

래스터라이저-프레임버퍼 혼합 설계기술을 이용한 고성능 렌더링 시스템 설계

최 춘 자, 박 우 찬, 한 탁 돈
연세대학교 컴퓨터과학과 미디어시스템 연구실

High Performance Rendering system using a Rasterizer Merged Frame Buffer

Chun-Ja Choi, Woo-Chan Park, Tack-Don Han
Media System Lab., Dept. of Computer Science, Yonsei University

요 약

3차원 그래픽 렌더링 파이프라인(3D Graphics Rendering Pipeline)은 크게 지오메트리 프로세싱(Geometry Processing)과 래스터라이제이션(Rasterization)으로 구성되어 있다. 본 논문에서는 래스터라이저와 프레임버퍼사이의 대역폭으로 인한 병목점을 분석하고, 그 한계를 극복해 낼 수 있도록 프로세서 메모리 집적구조를 이용하여 렌더링 시스템을 설계, 성능 분석한다. 프레임버퍼의 집적으로 인한 메모리 대역폭을 이용하기 위해, 각 픽셀 처리에 필요한 로직을 포함하는 SIMD 타입의 픽셀 처리 프로세서가 메모리 어레이와 밀결합(tightly coupled)된다. 제안하는 구조는 래스터라이저 로직과 프레임버퍼가 단일 칩으로 구성되었고, 텍스처 매핑, 범프 매핑, 안티알리아싱, 깊이 버퍼를 지원하며 초당 5백만 이상의 삼각형을 처리할 수 있는 고성능 렌더링 시스템이다.

1. 서론

3차원 그래픽 렌더링 파이프라인은 크게 지오메트리 프로세싱과 래스터라이제이션으로 구성되어 있다. 지오메트리 프로세싱은 주로 3차원 좌표계의 물체를 시점에 따라 변환하고 2차원 윈도우 좌표계로 투영하는 등의 처리과정을 의미한다. 래스터라이제이션은 2차원 좌표계의 이미지에 실제적인 색깔 값을 결정하고, 화면에 나타낼 영상 데이터를 저장하는 프레임버퍼에 갱신하는 처리를 한다. 이러한 3차원 그래픽 처리를 수행하는데 있어서는 극복해야 할 다음의 문제점들이 존재한다.

첫째, 지오메트리 프로세서에서의 부동 소수점 연산에 대한 요구로, 고품질의 해상도를 지원하기 위해서는 더욱 많은 부동 소수점 연산이 필요하게 된다. 둘째, 픽셀 처리를 위한 정수 연산으로, 각 픽셀 하나 하나에 대하여 다각형 고유의 색, 빛의 세기 투명 정도를 고려하여 최종 픽셀을 계산해 내기 위해 많은 정수 연산을 요구한다. 셋째, 프레임버퍼 메모리 대역폭의 병목이다. 계산된 모든 픽셀에 대해 프레임버퍼에 저장되어 있는 깊이(depth: Z)값을 읽은 다음, 새로 들어온 픽셀의 깊이 값과 비교하여 새로 들어온 값이 시점에서 더 가까이 있으면 프레임버퍼의 색깔 값을 새로 들어온 색깔 값으로 바꾸고, 그렇지 않으면 새로 들어온 값을 버린다. 이러한 읽기-수정-쓰기(read-modify-write) 패턴은 래스터라이저와 프레임버퍼사이 병목현상을 일으켜 전체적인 성능을 저하시킨다. 넷째, 렌더링과정중 3차원 그래픽 영상을 보다 현실적으로 만들기 위한 텍스처 매핑(Texture Mapping)으로 인한 메모리 지연과 대역폭 요구이다.

이러한 문제점들을 극복하기 위해 중요한 핵심기술로는 (1) 기하학 처리와 픽셀 처리를 위한 고성능 부동소수점 연산 및 정수 연산기 구조와 데이터패스 개발, (2) 래스터라이저와 프레

임버퍼 메모리의 대역폭 향상을 위한 프로세서-메모리 집적구조(로직과 메모리를 단일 칩에 집적함으로써 메모리 대역폭 문제를 해결하기 위한 새로운 접근 방법) 방식의 메모리 구조 설계와 데이터패스 개발이 필수적이다. 본 논문에서는 (2)의 적용을 통하여 렌더러와 프레임버퍼 사이의 메모리 대역폭 문제, 텍스처 매핑으로 인한 메모리 지연·대역폭 문제를 향상시킬 수 있는 구조를 제안하고, 다양한 3차원 그래픽 기능을 포함하는 고성능 렌더링 시스템을 설계한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련연구를 살펴보고, 3절에서는 3차원 그래픽 렌더링 처리과정에 대해 설명한다. 4절에서는 목표 렌더링 시스템 구조를 제안하고 5절에서는 래스터·프레임버퍼 유닛의 집적이슈에 대하여 논한다. 그리고 마지막 5절에서는 향후 연구 계획과 결론을 맺는다.

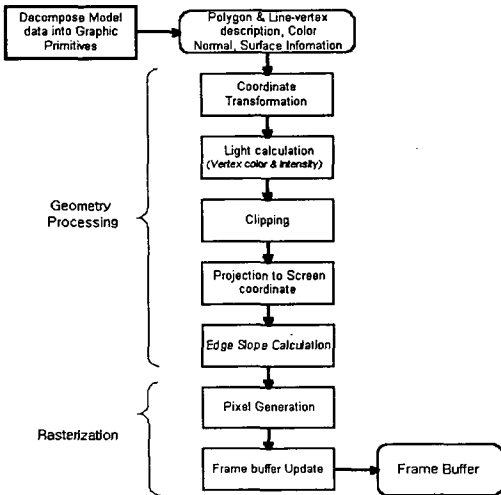
2. 관련 연구

본 절에서는 고성능 렌더링 아키텍처 및 프레임버퍼 메모리의 대역폭과 관련된 연구들을 살펴본다.

Reality Engine은 353개의 독립적인 프로세서를 가지고 초당 1 M이 넘는 안티알리아싱, 텍스처 매핑된 삼각형을 처리한다. 하지만 각 래스터 보드가 갖는 5개의 프래그먼트 생성기마다 모두 중복된 텍스처 메모리를 가지고 있다[4]. Neon은 중복된 메모리를 제거하여 통합된 프레임버퍼 메모리를 갖는다. 페이지 교차(page crossing)를 줄이고, 메모리 버스 턴어라운드(bus turnaround)를 줄이기 위한 배치(batch) 프로세싱, 페이지 프리페치(prefetch), 텍스처 캐시등을 두어 대역폭 요구를 줄임으로써 경쟁력 있는 성능을 이끌어 냈다[3]. Truga001은 단일 칩 렌더링 프로세서로 초당 천만 이상의 삼각형을 병렬로 렌더링함으로써 가상 현실 장면을 생성해 낸다[6].

3. 3차원 그래픽 렌더링 파이프라인

기본적인 그래픽 프리미티브를 렌더링하기 위한 과정이 [그림 1]에서 보여지고 있다. 각 시스템에 따라 이러한 렌더링 파이프라인을 다르게 나눌 수 있다. 그래픽 렌더링은 응용프로그램 처리로부터 폴리곤과 라인을 받아 아래에 보이는 처리 흐름에 따라 각 단계를 처리한다. 각 처리단계의 상세한 사항은 [2]에 기술되어 있다.



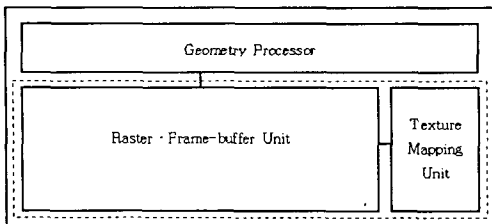
[그림 1] 3차원 그래픽 처리과정

본 논문의 렌더링 시스템은 고라우드(Gouraud) 범프 매핑, 안티알리아싱, 깊이 버퍼, 텍스처 매핑된 픽셀을 처리한다. 중요한 고려사항은 (1) 프로세서와 메모리 집적구조를 이용하여 래스터라이저와 프레임버퍼 사이에서 128MB/s 이상의 메모리 대역폭을 제공 (2) 텍스처 매핑을 위한 로직과 메모리를 래스터라이저와 분리시켜 텍스처 매핑에서의 메모리 접근과 텍스처 이미지 영역을 제거하였으며 (3) 안티알리아싱을 성공적으로 수행 (4) 파이프라인을 통하여 텍스처 매핑, 범프 매핑, 안티알리아싱, 픽셀 연산들을 병렬로 수행한다는 것이다.

4. 렌더링 시스템

4.1. 시스템 Top view

[그림 2]에서 렌더링 시스템의 전체 구성이 보여지고 있다.



[그림 2] 시스템 모델

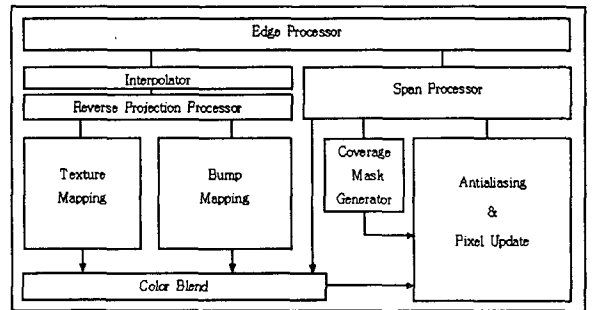
이 시스템은 지오메트리 프로세서와 래스터라이저, 프레임버퍼가 집적된 래스터·프레임버퍼 유닛과 텍스처 매핑의 방대한 텍스처 메모리 요구를 위해 단일 칩으로 구성된 텍스처 매핑 유닛으로 이루어진다. 래스터·프레임버퍼 유닛과 텍스처 매핑 유닛을 합하여 렌더링 프로세서라 부르게 한다.

4.2. 지오메트리 프로세서

본 논문에서는 다른 그래픽 가속기들과 달리 지오메트리 프로세서의 부동 소수점 연산을 가속하지는 않는다. 지오메트리 프로세서는 200MHz에서 동작하는 것을 가정하며 좌표와 법선벡터(normals)를 시각 좌표계로 변환, 범프매핑을 위한 물체 표면(object surface)의 굴곡화(perturbation), 광원 처리, 클립 좌표계로의 변환, 클리핑, 윈도우 좌표계로의 투영을 처리한다. 또한 관련된 텍스처 좌표는 삼차원 행렬에 의해 변환·윈도우 좌표계, 컬러와 연결되고 R,G,B,α,Z, 텍스처 좌표와 관련된 윈도우 좌표계 경사 정보가 계산된다. 이제 래스터라이제이션될 준비가 된 투영된 삼각형이 지오메트리 프로세서의 출력으로 나오고 버스를 통해 래스터라이저로 보내진다.

4.3. 렌더링 프로세서

래스터라이제이션과 텍스처 매핑을 포함한 전체 렌더링 프로세서 처리 흐름이 [그림 3]에 보여진다.

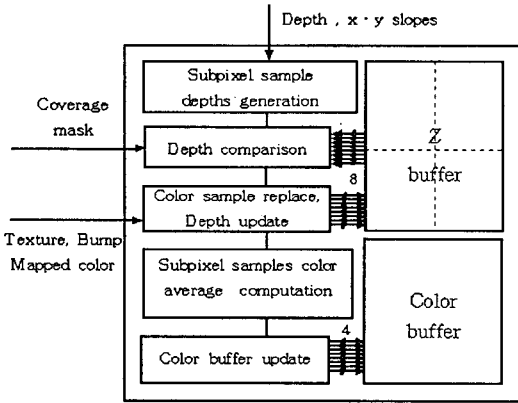


[그림 3] 렌더링 프로세서 처리 흐름도

에지 프로세서는 최상위(top) 정점에서 시작하여 해당 폴리곤 외각선의 양쪽 가지(branch)를 따라 모든 보간된 데이터를 출력한다. 이것은 수평 축에 의해 교차되어진 폴리곤 외각의 두 에지를 결정한다. 에지 프로세서 시작과 끝 x가 정해지면 보간연산기(Interpolator)에서는 해당 스캔에서의 모든 픽셀들에 대한 텍스처 매핑과 범프 매핑을 위해 u,v,w의 보간을 수행한다. 역 투영 프로세서(Inverse Projection Processor)는 각 픽셀에 대한 텍스처 맵과 범프 맵을 읽어오기 위해 공통으로 사용되는 주소를 생성한다. 그러는 동안 스캔 프로세서에서는 스캔의 각 픽셀들에 대한 수평 보간을 수행한다. 멀티 샘플 알리아싱 기법으로 안티알리아싱하기 위해 적용범위 마스크(coverage mask)를 생성한다[4]. 서브샘플 마스크는 8x8 서브 샘플의 바둑판(grid)상에서 선택된 4 샘플 위치에 대해 생성된다. 스캔 프로세서에서의 결과로 생성된 각 픽셀마다의 깊이 값과 x, y 기울기(slope)값들은 [그림 4]에서 같이 즉시 보내지고 각 서브 픽셀 위치에서의 깊이 값들을 다시 생성한다. 이는 프레임버퍼의 내용이 생성되는 즉시 다시 업데이트 되는 것을 가능하게 한다. 프래그먼트의 적용범위 마스크가 '1'인 각 샘플 위치에서 계산된 깊이 값은 프래그먼트에 저장되어 있던 깊이 값과 비교된다. 비교가 성공적이라면 서브샘플 위치에 있는 프래그먼트 컬러는 새로운 프래그먼트 컬러에 의해 대체되고 프래그먼트의 깊이값도 대체된다. 컬러 값이 업데이트 된다면 서브 픽셀 컬러들의 평균에 의한 총 픽셀 컬러 값을 다시 생성하여 컬러 프레임버퍼에 쓴다.

4.4 래스터라이저와 프레임버퍼 혼합설계

래스터라이저와 프레임버퍼를 집적하게 됨으로써 메모리 I/O 편 제약이 없어지므로 단위 시간 동안 접근 할 수 있는 메

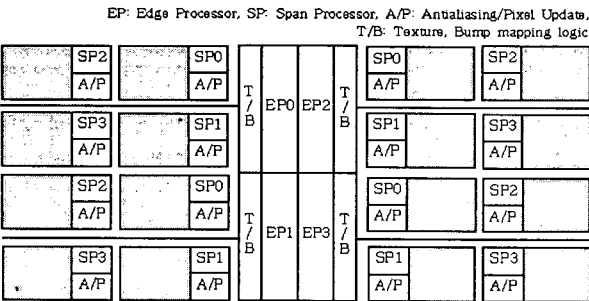


[그림 4] 안티알리아싱과 픽셀 업데이트 처리

모리 양, 즉 메모리 대역폭을 증가시킬 수 있게 된다. 그에 따른 메모리 대역폭을 효율적으로 사용하기 위해, 각 픽셀 처리에 필요한 로직을 포함하는 SIMD 타입의 픽셀 처리 프로세서는 메모리 어레이와 밀결합(tightly coupled)된다. 이때, 픽셀 처리 프로세서의 데이터 패스 폭(width)은 메모리 어레이의 폭과 동일하며 각 픽셀 처리 프로세서는 주어진 메모리 블록 주소에 의해 표시된 메모리 장소를 병렬로 접근하게 된다. 초기 시스템 모델에서 하나의 에지 프로세서는 네개의 스패 프로세서를 갖게 되며 따라서 한 클럭당 4개의 스패에서 4개, 총 16개의 픽셀을 생성해낸다[그림 5]. 프레임버퍼 메모리에는 각 픽셀에 대한 컬러(R,G,B, α -32bits)와 4개의 서브 샘플에 대한 깊이 값(32bits×4)들을 저장한다. 서브 샘플에 대한 깊이 값들은 한 사이클에 두 개씩 읽혀진다. 이러한 구성은 에지 프로세서, 픽셀 처리 연산 파이프라인 흐름과, 로드 밸런싱을 측정하여 다르게 구성될 수 있다.

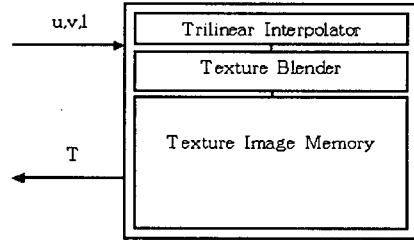
4.5 텍스처 매핑 유닛

텍스처 매핑은 방대한 메모리 요구와 자연으로 3차원 그래픽 가속기의 병목이 되고 있다. 이를 해결하기 위해 본 논문에서는 최신 프로세서-메모리 집적 공정을 이용하여 텍스처 매핑 로직과 텍스처 메모리를 집적하여 512MB/s 이상의 대역폭을 지원하는 단일 유닛으로 구성하였다[그림6].



[그림 5] 렌더링 프로세서와 프레임버퍼 구조

Neon은 작은 텍스처 캐쉬를 두어 텍스처 매핑에 요구되는 대역폭 요구를 줄였는데[3], 텍스처 메모리를 단일 칩으로 구성하면 버스 폭의 증가로 대역폭을 향상시킬 수 있으며 DRAM의 감지 증폭기(Sense Amplifier)를 캐쉬처럼 사용할 수 있게 되므로[7] 접근 시간을 상당히 향상시킬 수 있게 된다.



[그림 6] 텍스처 매핑 유닛

5. 래스터라이저와 프레임버퍼 메모리 집적 이슈

16Mb DRAM 이전까지는 1메탈 층만을 사용하였기 때문에 적어도 256Mb 세대인 0.25um 2-3 메탈 층을 가지는 공정이 약 100-200MHz CPU 로직을 구성할 수 있다. 목표 시스템 모델은 2-3 메탈 이상의 CMOS 공정을 가지는 256Mb DRAM 이상으로 한다. 래스터 로직은 가격대 성능비의 우위를 고려하여 DRAM 다이 크기의 15%~20% 정도를 차지하는 것을 목표로 한다.

6. 결론 및 향후 계획

본 논문에서는 3차원 그래픽 렌더링 처리과정의 병목점 즉 래스터라이저와 프레임버퍼 사이의 대역폭 문제를 향상시키기 위해 래스터라이제이션 로직과 프레임버퍼를 단일 칩에 집적하였고 텍스처 매핑 유닛을 분리하여 텍스처 매핑으로 인한 메모리 지연·대역폭 문제를 해결하였다. 또한 범프 매핑, 텍스처 매핑, 깊이 버퍼, 안티알리아싱을 지원하며, 초당 5백만 이상의 폴리곤을 처리 할 수 있는 렌더링 시스템을 설계하였다. 현재는 이를 검증하기 위한 시뮬레이터를 제작중에 있으며 향후에는 증가된 대역폭을 통해 픽셀 생성의 병렬성을 극대화시킬 수 있는 메모리 구조에 대한 분석도 병행하여 이루어질 것이다. 또한 메모리 집적 기술의 발달은 더 많은 서브샘플링을 가능케 하여 보다 나은 안티알리아싱의 구현이 가능할 것이다.

참고 문헌

- [1] I. Ernst, H. Russeler, H.Schulz, O. Wittig , "Gouraud Bump Mapping," Proceeding of the 1998 EUROGRAPHICS/SIGGRAPH Workshop on Graphics Hardware, Lisbon, Portugal, pp. 47-53.
- [2] J. Foley, A. van Dam, S. Feiner, J. Hughes," Computer Graphics Principles and Practice," 2nd edition, Addison-Wesley Publishing, 1990.
- [3] J. McCormack, R. McNamara, C. Gianos, N. P. Jouppi, T. Dutton, J. Zurawski, "Implementing Neon: A 256-bit Graphics Accelerator," IEEE Micro, March/April 1999, pp. 58-69.
- [4] K. Akeley, "RealityEngine Graphics," In Proceeding of SIGGRAPH '93, pp. 109-116.
- [5] P. S. Heckbert, H. P. Moreton, "Interpolation for Polygon Texture Mapping and Shading," State of the Art in Computer Graphics: Visualization and Modeling, Springer-Verlag, 1991, pp. 101-111.
- [6] T.Ikeda, J. Ma, "The Truga001: A Scalable Rendering Processor," IEEE Computer Graphics and Applications, March/April 1998, pp. 59-79.
- [7] Y. S. Kim, T. D. Han, S. D. Kim, "A Memory System for Merged Memory with Logic LSIs," Asia Pacific Conference on ASICs 1999, Seoul, Korea, pp. 384-387.