

64비트 환경에서 OpenCL을 이용한 볼륨 데이터의 기초 영상 생성

Generating Elemental Images for Volume Data using OpenCL over 64-bit environment

김도형, 박찬, 정지성, 김종오, 권순옥,
주성연, 박진아, 권신애, 권기철, 김남,
류관희
충북대학교

Do-Hyeong Kim, Chan Park, Ji-Seong Jeong,
Jong-Oh Kim, Sun-Ok Kwon, Seong-Yeon Ju,
Jin-A Park, Sin-Ae Kwon, Ki-Chul Kwon,
Nam Kim, Kwan-Hee Yoo
Chungbuk National University

요약

최근, OpenCL을 이용한 병렬 처리 기법을 통해 3D 볼륨 데이터의 기초영상을 생성하는 고속화 기법에 대한 연구가 활발히 진행되고 있다. 그러나 기존 연구를 통해 보면, 3D 볼륨 데이터의 용량이 클수록, 렌즈 어레이의 수가 많을수록 기초 영상을 생성하는데 많은 시간이 소요 되었고, 실시간으로 고용량의 볼륨데이터 집적영상을 생성하는데 문제가 있었다. 본 논문에서는 64비트 운영체제 환경에서 볼륨 데이터 기초영상을 생성하고 32비트 운영체제 환경과 비교·분석하였다.

I. 서론

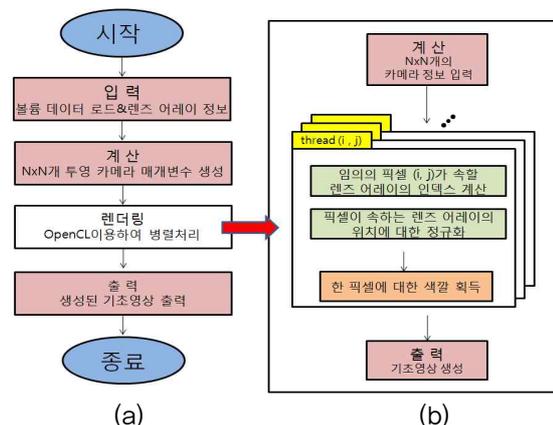
집적 영상(Integral imaging 혹은 IP, integral photography)는 1908년 Lippmann[1]에 의해서 처음으로 제안 되었고, 최근 디지털 기술과 더불어 초고해상도 영상 처리 기술의 발전에 힘입어 다시 주목되고 있는 무안경식 3D 디스플레이 기술 중 하나이다. 무안경식 3D 가시화를 위해서는 먼저 대상물인 3D 객체를 여러 개의 기초렌즈(elemental lens)로 구성된 렌즈 어레이(lens array)를 통해 보았을 때 만들어지는 기초영상(elemental imaging)을 구해야 한다. 그 후 생성한 기초영상에 동일한 렌즈 어레이를 동일한 형태로 올려놓으면 생성되는 집적영상의 관찰을 통해 안경을 쓰지 않고도 원하는 3D 영상을 볼 수 있다.

본 논문에서는 대량의 3D 데이터를 제어하기 위해 64비트 환경에서 기초 영상을 생성하는 프로그램을 개발한 결과를 소개하고 기존의 32비트 운영체제 환경과 비교하여 실험한 결과를 제시한다.

II. OpenCL을 이용한 기초영상 생성 기법

OpenCL은 표준화된, 다양한 플랫폼에 적용 가능한, C 언어에 기반한 병렬 컴퓨팅 API이다. OpenCL은 이기종(Heterogeneous) 컴퓨팅 디바이스로 구성된 시스템에 대하여 이식이 가능한 병렬 응용프로그램을 개발 할 수 있도록 설계되었다. 일반적으로 OpenCL 프로그램은 하나 이상의 OpenCL 디바이스에서 실행되는 커널(GPU)과 커널을 관리하는 호스트(CPU) 프로그램의 두 부분으로 구성되

어 있으며, 직접적인 병렬 프로세싱이 진행되는 커널 함수는 호스트 프로그램에서 개시하여 수행하게 된다. 최근 이러한 병렬처리 기법을 적용하여 집적영상을 생성하는 연구 결과가 발표되었다[2,3]. 그들은 디스플레이 패널 크기의 픽셀에 대하여 각 픽셀이 렌즈어레이에 속할 위치와 색을 결정하는 과정을 병렬로 처리하기 위해 OpenCL을 이용하였다. [그림 1]이 OpenCL을 이용하여 NxN개의 렌즈 어레이에 대해 기초영상을 생성하는 과정을 표현한 구조도이다.



▶▶ 그림 1. OpenCL을 활용한 기초영상 생성 과정

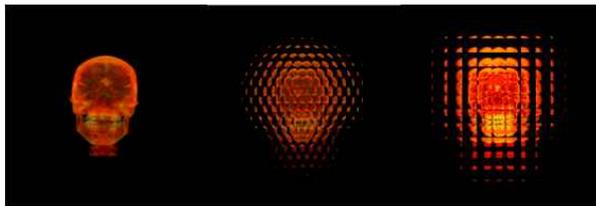
프로그램 실행 시 볼륨 데이터를 로드하고, 렌즈 수(N), 디스플레이 패널 크기 등을 입력 받는 입력과정, 입력받

은 값을 이용해 기초영상 사이즈, 뷰 행렬, 가상 카메라 정보를 계산하여 OpenCL 커널로 전달하는 계산과정, OpenCL 커널에서 병렬처리로 각 픽셀의 색을 결정하는 렌더링 과정, 그리고 색이 결정된 픽셀이 모여 기초영상을 디스플레이하는 출력 과정으로 이루어져있다. [그림 1](a)는 기초영상을 생성하는 전반적인 순서를 나타낸 구조도이며, [그림 1](b)는 4가지 과정 중 OpenCL을 활용하는 렌더링 과정에 대해 자세히 표현한 구조도이다. 출력영상의 크기에 대한 픽셀의 수만큼 스프레드를 생성해 병렬로 모든 픽셀의 색을 결정하는 과정을 보여준다.

III. 실험결과 및 결론

본 논문의 실험에 사용된 PC의 하드웨어는 Intel(R) Core(TM)i5-2500 CPU(a) 3.30GHz와 4GB RAM, NVIDIA Quadro 4000를 사용하였고, 각각 32비트, 64비트의 Windows 7 운영체제 환경에서 개발도구로 MS Visual Studio 2008을 사용하였고, 3차원 그래픽스 라이브러리로 OpenGL[4]을, 병렬처리 라이브러리로 OpenCL[5]을 사용하였다.

[그림 2]는 크기가 128(W)×256(H)×256(D)인 3D 볼륨 데이터에 대한 기본영상과 사각형, 육각형 가상 렌즈어레이를 통해 생성한 기초영상 결과이다.



▶▶ 그림 2. 기본영상, 육각형, 사각형 기초영상 생성결과

아래 표는 렌즈 어레이의 수를 고정 50×50[개]으로 고정하고, 렌즈 한 개당 포함하는 픽셀의 수를 변화시키며 32비트와 64비트 환경에서의 기초영상 생성 시간(단위 : 밀리초)을 비교한 표이다.

표 1. 기초영상의 해상도에 따른 실행시간 비교

기초영상의 한 렌즈에 대한 영상 크기[P×P]	사각형 64비트 (ms)	사각형 32비트 (ms)	육각형 64비트 (ms)	육각형 32비트 (ms)
10×10	45	40	96	93
20×20	116	103	356	358
30×30	242	237	787	746
40×40	423	390	1250	1219
50×50	625	625	1923	1923

32비트와 64비트 환경에서 3D 볼륨데이터의 기초영상 생성 성능을 비교한 결과 32비트와 64비트 환경에서 성능의 차이를 확인할 수 없었다. 그 이유는 OpenCL의

Heterogeneous 특성에 의해 프로세싱 성능은 디바이스에 영향을 받기 때문에 32비트 환경과 64비트 환경의 차이를 확인할 수 없는 것이다. 64비트 환경은 속도의 향상보다는 어드레스 공간의 확장과 대규모 운용이 가능한 시스템으로 봐야 하며, 64비트 환경은 볼륨데이터를 64비트 환경에 적합한 데이터 형으로 표현할 경우 현재 보다 더 큰 용량의 볼륨데이터의 처리가 가능 할 것으로 예측된다.

IV. 향후 과제

향후 연구로는 현재 32비트 환경에서 개발된 기초영상 생성 프로그램을 64비트 환경에 적합하도록 볼륨데이터의 데이터 형을 설정하여 동일한 디바이스 환경에서 32비트와 비교해 64비트에서 더 큰 용량의 볼륨 데이터를 활용한 기초영상 생성 프로그램을 개발하는 연구가 필요하다.

■ 감사의 글 ■

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2011-0025849)와 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No. 2012-0000479)을 받아 수행된 연구임.

■ 참고 문헌 ■

- [1] G. Lippmann, "La photographie integrale," C.R Academic Science. Vol.146, pp.446-451, 1908.
- [2] Ki-Chul Kwon, Chan-Park, Munkh-Uchral Erdenebat, Ju-Seong Jeong, Jeong-Hun Choi, Nam Kim, Jae-Hyeung Park, Young-Tae Lim, and Kwan-Hee Yoo, "High speed image space parallel processing for computer-generated integral imaging system", OPTICS EXPRESS, Vol.20, No.2, p 732-740, 2012
- [3] 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," Journal of Korea Contents Association, 10(9),1-9, 2010
- [4] <http://www.opengl.org>.
- [5] Aaftab Munshi, "The OpenCL Specification, Version: 1.1" Khronos OpenCL Working Group, 2010