

# 3차원 디스플레이를 위한 유니버설 가상 시점 영상 합성

최지윤\*□박종일\*

\*한양대학교 전자컴퓨터통신공학과

## 목 차

I. 서론	IV. GPGPU를 이용한 3차원 영상처리
II. 3차원 디스플레이 시스템의 구성	V. 결론
III. 유니버설 가상 시점 영상 합성	

## I. 서론

흑백 아날로그에서 시작되어진 방송 서비스는 디지털 방송으로 전환되면서 초고화질의 컬러 방송을 지향하고 있으며, 최근 3D 영화 아바타가 성공하고 국내외 가전사의 3DTV가 출시되면서 위성, 케이블, 지상파 방송에서 3D 실험 방송이 추진되고 있다.

3차원 콘텐츠를 제대로 즐기려면 이를 디스플레이 할 수 있는 기기가 있어야 한다. 현재 3차원 디스플레이 방식은 크게 안경식과 무안경식으로 나누어지는데, 적청 안경, 편광 안경, 셔터 안경 등을 사용하는 입체 디스플레이는 현재 상용화 되어 판매되고 있다. 그러나 이들 디스플레이는 안경을 써야만 하는 불편함 때문에 한계가 있어 무안경식 디스플레이에 관한 연구가 활발히 진행되고 있다. 무안경식 3차원 디스플레이 방식은 양안 시차를 이용한 공간 분할 방식과 허상 방식으로 크게 나누어진다.

기존 2차원 디스플레이에서 3차원 디스플레이로 넘어오면서 가장 중요한 차이점은 한 장면을 보여주기 위해 필요로 하는 요구 데이터양의 폭발적 증가에 있다. 기존 2차원 디스플레이에서 한 시점에서의 영상만이 필요했다면 3차원의 경우 보여주고자 하는 시점의 개수만큼 데이터가 추가되며 경우에 따라 각 시점의 깊이 지도와 각 시점들과의 기하학적 관계에 대한 데이터도 추가되기 때문이다. 따라서 특별한 대책 없이 기존 CPU만을 사용하여 증가한 데이터를 전송 및 처

리하는 데에는 속도 면에서 한계를 가지고 있다.

또한 현재 연구되고 판매되는 3차원 디스플레이의 경우 디스플레이 하고자 하는 3차원 콘텐츠의 종류, 디스플레이를 제작하는 제조사의 정책 및 디스플레이 사양 등에 의존적이며 상호 호환성이 극히 떨어지고 있다. 이런 문제들을 해결하는 것은 양질의 3차원 콘텐츠를 만드는 만큼이나 중요한 문제이기 때문에 이를 해결해야 할 필요가 있다. 본고에서는 이런 문제점들을 해결하기 위해, 3차원 그래픽 처리장치인 그래픽 카드를 이용하는 유니버설 가상 시점 영상 합성 유닛을 제안하고 관련 실험 결과를 제시한다.

본고의 구성은 다음과 같다. 2장에서 현재 연구되거나 상용화된 3차원 디스플레이 시스템의 종류를 소개하고, 3장에서 유니버설 가상 시점 영상 합성 유닛에 대한 개요와 다시점 영상 합성에 대해 설명한다. 4장에서 GPGPU(General-Purpose computation on Graphics Process Unit)에 대한 개념을 설명한 후, 이를 3차원 영상처리에 응용하여 고속화 시킨 사례에 대해서 소개하고 마지막으로 결론을 맺는다.

## II. 3차원 디스플레이 시스템의 구성

3차원 영상을 획득하기 위해서는 카메라가 최소 2대 이상 있어야 하며, 이를 재생하기 위해 각 눈에 다른 영상을 보여 줄 수 있는 디스플레이 장치가 필요하

다. 일반적으로 3차원 디스플레이는 양쪽 눈에 양안 시차가 있는 영상을 동시에 보여주는 디스플레이 장치이며 영상을 보여주는 방식에 따라 그림 1과 같이 세분화 하여 나눌 수 있다.

무안경식 입체 디스플레이의 대표적인 방식은 렌티큘러(Lenticular) 방식, 인테그랄 이미징(Integral Imaging) 방식 그리고 홀로그래피(Holography) 방식으로 나눌 수 있다.

렌티큘러 디스플레이 방식은 반원통의 투명 렌즈판을 수직적으로 배열하여 붙여 넣은 것으로 스크린 뒤에 렌티큘러 한 피치당 각 시점의 화소를 대응시켜 놓음으로써 관찰자가 입체로 보게 하는 원리이다.

Integral Imaging 방식은 촬영할 때 초소형 요소렌즈를 다수 배치하여 안경 없이 입체 영상을 볼 수 있게 하는 방식으로, 이 방식은 수평, 수직 모든 방향으로 자유롭게 입체감을 느낄 수 있는 반면, 인접한 요소렌즈

간의 간섭이 최소화 되도록 정밀한 조정이 요구되어 진다.

홀로그래피는 광파의 간섭과 회절을 이용한 파면의 기록 기술로서 레이저에서 나온 빔을 렌즈에 통과 시켜 구면파로 만들고, 이 구면파가 피사체에 부딪쳐서 반사한 산란광과 거울로 반사시킨 평행 광을 사진 필름에 닿게 하여 중첩시키면 두 빛이 겹쳐진 부분에 간섭무늬가 생기며 이를 현상화한 것을 홀로그래피라고 한다. 홀로그래피를 만들 때 참조광으로 사용했던 평행 광을 같은 방향으로 입사시켜 홀로그래피에 조명하면, 피사체가 있던 위치에 동일한 3차원 상이 재현되어 지는 원리이다.

어떤 방식이든 송신측은 3차원 영상 입력장치, 영상 처리장치, 전송장치로 구성되어 있으며 수신측은 수신 및 영상 복원, 입체 모니터 장치 등으로 구성되어 진다.

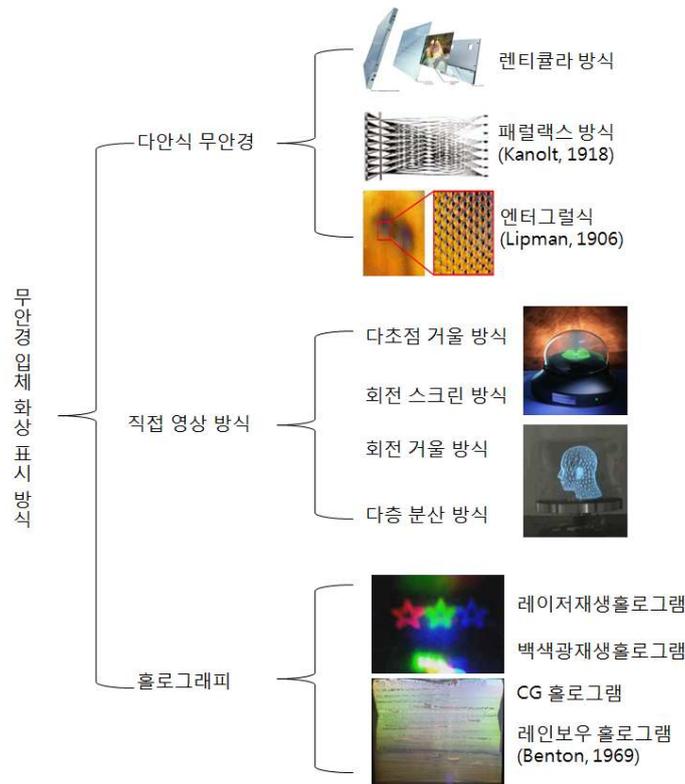


그림 1. 무안경식 입체 화상 표시 방식 분류도

### III. 유니버설 가상 시점 영상 합성 유닛

본 장에서는 유니버설 가상 시점 영상 합성 유닛 (Universal View Synthesis Unit, 이하 UVSU)에 대한 개략적인 개념을 설명하고 이를 이용한 다시점 영상 합성 방법에 대해 설명한다.

#### 3.1. UVSU 개요



그림 2. UVSU 개념도

3차원 디스플레이에서 영상을 출력하기 위해서는 다양한 시점에서의 영상 데이터를 입력 받아야 한다. 그림 2는 UVSU 개념도로서 임의의 시점에서 획득한 영상들을 입력 받아 목표 디스플레이에서의 데이터 출력의 변환을 목표로 함을 보여주고 있다. 이 때 고려해야 할 점으로 데이터 변환 및 연산의 고속화, 모든 디스플레이에 적용 가능한 범용성, 상호 디스플레이간 변환이 가능한 유연성과 개발자 입장에서 쉬운 영상 처리가 가능하도록 도와주는 접근성이다. 이를 위해 UVSU 개념을 제시하며, 고속처리를 위해 이를 GPU를 사용하여 구현한다.

다양한 종류의 작업을 처리하기 위해 범용으로 설계된 CPU와 달리 GPU는 3D 그래픽 데이터를 빨리 처리한다는 목적을 위해 최적화 되어있다. 최근에 GPU의 데이터처리 속도가 비약적으로 향상됨에 따라 GPU를 마치 CPU처럼 여러 용도로 사용하겠다는 GPGPU가 일반화되고 있다. 거의 모든 컴퓨터에 OpenGL을 지원하는 GPU가 달려 있어 상호간 호환성이 좋으며, 3차원 디스플레이와 같이 규격이 다른 장치와의 호환을 위한 디스플레이 출력의 유연성이 우수하다. 또 GPU 언어가 C/C++를 바탕으로 하고 있어 개발 접근성에도 유리하다.

#### 3.2. 영상 합성 방법

임의의 가상 시점에 대한 영상을 합성하기 위해서는 참조 시점들의 기하학적 관계를 활용한다. 참조 시점과 가상 시점 위치의 기하학적 관계는 행렬의 형태로 나타낼 수 있으며, 시점의 변화는 시점과 영상간의 관계를 표현하는 행렬간의 연산으로 나타낼 수 있다. 따라서 카메라와 영상의 기하 관계를 이용하여 중간 시점 영상을 구할 수 있다[1].

이러한 연산을 가상 시점 수만큼 반복하는 구조는 CPU로 처리하는 것보다 GPU의 스트림 프로세서를 활용한 연산을 통해 고속화함으로써 큰 속도 향상을 기대할 수 있다.

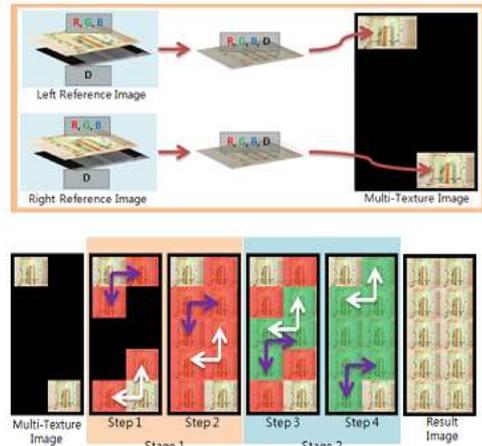


그림 3. 가상 시점 생성 순서도

그림 3은 위의 관계를 이용하여 두 시점의 색상 정보와 깊이 정보를 입력 받아 이를 각각의 텍스처로 저장 후 GPU로 보내 가상 시점을 생성하게 되고, 결과를 하나로 묶어서 출력하는 시스템을 보여주고 있다. 위의 그림에서, 붉은 영역은 해당 시점에서 가까운 기준 시점을 기준으로 전방 사상을 하게 되고 채워지지 않은 영역(hole)은 반대쪽 기준 시점의 전방 사상을 통해 채워짐을 의미한다 [2].

이 때 참조 시점의 정보를 토대로 병렬 처리하고 동시에 생성하고자 하는 가상 시점들을 병렬 처리를 통해 합성하게 되면 동시에 다수의 가상 시점에서의 영상을 합성할 수 있기 때문에 고속 처리가 가능하다.

#### IV. GPGPU를 이용한 3차원 영상처리

본 장에서는 3장에서 언급한 GPGPU의 기본적인 구조와 이를 실제 3차원 영상 처리에 응용하여 고속화 시킨 사례에 대해서 설명한다.

##### 4.1. GPGPU를 이용한 고속 영상 처리

그림 4는 고정된 그래픽스 파이프라인을 보여주고 있다. Vertex shader는 정점 단위의 연산을 수행하는 shader 모델을 의미한다. Vertex shader는 좌표계의 변환을 계산하여 변화된 좌표와 함께 출력한다. Fragment shader는 화소 단위의 연산이 이루어지며 일반적인 GPGPU 분야에서 처리할 데이터를 텍스처 형태로 메모리에 저장 후 fragment shader에서 이에 접근하여 데이터를 병렬 방식으로 나누어 처리하여 연산 속도가 높다.

하지만 고정된 그래픽스 파이프라인에서의 정형적인 연산으로는 다양한 그래픽 효과의 구현이 제약된다. 이를 해결하기 위해 NVIDIA에서는 기존의 그래픽스 파이프라인 구조에 종속되지 않고 GPU 자체를 병렬처리를 위한 프로세서로 사용할 수 있도록 한 CUDA를 발표하였으며, 대략적인 구조는 그림 5와 같다.

CUDA 프로그램은 호스트 코드(Host Code)와 디바이스 코드(Device Code)로 나뉘어진다. 호스트 코드는 CPU에서 실행되며 CUDA를 이용해 연산하기 전에 필요한 작업들과 연산이 끝난 후 결과물을 처리하는 작업을 수행한다. 디바이스 코드는 GPU상의 CUDA 코어에서 연산을 수행하게 된다. 이처럼 CUDA 코어 안에서 실행 될 디바이스 코드를 커널(Kernel)이라 하며, 하나의 커널은 하나의 스레드에서 실행된다. GPU안에 포함된 CUDA 코어는 하나의 스레드를 실행 할 수 있으며 여러 개의 스레드를 묶은 것을 스레드 블록(Thread Block)이라 하고 스레드 블록을 여러 개 묶은 것을 그리드라 한다. 그리드는 멀티프로세서의 실행을 위해 scheduling block에 의해 디바이스에서 실행이 된다. 각 멀티프로세서는 일괄 처리가 가능한 블록들을 처리 하며 한 블록은 하나의 프로세서로 처리가 가능하기 때문에 다수의 프로세서를 이용하여 고속 연산이 가능하게 된다.

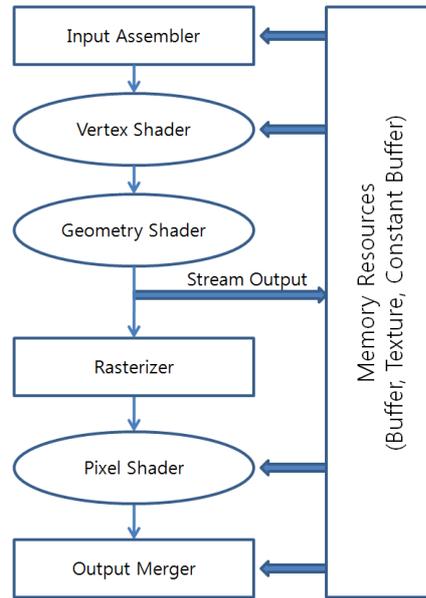


그림 4. 고정된 그래픽 파이프라인

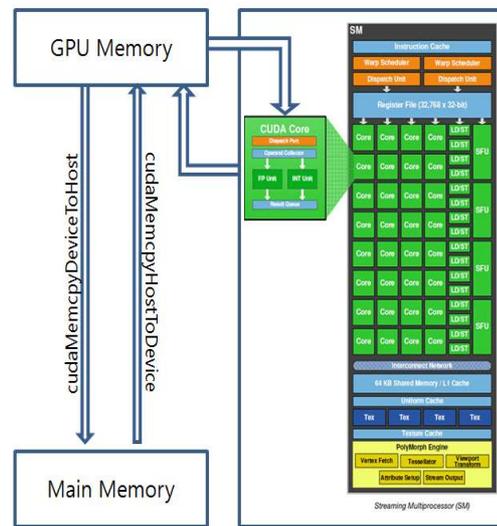


그림 5. CUDA 파이프라인

##### 4.2. 3차원 영상 처리 사례

그림 6은 CUDA를 이용한 개발 사례[3]의 일부로서 많은 응용분야에서 속도 향상을 가져 왔다. 이 절에서는 3차원 영상 처리를 GPGPU에 적용하여 속도가 향

상된 사례에 대하여 디스플레이 장치별로 분류 하여 개략적으로 소개하고 각 사례에서 CPU와 비교하여 속도 향상이 어느 정도 이루어졌는지에 대해서 설명한다.

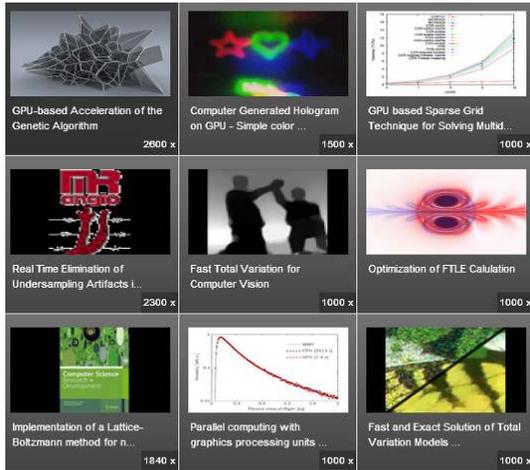


그림 6. CUDA를 이용한 고속화 사례

#### 4.2.1. 렌티큘러 방식에서의 응용사례

렌티큘러 방식은 그림 7에서와 같이 렌티큘러 렌즈를 세로로 배열시킨 렌티큘러 스크린을 통해 좌우 화

상을 굴절시켜 양 눈으로 각각의 화상을 보내는 방법을 사용한다. 렌티큘러 방식의 무안경식 입체 디스플레이에 입체감 있는 영상을 표현하기 위해서는 다중 시점 영상을 필요로 한다. 하지만 다중 시점을 획득하여 통신채널을 통해 전송하는 것은 현실적으로 극히 어려운 문제이다. 이 문제를 해결하기 위해 한정된 참조 영상과 깊이 정보를 토대로 임의의 가상 시점 영상을 실시간으로 생성하는 기술이 소개 되었다 [2][4]. 신흥창 등[4]은 CUDA를 이용하여 2개의 참조 시점을 가지고 8개의 가상 시점을 생성하는 영상 기반 가상 시점 합성을 제안하였다. 최지윤 등[2]은 3개의 참조 시점 영상의 기하학적 정보를 가지고 10개 시점의 가상 영상을 합성하였는데 고정된 그래픽 파이프라인의 프래그먼트 셰이더를 OpenGL Shading Language(OpenGLSL)를 이용하여 구현하였다. 두 방식 모두 3차원 디스플레이를 위해 필요한 가상 시점의 간격이 동일하다는 병렬성을 이용하여 해당 부분을 GPU로 병렬 처리하여 속도 향상을 가져왔으며 CPU에 비해 13배에서 최고 50배 가까운 속도 향상을 보였다.

#### 4.2.2. 집적영상(Integral Imaging) 방식의 사례

집적영상법은 무안경식 3차원 디스플레이 방식의 하나로, 1908년 Lippmann에 의해서 제안되었다[5]. 그림 8은 직접영상 방식의 개념도이다. 직접 영상법은 어

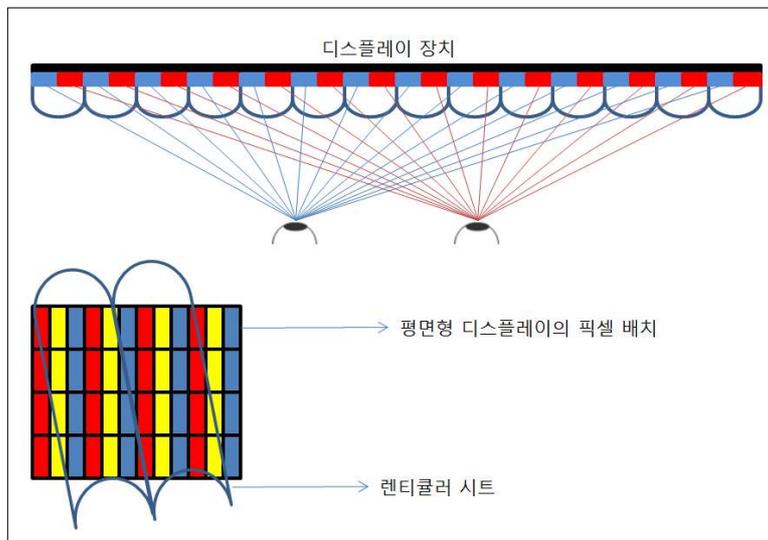


그림 7. 렌티큘러 방식의 디스플레이 개념도

러 개의 요소 렌즈들로 구성된 렌즈 어레이를 이용하여 대상물의 3차원 정보를 element image의 형태로 저장하고 그 기초 영상을 다시 렌즈 어레이를 통하여 3차원 영상을 표현하는 장치로서, 렌즈 어레이의 병렬성 때문에 GPGPU에 적합한 특징을 가지고 있다. C. H. Messom 등[6]은 고정밀 집적영상 알고리즘을 GPU로 구현하고 float형과 double형에서의 CPU와 GPU의 수행 시간을 비교하였는데, 작은 영상의 경우 CPU나 GPU에서의 속도 차이가 크지 않지만, 영상의 크기가 커질 경우 연산 수행 시간의 차이가 커졌으며 GPU로 구현한 것이 CPU에서 구현할 수 있는 가장 빠른 경우보다 10배 이상 빠르다고 하였다. 이는 영상 데이터의 용량이 클수록 GPU에서 처리하는 것이 CPU보다 훨씬 효율적임을 시사하고 있다.

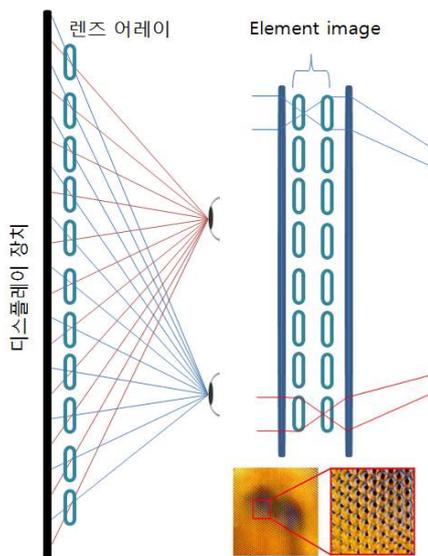


그림 8. 직접 영상 방식 개념도

#### 4.2.3. 홀로그래피 사례

홀로그래피란 D. Gabor가 제안한 방식으로서 간섭성이 좋은 빛(레이저)을 사용하는 방식이다. 레이저 광이 지나가는 광로에 선속분할기(Beam splitter)를 놓으면 투과하는 레이저 광과 반사하는 레이저 광으로 분리된다. 반사된 광은 볼록렌즈를 통해 대상을 비춘다. 이 때 투과된 빛을 중첩시키면 간섭무늬가 생성되고, 이를 다시 렌즈로 확대된 레이저 광을 비추면 촬영한

대상을 입체적으로 볼 수 있게 된다. 그림 9는 3채널 홀로그래피 방식의 개념도이다.

Shiraki 등[7]은 3채널의 홀로그램 영상을 재생시키기 위한 기본적인 CGH 알고리즘인 Fresnel 근사 알고리즘을 CUDA로 구현하였는데, 1,024개의 광원에 대한 CPU와 GPU의 연산 속도를 측정한 결과 GPU가 CPU에 비해 약 1,460배가 빠름을 보였으며, 나아가 홀로그램을 실시간(30 fps)으로 재생할 수 있음을 보였다. Shimobada 등[8]은 흑백 영상의 CGH 알고리즘을 OpenCL로 구현하였는데 725MHz의 1440개의 stream core를 내장한 AMD사의 RV870 GPU를 이용한 결과 CPU만 사용하였을 때보다 약 1,500배 이상의 속도 향상을 가져올 수 있음을 확인하였다.

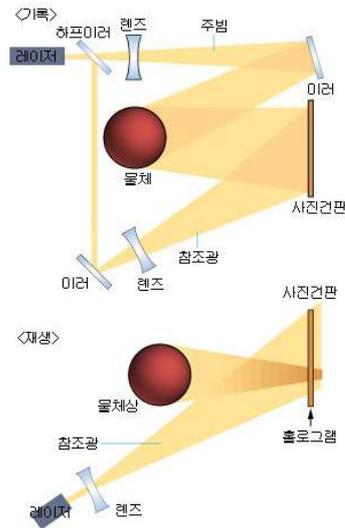


그림 9. 홀로그래피 개념도

#### V. 결 론

최근 아바타를 비롯한 3D 영화의 성공과 3D 입체 방송 서비스의 추진으로 3D 영상에 대한 관심이 급속히 증가하고 있다. 그러나 3D 콘텐츠는 기존 2D 방식에 비해 처리해야 할 데이터의 양이 폭발적으로 증가하므로 기존의 처리방식으로는 속도 면에서 근본적 한계를 가질 수밖에 없다. 다양한 종류의 3차원 콘텐츠를 다양한 3차원 디스플레이에 재현하기 위해서는 유연하

고, 효과적인 유니버설 가상 시점 영상 생성 방법이 요구되고 있다.

본고에서는 3D 디스플레이를 위한 유니버설 가상시점 합성 유닛(UVSU)을 구현하기 위한 요소기술을 설명하였다. GPU 프로그래밍을 활용하여 무안경식 3D 디스플레이를 위한 다시점영상을 고속 생성하는 방법을 기술하고, 몇가지 3D 디스플레이에 GPU 프로세싱이 적용된 사례에 대해서 살펴보았다. 성능이 비약적으로 발전하고 있는 GPGPU를 적절히 활용하면 고속 영상처리에 상당한 진전이 있을 수 있음을 최신 연구 사례를 통해 확인할 수 있었다. GPGPU를 활용한 고속 영상처리 기술은 향후 대용량 데이터 기반의 초고해상도 방송 시스템이나 3D 방송 분야에 큰 기여를 하게 될 것이다.

#### 참고문헌

[1] D. Scharstein, "View Synthesis Using Stereo Vision," Lecture Notes in Computer Science, Vol. 1583, 1999

[2] J. Choi, S.-W. Ryu, H.-C. Shin and J.-I. Park, "Real-time view synthesis system with multi-texture structure of GPU", Proc. of Consumer Electronics (ICCE), pp. 171-172, Jan, 2010.

[3] NVIDIA CUDA Zone <http://www.nvidia.com>

[4] H.-C. Shin, Y.-J. Kim, H. Park, and J.-I. Park, "Fast View Synthesis using GPU for 3D Display," In IEEE Trans. Consumer Electronics, vol. 54, no. 4, pp. 2068-2076, Nov. 2008.

[5] G. Lippmann, "La photographie integrale," C. R. Acad. Sci. 146, 446-451 (1908).

[6] C.H. Messom, A.L.C. Barczak, "High Precision GPU based Integral Images for Moment Invariant Image Processing Systems", Proc. of Electronics New Zealand Conference(ENZCON'08), 2008.

[7] A Shiraki, N. Takada, M. Niwa, Y. Ichihashi, T. Shimobaba, N. Masuda, T. Ito, "Simplified Electroholographic Color Reconstruction System Using Graphics Processing Unit and Liquid Crystal Display Projector," Opt.Express, 17, 16038-16045 (2009).

[8] T. Shimobaba, T. Ito, N. Masuda, Y. Ichihashi, N. Takada, "Fast calculation of computer-generated hologram using AMD GPU and OpenCL," Proc. of 3D System and Application(3DSA), pp. 246-249, May 2010.

#### 저자소개



최 지 윤 (JiYoun Choi)

2009년 2월 한양대학교 전자통신컴퓨터공학부 학사  
2009년 3월~현재 한양대학교 전자통신공학부 석사 과정

※관심분야 : 가상 현실, HCI, 3차원 영상 처리, 컴퓨터 그래픽스/비전, GPGPU



박 종 일 (Jong-H Park)

1987년 서울대학교 전자공학과 학사  
1989년 서울대학교 전자공학과 석사  
1995년 서울대학교 전자공학과 박사

1992년~1994년 일본 NHK방송기술연구소 객원연구원  
1995년~1996년 한국방송개발원 선임연구원  
1996년~1999년 일본 ATR지능영상통신연구소 연구원  
1999년~현재 한양대학교 전자통신컴퓨터공학과 교수  
※관심분야 : 가상현실, 증강현실, HCI, 3차원 영상처리, 컴퓨터그래픽스/비전