의 메시지를 기다린다. Send 쓰레드는 현재 프레임의 홀로그램 생성 결과를 전송하고 나면 다음 프레임의 홀로그램 생성 전송을 위해 Calculation 쓰레드의 메시지를 기다린다. 결국, Calculation 쓰레드가 현재 프레임의 홀로그램 생성을 수행하는 동안 Receive 쓰레드는 다음 프레임의 홀로그램 생성을 위한 광원 정보를 받을 수 있고, Send 쓰레드가 현재 프레임의 홀로그램 결과를 서버 PC로 보내는 동안 Calculation 쓰레드는 다음 프레임의 홀로그램 생성을 동시에 수행하게 된다.

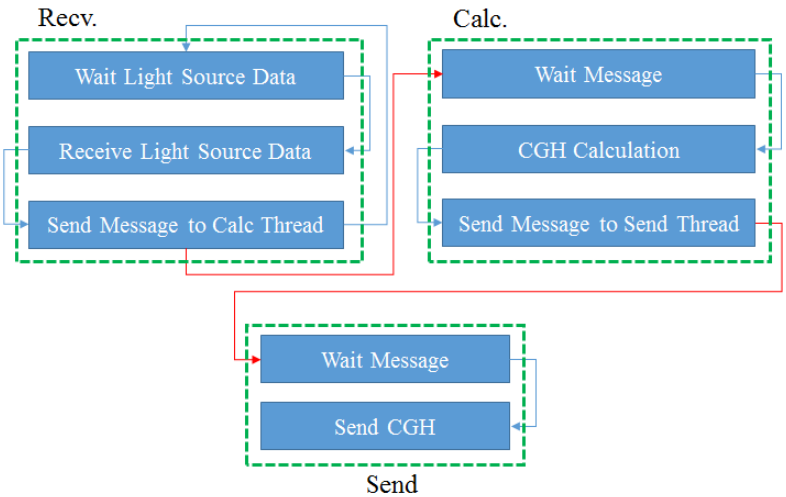

그림 4. 클라이언트 PC의 각 쓰레드에서의/사이의 프로세스 흐름
Fig. 4. Process flow in/between each thread of client PCs

Ⅳ. 실험 결과 및 분석

실험을 위해 한 대의 서버 PC와 4 대의 클라이언트 PC로 구성된 PC 클러스터를 구성하였다(표 1 참조). 각 PC는 기가비트 이더넷 허브(Cisco SG300-28)로 연결되었으며, 소켓 방식으로 TCP/IP 통신을 수행한다. GPU를 이용한 병렬처리를 위해 CUDA API[11]가 사용되었다. 기존 방법[9]과 제안 방법의 성능을 비교, 분석하기 위해 크게 세 가지 조건(클라이언트 PC 수, 광원 수, 홀로그램 해상도)을 각각 변화시켜 가면서 150 프레임을 가지는 비디오 홀로그램(windmill[9]) 생성 시간을 측정하였다. 추가적으로, 기존 방법에서 광원을 적응적으로 분배하는 것과 달리 모든 클라이언트 PC에 균일하게 분배하는 방법에 대해 제안 방법 적용에 의한 성능 변화를 분석하였다. 광원 분배 방식에 상관없이 기존 방법의 경우 모든 프로세스를 단일 쓰레드로 진행하지만, 공정한 비교를 위해 기존 방법의 구현 역시 제안 방법의 프로그램 구조를 그대로 사용하였다. 다만, 각 쓰레드가 순차적으로 진행되도록 하기 위해, 그림 4에서 Send 쓰레드는 홀로그램 결과를 서버 PC로 전송 완료하면 Receive 쓰레드로 메시지를 보내도록 하고, Receive 쓰레드는 Send 쓰레드의 전송 완료 메시지를 수신한 후 다음 프레임의 광원 정보를 받도록 하였다.

1. 클라이언트 PC 수의 변화에 따른 홀로그램 생성 시간

평균적으로 23K의 광원을 가지는 각 프레임에 대해 1024 × 1024 크기의 해상도를 가지는 홀로그램을 클라이언트 PC 수를 증가시키면서 생성하였다. 표 2는 기존 방법과 제안 방법의 생성 시간을 보여준다. 클라이언트 수에 상관없이 각 클라이언트의 성능에 맞춰 광원은 적절히 분배되었으며, 이는 각 클라이언트가 각 프레임의 홀로그램을 생성하기 위해 걸린 시간(식 (1)의 CGH 계산 시간)이 유사하다는 것을 통해 알 수 있다. 각 클라이언트의 홀로그램 생성 시간은 클라이언트 수가 증가할수록 감소했으나, 서버로의 홀로그램 전송 시간은 증가했다. 전송 시간 자체는 제안 방법이 더 크게 나타났으며, 이는 전송과 생성을 동시에 수행하기 때문인 것으로 판단된다. 그러나, 기존 방법은 생성과 전송이 순차적으로 진행되지만, 제안 방법은 전송과 생성이 동시에 진행되기 때문에 비디오 홀로그램 전체

표 1. 실험에 사용된 PC 클러스터의 사양
Table 1. Specifications of the PC cluster used in experiments

Role	GPU (Quantity)	CPU	RAM	OS
Server	GeForce GTX 770 (1)	i7-3770 3.4GHz	32GB	Windows Embedded 8.1 Industry Pro
Client_1	GeForce GTX 580 (2)	i7-4790 3.6GHz	32GB	Windows Embedded 8.1 Industry Pro
Client_2	GeForce GTX 980 (2)	i7-4790K 4.0GHz	16GB	Windows 8.1 Enterprise
Client_3	GeForce GTX 980 Ti (1)	i7-4790K 4.0GHz	32GB	Windows 10 Education
Client_4	GeForce GTX 680 (1)	i7-4770K 3.5GHz	8GB	Windows Embedded 8.1 Industry Pro

표 2. 클라이언트 PC 수에 따른 홀로그램 생성 시간 [단위: 초]
Table 2. Hologram generation time according to the number of client PCs [unit: sec]

# of clients		# of assigned light sources (L_i)	CGH calc. time per frame	Trans. time per frame		Total generation time		
				[9]	Proposed	[9] (= a)	Proposed (= b)	Ratio (= b/a)
2	Client_1	13K	0.575	0.043	0.041	95.352	88.659	0.93
	Client_4	10K	0.577	0.052	0.042			
3	Client_1	6K	0.265	0.061	0.112	55.696	41.772	0.75
	Client_3	12.5K	0.234	0.076	0.113			
	Client_4	4.5K	0.262	0.056	0.109			
4	Client_1	3.5K	0.169	0.092	0.135	39.989	28.957	0.72
	Client_2	10K	0.134	0.110	0.139			
	Client_3	7K	0.132	0.098	0.162			
	Client_4	2.5K	0.147	0.081	0.132			

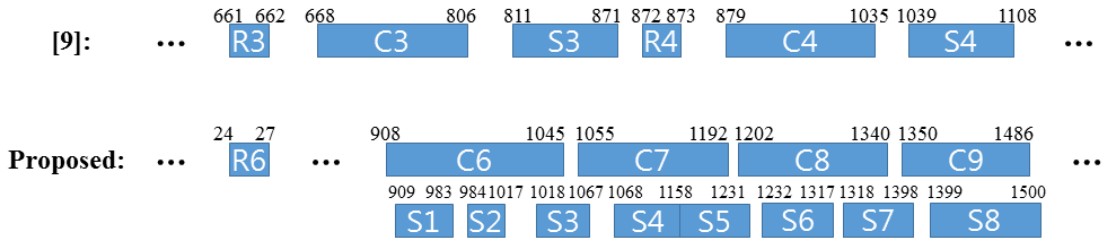

그림 5. Client_2의 각 쓰레드의 타임스탬프 다이어그램
Fig. 5. Time-stamp diagram of each thread of Client_2

생성 시간은 제안 방법이 더 작았다. 두 방법에 의한 홀로그램 생성 과정의 타임스템프 다이어그램을 보면(그림 5 참조), 기존 방법은 모든 과정이 순차적으로 진행되며 각 프레임의 생성(C*n*), 전송(R*n*)을 완료한 후에 다음 프레임의 생성이 진행되지만, 제안 방법은 각 프레임의 생성, 전송이 여러 프레임에 걸쳐 병렬로 진행됨을 알 수 있다. 두 방법의 전체 생성 시간 사이의 비율을 보면, 제안 방법의 성능은 클라이언트 수가 증가할수록 더욱 뚜렷했으며, 이는 클라이언트 수가 증가할수록 전송 시간이 늘어나고 전체 생성 시간은 줄어들기 때문에 전체 생성 시간에서의 전송 시간의 비중이 커지기 때문이다. 4 대의 클라이언트 PC를 이용할 경우, 비디오 홀로그램(150 프레임) 생성 시간의 28% 정도를 줄일 수 있었다.

2. 광원 수의 변화에 따른 홀로그램 생성 시간

4 대의 클라이언트 PC를 가지는 PC 클러스터를 이용하여 광원 수를 증가시키면서 1024 × 1024 크기의 해상도를 가지는 홀로그램을 생성하였다. 광원은 적응적 분배 방식에 의해 광원 수에 상관없이 client_1:client_2:client_3:client_4 = 0.15:0.44:0.30:0.11의 비율로 분배가 되었으며, 표 3에서 보는 것처럼 각 클라이언트에서의 홀로그램 생성 시간은 광원 수에 비례하여 증가하였다. 홀로그램 전송 시간의 경우, 클라이언트 수나 홀로그램 해상도가 변하지 않으면 거의 변하지 않기 때문에(실제로는 각 클라이언트 PC의 실행 환경이나 네트워크 환경의 변화로 인해 달라졌지만), 광원 수가 증가할수록 전체 홀로그램 생성 시간에 대한 전송 시간의 비중이 감소하게 되어 제안 방법의 성능은 줄어들었다. 5K의 광원 수에 대해 제안 방법은 비디오 홀로그램(150 프레임) 생성 시간을 약 37% 줄일 수 있었지만, 50K 이상의 광원 수를 가질 경우, 9% 정도를 줄이는 데 그쳤다.

3. 홀로그램 해상도의 변화에 따른 홀로그램 생성 시간

평균적으로 23K의 광원을 가지는 각 프레임에 대해 4 대의 클라이언트 PC를 가지는 PC 클러스터를 이용하여 서로 다른 해상도를 가지는 홀로그램을 생성하였다. 광원은 표 2에서와 같이 분배되었으며, 각 클라이언트 PC에서의 홀로그램 생성 시간과 서버로의 홀로그램 전송 시간은 홀로그램 해상도에 비례하여 함께 증가하였다. 결과적으로, 전체 홀로그램 생성 시간 대비 전송 시간의 비중은 홀로그램 해상도의 변화에 상관없이 거의 일정했으며, 제안 방법의 성능 역시 큰 변화가 없었다. 제안 방법을 통해 대략 28~33%

표 3. 광원 수에 따른 홀로그램 생성 시간 [단위: 초]
Table 3. Hologram generation time according to the number of light sources [unit: sec]

# of light sources	CGH calc. time per frame	Trans. time per frame		Total generation time		
		[9]	Proposed	[9] (= *a*)	Proposed (= *b*)	Ratio (= *b*/*a*)
~ 5K	0.033	0.158	0.138	36.231	22.900	0.63
~ 13K	0.080	0.140	0.148	37.961	23.683	0.62
~ 23K	0.146	0.095	0.142	39.989	28.957	0.72
~ 50K	0.311	0.113	0.128	69.803	63.737	0.91

표 4. 홀로그램 해상도에 따른 홀로그램 생성 시간 [단위: 초]
Table 4. Hologram generation time according to the hologram resolution [unit: sec]

Hologram resolution	CGH calc. time per frame	Trans. time per frame		Total generation time		
		[9]	Proposed	[9] (= *a*)	Proposed (= *b*)	Ratio (= *b*/*a*)
512 × 512	0.036	0.026	0.039	12.743	8.579	0.67
1024 × 1024	0.146	0.095	0.142	39.989	28.957	0.72
1536 × 1536	0.320	0.285	0.319	98.079	66.973	0.68

정도의 시간을 줄일 수 있었으며, 512 × 512, 1024 × 1024, 1536 × 1536의 해상도를 가지는 비디오 홀로그램(150 프레임)을 생성하는 데 각각 약 8.6초, 29초, 67초가 걸렸다.

4. 광원의 균일 분배 방식에 제안 방법 적용

앞의 실험에서는 각 클라이언트 PC의 성능을 고려하여 광원을 분배하는 적응적 분배 방식[9]을 사용함으로써, 각 클라이언트 PC가 주어진 광원에 대해 홀로그램을 생성하는 시간은 크게 차이가 나지 않았다. 그러나, 클라이언트 PC의 성능을 고려하지 않고 광원을 균일하게 분배하는 균일 분배 방식을 사용했을 때, 표 5에서 보는 것처럼 각 클라이언트 PC의 홀로그램 생성 시간은 크게 달라졌으며 성능이 떨어지는 클라이언트 PC로 인해 전체 비디오 홀로그램 생성 시간은 크게 증가하였다(적응적 분배 방식에 비해 대략 1.5배 증가). 다시 말해, 그림 6에서 보는 것처럼 성능이 좋은 클라이언트 PC에서 각 프레임의 홀로그램을 빠르게 생성, 전송하더라도 서버는 성능이 떨어지는 클라이언트 PC의 결과를 기다려야 하기 때문에 전체 비디오 홀로그램 생성 시간은 성능이 떨어지는 클라이언트 PC의 홀로그램 생성 시간에 의해 결정되었다. 반면, 홀로그램 전송 시간은 클라이언트 PC의 성능에 크게 영향을 받지 않기 때문에 적응 분배 방식에 비해 균일 분배 방식에서는 상대적으로 비중이 작았다. 결과적으로, 균일 분배 방식을 사용할 경우 제안 방법의 적용으로 인한 성능 개선은 줄어들었지만, 여전히 비디오 홀로그램 생성 시간을 크게 줄일 수 있었다(생성 시간이 약 15% 감소).

V. 결론 및 향후 과제

본 논문에서는 PC 클러스터를 이용하여 홀로그램을 생성하는 기존 방법을 개선하여 비디오 홀로그램을 고속 생성하기 위한 방법을 제안하였다. 즉, PC 클러스터를 이용할

표 5. 광원 분배 방식에 따른 홀로그램 생성 시간 [단위: 초]
Table 5. Hologram generation time according to the light source distribution method [unit: sec]

Distribution method	CGH calc. time per frame		Trans. time per frame		Total generation time		
			Sequential*	Parallel*	Sequential (= *a*)	Parallel (= *b*)	Ratio (= *b*/*a*)
Uniform†	Client_1	0.254	0.080	0.100	61.123	51.897	0.85
	Client_2	0.077					
	Client_3	0.110					
	Client_4	0.336					
Adaptive[9]	0.146		0.095	0.142	39.989	28.957	0.72

† The light sources are evenly distributed to the client PCs regardless of their computing power.
* Sequential: sending the resulting hologram to the server after finishing its generation, parallel: sending the t-th hologram while generating the (*t*+1)-th hologram.

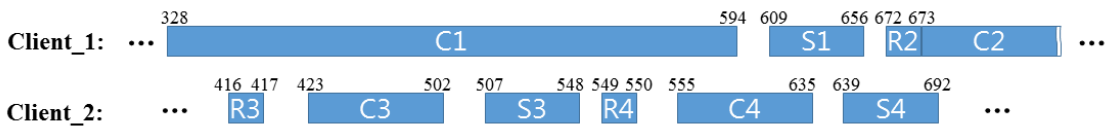

그림 6. 균일 광원 분배 방식에서의 Client_1과 Client_2의 각 쓰레드의 타임스탬프 다이어그램
Fig. 6. Time-stamp diagram of each thread of Client_1 and Client_2 when light sources are evenly distributed

경우, PC 사이의 데이터 전송 시간이 비디오 홀로그램의 고속 생성에 걸림돌이 되는데 이를 해결하기 위해 멀티쓰레드를 활용하는 방안을 제시하였다. 제안 방법을 적용함으로써, 데이터 전송으로 인한 비디오 홀로그램 생성 속도 저하는 거의 발생하지 않았다. 4 대의 클라이언트 PC와 한 대의 서버 PC로 구성된 PC 클러스터를 이용할 경우, 평균적으로 23K의 광원을 가지고 1024 × 1024 크기의 해상도를 가지는 비디오 홀로그램(150 프레임)을 생성하는 데 약 28.96초(기존 방법 대비 28% 감소)가 소요되었다.

향후, 보다 다양한 환경에서 제안 방법의 성능을 검증하는 실험이 뒷받침되어야 할 것이다. 특히, 네트워크 환경의 변화나 클러스터를 구성하는 PC 수나 성능 변화는 제안 방법의 성능에 큰 영향을 주기 때문에 이에 대한 분석이 필요하다.

참 고 문 헌 (References)

[1] H. Yoshikawa and J. Tamai, "Holographic image compression by motion picture coding," *Proc. of SPIE*, Vol. 2652, pp. 2-9, 1997.
[2] B. R. Brown and A. W. Lohmann, "Complex spatial filtering with binary masks," *Applied Optics*, Vol. 5, pp. 967-969, 1966.
[3] S. C. Kim and E. S. Kim, "Effective generation of digital holograms of three-dimensional objects using a novel look-up table method," *Applied Optics*, Vol. 47, No. 19, pp. D55-D62, 2008.
[4] T. Shimobaba and T. Ito, "An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition," *Comput. Phys. Commun.*, Vol. 138, No. 1, pp. 44-52, 2001.
[5] T. Nishitsuji, T. Shimobaba, T. Kakue, D. Arai, and T. Ito, "Simple and fast cosine approximation method for computer-generated hologram calculation," *Optics Express*, Vol. 23, No. 25, pp. 32465-32470, 2015.
[6] J. Song, J. Park, H. Park, and J.-I. Park, "Real-time generation of high-definition resolution digital holograms by using multiple graphic processing units," *Optical Engineering*, Vol. 52, No. 1, pp. 015803, 2013.
[7] T. Sugawara, Y. Ogihara, and Y. Sakamoto, "Fast point-based method of a computer-generated hologram for a triangle-patch model by using a graphics processing unit," *Applied Optics*, Vol. 55, pp. A160-A166, 2016.
[8] N. Takada, T. Shimobaba, H. Nakayama, A. Shiraki, N. Okada, M. Oikawa, N. Masuda, and T. Ito, "Fast high-resolution computer- generated hologram computation using multiple graphics processing unit cluster system," *Applied Optics*, Vol. 51, No. 30, pp. 7303-7307, 2012.
[9] J. Song, C. Kim, H. Park, and J.-I. Park, "Fast generation of a high-quality computer-generated hologram using a scalable and flexible PC cluster," *Applied Optics*, Vol. 55, No. 13, pp. 3681-3688, 2016.
[10] Messages and Message Queues, https://msdn.microsoft.com/ko-kr/library/windows/desktop/ms632590(v=vs.85).aspx (accessed May 23, 2017).
[11] CUDA Zone, https://developer.nvidia.com/cuda-zone (accessed May 27, 2017).

저 자 소 개

박 한 훈

- 2000년 : 한양대학교 전자통신전파공학과 공학사
- 2002년 : 한양대학교 대학원 전자통신전파공학과 공학석사
- 2007년 : 한양대학교 대학원 전자통신전파공학과 공학박사
- 2008년 ~ 2011년 : NHK방송기술연구소 박사후연구원
- 2012년 ~ 현재 : 부경대학교 전자공학과 부교수
- ORCID : http://orcid.org/0000-0002-6968-4565
- 주관심분야 : 증강현실, 인간컴퓨터상호작용, 3차원 영상처리/비전 등

김 창 섭

- 2017년 : 한양대학교 컴퓨터공학부 공학사
- 2017년 ~ 현재: 한양대학교 대학원 컴퓨터소프트웨어학과 석사과정
- ORCID : http://orcid.org/0000-0002-2524-4269
- 주관심분야 : 영상처리, 디지털 홀로그램, GPGPU

저 자 소 개

박 종 일

- 1987년 : 서울대학교 전자공학과 공학사
- 1989년 : 서울대학교 전자공학과 공학석사
- 1995년 : 서울대학교 전자공학과 공학박사
- 1992년 ~ 1994년 : 일본 NHK방송기술연구소 객원연구원
- 1995년 ~ 1996년 : 한국방송개발원 선임연구원
- 1996년 ~ 1999년 : 일본 ATR지능영상통신연구소 연구원
- 1999년 ~ 현재 : 한양대학교 공과대학 컴퓨터공학부 교수
- ORCID : http://orcid.org/0000-0003-1000-4067
- 주관심분야 : 증강현실, 계산사진학, 3차원 컴퓨터비전, 인간컴퓨터상호작용