

휴대용기기에 적합한 3차원 그래픽 렌더링 처리기의 파이프라인 설계.

우 현재, 정 중철, 이 문기

연세대학교 전자공학과

전화 : 02-2123-4731 / 핸드폰 : 017-708-3339

The design of 3D graphics rendering processor for portable device

Hyun-Jae Woo , Jong-chul Jeong, Moon-Key Lee
Dept. of Electrical and Electronic, Yonsei University
E-mail : woohj@spark.yonsei.ac.kr

Abstract

This paper proposes an 3D graphics rendering processor for portable device. One the most important factor is chip size for portable device, but the conventional 3D graphics rendering processor is not a suitable because the processor needs a lot of multiplication and division units. So the proposed architecture substitutes single precision floating point by 32 bit fixed point, and uses recursive units for the same operation such as color values(z, r, g, b, a) and texture values (s, t, u, v).

In this approach, we reduce numbers of multiplications and divisions by 66.1% and 75% respectively at the sacrifice of performance degradation by 2.12% .

I. 서론

최근 디지털 미디어 컨버전스 (digital media convergence)의 개념이 확산됨으로 인하여 디지털 미디어에 여러 가지 기능들이 통합되고 있는 추세이다. 이러한 추세는 개인용 Portable Device에서도 예외가 아니다. 이전의 Portable Device들에서는 단순한 텍스

트 위주의 문서나 정보를 위한 응용 프로그램 수준의 서비스를 제공하였으나, 최근에는 발달된 Display 기술을 바탕으로 보다 그래픽한 정보를 제공할뿐더러, SOC (System On a Chip) 기술의 도입을 통하여 PC에서의 컴퓨팅 기능을 Portable Device에서 제공할 수 있을 정도로 그 응용영역의 폭이 넓어지고 있다. 이에 따라 휴대용 게임기, 차세대 PDA 및 핸드폰 등에 3차원 영상지원이 요구되어지고 있다. 하지만 기존의 3차원영상 처리기는 저 전력, 소규모를 요구하는 휴대형 기기에는 적합하지 않다. 따라서 본 논문에서는 휴대형 기기에 적합한 3차원 영상 처리기를 제안한다.

제2장에서는 3차원 그래픽 가속기 파이프라인의 이해를 위한 장으로 각 단계에서 이루어지는 일과 절차에 대해 기술하였고, 3장에서는 제안하는 고정소수점 연산을 사용한 반복 연산을 수행하는 3차원 그래픽 가속기 파이프라인에 대하여 4장에서는 성능분석을 하며 마지막장은 본 논문의 결과로 이루어진다.

II. 3차원 그래픽 가속기의 이해

3차원 그래픽 가속기는 크게 기하학 연산 처리기와 렌더링 처리기로 이루어진다. 기하학 연산 처리기는 3차원영상 데이터의 크기 및 위치 변화와 그에 따른 빛의 변화를 처리하는 부분이다.

World Transform은 각 객체를 이루는 삼각형들의 정점의 좌표를 객체 좌표계에서 통합 좌표계로 변환하여

모든 삼각형들이, 즉 모든 객체들이 하나의 공간 안에 존재하도록 한다.

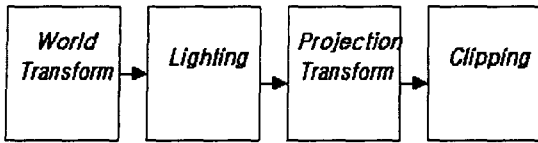


그림 1 기하학 처리기

World Transform을 한 삼각형은 Lighting 과정을 거치게 된다. 현실 세계는 여러 개의 빛이 존재하여 그 빛의 밝기와 색이 객체의 색상에 영향을 주며 객체 표면의 빛의 반사도 객체의 색상에 영향을 준다. Lighting 과정은 이러한 빛에 의한 삼각형의 색상 변화를 처리한다.

3차원 영상 데이터를 2차원 화면에 표현하기 위해서 이 삼각형들을 2차원 평면에 투영시키는 과정이 필요하다. 이를 Projection Transform이라고 한다. 2차원 평면에 투영시킬 때는 가상적인 Z축으로 시점과 삼각형 사이의 거리를 나타내어 삼각형간의 전후 순서를 결정한다. 마지막으로 Clipping 과정에서는 실제로 화면에 보여지는 영역인 View Volume 밖에 있는 삼각형들을 잘라내어 불필요한 연산을 줄여준다.

다음으로 렌더링 처리기는 삼각형 내부 픽셀의 색상을 결정하는 단계로 색상을 계산하고 텍스처를 입히는 과정이다. 렌더링처리기는 기하학처리기로부터 입력된 삼각형 데이터의 기울기 및 색상 정보의 증가분을 구하는 삼각형 셋업 과정, 변에 대한 정보를 구하는 처리기, 픽셀에 대한 색상정보를 구하는 스캔 처리기, 텍스처를 입히는 텍스처 매핑으로 구성된다[1][2][4].

III. 제안하는 휴대형 3차원 그래픽 렌더링 처리기 고찰

3.1고라우드 셰이딩 알고리즘 사용

렌더링 파이프라인은 셰이딩 알고리즘에 의해서 크게 달라진다. 셰이딩이라는 것은 폴리곤 내의 픽셀의 색상 값을 결정하는 방식을 의미한다. 셰이딩에는 플랫 셰이딩(Flat Shading), 고라우드 셰이딩(Gouraud shading), 풍 셰이딩(Phong Shading)이 있으며, 이 중 풍 셰이딩이 가장 고성능의 방식이나 각 픽셀에 대한 법선 벡터에 대한 정보를 처리해야 하기 때문에, 많은 연산량과 대역폭을 요구하게 되어 제안하는 렌더링 처리기에는 적합하지 않다. 플랫 셰이딩 알고리즘의 경

우에는 삼각형의 내부를 세 정점의 평균값을 취하여 처리하는 방식으로 하드웨어는 간단하나 성능이 현저하게 떨어진다. 따라서 제안하는 렌더링 처리기에서는 각 정점 값의 증분을 이용하여 삼각형의 내부를 연산하는 방식인 고라우드 셰이딩을 사용하였다.

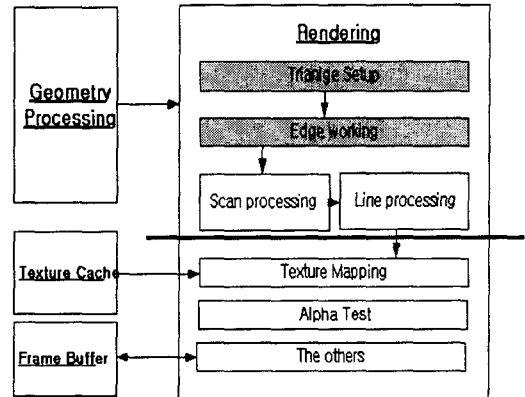


그림2 렌더링 처리기.

우선 삼각형 셋업은 각 삼각형 당 한번만 수행되며 삼각형 각 변의 기울기와 쉐더 값, 깊이 값 및 텍스처 매핑 관련 정보들의 증가분을 구한다. 다음 단계인 변 처리는 삼각형당 2번 수행하게 되고 x, y 및 색상정보 등의 시작 값을 결정한다. 스캔 처리는 변 처리기의 시작점을 가지고 그 라인의 증분을 구한다. 즉 라인 수만큼 연산을 수행한다. 마지막으로 라인 처리는 시작점과 증분을 가지고 삼각형 내의 모든 픽셀의 색상 값을 구하는 것으로 이전 픽셀 위치의 R, G, B, a, s, t, u, v 값에 x축에 대한 증분 값들을 더하여 결과 값을 구한다. 따라서 라인 처리는 해당라인의 픽셀수 만큼 연산을 수행하게 된다.

텍스처 매핑은 텍스처 이미지를 화면내의 객체 표면에 투영하는 것으로 텍스처 데이터를 각 픽셀에 매핑하는 것이다[3].

3.2 하드웨어 구현

위에서 설명한 것과 같이 하나의 삼각형 처리에서 삼각형 셋업은 1번 변 처리는 2번 수행하며 그 외의 스캔 처리나 라인처리 등은 삼각형의 크기에 따라 그 수행 회수가 달라진다.

휴대형 디바이스 환경에서의 3차원 그래픽은 한 프레임 표시하는 삼각형의 숫자가 적다(Low Polygon). 즉 삼각형 하나가 갖는 평균 픽셀 수는 커지게 된다.

따라서 위의 상관관계를 고려하면 삼각형 셋업과 변 처리기의 성능보다는 스캔 처리와 라인처리의 성능이 중요 시 됨을 알 수 있다. 따라서 삼각형 셋업과 변 처리기에서 동일연산을 수행하는 색상정보(z, r, g, b, a) 와 텍스처 정보(s, t, u, v) 처리를 각각 하나의 연산기단을 사용하여 반복 수행함으로써 연산기의 개수를 크게 줄일 수 있었다. 또 기존 삼각형 셋업에서 병렬적으로 사용되는 7개의 나눗셈기를 하나의 나눗셈기로 재귀적인 사용을 함으로 하드웨어의 크기가 큰 나눗셈의 문제를 해결할 수 있었다. 사용한 나눗셈기는 5단 파이프라인을 가진 테일러시리즈 멀티플리케이터 방식으로 결과 값이 최하위 비트(LSB)에서 한 비트의 오차를 갖지만 시뮬레이션을 통하여 그 오차가 무시 될 수 있음을 확인하였다. [표 1]에서 하드웨어의 크기에 주된 영향을 주는 곱셈기와 나눗셈기의 개수를 비교하였다.

		기존의 알고리즘	제안하는 알고리즘
삼각형 셋업	곱셈	65	21
	나눗셈	7	1
변 처리	곱셈	31	7
	나눗셈	0	0
스캔/라인 처리	곱셈	7	7
	나눗셈	1	1
Total	곱셈	103	35
	나눗셈	8	2

표 1 곱셈기와 나눗셈기의 개수 비교

이러한 재귀적인 연산시 다른 단들은 홀드되기 때문에 폴리곤당 34 사이클의 지연이 발생하게 되며 지연 후에는 매 사이클당 한 픽셀씩 처리하게 된다. 이러한 지연이 미치는 영향은 3.4절에 기술하였다.

3.3 제안하는 구조의 데이터 포맷

제한하는 3차원 그래픽 렌더링 처리기는 휴대형기기의 요구 사항인 소규모를 위하여 기존의 Single precision Floating-Point연산 대신 픽스트 포인트를 사용하였다.

제안하는 3차원 그래픽 렌더링 처리기는 삼각형의 x, y, z, r, g, b, a에 해당하는 일반연산 포맷과 0~1사이의 값만을 갖는 소수점에 민감한 format로 나뉜다. [그림3] 위의 포맷은 800*600의 화면과 32비트 트루컬

라를 지

원하도록 설계하였고 큐브, DYN, Plane, 및 화면에 최대로 표현할 수 있는 삼각형으로 검증하였다.[5][첨부]

32bit format

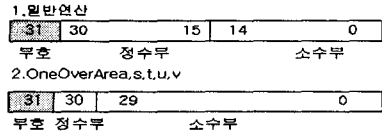


그림 3 제안하는 구조의 데이터 포맷

IV 성능고찰

[표2]는 케이크3의 데모를 800 * 600에 100프레임을 기준으로 프레임당 폴리곤수를 세어서 낸 평균 값이다.[6]

$$aver\ polygon\ size = \frac{screen\ size * aver\ depth\ complexity}{number\ of\ polygon}$$

수식1 폴리곤 당 픽셀 수

위의 수식에서 케이크3의 평균 깊이 복잡도는 3이며 아래의 표와 같다.

	평균 폴리곤수	평균 폴리곤당 픽셀수
데모1	898	1603.56

표 2 케이크3의 평균 폴리곤수

3.2절에서 설명한 것과 같이 제안하는 하드웨어에서는 삼각형당 34 사이클의 지연이 발생한다. 따라서 케이크3의 데모1의 경우 프레임 당 30532(898*34) 사이클이 지연된다. 평균 3의 깊이 복잡도를 갖으므로 한 프레임은 총 1440000(800*600*3)픽셀로 이루어진다. 즉 프레임당 총픽셀 수에 프레임당 지연 사이클을 더하면 제안하는 구조에서의 사이클수가 나온다. 1470532/1440000 *100=102.12% 따라서 2.12%의 성능 저하가 일어난다.

$$frame/sec = \frac{operation\ frequency}{screen\ size * depth\ complexity + the\ number\ of\ polygon * delay}$$

수식2 초당 프레임 수

제안하는 3차원 그래픽 렌더링 처리기는 현 FPGA검증 중에 있으며 36.2 Mhz의 동작 주파수를 갖는다. [수식2]에서 Delay는 34 이며 휴대용 환경에서의 폴리

관수가 3000폴리곤 내외라 가정하면 초당 23 프레임을 표현할 수 있다.

첨부] 데이터 포맷의 검증 (데이터 결과 영상)

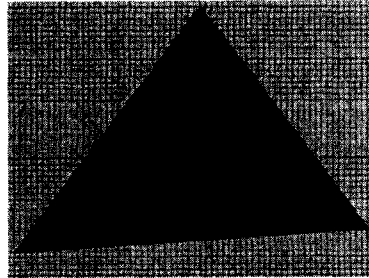
V. 결론 .

휴대형 기기의 대중화에 따라 3D 영상처리 하드웨어의 필요성은 증가 할 것이다. 휴대형 기기에서는 소규모의 칩이 요구되며 본 논문에서 살펴본 바와 같이 제안하는 고정소수점 연산을 사용한 반복 연산을 수행하는 3차원 그래픽 가속기 파이프라인으로 하드웨어의 크기가 큰 곱셈기와 나눗셈기를 각각 33.9%와 25%로 효과적으로 줄일 수 있었고 케이3의 데모의 경우 단지 2.12%의 성능감소만이 있었다.

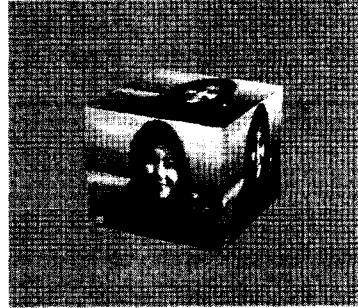
참고문헌(또는 Reference)

- [1] J.D Foley, A. Dam, S. K. Feiner and J. F. Hughes, computer Graphics principles and practice 2th Ed, Addison-Wesley, Massachusetts, 1997.
- [2] A.Kugler, "The setup for Triangle Rasterization, " 11th Euro-graphics Workshop in Computer Graphics Hardware, August 1996.
- [3] Tomas Möller and Eric Haines "Real-Time Rendering. p99~120: Texturing
- [4] COMPAQ, "Neon: A (Big) (Fast) Single-Chip 3D Workstation Graphics Accelerator."
<http://www.research.compaq.com/wrl/admin/publications.html>
- [5] Mesa library
<http://www.ssec.wisc.edu/~brianp/Mesa.html>
- [6] Quake 3
<http://www.idsoftware.com/games/quake/quake3-arena>
- [7] plane
<http://www.gldoman.com/Programs/Terra3d.htm>

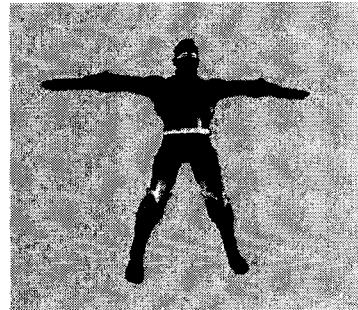
1.표현가능한 최대 크기의 폴리곤



2.메사 큐브



3.메사 DYN



4.Plane [7]

