

특집논문 (Special Paper)

방송공학회논문지 제19권 제5호, 2014년 9월 (JBE Vol. 19, No. 5, September 2014)

<http://dx.doi.org/10.5909/JBE.2014.19.5.590>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

## 디지털 홀로그램의 비디오 서비스를 위한 시스템 설계

이 윤 혁<sup>a)</sup>, 서 영 호<sup>b)</sup>, 김 동 육<sup>a)†</sup>

## System Architecture for Digital Hologram Video Service

Yoon-Hyuk Lee<sup>a)</sup>, Young-Ho Seo<sup>b)</sup>, and Dong-Wook Kim<sup>a)†</sup>

### 요 약

본 논문은 디지털 홀로그램 비디오를 서비스할 수 있는 서비스 시스템을 제안하는 것을 목적으로 한다. 이 시스템은 현존하는 2차원 또는 3차원 영상/비디오를 서비스하는 시스템의 프레임, 즉 데이터 획득, 처리, 전송, 수신, 복원의 과정을 그대로 따른다고 가정한다. 이 시스템은 수직 리그로부터 획득한 영상을 보정하는 전처리 과정을 수행하고, 시청자의 시점에 해당하는 가상 시점의 영상을 생성하고 홀로그램을 생성한다. 이전 연구에서 구현하지 못한 홀로그램에 대한 신호처리 요소들을 추가하고 성능을 개선하였다. 본 논문에서는 GPU를 늘리면서 이전 버전<sup>[25]</sup>의 홀로그램 생성시간을 1/3로 감소 시켰고 추가적으로 홀로그램의 압축, 워터마킹, 암호화 모듈을 추가하였다.

### Abstract

The purpose of this paper is to propose a service system for a digital hologram video, which has not been published yet. This system assumes the existing service framework for 2-dimensional or 3-dimensional image/video, which includes data acquisition, processing, transmission, reception, and reconstruction. This system includes acquisition of color and depth image pairs from a image acquisition system with vertical rigs, rectification of acquired image pairs and generating digital hologram. Also it is designed to reduce the CGH (computer-generated hologram) generation time to 1/3. It also includes some additional and optional functions such as watermarking, compression, and encryption.

Keyword : computer-generated hologram, vertical rig system, hologram signal process, hologram video service system.

a) 광운대학교 전자재료공학과(Kwangwoon Uni. Dept. of Electronic Material Engineering)

‡ Corresponding Author : 김동우 (Dong-Wook Kim)  
E-mail: dwkim@kw.ac.kr  
Tel: +82-2-940-5167

※ 본 연구는 2014년 정부(교육과학기술부)의 재원으로 한국연구재단 지원을 받아 수행된 연구임 (NRF-2013R1A1A2057798).

※ 이 논문의 연구결과 중 일부는 “3D Systems and Applications (3DSA 2014)”에서 발표한 바 있음.

· Manuscript received July 15, 2014 Revised August 18, 2014 Accepted August 18, 2014

### I. 서 론

홀로그램은 빛의 세기뿐만 아니라 위상정보까지 기록할 수 있기 때문에 원래의 3차원 입체상을 공간에 자연스럽게 재현할 수 있는 가장 이상적인 영상 시스템 중의 하나이다 [1]. 최근 들어 3차원 입체 디스플레이 기술이 보편화되면서 디지털 홀로그램 비디오를 이용한 홀로그래픽 방송 및 TV에 대한 관심이 높아지고 있다. 지금까지 기초적인 수준이

지만 홀로그래픽 TV에 대한 연구도 진행되어 왔다<sup>[2~4]</sup>. 홀로그래픽 TV에 대한 연구는 크게 디지털 홀로그램의 획득과 디스플레이로 구분한다. 여기에서는 디지털 홀로그램의 획득에 대한 논의에 집중한다. 디지털 홀로그램을 획득하는 것은 광학적으로 촬영하는 방법과 컴퓨터 생성 홀로그램을 이용하여 생성하는 방법 등이 있다. 하드웨어 시스템적인 측면에서 디지털 홀로그램 비디오를 생성하는 기법에 대한 여러 연구가 있다<sup>[2~4]</sup>. 그러나 아직 홀로그램을 방송 등으로 서비스하기에는 복원영상의 크기와 선명도에서 많이 부족하다. 따라서 홀로그램, 즉 디지털 홀로그램을 서비스하는 시스템에 대해서는 구체적인 연구가 진행된 사례가 없고, 발표된 바도 없다. 단, 최근에 시청자의 눈을 추적하여 초점에 해당하는 부분만 홀로그램으로 디스플레이하는 부분 홀로그램 방법으로 홀로그램 생성 및 디스플레이 시스템을 발표한 적은 있다<sup>[5]</sup>.

본 논문은 디지털 홀로그램을 서비스하기 위한 시스템을 궁극적인 목표로 한다. 이 시스템은 수직리그 시스템으로부터 컬러 및 깊이 영상을 시청자의 시점에 따라서 홀로그

램을 생성 및 서비스하는 시스템을 구현하였다. 2장에서는 시스템의 개요를 설명하고 3장에서는 시스템의 세부 기능에 대하여 설명하였다. 4장에서는 2~3장에서 설명한 기능을 LabVIEW를 통하여 구현한 결과를 보였다.

## II. 시스템의 개요

그림 1은 본 논문에서 구현한 시스템의 흐름도이다. 이 시스템은 컴퓨터 생성 홀로그램(Computer Generated Hologram, CGH)을 대상으로 하기 때문에 홀로그램 생성을 위한 데이터 획득과 생성한 홀로그램의 후처리 과정이 포함된다. 홀로그램 생성을 위한 데이터를 획득하기 위해 HD급의 컬러 카메라<sup>[6]</sup>와 ToF(Time of Flight)<sup>[7]</sup> 형태의 깊이 카메라를 수직 형태로 구성된 3쌍의 수직리그 시스템으로부터 3개의 시점의 컬러 및 깊이 영상을 획득한다. 정확한 깊이 영상을 위해 ToF 형태의 깊이 카메라를 사용하는데 일반적으로 ToF 형태의 깊이 카메라는 해상도가 작기 때문에

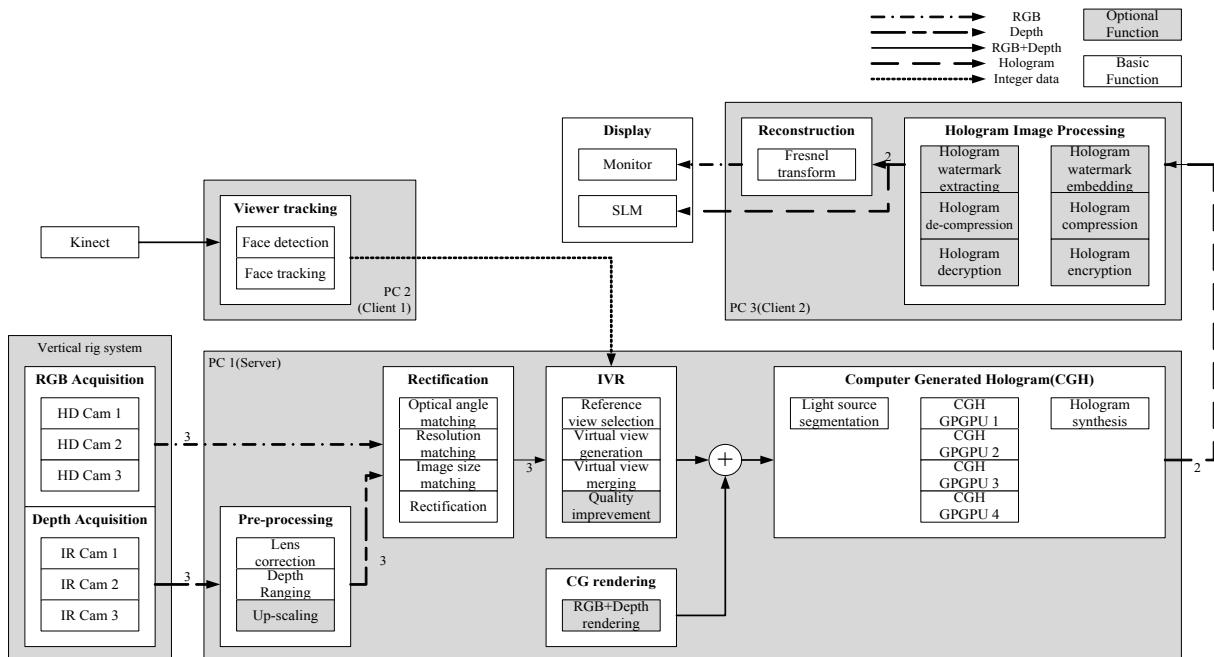


그림 1. 홀로그램 서비스를 위한 시스템의 흐름도  
 Fig. 1. System flow for hologram service

별도의 전처리 과정을 통하여 고 해상도의 깊이 영상을 획득할 수 있다. 서로 다른 카메라를 이용하여 각각의 광각이 다르고 각 시점은 수평방향으로 가정하기 때문에 3쌍의 영상의 보정(Rectification)과정이 필요하다.

또한 시청자의 시점에 따라 자연스러운 시점의 홀로그램을 생성하기 위해 얼굴 추적과 중간시점 렌더링(Intermediate View Rendering, IVR)<sup>[8]</sup> 과정을 수행한다. 얼굴 추적은 Microsoft의 Kinect를 이용하여 한 쌍의 컬러 및 깊이 영상을 이용하여 SDK내의 함수를 이용하여 구현하였다<sup>[9]</sup>. 또한 다양한 콘텐츠 합성을 위하여 컴퓨터 그래픽(Computer Graphic, CG)기반의 콘텐츠를 전 방향에서 합성할 시점에 대한 추출하여 합성하는 과정을 추가적으로 수행한다<sup>[10]</sup>.

컴퓨터 생성 홀로그램(Computer Generated Hologram, CGH)는 많은 양의 연산량이 필요하다<sup>[11]</sup>. 이를 고속화하기 위해 다중 GPU와 PCIe 확장슬롯을 이용하여 고속화 하였다. 또한 생성한 홀로그램의 정보 보안을 위하여 워터마킹<sup>[12]</sup>, 압축<sup>[13]</sup>, 암호화<sup>[14]</sup> 과정을 추가적으로 수행할 수 있다. 홀로그램을 Fresnel 변환<sup>[15]</sup>을 통하여 모니터 또는 SLM(Spatial Light Modulation)장치<sup>[16]</sup>를 이용하여 디스플레이 할 수 있도록 구현하였다. 구현한 시스템은 총 3대의 PC를 이용하여 구현하였고 PC간의 데이터 전송은 TCP/IP를 통하여 전송하는 구조로 구성하였다. 그림 1에 기본 함수 및 추가적인 함수를 나타냈고, 각 모듈 간 데이터 형태 및 시점의 수를 나타냈다.

라는 850nm의 적외선을 이용하여 11bit의 거리 정보를 표현할 수 있다. 또한 고 해상도의 컬러 영상을 획득하기 위해 FullHD(1920x1080)급 데이터를 획득할 수 있는 RGB 카메라를 이용하였다<sup>[6]</sup>. 서로 다른 카메라를 이용하기 때문에 컬러 영상의 객체와 깊이 카메라로부터 획득한 거리 정보는 서로 틀어지는 현상이 생기게 된다. 따라서 두 정보의 오차를 줄이기 위하여 콜드미러를 이용하여 카메라 쌍을 수직으로 구성한다<sup>[17]</sup>. 그림 2는 수직리그의 구조를 나타냈다. 콜드미러를 45도 기울여 가시광선(380nm ~ 750nm)에 해당하는 파장은 반사되어 컬러 카메라로 입력되고 850nm 파장의 적외선은 투과하여 깊이 카메라로 입력된다. 콜드미러는 그림 3과 같은 특성을 가지고도록 제작하여 약간의 가시광선의 손실이 생기지만 적외선의 손실을 최소화 하여 신뢰성 높은 거리 정보를 획득할 수 있도록 하였다.

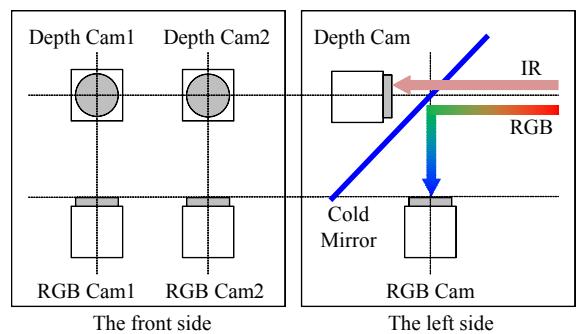


그림 2. 카메라 시스템의 구조 및 동작  
Fig. 2. The structure and operation of the camera system

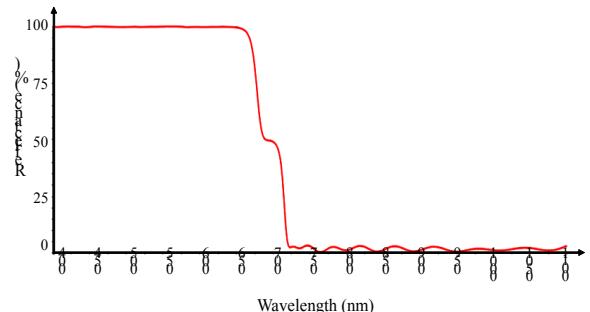


그림 3. 콜드 미러의 반사를 특성  
Fig. 3. The reflectance of the cold mirror

### III. 시스템의 동작

#### 1. 수직 리그 시스템

향상된 홀로그램을 획득하기 위해서 객체의 정밀한 실제 거리와 고해상도의 빛의 세기가 필요하다. 특히 깊이 정보의 신뢰도가 높을수록 향상된 홀로그램을 생성할 수 있다. 깊이정보를 얻는 방법으로 Microsoft사의 Kinect<sup>[9]</sup> 및 ToF 형태의 깊이 카메라를 사용할 수 있는데 신뢰도를 높이기 위하여 MESA Imaging의 SR4K<sup>[7]</sup>을 사용하였다. 이 카메

## 2. 컬러 영상 + 깊이 영상 쌍 생성(영상 보정)

앞 절에서 획득한 정보는 서로 다른 카메라를 사용하여 컬러 영상과 깊이 영상의 일치하지 않는다. 따라서 각각 영상에 몇 가지 영상 보정 과정을 통하여 일치시키는 과정을 수행한다. 첫 번째로 깊이 카메라로부터 획득한 정보는 그림 4(a)와 같이 렌즈 왜곡이 발생한다. 왜곡을 보정하기 위해 OpenCV<sup>[18]</sup>내의 렌즈 교정(Lens Correction) 함수를 통하여 교정하였다. 그림 4(b)는 렌즈 교정이 적용된 영상이다.

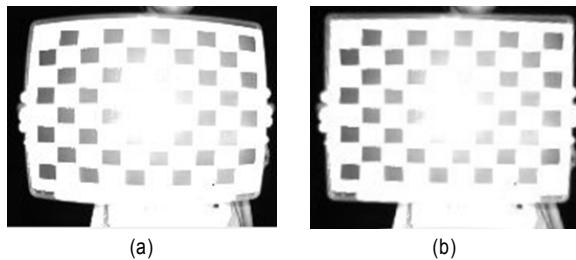


그림 4. 렌즈 교정 (a)전,(b)후  
 Fig. 4. Lens correction (a) before, (b) after

또한 깊이 카메라로부터 획득한 깊이 정보는 QCIF (176x144)의 해상도로 출력한다<sup>[7]</sup>. 고 해상도의 광원 정보를 생성하기 위해서 업 샘플링 과정을 수행한다. 깊이 정보를 웨이블릿 변환 기반의 업 샘플링을 그림 5에 나타냈다. 웨이블릿 영역에서 수행하는 여러 고해상도 기법들의 핵심은 웨이블릿 영역에서의 고주파 대역을 의미하는 부대역을 구하는 것이다. 이산 웨이블릿 변환 (Discrete Wavelet Transform, DWT)에서 기본적으로 수행되는 다운 샘플링 과정을 수행하지 않음으로써 획득한 고주파 대역의 부대역과 웨이블릿 0-화소 삽입 확장 (Wavelet Zero Padding: WZP)의 결과를 DWT 하여 얻은 부대역을 합성하여 고주파 부대역을 획득한다. Inverse DWT를 이용하여 입력된 영상과 획득한 고주파 부대역을 합성하여 입력된 영상에 비하여 해상도가 4배 향상된 영상을 얻는다. 마지막으로 생성된 고주파 부대역이 완벽하지 않기 때문에 나타나는 블록 현상을 줄여주기 위해서

블록 현상 제거 필터를 적용한다<sup>[19]</sup>.

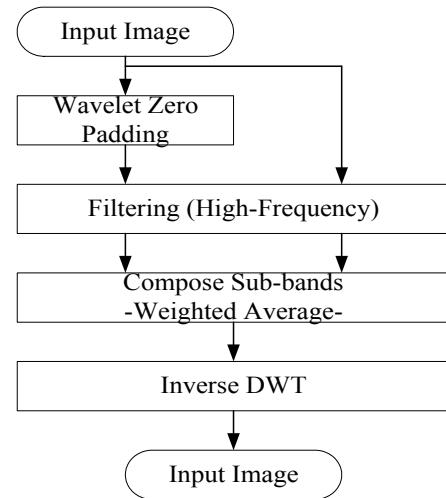


그림 5. DWT 기반 업 샘플링  
 Fig. 5. DWT-based up sampling

두 번째로 컬러 영상과 깊이 영상은 서로 다른 카메라로 찍기 때문에 화각이 다르다. 따라서 화각을 맞추기 위해 상대적으로 큰 해상도인 컬러 영상에 그림 6과 같이 몇 가지 기하학적 처리를 수행한다. 컬러 영상에 회전(Rotation), 이동(Shifting), 해상도 조절(Re-Sizing) 처리 후 깊이 영상과 같은 해상도로 만든다. 그림 6(e)는 깊이 영상과 컬러 영상의 중심을 기준으로 식 (1)을 이용하여 합성하여 나타내었다.  $C(x,y)$ ,  $D(x,y)$  그리고  $S(x,y)$ 는 각각  $x, y$  좌표에서의 컬러, 깊이 및 합성한 영상이다.

$$S(x,y) = \frac{C(x,y) + D(x,y)}{2} \quad (1)$$

위 두 과정을 통하여 컬러 영상과 깊이 영상을 일치 시킬 수 있다. 다음 과정인 중간시점 렌더링을 위한 가정으로 각각의 시점은 수평이 일치되어야 한다. 따라서 3시점의 수평을 맞추어야 한다. 그림 7은 시점의 수평을 맞추기 위한 과정으로 각각의 깊이 영상을 기준으로 같은 y축 좌표에 기준이 되는 선을 통하여 3시점의 컬러+깊이 영상 쌍의 수평을 맞춘다.

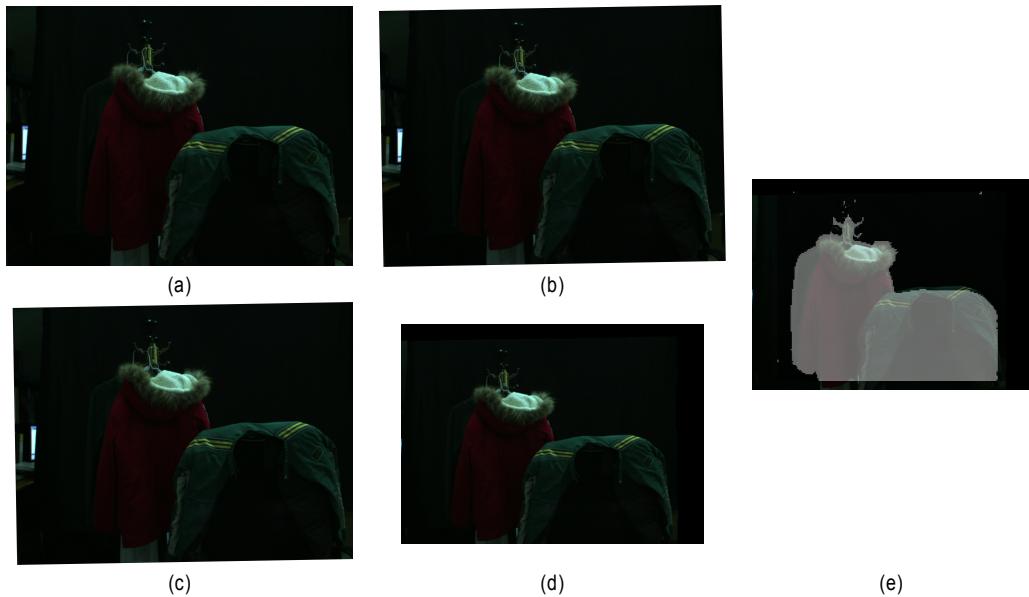


그림 6. 컬러영상의 기하학적 처리(a)원본 (b)회전, (c)이동, (d)해상도 조절, (e)합성

Fig. 6. Geometric process for color images (a) origin (b)rotation, (c)shifting, (d)re-sizing, (e)synthesis



그림 7. 컬러 + 깊이 영상 쌍의 수평조절

Fig. 7. Horizontal matching color + depth image pairs.

### 3. 시청자 시점 영상 렌더링

시청자의 시점에 해당하는 영상을 생성을 위해 우선 시청자의 위치를 추적하고 해당 위치의 좌표를 이용하여 3시점의 컬러+깊이 영상 쌍을 이용하여 중간 시점 영상을 렌더링 하는 방식을 택하였다. 그림 8에 시청자 시점 영상 렌더링 방법을 도식화 하였다. 시청자의 위치에 따라서 총 21개의 시점을 생성할 수 있고 5가지의 방법을 통하여 생성할 수 있다. Kinect로부터 시청자의 좌표를 구하면 좌표에 따

라 21개로 나누고 중간, 가장 왼쪽과 오른쪽은 해당 시점에 컬러+깊이 영상 쌍을 출력하고 나머지 좌표일 경우에는 가장 가까운 두 개의 시점을 이용하여 중간시점 영상을 생성하여 출력한다. 시청자의 위치를 추적하기 위해 Microsoft 사에서 제공하는 Kinect용 SDK내의 추적 프로그램<sup>[9]</sup>을 이용하였다. 또한 중간 시점 생성은 3D 워핑 기법을 이용하여 구현하였다<sup>[8]</sup>. 3D 워핑 기법을 통하여 각 시점에 영상을 이용하여 해당 좌표의 시점의 영상을 각각 생성하고 식 (2)와 같이 시청자의 좌표에 따라 가중치를 주어 두 영상을 합성

하였다.  $V_1(x,y)$ 와  $V_2(x,y)$ 는 좌우 시점으로부터 생성한 가상 시점이고  $w$ 는 시청자의 시점에 대한 가중치이다.

의 좌표이고  $dx, dy$ 는 컬러 영상의 화소의 크기이고  $z$ 는 해당 좌표의 실제 거리에 해당한다.  $u, v$ 는 홀로그램의 좌

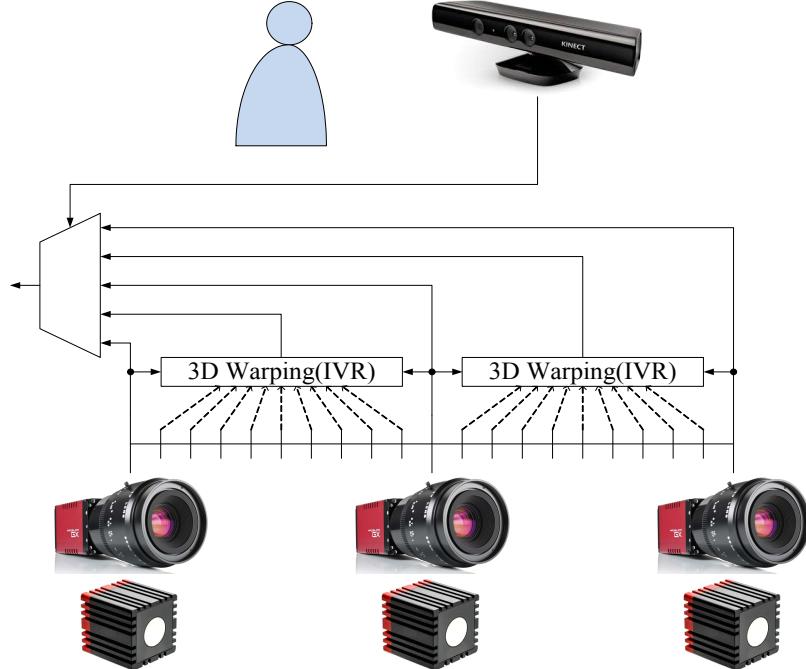


그림 8. 시청자 시점 영상 렌더링  
 Fig. 8. Viewer's view position image rendering

$$S(x,y) = \begin{cases} V_1(x,y), & V_2(x,y) = NLL \\ V_2(x,y), & V_1(x,y) = NLL \\ wV_1(x,y) + (1-w)V_2(x,y), & \text{other} \end{cases} \quad (2)$$

합성한 영상은 두 시점으로부터 생성하는데 이때 시점 변화로 인하여 생기는 홀(hole) 영역은 가장 근접한 배경화 소로 채웠다.

#### 4. CGH 생성

CGH는 객체에서 반사되는 객체파와 참조파간의 간섭현상을 모델링하여 식 (3)과 같이 나타낸다.  $N$ 은 컬러 영상의 화소 값이 0이 아닌 유효한 화소의 수이고  $x, y$ 는 컬러영상

표이고  $du, dv$ 는 홀로그램의 화소의 크기(실제 SLM의 화소크기)이다.  $\lambda$ 는 사용한 참조파의 파장이다.  $I$ 와  $g_j$ 는 각각 홀로그램의 화소 값과 컬러 영상의 화소 값을 나타낸다. 식 (3)을 Fresnel 근사<sup>[22]</sup>하여 식 (4)와 같이 표현할 수 있다.  $\theta_z$ 는  $z/\lambda$ 이고  $\theta_x, \theta_y$ 는 각각  $(udu - xdx)^2$ ,  $(vdv - ydy)^2$ 이다.

$$I(u,v,0) \approx \sum_j^N g_j(x,y,z) \exp\left[-j2\pi\left(\theta_z + \frac{\theta_x + \theta_y}{2\lambda z}\right)\right] \quad (4)$$

식 (4)에서 볼 수 있듯이 10k 광원에 대하여 1024×1024의 홀로그램을 연산하기 위해서는 100억회 이상의 하나의 홀로그램에 대한 하나의 광원의 연산을 수행해야한다. 이

$$I(u,v,0) = \sum_j^N g_j(x,y,z) \exp\left[\frac{-j2\pi}{\lambda} \sqrt{(udu - xdx)^2 + (vdv - ydy)^2 + z^2}\right] \quad (3)$$

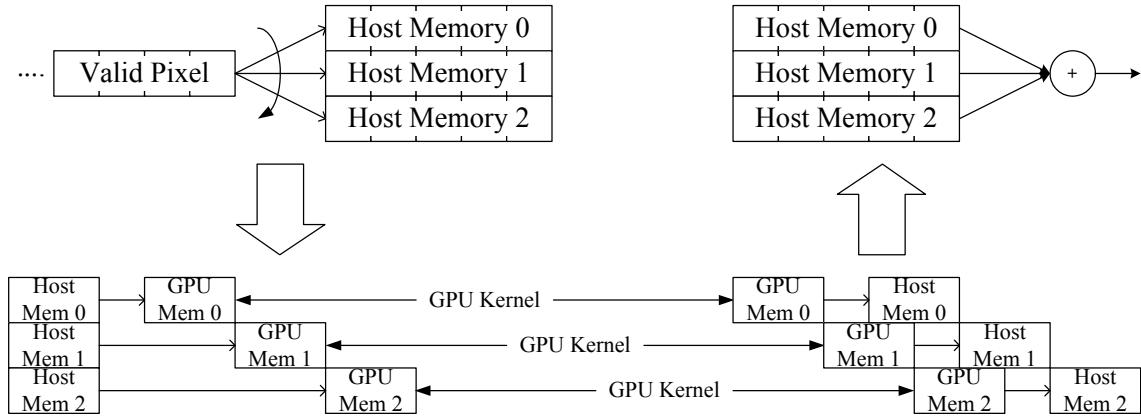


그림 9. 다중 GPU를 이용한 CGH 연산방법  
Fig. 9. Method for CGH calculation using Multi-GPU

와 같이 매우 많은 연산을 고속화를 위해 본 논문에서는 다중 GPU(Graphic Process Unit)를 이용한 병렬 처리를 통하여 구현하였다. 그림 9는 다중 GPU를 이용한 연산을 나타낸 것으로 하나의 헤로그램 화소에 대하여 모든 광원의 연산을 수행하는 것이 효율적이다<sup>[11]</sup>. 유효한 광원을 동일한 양으로 GPU의 개수로 분배하고, 각각을 GPU로 전송한다. 이때 하나의 GPU에서 연산할 광원의 정보가 모두 GPU로 전송이 되면 커널이 실행되는데 이 커널이 실행되는 동안 호스트에서 다른 GPU로의 데이터 전송이 가능하다. 모든 GPU에서 커널이 종료되면 헤로그램 정보를 호스트에서 누적하여 출력한다.

## 5. 헤로그램 복원

헤로그램을 복원하는 방법은 크게 두 가지로 나뉜다. 첫 번째 방법은 광학 장치를 이용하여 복원하는 방법으로 회절 현상에 의해 SLM에 헤로그램을 로드하고 생성할 때와 동일한 파장의 빛을 조사하여 복원한다. 복원한 헤로그램은 생성할 때와 같은 거리, 크기의 영상으로 복원이 된다. 이러한 광학적 복원 방법인 회절현상을 수학적으로 모델링 하여 식 (5)와 같은 Fresnel 변환을 이용하

여 시뮬레이션 할 수 있다<sup>[21,22]</sup>.  $F(x,y)$ 는  $x,y$ 좌표의 복원된 영상의 밝기 값이고  $f(u,v)$ 는  $u,v$ 좌표의 입력 헤로그램의 밝기 값이다.  $dx, dy$ 는 복원 영상의 화소의 크기이고,  $du, dv$ 는 입력 헤로그램의 화소의 크기이다.  $\lambda$ 는 조사되는 빛의 파장이고,  $z$ 는 입력 헤로그램과 복원영상 간의 거리이다.

식 (5)에서 복원 영상의 화소 크기를  $dx = (\lambda z)/(duM)$ ,  $dy = (\lambda z)/(dvN)$ 라고 가정하면 식 (6~7)과 같이  $g(u,v)$ 과  $G(x,y)$ 로 정의하고 식 (8)과 같이 나타낼 수 있다. 식 (8)은 이산 푸리에 변환(DFT)과  $G(x,y)$ 의 곱으로 나타낼 수 있고 식(9)와 같이 고속 푸리에 변환(FFT)을 이용한 식으로 변형할 수 있어 고속화가 가능하다<sup>[22]</sup>.

$$g(u,v) = f(u,v) \exp^{j \frac{\pi}{\lambda z} \{(udu)^2 + (vdv)^2\}} \quad (6)$$

$$G(x,y) = \exp^{j \frac{\pi}{\lambda z} \{(x dx)^2 + (y dy)^2\}} \quad (7)$$

$$F(x,y) = G(x,y) \sum_{v=0}^N \sum_{u=0}^M g(k,l) \exp^{-j2\pi ux/M} \exp^{-j2\pi vy/N} \quad (8)$$

$$\begin{aligned} & F(x,y) \\ &= \exp^{j \frac{\pi}{\lambda z} \{(x dx)^2 + (y dy)^2\}} \text{FFT} \left[ f(u,v) \exp^{j \frac{\pi}{\lambda z} \{(udu)^2 + (vdv)^2\}} \right] \end{aligned} \quad (9)$$

$$F(x,y) = \exp^{j \frac{\pi}{\lambda z} \{(x dx)^2 + (y dy)^2\}} \sum_{v=0}^N \sum_{u=0}^M f(u,v) \exp^{j \frac{\pi}{\lambda z} \{(udu)^2 + (vdv)^2\}} \exp^{-j \frac{2\pi}{\lambda z} (udu x dx + vdy y dy)} \quad (5)$$

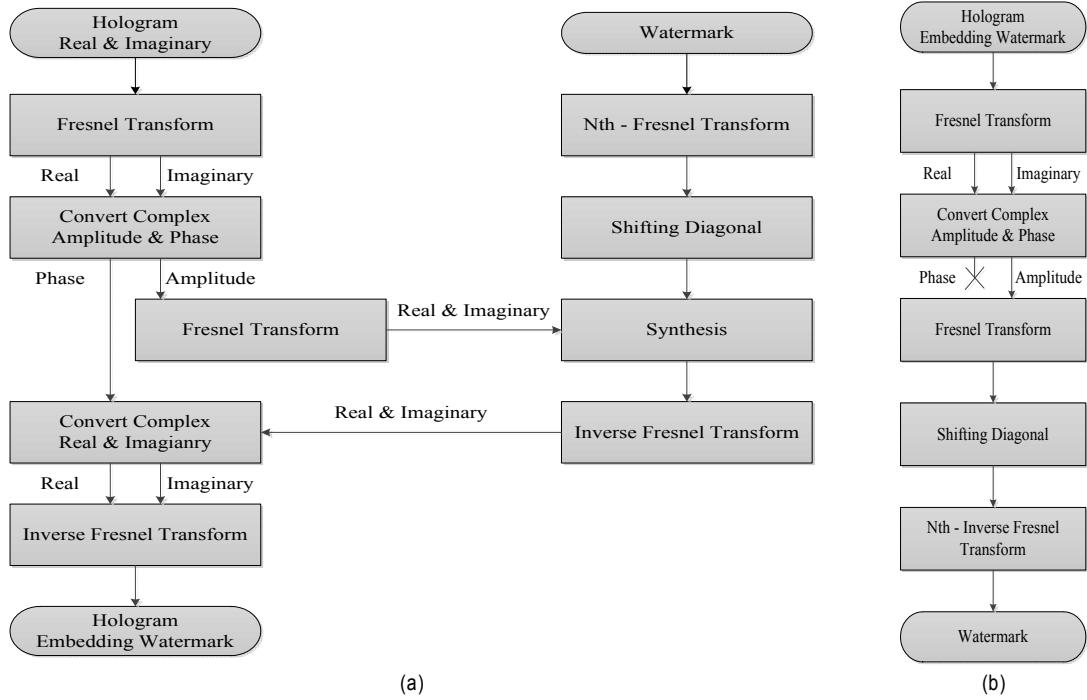


그림 10. 홀로그램 워터마킹 알고리즘 (a)삽입, (b)추출  
 Fig. 10. Watermarking algorithm for hologram (a)embedding, (b) extracting

## 6. 홀로그램 워터마킹

홀로그램의 정보 보호를 위해 시스템에서 사용한 워터마킹은 그림 10에 나타내었다. 일반적인 2D 영상에 Fresnel 변환을 수행하면 그림 11(b)와 같이 가운데로 에너지가 집중되는 현상이 생긴다<sup>[12]</sup>. 또한 홀로그램을 복원하면 그림 11(c)와 같이 객체형태의 2D 형태의 영상을 구할 수 있다. 따라서 복원 영상에 Fresnel 변환을 적용하면 2D영상에 Fresnel 변환을 적용한 것과 같이 가운데로 에너지 집중되는 현상이 생긴다. 이때 집중되지 않은 영역이 워터마크 삽입 영역이 된다. 워터마크를 그대로 워터마크 영역에 삽입 할 경우 홀로그램을 복원하면 워터마크 자체에 Fresnel 변환을 적용한 영상이 나타날 수 있으므로 다른 Fresnel 변환 파라미터를 이용하여 2번 이상의 Fresnel 변환을 수행한다. 그림 11(d)에 복원 영상과 중복되지 않는 영역에 워터마크를 삽입한 결과를 나타 내었다. 만약 이때 1번 적용할 경우 홀로그램 복원 영상과 같은 차수의 Fresnel 변환이 수행되므로 워터마크가 나타난다<sup>[12]</sup>.

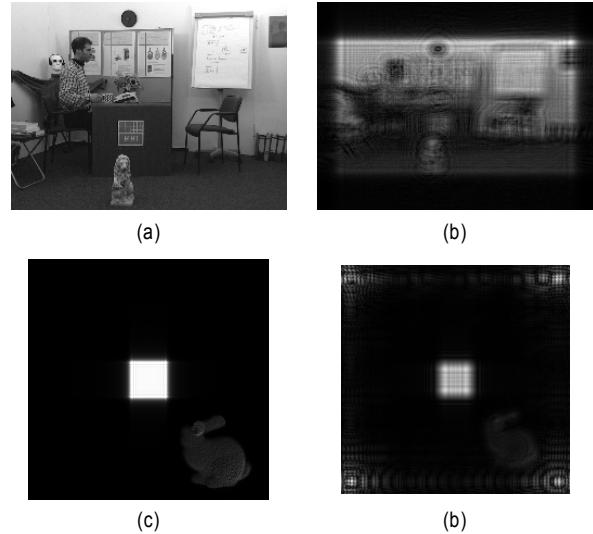


그림 11. Fresnel transform (a)origin 2D image, (b) transform applied 2D image, (c) transform applied hologram, (d) embedding result after applied transform for watermark  
 Fig. 11. Fresnel transform (a)origin 2D image, (b) transform applied 2D image, (c) transform applied hologram, (d) embedding result after applied transform for watermark

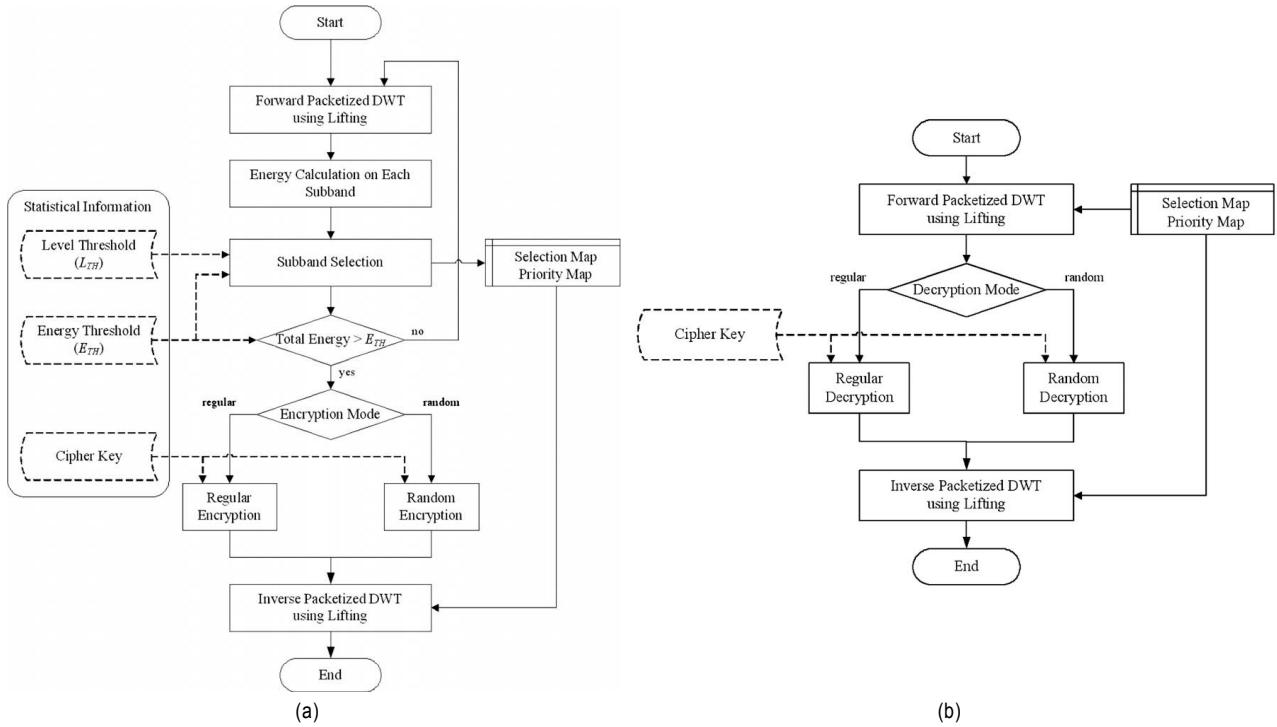


그림 12. 홀로그램 암호화를 위한 알고리즘 (a)암호화, (b)복호화  
Fig. 12. Encryption algorithm for hologram(a) encryption, (b) decryption

워터마크 추출은 워터마크를 삽입한 홀로그램에 Fresnel 변환을 2번 적용한 뒤 삽입 영역을 가운데로 오도록 재배열한 뒤 워터마크에 적용했던 Fresnel 변환 수만큼 역변환을 수행하여 추출한다<sup>[12]</sup>.

## 7. 홀로그램 암호화

홀로그램의 정보 보호를 위해 시스템에서 사용한 암호화 알고리즘을 그림 12에 나타내었다<sup>[24]</sup>. 암호화를 위하여 첫째로 웨이블릿 변환을  $n$  레벨 쿼드트리로 수행을 한다. 각 밴드별 에너지를 계산을 통하여 내림차순으로 밴드를 재정렬 한다. 그림 13(b)는 (a)를 4레벨 쿼드트리로 웨이블릿 변환을 한 결과 결과이다. 표 1은 그림 13(b)의 에너지 구하고 정렬한 테이블이다. 입력되는 암호화 강도에 대한 파라미터에 의해 표 1에 상위 밴드를 암호화할 밴드로 결정 후 블록 암호화를 통하여 암호화를 수행한다. 이때 선택된 밴드는

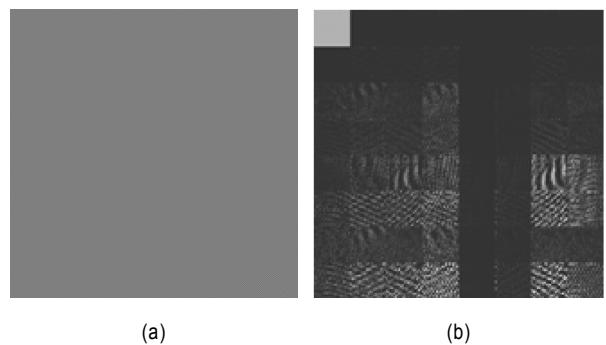


그림 13. 홀로그램의 웨이블릿 변환(a)홀로그램,(b)4-레벨 DWT  
Fig. 13. Hologram wavelet transform (a) hologram, (b) 4-level DWT

드는 암호화키와 함께 복호화를 위한 데이터이다. 암호화를 수행 후 역 웨이블릿 변환을 수행한다. 복호화 과정은 웨이블릿 변환 이후 선택된 밴드를 이용하여 암호화된 영역에 대하여 복호화를 수행하고 역 웨이블릿 과정을 수행 한다.

표 1. 에너지 테이블  
 Table 1. Energy table

Quad-DWT			Re-arrange table		
Sub-band	Average Energy	Ratio	Ratio	Average Energy	Sub-band
0	16201.034	38.357	38.357	16201.034	0
1	224.675	0.532	25.979	10972.849	14
2	6.705	0.016	9.627	4066.018	15
3	567.815	1.344	9.029	3813.564	7
4	625.367	1.481	8.863	3743.395	6
5	1533.207	3.630	3.630	1533.207	5
6	3743.395	8.863	1.481	625.367	4
7	3813.564	9.029	1.344	567.815	3
8	0.268	0.001	0.651	274.797	11
9	5.734	0.014	0.532	224.675	1
10	13.869	0.033	0.316	113.677	12
11	274.797	0.651	0.129	54.296	13
12	113.677	0.316	0.033	13.869	10
13	54.296	0.129	0.016	6.705	2
14	10972.849	25.979	0.014	5.734	9
15	4066.018	9.627	0.001	0.268	8

## 8. 홀로그램 압축

고 해상도의 홀로그램을 전송을 하려면 많은 양의 비

트 용이 필요 하다. 따라서 압축을 통하여 홀로그램의 비트 용량 줄일 필요성이 있다. 변환된 계수를 2차원 영상으로 취급하면 객체영상과 유사한 형태의 영상 정보가 나타나고, 디지털 홀로그램의 위치적인 특성에 따라서 객체가 이동하는 것과 같이 관찰된다. 이는 분할영역마다 한 대의 카메라가 있다고 가정하고, 해당위치에서 각 카메라로 객체를 촬영한 것과 유사한 결과이다. 즉, 다시점 영상을 획득하는 것에 해당한다. 따라서 분할된 홀로그램들의 위치적인 특성을 시간적인 변화로 사상할 경우에 전체 홀로그램은 하나의 비디오 스트림으로 취급할 수 있다. 분할된 홀로그램은 약간의 차이는 있지만 유사한 정보를 포함하기 때문에 데이터의 차이가 크지 않고, 시간적 중복 성(temporal redundancy)으로 간주할 수가 있어서 동영상 압축 도구를 이용하여 효율적으로 압축할 수 있다<sup>[13]</sup>.

그림 14와 같이 3D 객체는 다수의 홀로그램으로 저장되고 각각은 일반적인 비디오 영상에서 하나의 프레임에 해당된다고 볼 수 있다. 각 프레임을 분할영역(Segmentation)으로 나누고 주파수변환(Transform) 과정을 거친 후에 분할영역들 간의 상관도를 고려하여 분할영역을 새로운 프레임 단위로 하는 비디오 시퀀스를 구성한다. 구성된 비디오 시퀀스는 계수분류(Classification) 및 정규화(Normalization) 과정을 통해 비디오 압축 도구를 적용하기 위한 정보들로 재분류되고 부호화 과정을 통해서 압축된 비트스트림을 생성한다<sup>[13]</sup>.

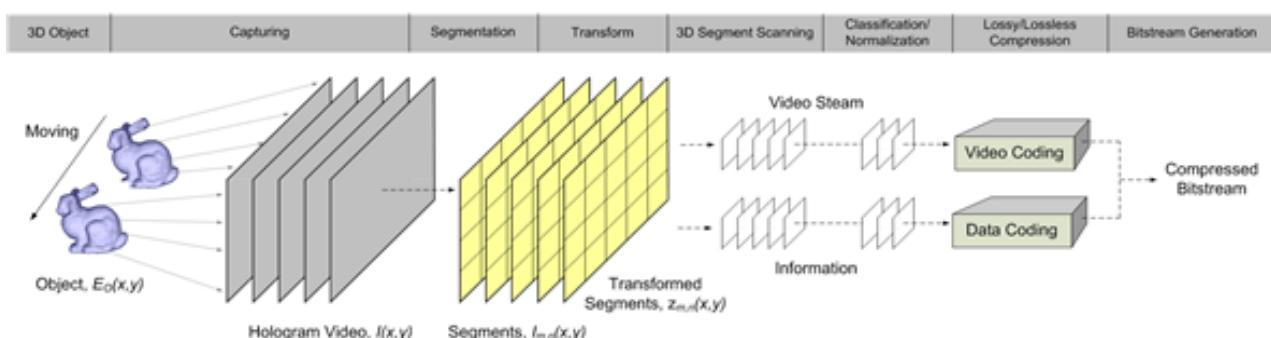


그림 14. 홀로그램 압축

Fig. 14. Hologram compression

## IV. 구현 결과

### 1. 시스템 구현

구현한 시스템은 National Instrument사의 LabVIEW® 환경에서 구현하고, 앞장에서 설명한 각 기능은 DLL(Dynamic Linking Library) 및 LabVIEW내의 함수를 이용하여 링크 시켰다. 먼저 각 PC간의 데이터 전송은 LabVIEW내의 TCP/IP 모듈을 통하여 실시간 데이터 전송이 가능하도록 하였다. 내부 네트워크는 1Gbps(Giga-bit per second)로 전송이 가능한 랜 케이블을 스위칭 허브에 연결하여  $1,024 \times 1,024$ 크기의 홀로그램(실수 + 허수)을 초당 64프레임 전송할 수 있는 네트워크 환경을 구축하였다.

#### 1.1 수직리그 시스템

수직 리그로부터 데이터 획득 과정 중 컬러 영상획득은 LabVIEW의 카메라 획득 모듈을 통하여 구현하고 LabVIEW내 드라이버가 존재하지 않는 깊이카메라는 DLL을 통하여 LabVIEW로 링크 하였다. MESA Image사의 SR4000카메라는  $176 \times 144$ (QCIF)의 해상도의 11-bit의 데이터를 획득가능하다. 서로 다른 카메라를 동시에 동작 시킬 경우 각 카메라의 적외선간의 간섭현상이 생기므로 카메라의 변조 주파수를 29MHz, 30MHz, 31MHz로 조절하여 구현하였다. 또한 작은 깊이영상의 해상도를 증가시키

기 위한 업 샘플링 모듈과 렌즈 왜곡조정 모듈을 SR4000획득 모듈에 포함 하였다. 그림 13은 컬러 및 깊이 영상을 획득하는 GUI를 나타내었다. 카메라 파라미터에서 획득한 영상에 해당하는 카메라의 위치, 업 샘플링 유무, 내 외부 카메라 파라미터 그리고 깊이의 범위 및 시스템에서 사용할 컬러영상의 색을 결정할 수 있다.

#### 1.2 컬러 및 깊이 영상 쌍 생성 및 보정

수직 리그 시스템으로부터 획득한 컬러 및 깊이 영상을 보정하는 모듈은 DLL을 통하여 LabVIEW로 링크하여 구현하였다. 그림 14는 영상의 보정을 수행하는 과정으로 내부 파라미터는 컬러영상을 깊이영상과 일치시키기 위한 파라미터로 이동, 해상도조절, 회전을 적용시킬 수 있으며 일치가 되었는지 확인하기 위한 알파 블렌딩 파라미터가 있다. 외부 파라미터는 세 쌍의 영상 쌍을 수평을 맞추기 위해 영상 이동을 적용시킬 수 있고 수평이 맞추어졌는지 확인하기 위한 라인의 색, 위치 및 확대를 할 수 있다.

#### 1.3 시청자 시점 렌더링

영상보정을 통하여 생성한 영상 쌍을 해당 시점의 영상을 생성하는 모듈은 DLL을 이용하여 링크하였고, 시청자의 시점을 찾는 얼굴 추적모듈과 3D 모델을 렌더링하는 모듈은 실행 파일로 링크하였다. 출력을 위한 시점은 오토 버

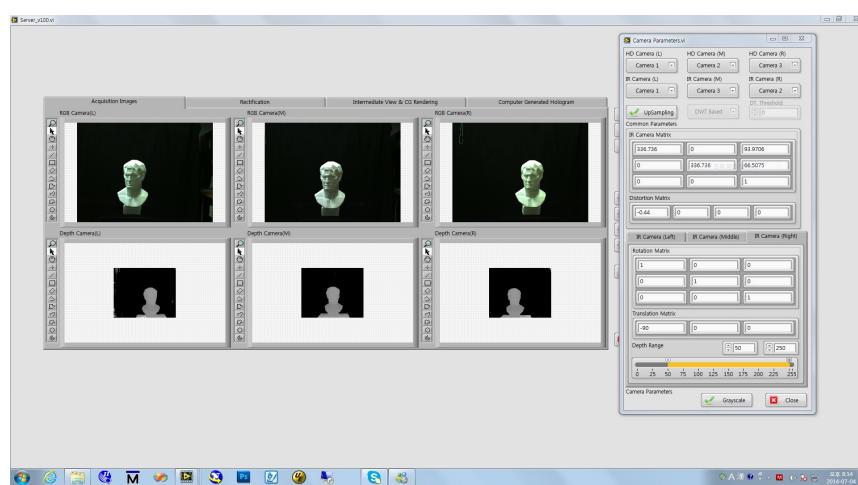


그림 15. 컬러 및 깊이 영상 획득을 위한 GUI  
Fig. 15. GUI for color and depth acquisition

튼을 통하여 시청자의 시점을 추적 또는 사용자가 직접 좌표를 입력할 수 있도록 되어있고 CG 렌더링 버튼을 통하여 3D 모델을 합성 유무를 결정할 수 있다.

#### 1.4 홀로그램 생성

시청자의 시점에 해당하는 컬러 및 깊이 영상을 이용하여 홀로그램을 생성하는 모듈은 cuda를 이용하여 GPU를 사용할 수 있는 프로그램을 DLL로 링크하여 구현하였다.

그림 16은 CGH를 위한 GUI로 광학 장치의 화소 크기 및 사용한 레이저의 파장을 입력할 수 있다. 또한 홀로그램의 크기와 위치, 컬러 및 깊이영상(광원)의 배율 및 위치를 입력 가능하도록 구현하였다.

#### 1.5 홀로그램 복원

생성한 홀로그램을 광학적으로 복원을 위한 모듈은 CUDA를 이용하여 GPU를 사용할 수 있는 프로그램을

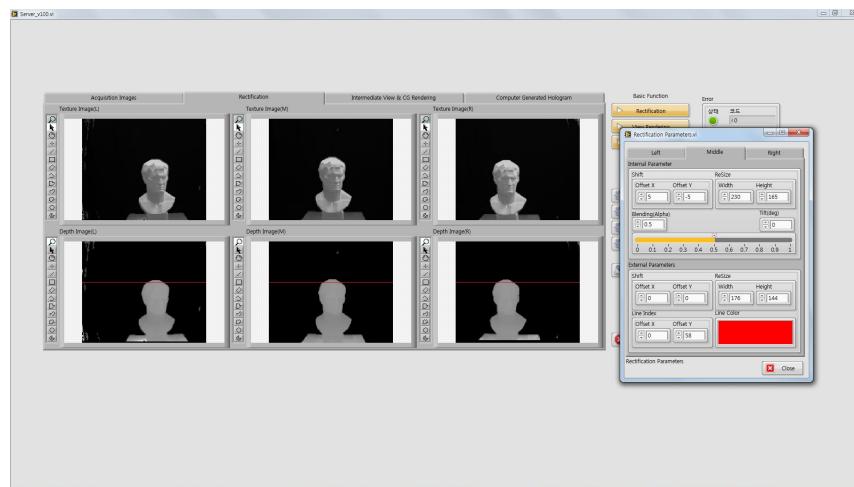


그림 16. 영상보정을 위한 GUI  
 Fig. 16.. GUI for rectification

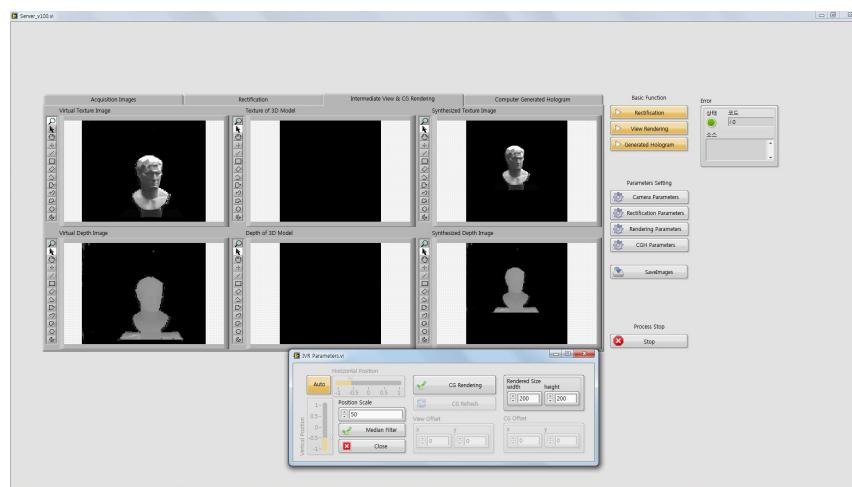


그림 17. 중간시점 영상 렌더링을 위한 GUI  
 Fig. 17. GUI for Intermediate view rendering

DLL로 링크하여 구현 하였다. 그림 17은 Fresnel 변환을 이용한 복원한 결과로 광학장치의 화소 크기 및 사용한 레이저의 파장을 입력할 수 있고, 광학장치로 부터의 거리를 입력할 수 있다.

### 1.6 추가적인 기능

홀로그램의 신호처리를 위한 모듈(압축, 워터마킹, 암

호화)은 DLL을 이용하여 구현하였다. 그림 18에 파라미터의 GUI를 나타냈다. 압축 파라미터는 DCT의 크기 및 H.264/ AVC의 QP값을 조절할 수 있고, 암호화는 암호화 키 및 DWT 레벨 그리고 에너지 제한 값을 입력할 수 있다. 워터마킹 파라미터는 워터마크의 거리 및 사이즈 그리고 워터마크 이후 공격에 대한 값을 입력할 수 있다.

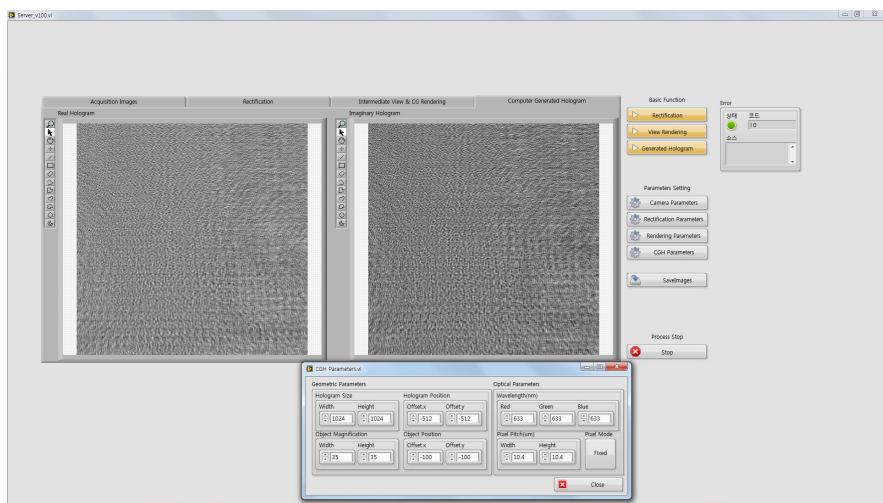


그림 18 홀로그램 생성을 위한 GUI  
Fig. 18. GUI for CGH

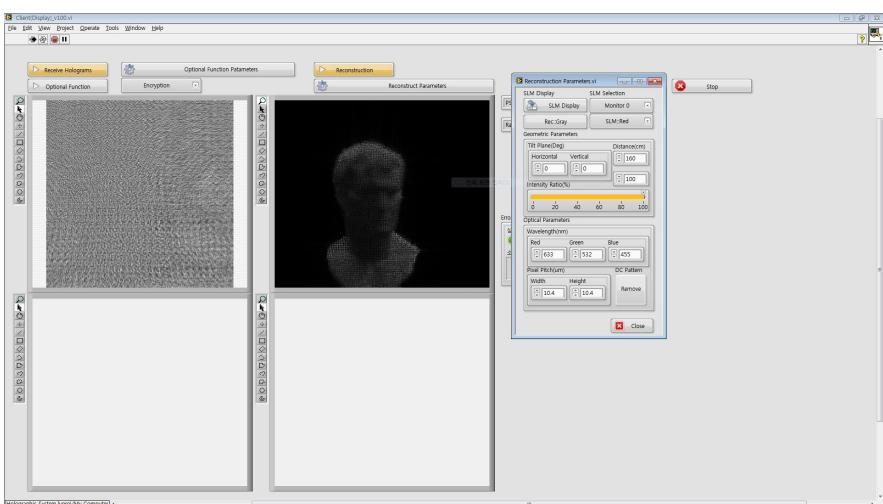


그림 19. 홀로그램 복원을 위한 GUI  
Fig. 19. GUI for hologram reconstruction

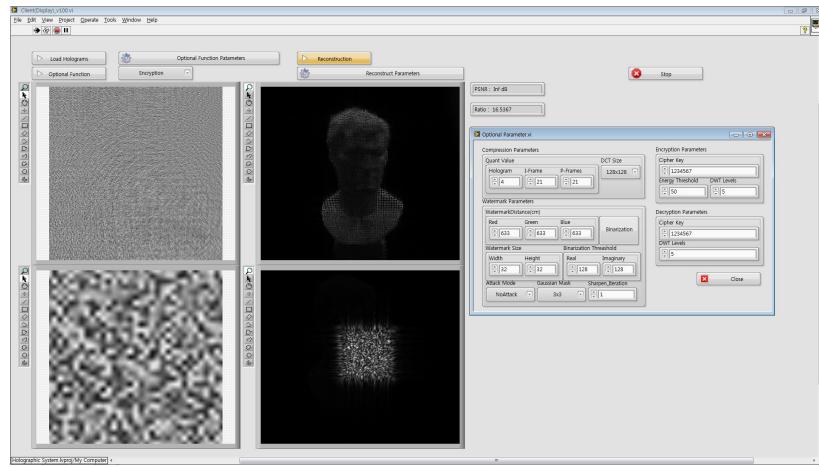


그림 20. 추가적인 함수를 위한 GUI  
 Fig. 20. GUI for optional functions

## 2. 시스템 성능

구현한 시스템은 표 2에 주요 함수의 성능을 나타냈다. 함수 종류는 필수 요소와 추가적 기능으로 분류하였다. 추가적 함수를 제외한 필수 함수는 초당 15 프레임을 생성할 수 있다(10,000개의 광원). 이는 홀로그램을 생성하는 시간에 따라 달라지는데 고성능 GPU 및 많은 수의 GPU를 이용할 경우 개선될 수 있다.

표 2. 주요 함수의 성능

Table 2. Performance for the major functions

Function Type	Function Name	Runtime (ms)	Resource
Basic	RGB/Depth Images acquisition	25/2	S/W
Option	Depth map up-sampling	29	S/W
Basic	Rectification	31	S/W
Basic	Intermediate view rendering	15	S/W
Option	CG Rendering	2	S/W
Basic	CGH generation (10k Light sources)	58	4 GPGPUs
Basic	Hologram reconstruction (optical)	-	SLM
Option	Hologram reconstruction(PC)	296	1 GPGPU
Option	Hologram compression/decompression	27835/85	S/W
Option	Hologram encryption /decryption	463/447	S/W
Option	Hologram watermark embedding/extrating	235/223	S/W

## V. 결 론

본 논문에서는 수직 리그 시스템에서 획득한 컬러 및 깊이 영상을 이용하여 시청자의 시점에 따른 홀로그램을 생성하기 위한 시스템을 구현하였다. 수직 리그로부터 3쌍의 컬러 및 깊이 영상 쌍을 입력받고 ToF(Time of Flight)카메라의 특성으로 상대적으로 작은 깊이 영상은 업 샘플링을 통하여 고해상도로 크게 만들고 렌즈 왜곡 보정을 통하여 고화질의 컬러 및 깊이 영상 쌍을 생성한다. 또한 일치하지 않은 영상 쌍과 수평이 일치하지 않는 3세트의 영상 쌍의 수평을 맞추기 위하여 영상 교정(Rectification)을 통하여 홀로그램을 생성하기 위한 영상 쌍을 생성한다. 시청자의 시점에 따라 홀로그램을 생성하기 위하여 Kinect를 통한 얼굴 추적 알고리즘을 통하여 시청자의 위치를 찾아서 3세트의 영상 쌍을 이용하여 가상 시점의 영상을 생성한다. 생성한 가상 시점의 영상 쌍을 이용하여 홀로그램 생성하여 가상 시점에 해당하는 홀로그램을 생성한다. 연산량이 많은 홀로그램의 고속 생성을 위하여 PCIe 확장 슬롯을 이용하여 다중 GPGPU 기법을 적용하여 홀로그램을 생성한다. 변환된 홀로그램은 광학 장치로 복원 및 Fresnel 변환을 통하여 시뮬레이션 상으로 복원할 수 있다. 또한 생성한 홀로그램을 기존의 압축, 워터마킹, 암호화 기법을 적용할 수 있는 모듈을 포함하였다.

구현한 시스템은 차세대 디지털 휠로그램 방송 영상 시스템 등을 위한 것으로 이와 같은 시스템의 연구와 개발에 대한 국내연구가 많지 않은 상황에서 차세대 영상 시대를 위해 여러 분야의 연구와 개발에 참고가 될 수 있기를 기대 한다.

### 참 고 문 헌 (References)

- [1] HILAIRE, PIERRE ST, "HOLOGRAPHIC VIDEO: The ultimate visual interface?", Optics and Photonics News, Vol. 8 Issue 8, pp.35- (1997).
- [2] Quentmeyer, Tyeler; Plesniak, Wendy J; Bove Jr , V Michael, "Computing real-time holographic video content with off-the-shelf PC hardware," Frontiers in Optics (FiO) 2004.
- [3] Y. Ichihashi, H. Nakayama, T. Ito, N. Masuda, T. Shimobaba, A, Shiraki, and T. Sugie, "HORN-6 special-purpose clustered computing system for electroholography", Optics Express, vol. 17, no. 16, pp. 13895-13903, Aug, 2009
- [4] Tsang, Wai Ming; Cheung, Wai Keung; Poon, Ting-Chung, "Realtime video holographic system based on range camera, sub-lines and integrated Fresnel lines," Digital Holography and Three-Dimensional Imaging (DH) 2011.
- [5] N. Costa and A. Cartaxo, Advances in Lasers and Electro Optics, INTECH, April 2010.
- [6] Pike User Manual, <http://www.alliedvisiontec.com/emea/products/cameras/firewire/pike.html>, ALLIED VISION, Aug. 2012.
- [7] SR4000 Data Sheet, <http://www.mesa-imaging.ch/prodview-4k.php>, MESA Imaging, Oct. 2010.
- [8] Min Soo Ko, Dong Wook Kim, Douglas L. Jones, Jisang Yoo, "A New Common-hole Filling Algorithm for Arbitrary View Synthesis", 4th International conference of 3D systems and applications, pp.242-245, Jun. 2012.
- [9] Kinect for windows SDK, <http://www.microsoft.com/en-us/kinectforwindowsdev/Downloads.aspx>, Microsoft.
- [10] Byung-Kuk Seo, Jungsik Park, Hanhoon Park, Jon-Il Park, "Real-time visual tracking of less textured three-dimensional object on mobil platforms", Optical Engineering, Jan. 2013.
- [11] YoonHyuk Lee, Dong Wook Kim, Young Ho Seo, "High-Speed Generation Technique of Digital Holographic Contents based on GPGPU", Journal of the Korea Society of Digital Industry and Information Management, March. 2013.
- [12] Yoon Hyuk Lee, Hyun Jun Choi, Young Ho Seo, Dong Wook Kim, "Digital Hologram Watermarking with the Fresnel Diffraction Model", International conference of 3D systems and applications 2014, May. 2014.
- [13] Y. H. Seo, H. J. Choi, J. S. Yoo, G. S. Lee, C. H. Kim, S. H. Lee, S. H. Lee, and D. W. Kim, "Digital hologram compression technique by eliminating spatial correlations based on MCTF," OPTICS COMMUNICATIONS, Vol. 283, pp. 4261-4270, July 2010.
- [14] Y. H. Seo, H. J. Choi, and D. W. Kim, "Digital Hologram encryption using discrete wavelet packet transform", OPTICS COMMUNICATIONS, Vol.282, pp. 367-377, Feb. 2009.
- [15] U. Schnar and W. Jueptner, "Digital Holography", Springer, Berlin, Germany, 2005.
- [16] SLM data sheet, <http://holoeye.com/spatial-light-modulators/slm-pluto-phase-only>, HOLOEYE.
- [17] Y.H. Seo, Y.H. Lee, J.M. Koo, W.Y. Kim, J.S. Yoo and D.W. Kim, "Digital Holographic Video Service System for Natural Color Scene", pp.113106, Vol. 52(11), Nov. 2013.
- [18] OpenCV Documentation, <http://opencv.org/>.
- [19] Jong Myeong Lim, Jisang Yoo, "Super-resolution Algorithm using Discrete Wavelet Transform for Single-image", Journal of broadcast engineering, The Korean Society of Broadcast Engineering, pp. 344-353, Vol. 17(2), March. 2012.
- [20] H. Yoshikawa, S. Iwase and T. Oneda, "Fast Computation of Fresnel Holograms employing Differences", Proceeding of SPIE, Vol.3956, 2000.
- [21] U. Schnar and W. Jueptner, Digital Holotaphy, Springer, Berlin, Germany, 2005.
- [22] Fresnel diffraction, [http://en.wikipedia.org/wiki/Fresnel\\_diffraction](http://en.wikipedia.org/wiki/Fresnel_diffraction), wikipedia.
- [23] H. J. Choi, Y. H. Seo, J. S. Yoo, and D. W. Kim, "Digital watermarking technique for holography interference patterns in a transform domain," OPTICS and LASERS in ENGINEERING, Vol. 46, No. 4, pp. 343-348, Jan. 2008.
- [24] Y. H. Seo, H. J. Choi, D. W. Kim, "Digital Hologram Encryption using Discrete Wavelet Packet Transform", Optics Communications, Vol. 282, pp.367-377, Oct. 2009.
- [25] 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, Vol.52, No.11, pp. 113106-1 - 113106-8, Nov. 2013.

---

## 저자 소개

---



이 윤 혁

- 2012년 : 광운대학교 전자재료공학과 졸업(공학사)
- 2014년 : 광운대학교 일반대학원 졸업(공학석사)
- 2014년 ~ 현재 : 광운대학교 일반대학원 박사과정
- 주관심분야 : 디지털 홀로그램, SoC 설계



서 영 호

- 1992년 : 광운대학교 전자재료공학과 졸업(공학사)
- 2001년 : 광운대학교 일반대학원 졸업(공학석사)
- 2004년 : 광운대학교 일반대학원 졸업(공학박사)
- 2003년 ~ 2004년 : 한국전기연구소 연구원
- 2005년 ~ 2008년 : 한성대학교 조교수
- 2008년 ~ 현재 : 광운대학교 교양학부 부교수
- 주관심분야 : 실감미디어, 2D/3D영상 신호처리, 디지털 홀로그램



김 동 융

- 1983년 : 한양대학교 전자공학과 졸업(공학사)
- 1985년 : 한양대학교 전자공학과 졸업(공학 석사)
- 1991년 : Georgia 공과대학 전기공학과 졸업(공학박사)
- 1992년 ~ 현재 : 광운대학교 전자재료공학과 교수
- 1996년 ~ 2001년 : 반도체 설계 교육센터(IDEC) 광운대지역 운영위원
- 1997년 ~ 2000년 : 대학산업기술 자문단 자문위원
- 1998년 ~ 2001년 : IEEE SSCS/EDS Joint Chapter of Korea Section 학술위원장
- 1999년 ~ 2001년 : AP-ASIC 학술위원장
- 2001년 : 대한전자공학회 교육연구위원회 부위원장
- 2005년 ~ 2006년 : 대한전자공학회 협동이사
- 2006년 ~ 2008년 : 3차원영상협회 이사
- 2009년 : 실감미디어산업협회 이사
- 2011년 ~ 2013년 : 한국방송공학회 상임이사
- 주관심분야 : 3D 영상처리, 디지털 홀로그램, 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication