

3차원 그래픽 가속 하드웨어 설계를 위한 그래픽 파이프라인 시뮬레이터 구현

이원중^U 박우찬 한탁돈
연세대학교 컴퓨터과학과
(wjlee, wcpark, hantack)@kurene.yonsei.ac.kr

The Implementation of Graphic Pipeline Simulator for 3D Graphic Accelerator Hardware Design

Won-Jong Lee^U Woo-Chan Park Tack-Don Han
Dept. of Computer Science, Yonsei University

요 약

고성능의 3차원 그래픽 가속기 설계를 위해서는 어플리케이션, 하드웨어 구조, 수행모델 채택, 설계비용 등의 다양한 고려사항이 요구되고 따라서 각 모델에 따른 사전 시뮬레이션 환경구축은 반드시 필요하다. 이에 본 논문에서는 기본적인 3차원 그래픽 파이프라인 작업을 수행하여 다양한 결과를 보여주는 이식성 높은 시뮬레이션 환경을 제공함으로써 3차원 그래픽 가속하드웨어 세부모델 설계에 필요한 설계 고려사항을 효과적으로 제시할 수 있게 하였다.

1. 서 론

최근까지 3차원 그래픽스 분야에서 고화질의 영상을 보다 빠르게 생성하기 위한 다양한 연구가 진행되어왔다. 반도체기술의 발달로 처리속도, 집적도의 향상을 통해 방대한 부동소수점 정수연산 그리고 실시간 프레임처리를 위한 잦은 메모리 접근을 허용하는 3차원 그래픽 가속 하드웨어 설계가 가능하게 되었다. 하지만 범용 프로세서와 달리 이러한 그래픽 가속기 설계는 사용 어플리케이션, 수행모델 채택, 설계비용 등의 다양한 고려사항이 존재한다. 또한 중앙처리장치의 부담을 덜기 위한 별도의 기하학 연산기 설계와 이에 따른 고성능 부동소수점 연산기의 설계, 방대한 양의 기하학 데이터를 빠르게 전송하기 위한 제어부 데이터버스 설계, 프레임버퍼와 텍스처 메모리의 대역폭 증가를 위한 캐시시스템 설계, 고속의 렌더링을 위한 병렬구조 등 병목점을 해결하는 하드웨어 전반적인 설계 고려사항이 요구된다.[1]

따라서 가속기를 설계하는데 가장 중요한 것은 두 개의 핵심 처리부인 기하학 연산기(Geometry Processor)와 렌더링 유닛(Rendering Unit)[2]의 성능의 균형을 이루도록 하는 것이다. 이를 위해 설계 이전단계에서 기하학 연산기, 렌더링 유닛, 텍스처 로직, 메모리와 캐시시스템 등의 세부모델별 하드웨어 성능을 측정할 수 있는 시뮬레이션 환경의 구축이 필요하고 또한 이러한 시뮬레이션 환경은 모델링한 하드웨어의 구조를 반영할 수 있어야 하며 그 처리결과와 정확성을 보장할 수 있어야 한다.

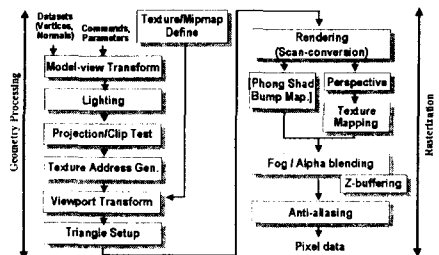
본 논문에서 제시하는 3차원 그래픽 파이프라인 시뮬레이터는 기하학 처리와 렌더링의 기본적인 그래픽 파이프라인 연산을 수행하여 입력 벤치마크(Benchmark)별 수행시간 프로파일링, 기하학 연산 대역폭, 렌더링시의 프레임별 벤치마크 특성, 전체 수행시간 등을 보여줌으로써 차후에 설계할 가속기 세부모델의 설계방향을 제시할 수 있게 하였고 해당 모듈에 적합하게 시뮬

레이터를 수정할 수 있게 하여 이식성있는 실험환경을 제공하였다. 또한 SuperScalar 수행모델 시뮬레이터인 SimpleScalar[5]와 연동하여 전체 수행시간을 측정하였고, 최종적으로 렌더링된 이미지를 보여줌으로써 시뮬레이터를 통한 수행과정의 정확성을 검증할 수 있게 하였다.

본 논문의 구성은 다음과 같다. 2절에서는 기하학 연산, 렌더링의 기본적인 3차원 그래픽 파이프라인 처리과정과 관련연구를 살펴보고, 3절에서는 구현된 시뮬레이터의 전체 구조를 설명한다. 4절에서는 다양한 시뮬레이션 실험결과를 보여주고 마지막으로 5절에서는 결론 및 향후 연구계획에 대하여 논한다.

2. 3차원 그래픽 파이프라인 및 관련 연구

3차원 어플리케이션은 궁극적으로 생성시키고자 하는 객체에 대한 정보(위치, 색, 법선벡터 등)를 데이터베이스에서 불러와 3차원 그래픽 파이프라인에 전달하게 된다. [그림]과 같이 3차원 그래픽 파이프라인은 객체에 대한 위치정보 변환, 광원에 따른 삼각형 색상정보 계산 등의 기하학 처리(Geometry Processing)를 수행한 후 생성된 삼각형을 이미지의 최종 단위인 픽셀로 변환하여, 각 픽셀의 색상값을 계산하는 렌더링(Rendering)과정을 거쳐 최종 프레임 이미지를 생성하게 된다.[2]



[그림 2] 3차원 그래픽 파이프라인

* 이 논문은 과학기술부에서 지원하는 국가지정연구실사업과제 연구비에 의하여 연구되었음.

Stanford 대학에서는 텍스처 매핑에 적합한 캐시시스템[3] 설계를 위해 3차원 그래픽 파이프라인 소프트웨어를 구현하였다. 이 소프트웨어를 이용하여 모델링된 데이터를 입력받아 파이프라인 처리를 거치는 동안 메모리 접근 트레이스를 생성한 후 별도의 캐시 시뮬레이터의 실험을 통하여 최적의 캐시구조를 제안하였고 이에 따른 렌더링 유닛의 성능향상을 보여주었다. 또 Savage 그래픽 가속기를 제작한 S3사에서는 SuperScalar 수행모델의 렌더링 엔진[4]을 설계하기 위해 자체적으로 정확한 수행시간을 측정할 수 있는 시뮬레이터를 개발하여 SuperScalar 3차원 그래픽 엔진의 병렬성으로 인한 성능향상을 검증하였다.

3. 시뮬레이션 환경

본 논문에서는 3차원 그래픽 파이프라인의 기본적인 단계를 수행하여 단계별 또는 전체적인 실험결과를 보여주는 효과적인 실험환경을 제시한다. 이 실험환경에서는 기하학 처리단계, 렌더링 단계를 수행하면서 수행시간별 프로파일링, 대역폭, 벤치마크 특성을 보여주고 최종적으로 렌더링된 이미지를 보여줌으로써 시뮬레이션 처리결과와 정확성을 검증할 수 있게 하였다. 그리고 SuperScalar 시뮬레이터인 SimpleScalar를 적용한 후 SuperScalar 수행모델에서의 수행시간을 측정하여 기하학 처리 프로세서, 렌더링유닛 설계시 요구되는 설계고려사항을 제공하였다.

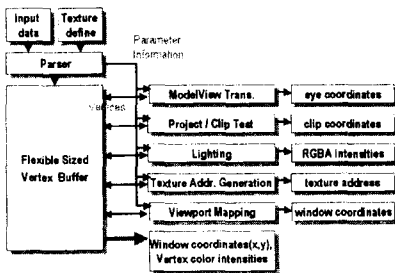
3.1 입력 벤치마크

OpenGL 성능측정시 사용되는 SPECviewperf[6]의 벤치마크들을 시뮬레이터의 입력으로 사용하였다. Awadv, DRV, DX, Light, CDRS 다섯 개의 벤치마크에 대한 세부사항은 [표 1]과 같다.

[표 1] 입력 벤치마크들의 특성

Data Set	Frames	Total Primitives	Vertex/Primitive
Awadv.	50	18,437	3.39
DRV	50	21,526	7.03
DX	50	976	95.84
Light	50	450,029	4.00
CDRS	50	75,136	3.74

3.2 기하학 연산 처리부

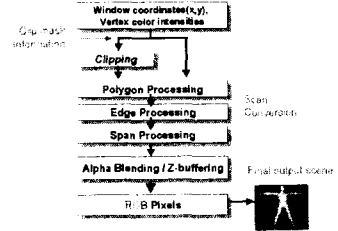


[그림 2] 시뮬레이터의 기하학 연산 처리부

기하학 처리 실험환경은 [그림 2]와 같이 구성하였다. 정점, 법선벡터, 색상등의 벤치마크 관련정보와 실제 기하학 연산 각 단계에서 필요한 입력 파라미터 그리고 텍스처 매핑시 사용된 텍스처 정보가 입력되면, 입력 파라미터값을 해독하는 파서를 통해 정점들은 정점버퍼(Vertex Buffer)에 임시로 저장된 후 파이프라인의 첫 단계인 모델뷰변환 단계로 전달된다. 이때 파서를 통해 해독된 파라미터들을 이용하여 각 단계에 필요한 행렬을 생성하게 되고 각 단계별로 파라미터정보를 전달해 실제 각 단계 처리시 이용하게 된다. 한 단계의 작업이 끝나게 되면 중

간 결과가 저장되고 변환된 정점, 색상정보들은 다시 정점버퍼에 저장되어 다음 단계의 입력으로 사용되게 된다.

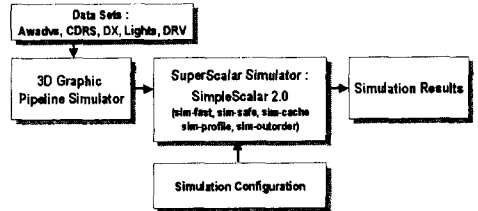
3.3 렌더링 처리부



[그림 4] 시뮬레이터의 렌더링 처리부

기하학 처리가 끝난 후 정점의 좌표와 정점의 색상값은 클립 테스트 단계에서 결정되어진 클리핑 될 대상에 해당하는 정점에 관한 클립마스크 정보를 이용하여 삼각형을 잘라내게 되고 이에 따라 새로이 생성된 또 다른 삼각형들과 함께 주사변환(Scan Conversion)과 블렌딩(Blending) 그리고 깊이정보 비교(Z-buffering) 단계를 거쳐 픽셀을 생성하여 최종이미지를 출력하게 된다.[그림 3]

3.4 SimpleScalar와의 연동



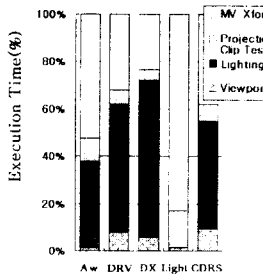
[그림 5] SimpleScalar와의 연동

SuperScalar 수행구조의 범용 프로세서 시뮬레이터인 SimpleScalar는 벤치마크 프로그램을 입력받아 수행시간, CPI, 캐시시스템 성능, 메모리 접근 패턴 등의 다양한 결과를 보여준다. 3차원 그래픽 파이프라인 시뮬레이터를 SimpleScalar의 입력으로 사용하여 SuperScalar 수행모델에서의 벤치마크별 그래픽 엔진의 수행시간을 측정할 수 있게 하였다.

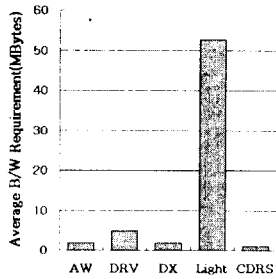
4. 실험 결과

기하학 연산은 대부분 부동소수점연산을 수행하므로 하드웨어 구성시 고성능의 부동소수점 연산기 설계가 중요하다. [그림 5]는 기하학 연산의 각 단계에서의 부동소수점 연산의 수행시간을 프로파일링한 결과이다.[7] Light는 래디오시티(Radiosity)를 이용 색상정보를 생성하기 때문에 라이팅(Lighting)단계를 수행하지 않는다. Light 벤치마크를 제외한 다른 벤치마크들의 경우 전체수행시간의 70~80% 이상이 라이팅과 변환(Transformation) 단계에서 이루어진다. 따라서 해당 단계를 수행하는 하드웨어 설계가 중요한 것을 알 수 있다.

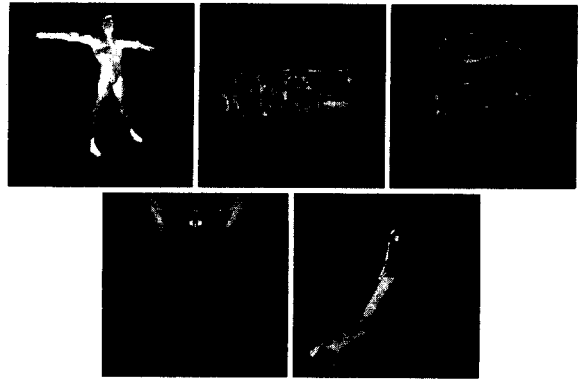
[그림 6]은 벤치마크들의 정점들로 인해 발생하는 기하학 처리 대역폭을 측정된 결과이다. Light의 경우 매우 많은 정점으로 인해 50Mbytes 이상을 처리하게 되는데 이러한 대역폭에 의한 병목점을 해결하기 위해서는 삼각형 메쉬구조의 프리미티브를 이용하거나 기하학 모델 압축기법 등을 이용해야만 한다.



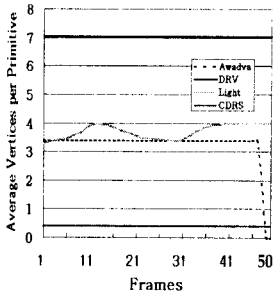
[그림 6] 기하학 연산 프로파일링



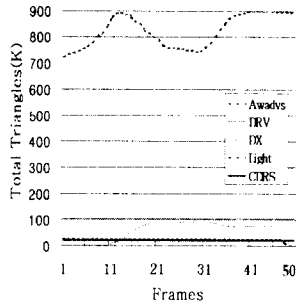
[그림 7] 프레임당 평균 기하학 연산 대역폭



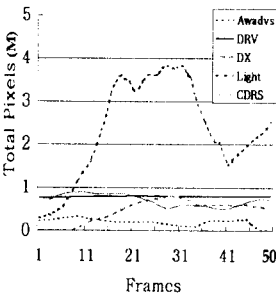
[그림 11] 최종 출력된 벤치마크의 이미지들, 위에서부터 차례대로 AWadvs, DRV, DX, Light, CDRS



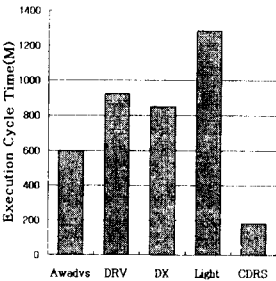
[그림 8] 연속된 프레임에서의 프리미티브 당 평균 정점의 개수



[그림 9] 연속된 프레임에서 총 삼각형수



[그림 10] 연속된 프레임에서의 총 픽셀수



[그림 11] SimpleScalar 시뮬레이션 환경에서의 벤치마크별 수행시간

[그림 7], [그림8], [그림9]는 벤치마크별로 시점의 변화에 따른 50개의 연속된 프레임을 렌더링 하면서 측정된 실험결과들이다. 그림에서처럼 프레임간에 발생하는 삼각형이나 픽셀수의 변화량은 커지게 되므로 프레임을 갱신하기 위한 빈번한 메모리 접근이 발생한다. 따라서 고성능의 렌더링 유닛을 설계하기 위해서는 프레임버퍼를 구성하는 메모리의 대역폭 병목현상을 해결하도록 해야한다.

[그림10]는 Out-of-Order issue가 가능한 SuperScalar 수행모델에서 수행시킨 벤치마크별 수행시간을 보여준다. 수행시간은 정점의 개수, 라이팅 단계수행의 유무 등의 벤치마크 복잡도에 따라 결정되는 것을 알 수있다.

[그림11]은 최종적으로 생성된 벤치마크의 이미지이다. 이를 통해 구현된 3차원 그래픽 파이프라인 시뮬레이션 환경의 정확성을 검증할 수 있었다.

5. 결론 및 향후계획

본 논문에서 제시한 3차원 그래픽 파이프라인 시뮬레이터는 기본적인 그래픽스 연산을 수행하도록 함으로써 향후 설계할 가속연진에 적합한 시뮬레이션 환경을 위한 프레임워크로 설계되었다. 기본적인 벤치마크의 특성분석, 기하학 연산 수행시간 프로파일링, 기하학 모델 대역폭, 렌더링 수행과정 그리고 정적 수행모델과의 연동수행으로 인한 수행사이클측정 등을 통해 다양한 실험결과를 보여주었다.

현재 텍스처매핑, 범프매핑에 관련된 시뮬레이션 환경을 추가보완 중이며 더욱 다양한 실험결과를 보여줄 수 있도록 연구를 진행할 것이다. 차후 본 시뮬레이터는 기하학 모델 압축, 텍스처 매핑에 적합한 캐쉬구조 설계, 병렬 렌더링 같은 폭넓은 연구에 적용되어 고성능의 효율적인 그래픽 가속기 구조를 제시할 수 있게 할 것이다.

6. 참고문헌

- [1] N. Trevett, "Challenges and Opportunities for 3D Graphics on the PC," SIGGRAPH/Eurographics Workshop on Graphics Hardware, keynote 1999
- [2] J.D Foley, A. van Dam, S. K. Feiner, J. F. Hughes. *Computer Graphics:Principles and Practice*, 2nd ed. in C. Addison-Wesley, 1996.
- [3] Z. S. Hakura, A. Gupta "The Design and Analysis of a Cache Architecture for Texture Mapping," *24th International Symposium on Computer Architecture*, 1997
- [4] A. Wolfe, D. B. Noonburg, "A Superscalar 3D Graphics Engine," in *MICRO, International Symposium on Microarchitecture*, 1999
- [5] D. Burger, T. M. Austin, S. Bennett. "Evaluating Future Microprocessors : the SimpleScalar Tool Set," Technical Report 1308, Computer Sciences Department, University of Wisconsin, Madison, WI, July 1996.
- [6] SPECviewperf™ 6.1.2, <http://www.specbench.org/gpc/opc.static/opcview.htm>
- [7] Chia-Lin Yang, B. Sano, and Alvin R.Lebeck, "Exploiting instruction level parallelism in geometry processing for three dimensional graphics applications," in *MICRO, International Symposium on Microarchitecture*, 1998