



정보산업동향

실시간 광선 추적 GPU 기술 동향



박우찬 (세종대학교)

-
- 목 차 »
1. 서론
 2. 래스터 방식 vs. 광선 추적 방식
 3. 실시간 처리 방안
 4. 결론
-

1. 서론

현실감 있는 3차원 그래픽 영상을 지원하기 위해서는 방대한 양의 데이터와 대규모 연산을 필요로 하기 때문에 고성능의 전용 GPU(graphics processing unit)가 필요하다. GPU 기술은 PC 분야는 물론이고 고성능 스마트폰, 태블릿, DTV 등 첨단 분야의 핵심 요소기술로 자리를 잡았다. 하지만, 국내에서는 Nvidia 사, Imagination Technology 사, ARM 사 등의 해외 기술에 전적으로 의지하고 있다.

현재 거의 모든 GPU는 스캔 컨버전(scan conversion)을 기반으로 하는 래스터(rasterization) 방식을 채택하고 있다. 이와 같은 래스터 방식으로 생성된 영상은 지역 조명(local illumination)을 바탕으로 두고 있기 때문에 현실감 있는 영상을 생성하는데 한계가 분명하다. 이와 반대로, 광선 추적(ray tracing) 방식은 각 픽셀에 대해서 광선을 생성하여 빛의 경로를 추적하고 3차원 가상 객체에 대한 빛의 효과를 계산하는 방식이

기 때문에 전역 조명(global illumination) 효과가 가능하다. 또한 광선 추적 방식은 영상의 정확도, 콘텐츠 제작의 용이성 등의 다양한 장점을 가지고 있다^[1,2]. 하지만, 광선 추적은 엄청난 양의 계산 및 메모리 대역폭을 요구하기 때문에 과거에는 실시간 처리가 불가능하다고 여겨졌다.

반도체 기술의 발전으로 인하여 광선 추적을 실시간으로 처리하고자 하는 연구들이 생겨나고 있으며, 향후에는 광선 추적 방식이 래스터 방식을 대체할 것으로 예상하고 있다^[1]. 최근에는 국내의 적으로 향후 2~3년 이내에 실시간 광선 추적 방식이 상용화 될 것이라는 전망이 국내외적으로 기사화 되고 있다.

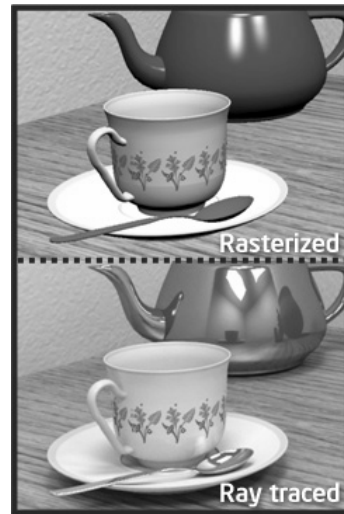
근래 들어 Intel 사의 MIC(many integrated core)와 Nvidia 사의 GPU 등 매니 코어의 등장으로 인하여 실시간 광선 추적에 대한 다수의 연구가 발표되고 있다^[3,4]. 또한, 국내외 벤처 회사에서 전용 하드웨어 방식을 사용화 되고 있으며, 이에 대한 기사가 전 세계적으로 주목을 받을 정도

로 본 분야의 기술적 사업적 영향력이 매우 크다. 본 글에서는 광선 추적 방식에 대한 소개를 하며, 최근 동향을 살펴보고 향후 전망에 대하여 논하고자 한다.

2. 래스터 방식 vs. 광선 추적 방식

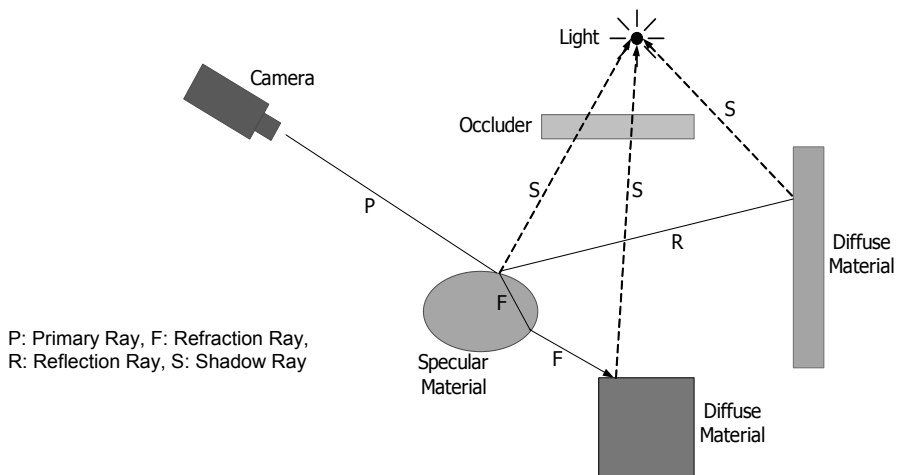
(그림 1)에서는 래스터 방식과 광선 추적에 대한 영상의 차이를 잘 보여주고 있다 [5]. 현재의 표준 3D API인 OpenGL 및 Direct3D를 이용하는 렌더링은 래스터 방식으로써 이는 지역 조명에 따른 방식이다. 이는 광원에서 직접적으로 발생한 light만이 영상에 영향을 준다는 의미이다. 광선 추적 방식은 전역 조명 효과를 얻을 수 있는 렌더링 방식이다. 이는 다른 물체에서 반사되거나 굴절된 light도 현재 물체의 영상에 영향을 준다는 것을 의미한다. 이로 인하여 반사, 굴절, 그림자 효과가 자연스럽게 제공되기 때문에 현실감 있는 3D 영상을 생성할 수 있으며, 현재 고화질 3D 애니메이션 및 특수 효과 등에 사용된다.

(그림 2)는 광선 추적이 수행되는 예이다. 먼저



(그림 1) 래스터 vs. 광선 추적

각 픽셀 당 카메라의 위치로부터 일차(primary) 광선을 생성하며 이 광선과 만나는 물체를 찾기 위한 계산을 수행한다. 광선과 만나게 된 물체가 반사(reflection)나 굴절(refraction)의 성질이 있으면 광선과 물체가 만난 위치에서 반사 효과를 위한 반사 광선 혹은 굴절 효과를 위한 굴절 광선을 생성하고 또한 그림자(shadow) 효과를 위하여 조



(그림 2) 광선 추적의 처리 과정의 예

명 방향으로 그림자 광선을 생성한다. 반사 광선과 굴절 광선은 이차(secondary) 광선이라고 불리며 이들은 각각 자신과 만나는 물체를 찾기 위한 계산을 수행한다. 이러한 과정은 재귀적으로 수행된다.

광선 추적은 크게 광선을 발생하는 광선생성부, 발생한 광선과 교차하는 물체를 탐색하는 탐색부, 광선과 교차된 부분에 대한 색깔 값 계산 및 텍스처 매핑(texture mapping)을 수행하는 셰이딩(shading) 부로 구성된다. 여기서, 절대적으로 많은 양의 연산과 메모리 대역폭을 요구하는 부분은 탐색부이다. 탐색과정을 가속하기 위한 일반적인 방법으로 그래픽 primitive(예, 삼각형)들을 트리 형태로 구성하며, 이를 가속 구조체라고 한다. 광선 추적을 가속하기 위한 연구들의 상당 부분은 가속 구조체에 대한 것이다.

3. 실시간 처리 방안

광선 추적은 각 픽셀에서 생성된 광선들이 독립적으로 처리될 수 있기 때문에 알고리즘적으로 병렬처리가 용이하다. 이러한 이유로 광선 추적은 병렬 컴퓨터에서 소프트웨어 적으로 처리가 용이하다. 하지만, 이러한 방식은 실시간 처리에는 한계가 있다. 최근에는 전용 하드웨어 방식이 시도되고 있다. 이번 장에서는 광선 추적을 가속하는 방식들을 살펴본다.

3.1 소프트웨어 방식

최근 CPU 및 GPU의 성능이 비약적으로 발전하고 코어의 수도 늘어나고 있는 추세이다. 이로 인하여 CPU 혹은 GPU에서 광선 추적을 소프트웨어로 가속하는 연구들이 시도되고 있다. 하지만, 실시간 광선 추적을 소프트웨어로 처리하는

것은 성능 면에서 매우 어렵다.

CPU 방식으로는 Intel 사에서 발표한 cloud-based ray tracing이 있다^[3]. 이는 parallel visual processing을 위한 Intel MIC 프로세서에 광선 추적 방식을 구현한 사례이다. 이 프로세서는 32개의 코어로 구성되어 있으며, 한 코어당 4개의 thread를 구동할 수 있다. 하지만, 여전히 고화질의 광선 추적 방식을 실시간으로 처리하기에는 성능면에서 부족하다.

GPU 방식으로는 Nvidia 사에서 발표한 OptiX라는 플랫폼이 있다^[4]. OptiX에는 광선 추적을 가속할 수 있는 API가 제공되며, 이를 사용하여 광선 추적의 성능을 증가시키는 연구가 다수 발표되었다. 하지만, 최신 GPU에는 1천개가 넘는 코어를 제공함에도 불구하고, GPU는 구조적으로 고화질의 광선 추적에 대해서는 코어의 효율이 급격하게 떨어질 수 있다. 따라서, 이 경우 여전히 실시간 광선 추적의 지원은 성능면에서 부족하다.

광선 추적 가속을 위한 구조로써 [5]에서는 22nm 반도체 공정 하에서 4GHz의 128개의 코어와 32Mbyte L2 캐시를 내장하고 최대 4Tera flops 성능의 구조인 Copernicus를 제시 하였다. 시뮬레이션 상에서 최대 초당 74M 광선 처리 성능을 예상하고 있다. 이는 실시간 처리하기에는 매우 부족한 성능이다.

3.2 하드웨어 방식

실시간 광선 추적을 지원하기 위한 방안은 전용 하드웨어를 제작하는 것이 가장 유력하다. 하지만, 광선 추적에 대한 전용 하드웨어에 대한 연구는 기술적인 어려움으로 인하여 전 세계적으로 초보적인 수준이며, 이에 대한 연구 사례는 매우 드물다.

근래 들어 독일의 Saarland 대학에서 전용 하드웨어 방식의 광선 추적 가속기를 발표하였다^[6]. 하지만 이는 기능면에서 광선 추적이 아닌 광선 투사(ray casting) 위주이다 보니 광선 추적의 다양한 기능을 고성능으로 지원하기에는 여러 가지 면에서 부족하다.

최근 들어 미국 회사인 코스틱그래픽스(www.caustic.com) 사는 광선 추적을 가속하는 플랫폼을 발표하였다. 이는 PCI express card 형태의 하드웨어와 CPU 혹은 GPU에서 수행되는 소프트웨어가 결합된 솔루션이다. 이는 애니메이션이나 특수 효과에 적합한 구조이다. 전체 광선 추적 파이프라인의 일정 부분만이 하드웨어로 구현되어 있기 때문에, 실시간 광선 추적에는 부족한 성능을 보여주고 있다. 이 회사는 2010년 말에 모바일 GPU의 선두 기업인 Imagination Technology 사에 인수되었으며, 몇 년 이내에 모바일 광선 추적 GPU를 출시할 것으로 알려지고 있다.

가장 주목할 사례로 국내의 벤처기업인 실리콘아츠(www.siliconarts.com) 사는 실시간 광선 추적을 지원하는 반도체 IP인 RayCore® IP를 작년에 출시하였다. 이는 전 세계 최초의 광선 추적 IP이며, 광선 추적 방식의 주요 기능이 모두 지원된다. 이는 코스틱그래픽스와 달리 광선 추적 파이프라인 전체가 하드웨어로 구현되었고 가속 구조체 탐색을 효과적으로 수행하는 하드웨어 구조를 구현함으로써, 스마트폰이나 DTV에 내장할 수 있는 정도의 하드웨어 면적 크기, 성능, 저 전력화를 유일하게 달성하였다. 현재 API는 OpenGL ES 1.1 수준이다. 현재 글로벌 기업들과의 계약이 진행되고 있는 것으로 알려져 있으며, 향후 1-2년 내에 본 기술이 탑재된 스마트폰용 혹은 게임 콘솔용 반도체가 상용화 될 것으로 예상된다.

4. 결론

실시간 광선 추적의 지원은 전용 하드웨어를 사용하는 방식이 가장 유력하다. 하지만, 이에 대한 연구의 사례가 매우 드물고 기술적인 어려움으로 인하여 전 세계적으로 초기 단계에 머무르고 있다. 한편, 국내의 한 벤처기업에서 실시간 광선 추적 IP가 유일하게 발표되었다.

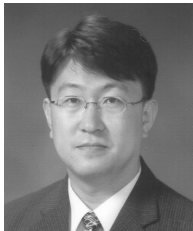
현재 시스템 반도체를 제작하는 국내의 주요 기업들은 GPU를 도입하고 있으며 이로 인한 기술 종속이 매우 심각하다. 한편, 실시간 광선 추적은 현재의 GPU를 대체할 차세대 기술로 확실히 되고 있으며, 전용 하드웨어에 대한 연구는 전 세계적으로 초기 단계이다. 게임 분야 및 시스템 반도체 분야에 대한 경쟁력을 가지고 있는 국내의 현실로 보았을 때 실시간 광선 추적 GPU는 분명히 기술적 및 산업적으로 향후 커다란 기회의 분야라고 생각된다.

참고 문헌

- [1] J. Hurley, "Ray Tracing Goes Mainstream," *Intel Technology Journal*, Vol.9, No.2, pp.99-108, May, 2005.
- [2] K. Suffern, *Ray Tracing from the Ground Up*, Wellesley, AK Peters Ltd., 2007.
- [3] D. Pohl, "Experimental Cloud-based Ray Tracing Using Intel® MIC Architecture for Highly Parallel Visual Processing," *Intel Software Network*, Feb., 2011.
- [4] S. G. Parker et al., "OptiX: a General Purpose Ray Tracing Engine," *ACM Transactions on Graphics*, Vol.29, No.4, article number 66, July, 2010.
- [5] V. Govindaraju et al., "Toward a Multicore Architecture for Real-time Ray-tracing," In *Proceedings of International Symposium on Microarchitecture*, pp.176-187, Nov., 2008.

- [6] S. Woop, J. Schmittler, and P. Slusallek, "RPU: A Programmable Ray Processing Unit for Real-time Ray Tracing," In *Proceedings of SIGGRAPH*, pp.434-444, 2005.

저 자 약 력



박 우 찬

이메일 : pwchan@sejong.ac.kr

- 1993년 연세대학교 전산학과(학사)
- 1995년 연세대학교 컴퓨터학과(석사)
- 2000년 연세대학교 컴퓨터학과(박사)
- 2001년~2003년 연세대학교 연구교수
- 2003년~2008년 세종대학교 컴퓨터공학과 조교수
- 2008년~현재 세종대학교 컴퓨터공학과 부교수
- 관심분야: GPU 구조, 실시간 Ray Tracing 가속기, 컴퓨터 구조, 실시간 렌더링, ASIC 설계