

# 육각형 렌즈 어레이로부터 요소영상을 생성하기 위한 병렬 처리 기법

## Parallel Processing Method for Generating Elemental Images from Hexagonal Lens Array

김도형\*, 박찬\*\*, 정지성\*\*, 권기철\*\*\*, 김남\*\*\*, 류관희\*  
충북대학교 디지털정보융합학과\*, 충북대학교 정보산업공학과\*\*, 충북대학교 정보통신공학부\*\*\*

Do-Hyeong Kim(dobong87@cbnu.ac.kr)\*, Chan Park(szell@cbnu.ac.kr)\*\*,  
Ji-Sung Jung(farland83@cbnu.ac.kr)\*\*, Ki-Chul Kwon(jshajsha@yahoo.co.kr)\*\*\*,  
Nam Kim(namkim@cbnu.ac.kr)\*\*\*, Kwan-Hee Yoo(khyoo@cbnu.ac.kr)\*

### 요약

대부분 집적영상 기법에서는 사각형 렌즈 어레이가 사용되고 있으며, 이로 인해 집적된 빛의 분포는 사각 격자 형태로 기록된다. 그러나 육각형 렌즈 어레이를 사용하면, 사각형 렌즈 어레이보다 더 높은 밀도의 빛의 분포와 이상적인 원형 렌즈에 가깝게 이미지를 기록 또는 재생 할 수 있다[4]. 육각형 렌즈 어레이 요소영상을 병렬 처리 기법을 사용하여 생성하기 위해서는 요소영상을 구성하는 각 화소에 대하여 그 화소가 속할 육각형 렌즈를 결정해야하고, 이 과정은 화면에 출력되는 모든 화소에 대하여 진행 되며 많은 계산량이 요구된다. 본 논문에서는 3D 볼륨 데이터를 사용하여 육각형 렌즈 어레이에 대한 요소영상을 생성하기 위해 OpenCL를 사용한 병렬 처리 기법을 제안한다. 제안 된 방법을 위한 실험에는 Male [128×256×256 화소] 볼륨데이터를 사용하였으며, 실험 결과 20×20개의 육각형 렌즈 어레이에 대해 요소영상을 초당 20~60장 생성할 수 있었다.

■ 중심어 : | 3차원 디스플레이 | 육각형 렌즈 어레이 | 집적영상 | 볼륨 데이터 | OpenCL |

### Abstract

According that most integral imaging techniques have used rectangular lens array, this integrated distribution of light is recorded in the form of a rectangular grid. However, hexagonal lens array gives a more accurate approximation of ideal circular lens and provides higher pickup/display density than rectangular lens array[4]. Using the parallel processing technique in order to generate the elemental imaging for hexagonal lens array, each pixel that compose the elemental imaging should be determined to belong to the hexagonal lens. This process is output to the screen for every pixel in progress, and many computations are required. In this paper, we have proposed parallel processing method using an OpenCL to generate the elemental imaging for hexagonal lens array in 3D volume data. In the experimental result of proposed method show speed of 20~60 fps for hexagonal lens array of 20x20 sizes and input data of Male[128x256x256] volume data.

■ keyword : | 3D Display | Hexagonal lens array | Integral Imaging | Volume Data | OpenCL |

\* 본 연구는 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2011-0025849)과 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No. 2012-0000479)에 의해 수행되었음.

접수번호 : #120320-003

심사완료일 : 2012년 05월 21일

접수일자 : 2012년 03월 20일

교신저자 : 류관희, e-mail : khyoo@cbnu.ac.kr

## I. 서론

집적영상(Integral imaging 혹은 IP, integral photography)는 1908년 Lippmann[1]에 의해서 제안되었지만 렌즈 어레이(lens array)의 제작을 위한 정밀 가공기술과 이미지에 대한 분해능의 사진 기술의 부재로 오랫동안 잊혀져왔다. 그러나 최근 디지털 기술과 더불어 초고해상도 영상처리 기술의 발전에 힘입어 다시 주목되고 있는 3D 디스플레이 기술 중 하나이다. 직접영상은 촬영하고자 하는 물체에 대해 영상센서 앞에 렌즈 어레이(lens array 혹은 fly's eye lens)를 놓는다. 이렇게 획득된 영상은 수많은 요소영상이 만들어지고, 이를 재구성하여 만들어진 영상 앞에 렌즈 어레이를 사용하면 3D 영상을 볼 수 있다.

집적영상은 무안경 방식의 완전시차(full-parallax), 연속시점(continuous viewpoint)을 제공하며 점광원(point light source)을 이용하거나 렌즈 어레이를 이용하여 영상을 획득 또는 재생을 하는 방식의 기술이다. 각각의 요소 렌즈를 통해 얻어진 시차 정보는 요소영상(elemental image)라 칭하며 획득할 때의 렌즈 어레이와 동일한 조건으로 재생하는 방법(광학적 방법)과 컴퓨터 그래픽을 이용하여 요소영상을 생성하여 재생하는 방법으로 나눌 수 있다. 본 논문에서는 가상공간에 존재하는 3차원 객체를 대상으로 가상으로 설정한 렌즈 어레이를 통해 보여 지는 요소영상을 컴퓨터를 통해 생성하는 기법(CGII, computer generated integral imaging)[2]에 관한 것이다.

지금까지 대부분의 연구는 컴퓨터를 이용하여 사각형 렌즈 어레이로부터 요소영상을 효율적으로 얻는 기법들이었다. 대표적인 기법으로는 요소영상을 구성하는 각 화소의 영상을 구하는 PRR(point retracing rendering)[3], 하나의 렌즈에 가상 카메라를 설정하여 요소영상을 구하는 MVR(multiple viewpoint rendering)[3], 모든 렌즈를 통해 지나가는 벡터의 유형이 한정되어 있는 특성을 이용하는 PGR(parallel group rendering)[3], 렌즈를 통해 생성되는 벡터를 일정하게 근집화 하여 처리하는 VVR(viewpoint vector rendering)[3] 등이 있다. 최근 들어 사각형 렌즈로부터 나타나는 요소영상을

OpenCL를 이용하여 생성하는 병렬 처리 기법이 권기철 등[5]에 의해 제시되었다. 이 방법에서는 컴퓨터에서 제공하는 최대한의 스레드(thread)를 생성한 후, 각 스레드에 요소영상을 구성하는 특정 화소의 색깔(color)를 구할 수 있도록 하였다. 사각형 렌즈 어레이를 구성하는 각 렌즈는 3차원 공간의 가상 카메라로 재현할 수 있어, 스레드에서 처리해야 하는 작업은 렌즈의 중심을 통과하는 광선과 뷰 볼륨(view volume)에 존재하는 3차원 객체의 교차점을 구하는 것이다. 권기철 등[5]은 이 문제를 매우 효과적으로 해결할 수 있는 기법을 제시하였다.

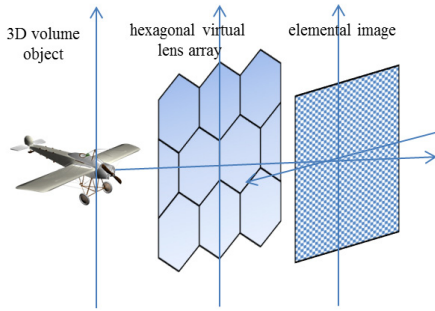
사각형 렌즈 어레이와 비교하여 육각형 렌즈 어레이를 사용하면, 사각형 렌즈 어레이 보다 더 높은 밀도의 빛 분포와 이상적인 원형 렌즈에 가깝게 이미지를 기록 또는 재생 할 수 있다[4]. 그러나 기존에는 사각형 렌즈 어레이에 대한 요소영상 생성에만 OpenCL을 이용한 병렬 처리 기법이 연구 되었으며, 육각형 렌즈 어레이에 대한 OpenCL을 이용한 병렬 처리 연구는 없었다. 육각형 렌즈 어레이로 확장하려면 뷰 볼륨 형태가 달라져 권기철 등[5]이 제시한 방법을 그대로 사용하기란 여러 문제가 발생한다. 본 논문에서는 이를 해결하기 위해 방법을 제시하고, 육각형 렌즈 어레이로부터 요소영상을 병렬 처리로 생성할 수 있는 기법을 제시한다.

본 논문의 제 2장에서는 육각형 렌즈 어레이의 형태와 뷰 볼륨 처리 기법과 요소영상을 생성하기 위한 병렬 처리 기법을 제시하고, 제 3장에서는 실험 결과를 제시한다. 제 4 장에서는 논문 결과를 요약하고 향후 연구 과제를 제시한다.

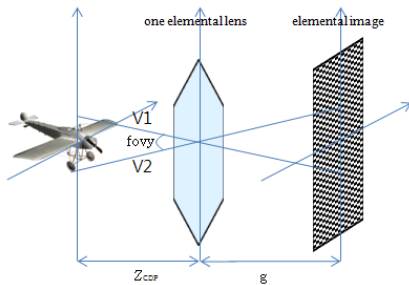
## II. 육각형 렌즈 어레이에 대한 요소영상

[그림 1]의 (a)에서 보는 바와 같이 무안경식 3D입체를 가시화하기 위해서는 픽업 과정을 통해 요소영상을 생성하여야 한다. 현재 렌즈 어레이에 부착한 3D 입체 카메라가 없으므로 연구 개발을 위해 컴퓨터를 이용한 요소영상 생성 기법이 많이 사용된다[3]. [그림 1]의 (b)에서 보는 바와 같이 컴퓨터를 이용하여 육각형 렌즈

어레이로부터 요소영상을 생성하는 과정을 살펴보면, 육각형 렌즈들이 격자모형으로 배치되고 이 렌즈 어레이는 각 렌즈의 중심에 가상의 원근 카메라를 놓고, 그 중심에서 렌즈의 끝 부분을 연결하는 두 벡터(V1, V2)에 의해 가상카메라의 시야각(fovy,  $\theta$ )이 결정된다. 그리고 디스플레이와 일정한 간격(g)을 띄워 배치하면 렌즈 어레이와 물체간의 거리(Zcdp)에 의해 가상 카메라의 초점 거리(f)가 결정된다. 렌즈 어레이를 구성하는 렌즈를 통해 본 3차원 객체가 대응하는 디스플레이 영역에 나타난다. 지금까지 설명한 내용은 [그림 1]에 잘 나타나 있다.



(a) 육각 렌즈 어레이에 대한 집적영상 픽업 개념도



(b) 집적영상 픽업에 대한 파라미터 정의

그림 1. 육각형 렌즈 어레이로부터 생성 될 요소영상기법에 서 요소영상을 얻는 픽업 과정의 기본 개념도

$N \times N$  개의 육각형 렌즈로 구성된 육각형 렌즈 어레이를 통해 만들어지는 요소영상은 각 렌즈로부터 얻어진 영상을 합쳐서 만들어진다. 디스플레이의 해상도(resolution)에 따라 각 렌즈에 대한 영상의 크기가 화소사이즈(pixel size)로 결정된다. [그림 2]는  $3 \times 3$ 개 육

각형 렌즈로 구성된 렌즈 어레이이며, 하나의 렌즈는  $\Delta x \times \Delta y$  화소로 구성된다.

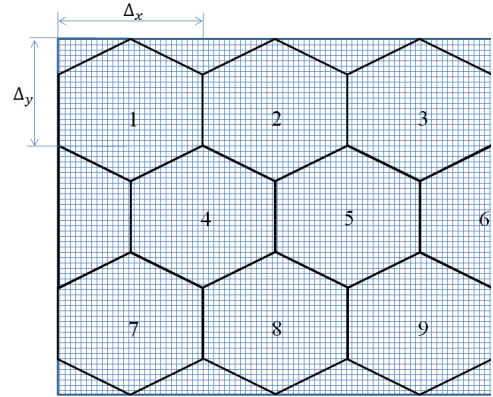


그림 2.  $3 \times 3$ 개의 육각형 렌즈로 구성된 렌즈 어레이로부터 생성된 요소영상 형태

### III. 육각형 요소영상 생성 기법

[그림 3]은 제안 된  $N \times N$ 개의 육각형 요소영상 생성 구조도를 나타낸다. [그림 3]의 (a)는 육각형 요소영상을 생성하는 계략적인 구조도이다. 프로그램 실행 시 볼륨 데이터를 로드하고, 렌즈 수(N), 디스플레이 패널 사이즈, 렌즈 어레이와 물체간의 거리(Zcdp), 가상카메라의 초점거리(f) 등을 입력 받는 입력과정, 입력받은 값을 이용해 요소영상 사이즈, 뷰 행렬, 가상 카메라 정보를 계산하여 OpenCL[10] 커널로 전달하는 계산과정, OpenCL 커널에서 병렬 처리로 각 화소의 색을 결정하는 렌더링과정, 그리고 색이 결정된 화소가 모여 요소영상을 디스플레이하는 출력과정으로 이루어져있고, 권기철 등[5]이 제시하였던 요소영상 생성 방법 구조와 동일한 과정을 거친다. 전체적인 과정은 동일하지만 육각형 렌즈 어레이에 대한 요소영상을 생성 할 때는 계산과정과 렌더링 과정에서 사각형 렌즈 어레이에 대한 요소영상을 생성 할 때보다 복잡한 과정이 추가된다. 그 이유는 [그림 4]의 사각형 렌즈와 육각형 렌즈의 차이를 통해 확인할 수 있다. 사각형 렌즈 어레이에 대한 요소영상을 생성할 때는 입력을 통해 얻은 요소영상의 패널 사이즈와 가로세로 렌즈 어레이 개수에 따라 각

화소가 위치하게 될 렌즈의 인덱스가 명확하게 계산되지만, 육각형 렌즈 어레이에 대한 요소영상을 생성할 때는 육각형 렌즈를 사각형으로 표현했을 때, [그림 4]에서 보는 바와 같이 임의의 ★ 화소가 특정 위치에 있을 경우 육각형의 인덱스가 중복될 수 있는 문제가 발생하여 이를 해결하기 위해 계산과정에서 입력받은 렌즈 어레이 정보에 따라 모든 육각형 렌즈의 중심점을 계산하고, OpenCL 커널로 전달해 렌더링 과정에서 화소가 속하는 인덱스를 찾는 연산을 수행해야 한다.

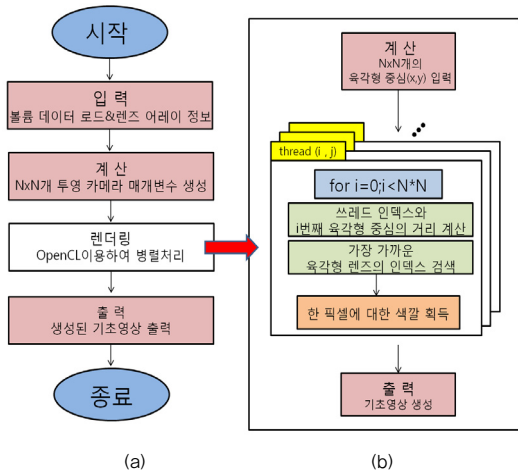


그림 3. (a) 제안 된 병렬 처리를 이용한 요소영상 생성 구조도, (b) Render 과정을 상세하게 표현한 구조도

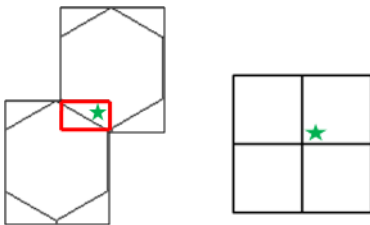


그림 4. 육각형 렌즈와 사각형 렌즈 어레이의 차이

계산과정에서  $N \times N$ 개 육각형의 중심점을 계산하는 방법은 [그림 5]에서 볼 수 있듯이 정육각형 어레이의 특성을 이용해 얻을 수 있다.



그림 5. 정육각형의 특성과 어레이 인덱스

육각형 렌즈 어레이는 사각형 렌즈 어레이와 달리 짝수 행과 홀수 행이 엇갈린 형태로 되어 있기 때문에 짝수 행과 홀수 행을 분리해서 중심점을 계산한다. 수식은 아래와 같다. 수식에서  $Center_x$ ,  $Center_y$  는 육각형 중심점의  $(x, y)$  좌표를 의미하며,  $s$ 는 정육각형의 한 변의 길이,  $HLens$ ,  $VLens$ 는 육각형 렌즈 어레이의 가로, 세로 렌즈 수를 의미한다. 육각형 렌즈 어레이에 대한 2차원 행렬  $(i, j)$ 에 대하여 병렬 처리를 위한 행렬의 인덱스는 선형 주소를 사용하여 접근되어야 하기 때문에  $idx$ 를 사용하여 행-우선(row-major) 방식에 따라 2차원 메모리를 선형 주소로 변환하여 데이터를 저장한다.

$$idx = i \times VLens + j \text{ 일 때}$$

(1) 짝수 행

$$i = 0, 2, 4, \dots, HLens - 2$$

$$0 \leq j < VLens$$

인  $i, j$ 에 대하여

$$Center_x[idx] = s \times \cos \frac{\pi}{6} \times (2j + 1)$$

$$Center_y[idx] = s \times (3i + 1)$$

(2) 홀수 행

$$i = 1, 3, 5, \dots, HLens - 1$$

$$1 \leq j \leq VLens$$

인  $i, j$ 에 대하여

$$Center_x[idx] = s \times \cos \frac{\pi}{6} \times 2j$$

$$Center_y[idx] = s \times \left( 3i - \frac{1}{2} \right) \text{ 이다.}$$

계산된 육각형 어레이의 중심점은 OpenCL 커널로 전달되어 특정 화소가 속하는 육각형 렌즈의 인덱스를 계산하는데 사용된다. [그림 6]과 같이 육각형 렌즈 어레이의 모든 중심점과 임의의 화소에 대한 거리를 계산하여 가장 가까운 거리에 있는 육각형 렌즈를 찾고, 화소의 색을 결정한다. [그림 6]에서는 임의의 화소 ★이 ②번과 ③번 렌즈가 겹쳐있는 부분에 위치하여 있고, 중심점과의 거리를 계산한 결과 ②번 렌즈에 속하게 되는 것을 알 수 있다. Render 과정에서 이러한 과정을 모든 화소에 대하여 병렬 처리 한다.

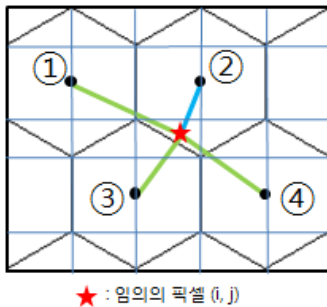


그림 6. 임의의 화소가 속하는 육각형 렌즈의 인덱스 결정 방법

병렬 처리를 통해 얻어진 모든 화소가 모여 디스플레이 패널을 통해 격자 모양의 육각형 렌즈 어레이 요소영상이 화면에 출력 된다.

#### IV. 실험결과

본 논문에서 제안한 방법을 실험하기 위해 MS Windows7 운영체제 PC환경에서 개발 도구로 MS Visual Studio 2008를 사용하였고, 3차원 그래픽스 API로 OpenGL[8]를 병렬처리 API로 OpenCL[9]을 사용하였다. PC의 하드웨어는 Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz와 4GB RAM, NVIDIA GeForce GTX 550 Ti를 사용하였다. 실험을 위해서 bucky, mummy, male, CTA 4가지 종류의 볼륨 데이터를 활용하여 요소영상을 생성하였다.

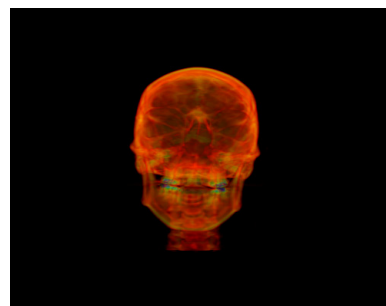
본 논문에서 제시한 육각형 렌즈 어레이에 대한 요소

영상을 서로 다른 4가지 볼륨 데이터, 렌즈의 개수, 그리고 렌즈 1개당 포함하는 화소 수의 경우로 나눠 실험을 진행 하였다. [표 1]은 본 논문에서 제시한 육각형 요소영상과 사각형 요소영상의 실행 속도를 비교하는 것이다. 두 실험은 20×20의 렌즈 어레이에 대한 요소영상 생성 속도를 나타낸 것이며, 육각형이 사각형과 비교해 느린 생성 속도를 나타냈다. 이러한 결과는 육각형 요소영상을 생성할 때 육각형 렌즈 어레이의 특성상 화소에 대응하는 렌즈의 인덱스를 찾는 과정에서 사각형 요소영상 생성 때 보다 각 화소 당 [가로 렌즈 수×세로 렌즈 수]만큼의 반복 연산을 더 진행하기 때문이다.

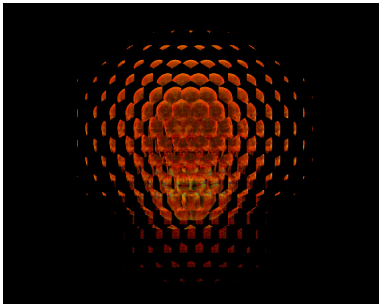
표 1. OpenCL 기반 볼륨 데이터의 크기와 렌즈 형태에 따른 요소영상 생성 시간 (렌즈 수: 20×20)

볼륨데이터	육각형렌즈[sec/fps]	사각형렌즈[sec/fps]
Bucky [32×32×32 화소]	0.016/62.50	0.011/90.91
Mummy [256×128×128 화소]	0.038/25.72	0.028/35.71
Male [128×256×256 화소]	0.047/21.28	0.031/32.26
CTA [512×512×79 화소]	0.049/20.14	0.034/29.41

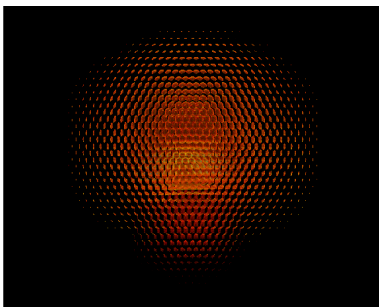
[그림 7]은 Male[128×256×256 화소] 볼륨 데이터에 대해 렌즈 수에 따라 생성된 요소영상 그림이며, [그림 8]은 Male 볼륨데이터를 사용하여 생성되는 요소영상의 렌즈 수를 증가시키며 실행시간을 비교한 그래프이다. 그래프에서 가로축은 렌즈의 개수를 나타내고, 세로축은 요소영상을 생성하는데 소요되는 시간(단위 : 밀리 초)을 나타낸다. 렌즈의 수가 증가할수록 실행 시간이 증가하였다.



(a) 볼륨데이터 원 영상



(b) 20×20 육각형 요소영상



(c) 50×50 육각형 요소영상

그림 7. [128×256×256 화소]크기의 볼륨 데이터에 대한 요소영상 생성

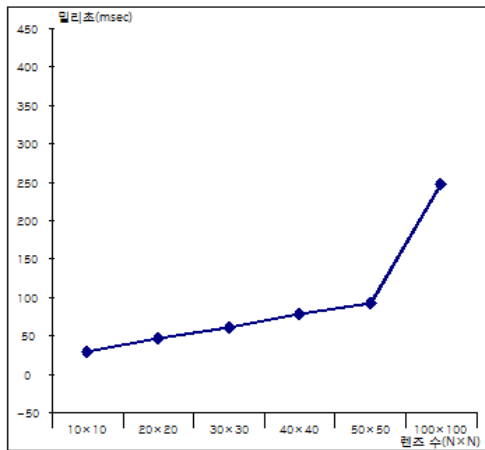


그림 8. 렌즈 수에 따른 실행 시간 (Male - 128×256×256 화소)

[표 2] 와 [그림 9]는 Male[128×256×256 화소] 볼륨 데이터에 대하여 렌즈 한 개당 포함하는 화소의 수에 따른 실행속도 결과와 그래프이다. 그래프에서 가로축

은 렌즈의 렌즈 한 개당 포함하는 화소의 수이고, 세로 축은 요소영상을 생성하는데 소요되는 시간(단위 : 밀리 초)을 나타낸다. 렌즈 어레이의 수는 30×30과 50×50으로 실험하였다. 렌즈 당 화소 수가 증가할수록 실행 시간이 급격히 증가하였고, 렌즈 수 50×50의 경우 46×40화소와 58×50화소에 대해서는 메모리 부족(out of memory)이 발생하였다.

표 2. 요소영상 크기에 따른 수행 시간 (Male - 128×256×256 화소)

요소영상의 한 렌즈에 대한 영상크기 [ $\Delta x \times \Delta y$ ]	렌즈 어레이 수[N×N]	
	30×30 [sec/fps]	50×50 [sec/fps]
12×10	0.030/32.25	0.186/5.35
23×20	0.140/7.14	0.641/1.56
35×30	0.265/3.77	1.333/0.75
46×40	0.467/2.14	메모리 부족
58×50	0.704/1.42	메모리 부족

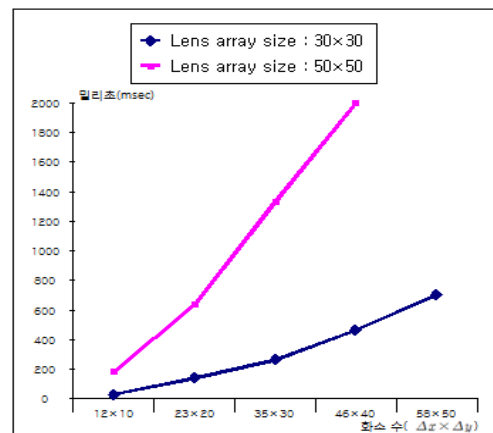


그림 9. 요소영상의 크기에 따른 실행 시간 (Male - 128×256×256 화소)

## V. 결론 및 향후 연구과제

본 논문에서는 OpenCL을 사용하여 볼륨데이터에 대한 육각형 렌즈 어레이로부터 요소영상을 컴퓨터로 생

성하는 병렬 처리 기법을 제안하였다. 제안한 병렬 처리 기법을 사용하여 실험 한 결과 20×20개의 육각형 렌즈 어레이에 Male [128×256×256 화소] 볼륨데이터를 이용해 초당 20~62장의 요소영상을 생성하였다.

그러나 렌즈의 개수(N)와 렌즈 당 포함하는 화소의 수(P)가 증가함에 따라 요소영상 생성 시간이 증가하였다. 고해상도 집적영상을 제작하기 위해서는 렌즈 어레이를 구성하는 렌즈의 수가 많아야 한다. 현재 본 논문에서는 렌즈 어레이 수를 최대 100×100개까지 실험 하였지만 적어도 지금의 실험보다는 많은 수의 렌즈 어레이가 필요하며, 그에 따라 연산 시간이 늘어나게 되어 실시간으로 집적영상을 생성하는데 문제점이 남아있다.

향후 연구로는 현재 실험에 사용되었던 볼륨데이터보다 큰 1024×1024×1024화소 정도의 고용량 데이터를 이용해 집적영상을 생성할 수 있는 기법과 렌즈의 개수와 영상의 해상도에 대한 영향을 최소화할 수 있는 실시간 집적영상 생성 기법에 대한 연구가 필요하다.

### 참 고 문 헌

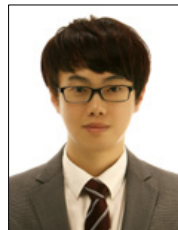
[1] G. Lippmann, "La photographie integrale," C.R Academic Science, Vol.146, pp.446-451, 1908.  
 [2] S. W. Min, K. S. Park, B. Lee, Y. J. Cho, and M. S. Hahn, "Enhanced Image Mapping Algorithm for Computer-Generated Integral Imaging System," Japanese Journal of Applied Physics, Vol.45, No.28, pp.L744-L747, 2006.  
 [3] Y. Igarashi, H. Murata, and M. Ueda, "3D Display System using a Computer Generated Integral Photography," Japan, Journal Applied Physics, Vol.16, pp.1683-1684, 1978.  
 [4] J. I. Ser, "A Study on the Properties of an Elemental Image depending on the Shape of Elemental Lens and the pick-up Method in the Integral Imaging," Journal of Telecommunication and information, Vol.10, pp.33-39, 2006.  
 [5] K. C. Kwon, C. Park, Munkh-Uchral Erdenebat,

J. S. Jeong, J. H. Choi, N. Kim, J. H. Park, Y. T. Lim, and K. H. Yoo, "High speed image space parallel processing for computer-generated integral imaging system," Optics Express, Vol.20, No.2, pp.732-740, 2012.  
 [6] Y. H. Jang, C. Park, J. S. Jung, J. H. Park, N. Kim, J. S. Ha, and K. H. Yoo, "Integral Imaging Pickup Method of Bio-Medical Data using GPU and Octree," International Journal of Contents, Vol.10, No.6, pp.1-9, 2010.  
 [7] Y. H. Jang, C. Park, J. H. Park, N. Kim, and K. H. Yoo, "Parallel Processing for Integral Imaging Pickup Using Multiple Threads," International Journal of Contents, Vol.5, No.4, pp.30-34, 2009.  
 [8] <http://www.opengl.org>.  
 [9] Aaftab Munshi, *The OpenCL Specification, Version: 1.1*, Khronos OpenCL Working Group, 2010.  
 [10] NVIDIA, *OpenCL programming guide for the CUDA architecture*, Ver.2.3, 2009.  
 [11] NVIDIA, *CUDA C programming guide*, Ver. 3.3.3, 2010.

### 저 자 소 개

김 도 형(Do-Hyeong Kim)

준회원



- 2012년 2월 : 충북대학교 컴퓨터 교육과(이학사)
- 2012년 3월 ~ 현재 : 충북대학교 디지털정보융합학과 석사과정

<관심분야> : 컴퓨터 그래픽스, 이러닝, 유러닝, 멀티미디어, LCMS, LMS



박 찬(Chan Park)

정회원



- 2003년 2월 : 충북대학교 컴퓨터 교육과(이학사)
- 2007년 2월 : 충북대학교 컴퓨터 교육과(교육학석사)
- 2012년 3월 : 충북대학교 정보산업공학과(공학박사)

<관심분야> : LMS, LCMS, 이러닝, 유러닝, 멀티미디어, 컴퓨터 그래픽스

정 지 성(Ji-Sung Jung)

준회원



- 2009년 2월 : 충북대학교 컴퓨터 교육과(이학사)
- 2011년 2월 : 충북대학교 정보산업공학과(공학석사)
- 2011년 3월 ~ 현재 : 충북대학교 정보산업공학과 박사과정

<관심분야> : LMS, LCMS, 이러닝, 유러닝, 멀티미디어, 컴퓨터 그래픽스

권 기 철(Ki-Chul Kwon)

정회원



- 1996년 2월 : 상주대학교 전기전자공학과(공학사)
- 2000년 2월 : 충남대학교 전자공학과(공학석사)
- 2005년 2월 : 충북대학교 정보통신학과(공학박사)

- 2002년 ~ 2008년 : 프리즘테크 부설연구소 연구원
- 2008년 ~ 현재 : 충북대학교 연구교수

<관심분야> : 디지털 영상처리 및 의료영상처리, 입체 영상 처리 시스템

김 남(Nam Kim)

종신회원



- 1981년 2월 : 연세대학교 전자공학과(공학사)
- 1983년 2월 : 연세대학교 전자공학과(공학석사)
- 1988년 8월 : 연세대학교 전자공학과(공학박사)

• 1992년 8월 ~ 1993년 8월 : 미국 Stanford 대학교 방문교수

• 2003년 3월 ~ 2001년 2월 : 미국 California Technology Institute 방문교수

• 1989년 ~ 현재 : 충북대학교 전기전자컴퓨터공학부 교수

<관심분야> : 광정보처리, 광통신, 이동 통신 및 전파, 마이크로파 전송 선로 해석, EMI/EMC 및 전자파 인체보호 규격

류 관 희(Kwan-Hee Yoo)

종신회원



• 1985년 8월 : 전북대학교 전산통계학과(이학사)

• 1988년 2월 : 한국과학기술원 전산학과(공학석사)

• 1995년 8월 : 한국과학기술원 전산학과(공학박사)

• 1988년 1월 ~ 1997년 8월 : 데이콤 선임연구원

• 2003년 7월 ~ 2005년 2월 : 카네기멜론대학교 로보틱스연구소 교환교수

• 1997년 8월 ~ 현재 : 충북대학교 디지털정보융합학과 교수

<관심분야> : 컴퓨터그래픽스, 인공지능, 3D게임, 메디컬그래픽스