

CUDA를 이용한 집적영상에 대한 3D 시차영상 재생 3D Viewing Images Reconstruction for an Integral Image using CUDA

류관희, 김도형, 정지성, 박찬, 권기철, 김남, 백낙훈*
충북대학교, 경북대학교*

Kwan-Hee Yoo, Do-Hyeong Kim, Ji-Seong Jeong,
Chan Park, Ki-Chul Kwon, Nam Kim,
Nakhoon Baek*

Chungbuk National University,
Kyungpook National University*

요약

본 논문에서는 무안경식 3D 디스플레이 영상으로 사용되는 집적영상(Integral Imaging)이 주어졌을 때, CUDA를 이용하여 사용자 시점에 따라 다르게 나타나는 3D 영상을 효율적으로 재생하기 위해 기법을 제안한다. 제시된 기법을 사용하여 생성된 3D 시차 영상(Viewing Images)은 기존 CPU 기반 3D 영상 재생 기법보다 매우 빠르게 생성되었다.

I. 서론

일반적으로 3D 산업 활성화의 핵심 기술이 3D 디스플레이의 산업화이다. 3D 입체 디스플레이를 위해 안경을 끼고 3D 영상을 볼 수 있는 안경식 3D 디스플레이와 안경을 끼지 않고 3D 영상을 볼 수 있는 무안경식 3D 디스플레이가 출현하고 있다. 무안경식 3D 디스플레이 기법은 Lipmann[1]에 의해 제안되었으며, 그 원리는 다음과 같다. 먼저 대상물인 3D 객체를 여러 개의 기초렌즈(elemental lens)로 구성된 렌즈어레이(lens array)를 통해 보았을 때 만들어지는 영상인 집적영상(integral imaging)을 구한 후, 그 집적 영상에 동일한 렌즈어레이를 동일한 형태로 올려놓고 집적 영상을 관찰하면 원하는 3D 영상을 사용자가 안경을 착용하지 않고 볼 수 있다. 무안경식 3D 디스플레이에 대한 연구는 렌즈어레이가 만들어져 있다고 가정하고 그 렌즈어레이에 적합한 질 높은 집적영상을 효율적으로 어떻게 생성할 것인지에 대해 진행되고 있다. Igarashi 등 [2]은 컴퓨터를 통해 집적영상을 생성하는 기법을 제시하였으며, Park 등[4]은 CUDA를 활용하여 집적영상을 실시간으로 생성하는 기법을 제시하였다.

그러나 렌즈어레이가 널리 보급되고 있지 않은 상황에서 생성된 집적영상에 대해 시점에 따라 다르게 3D 영상을 볼 수 있는 재생에 관한 연구도 활발히 진행되고 있다[3]. 집적영상으로부터 3D 시차 영상(3D viewing images)의 재생에 관한 연구도 집적영상의 생성에 관한 연구와 유사하게 질 높은 3D 시차영상을 효율적으로 생성하는 측면에서 이루어지고 있다. Wang 등[3]은 인접한 집적영상의 픽셀 값들의 보간을 통해 질 높은 3D 시차영상을 생성하는 기법을 제시하였다.

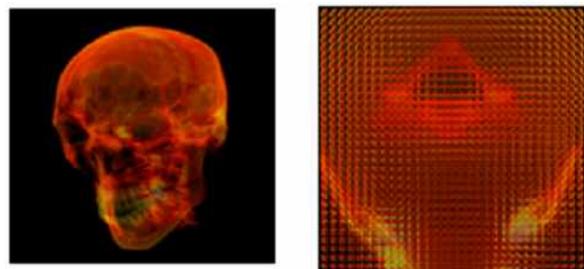
본 논문에서는 Wang 등과는 다른 측면, 즉 주어진 집적영상으로 3D 시차영상을 효율적으로 생성하는 측면에서 연구를 진행하였다. Park 등[4]이 집적영상을 효율적으로 생성하기 위해 CUDA를 적용한 것과 같이 본 논문에서도

CUDA를 이용하여 주어진 집적영상으로 매우 효율적으로 3D 시차영상을 생성하는 기법을 제시하고자 한다.

II. CUDA를 이용한 집적영상으로부터 3D 시차영상 생성 기법

1. 집적영상

무안경식 3D 디스플레이에 사용될 집적영상은 렌즈어레이의 형태에 적합하게 생성된다. <그림 1>은 CT를 통해 입력한 사람 머리를 나타내는 128 x 256 x 256 크기의 볼륨데이터에 대해 30x30개의 렌즈로 구성된 렌즈어레이를 통해 얻어진 집적영상이다.



▶▶ 그림 1. 128x256x256 볼륨데이터에 대해 30x30 렌즈 어레이로부터 얻어진 집적영상

2. CUDA를 이용한 3D 시차영상 생성 기법

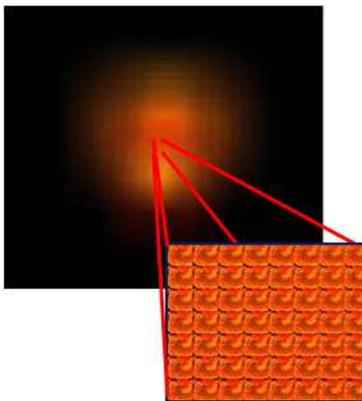
$N \times N$ 렌즈로 구성된 집적영상의 크기가 $S \times S$ (여기서 $S > N$ 이다)라고 가정하면 한 렌즈에 해당하는 영상의 크기는 $(S/N \times S/N)$ 이다. 여기서 S 는 N 보다 크다. 특정한 시점 (V_x, V_y) 이 주어졌다고 할 때 각 렌즈에 대응하는 $(S/N \times S/N)$ 크기의 영상으로부터 그 시점에 해당하는

픽셀을 추출하여 이들 픽셀을 차례대로 모으면 그 영상이 시점 (V_x, V_y)에 대한 3D 시차영상이다. 따라서 집적영상으로부터 시점에 따른 3D 시차영상은 집적영상의 픽셀 재배치를 통해 얻어질 수 있다.

본 논문에서는 CUDA [5]를 이용하여 집적영상으로부터 3D 시차영상을 효율적으로 생성하기 위해 집적영상, 렌즈의 개수, 영상의 크기를 CUDA 커널(Kernel)에 제공한다. CUDA 커널에서는 블록 크기와 스레드 크기에 따른 병렬 처리를 수행한다. 병렬 처리를 위한 하나의 명령은 집적영상의 특정위치의 픽셀을 3D 시차영상에 특정 위치로 옮기는 작업을 수행한다. 이러한 픽셀 재배치 명령이 집적영상을 구성하는 모든 픽셀에 대해 병렬 처리되므로 집적영상으로부터 3D 시차영상 생성이 매우 효율적으로 이루어진다.

III. 실험 결과

본 논문에서 제시된 기법은 QUADRO 4000인 그래픽 카드와 RAM 4GB인 PC환경에서 MFC VC++, OpenGL, OpenCL를 통해 구현되었다. <그림 2>의 집적영상은 <그림 1>의 왼쪽에 있는 사람 뇌 CT영상을 500x500 렌즈로 구성된 렌즈어레이로부터 만든 15,000x15,000크기의 집적영상이다. 따라서 한 렌즈에 대응하는 영상의 크기는 30x30이다. 또한 아래 빨간색 그림은 특정 부위의 영상을 나타낸다.



▶▶그림 2. 128x256x256 볼륨데이터에 대해 150x150 렌즈 어레이로부터 얻어진 15000x15000크기의 집적영상

<그림 3>은 <그림 2>의 집적영상으로부터 생성한 3D 시차영상들을 보여주고 있다. 시점이 영상이 나타나는 영역에서 움직인다고 가정할 때, <그림 3> (a), (b), (c)는 각각 시점이 (10,10), (10, 50), (10, 100) 위치에 있을 때의 3D 시차영상을 보여주고 있다.



(a)(10,10) 위치 (b) (10,50) 위치 (c) (10,100)위치

▶▶ 그림 3. 집적영상으로부터 생성한 3D 시차영상

본 논문에서 제시한 CUDA 기반 병렬 처리 기법을 이용하여 집적영상으로부터 3D 시차영상을 생성한 시간을 비교하기 위해 CPU 기반 처리 기법과 비교하였다. <표 1>이 그 결과를 보여주고 있고, 표에서 보는 바와 같이 본 논문에서 제안한 기법이 CPU 기반 기법에 비해 매우 효율적임을 알 수 있다.

표 1. CUDA 기반 3D 시차영상 생성 시간

집적영상크기	렌즈개수	CPU기법	제안기법
2000x2000	250x250	119ms /8,166ms	2ms /24ms
2000x2000	500x500	631ms /7,782ms	7ms /19ms
2500x2500	250x250	119ms /12,441ms	2ms /39ms
2500x2500	500x500	632ms /11,756ms	7ms /26ms

V. 결론

본 연구에서는 CUDA 기반하여 주어진 집적영상으로 3D 시차영상을 생성하기 위한 병렬 처리 기법을 제시하였다. 시험 결과 기존의 기법에 비해 매우 효율적임을 알았으며, 이 기법을 확장하면 질 높은 3D 시차영상을 효율적으로 생성할 수 있을 것이다.

■ 감사의 글 ■

본 연구는 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2010-0021853)과 교육과학기술부의 지역거점연구단육성사업/충북BIT연구중심대학육성사업단 지원을 받아 수행된 연구결과임.

■ 참고 문헌 ■

- [1] G. Lippmann, "La photographie integrale," *C.R Academic Science*, Vol.146, pp.446-451, 1908.
- [2] Y. Igarashi, etc, "3D Display System using a Computer Generated Integral Photography," *Jpn. J. Appl. Phys*, Vol.17, pp.1683-1684, 1978
- [3] Yu Wang, Three-Dimensional Computational Integral Imaging Reconstruction by Use of Pixel Blocks Interpolation, *ISECS International Colloquium on Computing, Communication, Control, and Management*, 2008.
- [4] C. Park, etc, "Acceleration Method for Integral Imaging Generation of Volume Data based on CUDA," *Journal of Korea Contents*, Vol. 11, No. 3, pp.9-17, 2011.
- [5] nVidia, *OpenCL programming guide for the CUDA architecture*, Version 2.3, 2009.