

그래픽스 하드웨어 구매 가이드(1)

컴퓨터 그래픽스 산업의 급속한 성장과 분화는 엔지니어들에게는 축복이다. 경쟁에 의해 값이 싸졌고, 간단한 2차원 도면에서 쉐이딩(shading)된 솔리드에 이르기까지 많은 이미지 타입들을 가지고 작업하는 것이 더욱 쉬워졌다.

물론 단점은 구매 결정이 더 복잡해졌다는 것이다. 우리들 대부분은 전자 회로의 배치를 하는데 있어서는 3차원 가속기를 필요로 하지 않는다는 것을 알지만 업자들이 전자 부분 엔지니어들에게 구경을 구매하도록 말하는 것을 보아 왔다. 그래픽스 하드웨어를 구매하는데 있어 다음과 같은 의문이 생긴다.

- 그래픽스 하드웨어의 가격들이 왜 그렇게 천차만별인가, 또한 이용할 수 있는 다른 종류의 제품들은 어떤 것들이 있는가?
- 그래픽스 시스템에서 필요로 하는 특징은 어떤 것들인가? 예를 들자면, 쉐이딩된 이미지들을 위해 형상 가속기들이나 텍스처 매핑 하드웨어를 필요로 한가?
- 솔리드 모델링 작업에 3차원 "제임 가속기"를 사용할 수 있는가?
- 그래픽스의 속도를 향상시키기 위해서는 어떤 종류의 가속기를 사용할 것인가?
- 호스트 컴퓨터의 CPU는 그래픽스에서 무슨 역할을 하는가?
- 필요로 하는 그래픽스 메모리가 얼마인지를 어떻게 알 수 있는가?
- CAD 응용의 종류가 그래픽스 하드웨어 구매에 어떤 영향을 미치는가?
- 제품들을 비교하는데 있어서의 최고의 기준은 무엇인가?

이런 질문들에 답하기 위하여, 우리는 기본적인 것부터 시작하기로 한다.

모든 컴퓨터 그래픽스 서브시스템은 그래픽스 작업에 대해 호스트 CPU의 부하를 덜어 주는 특수한 회로를 얼마나 사용하는가에 따라 특정 지어진다.

소량을 하는 시스템들에 대해서는 "소프트웨어적"으로 그래픽스를 한다고 말한다. 반면 대량을 하는 시스템들은 "하드웨어적"으로 한다고 칭한다. 구체적으로 말하자면, 우리가 이야기하고 있는 주요 하드웨어들은 메모리 버퍼들, 내장된 프로세서들과 컨트롤러들, 그리고 이들을 연결시키는 버스들이나 전자 채널들이다.

전산화된 이미지는 화소(pixel or picture element)라고 불리는 작고 이산적인 빛의 영역들로 구성된다. 화소들은 타일들을 가지고 모자익을 만드는 것과 상당히 유사한 방식으로 특정 장면(scene)을 구성한다. 그러나 CAD 시스템은 화소의 단위가 아니라 평면, 원통, 스플라인 곡면과 같은 기하학적 형상 요소를 고려한다. 따라서 컴퓨터 그래픽스란 응용의 기하학적 형상을 적절한 화소의 배열로 변환하여 작업자가 2차원 혹은 3차원 이미지를 인식할 수 있도록 하는 것이다.

◎ 프레임 버퍼

모니터에 화소들을 정의하기 위해 사용되는 데이터는 프레임 버퍼라는 특별한 메모리 영역에 저장된다. 프레임 버퍼는 디스플레이를 위해 필요한 데이터가 크기 때문에 이용되게 되었다. 만약 버퍼가 없다면, 화면상의 이미지가 리프레쉬될 수 있도록 자주 데이터를 전송하는 동안 메인 시스템의 버스는 막힐 수도 있다.

시스템의 가장 기초가 되는 것은 "dumb frame buffer"라고 불리는 것을 갖는 것이다. Dumb 터미널 처럼 dumb frame buffer도 CPU에 의해 제어되고 CPU에서 나오는 데이터에 의해 채워진다. Dumb frame buffer의 시대는 지나갔다. 만약 GUI를 통해 컴퓨터와 대화하고자 한다면, 화소의 그룹들을 이동시킬 수 있는 그래픽스 칩(graphics chip)의 지원을 받는 프레임 버퍼가 필요하다. "block operations"는 윈도우 크기의 변화, 메뉴의 이동 그리고 모델들의

panning이나 zooming을 포함한다. 이러한 어댑터들은 GUI 가속기라고 불리기도 한다.

컴퓨터를 이용한 제도, 전자 설계, 3차원 와이어 프레임 모델링과 같은 작업에는 그래픽스 부시스템이 거의 요구되지 않는다. 윈도우즈 시장에서, 약 200\$정도면, 좋은 GUI 가속기를 구입할 수 있다. 그러나 솔리드나 꼭면 모델들, 대규모의 건축 모델들, 또한 화학 플랜트 모델들은 그래픽스 부시스템을 훨씬 더 필요로 한다.

◎ 2차원과 3차원

모든 그래픽스 시스템들은 일반적으로 "graphics pipeline"으로 지칭되는 단계들을 갖추어야 한다. 이러한 단계는 다음과 같은 것들로 구성된다.

모델의 생성 : 컴퓨터의 중앙처리장치는 응용에 의해 생성되는 데이터를 조립하고 관리한다. 이 데이터는 2차원의 지도나 도면, 또는 3차원의 솔리드, 꼭면, 유향요소, 분석을 통해 나온 데이터의 집합들로 구성될 수 있다. 그래픽스 프로세서는 파이프라인에서 가장 강도 높은 작업인 이 단계에 도움을 주지 않는다.

전달(hand-off) : 사용자가 데이터 이미지의 갱신을 요구할 때 응용은 그래픽스 프로그래밍 인터페이스(응용 프로그래밍 인터페이스 혹은 API)를 통하여 디스플레이 프로세서에 명령을 보낸다.

변환 : 응용의 3차원 데이터는 좌표 변환되어 17인치나 20인치 디스플레이의 2차원 평면상에 매핑되어야 한다.

래스터라이징(rasterizing) : 변환된 이미지는 이산형 화소의 형태로 프레임 버퍼에 저장되기 위해 재구성되어야 한다.

디스플레이 : 프레임 버퍼의 데이터를 읽어 디지털-아날로그 변환기(RAM-DAC)를 통해 신호로 변환되어 모니터를 구동한다.

컴퓨터 그래픽스 전문가들은 파이프라인에서 각 단계의 호칭과 보조 단계의 위치에 이견을 갖고 있다. 위의 목록은 "The X Journal"의 9/10월호에 Silicon Graphics사의 Mark Kilgard가 기고한 OpenGL 그래픽스에 관한 글을 각색한 것이다. 일관성 없는 용어 때문에 그래픽스 하드웨어를 고르는 일이 더욱 복잡하다.

파이프라인이라는 용어는 관을 통해 변화없이 흐

르는 액체를 연상시킨다. 그것은 2차원에 있어서는 좋지만 3차원에 있어서는 다른 몇 가지가 추가되어야 한다. 파이프라인이 복합 반응기로 우회하고, 탱크에 머물게 되는 것을 상상해 보라. 솔리드 모델을 렌더링하는 추가적 노력은 이와 비슷하다. 2차원과 3차원 데이터는 파이프라인에서 다른 경로를 갖는다. 그것들은 기본적으로 다른 두개의 공칭이고, 이러한 이유로 상당히 다른 가격표를 붙이게 된다. 3차원 그래픽스 프로세서들의 비싼 이유를 이해하고 사용자의 요구에 적합한 사양을 결정하기 위해서는 그래픽스 파이프라인의 주요 기능인 변환, 셋업, 래스터라이징을 이해하여야만 한다.

◎ 변환

CAD 시스템은 실세계 형상을 이용한다. 그것은 물체를 인치나 피트 혹은 미터 단위로 스케일링한다. 그러나 그래픽스 부시스템은 디스플레이-윈도우 좌표계의 개념으로 생각한다. 2차원 직선을 윈도우 시스템으로 매핑하는 것은 비교적 쉬운 과정이다. 그러나, 3차원 솔리드 모델을 평평한 디스플레이 시스템으로 렌더링하는 것이 훨씬 더 복잡하다.

3차원 변환 과정은 화가가 그림을 그리는 것과 상당히 유사하다. 화가가 50피트의 나무를 그릴 때, 마음속으로 계산한 나무의 이미지를 캔버스에 5인치나 10인치로 그려야만 한다. 그는 이미지를 세계 좌표계에서 캔버스 좌표계로 변환한다. 또한 그 화가는 나무가 캔버스의 위쪽 끝과 측면 끝에서 약간의 거리를 가져야 한다는 것을 계산한다. 그는 그 그림의 외각에 있는 모든 것을 무시한다. 이와 동일한 과정이 컴퓨터 그래픽에서는 클리핑(clipping)으로 불린다. 마지막으로 화가는 원근법을 이용해 나무를 캔버스 위로 투영한다.

색상도 컴퓨터 화면으로 매핑되어야만 한다. 화가가 나무를 그리기 위해 팔레트에서 색을 선택하는 것처럼 그래픽스 프로세서도 디스플레이 이미지의 각 점에 적절한 색을 계산해야 한다.

변환 단계에서 사용된 형상은 응용 데이터베이스의 형상이 아니다. 그래픽스 시스템은 "primitives"라는 것을 이용한다. 일반적으로 2차원 그림을 그리는 데 있어서, 이러한 primitives는 짧은 선분(벡터)들이며, 3차원 이미지들은 삼각형 primitives에 의해 묘사되어진다.

삼각형의 각 꼭지점에서 그래픽스 시스템은 적절한 색과 셰이딩, 투명도, 그리고 가능하다면 질감을 계산해야만 한다. 또한, 곡면의 각 꼭지점에 대해 이러한 값들의 각각의 변화율을 계산해야만 한다. 후자는 셋업(setup)으로 불리기도 하는데 그래픽스 파이프라인의 다음 단계인 래스터라이징을 위해 필요하다.

◎ 래스터라이징

컴퓨터는 두 가지의 기본적인 작업을 통해 래스터라이징을 한다. 첫번째는 주어진 primitive가 차지할 화소들을 결정하는 것으로 주사 전환(scan conversion)이라 부른다. 두번째는 적절한 색으로 각 화소를 채우는 일인데 이 단계를 주사 생성(scan generation)이라 한다. 이 두 단계를 합쳐서 edge-pixel 보간이라 부르기도 한다.

주사 전환을 위해서 컴퓨터는 셋업에서 계산된 변화율을 필요로 한다. 2차원 셋업 간단하지만 3차원 primitives에 있어서는 삼각형 변의 위치를 알아야만 한다. 예를 들면, 래스터라이징 부시스템은 화소들을 몇 개나 늘리거나 줄여야 할지를 알아야만 하고, 꼭지점에서 오른쪽이나 왼쪽으로 조금 떨어진 삼각형의 변을 찾아야 한다. 또한 다양한 색, 빛의 투영, 텍스처, 그리고 응용에서 요구하는 다른 특징들의 변화율을 필요로 한다.

래스터라이징 작업은 응용이 요구하는 현실성을 높인다. 안티앨라이징된 선으로 2차원 그림을 만드는 것은 들쭉날쭉한 대각선들(jagged diagonals)로 그림을 렌더링하는 것보다 더 많은 일이 요구된다. 컴퓨터가 3차원 와이어 프레임을 렌더링하는 것은 단순한 선으로 그림을 그리는 일보다는 훨씬 더 힘든 일이다. 솔리드들은 그래픽스 프로세싱 양을 증가시키고 다양한 셰이딩 알고리즘들의 복잡성 때문에 계산량도 증가된다.

Flat 셰이딩은 주어진 primitive에 있는 각 화소가 동일한 색을 갖기 때문에 상당히 단순하다. 그러나 더 부드러운 Gouraud 셰이딩을 하려면 컴퓨터는 주어진 primitive내의 각 화소들이 갖는 다른 색들을 보간해야만 한다. 투명도를 나타내거나 실제 곡면의 재질이나 텍스처를 매핑을 하기 위해서는 더 많은 계산을 필요로 한다.

셋업과 래스터라이징은 대칭적이다. 그래픽스 파

이프라인의 시작 부분에서는 대부분의 형상이 실수로 표현된다. 그러나 일반적으로 컴퓨터는 셋업의 마지막에서 실수를 정수로 변환한다.

◎ 게임들과 CAD

그래픽스 프로세서의 기능들에 대해 간략하게 이해했다고 해도 그것의 속도가 얼마나 빠른지를 알기는 쉽지 않다. 다양한 종류의 그래픽스 어댑터를 만드는 Diamond Multimedia사의 Todd Reddick은 컴퓨터 구매자들은 게임 기반 어댑터와 응용 기반 어댑터를 구분해야 한다고 제안한다.

게임 카드들은 마치 그것들이 더 좋은 업데이터인 것처럼 보이게 하는 몇 가지 특징을 가지고 있다. 그것들은 워크스테이션 계열에 이용되는 그래픽스 파이프라인을 위해서 선별적으로 하드웨어의 확장을 제공하지만 솔리드 모델링 환경에서의 그것들의 능력은 실망스럽다.

Diamond Stealth 3D 2000과 Number Nine의 9FX Reality 가속기는 모두 S3의 Virge 그래픽스 프로세서에 기반을 둔다. Reddick의 말에 따르면 Virge는 3차원 그래픽스에서 은신 제거를 위해 Z축을 비교하는 회로를 갖고 있다. 또한 Virge는 3차원 곡면에 텍스처 매핑을 하고, 비디오 이미지들의 재생시 필요에 따라 색을 변환하며, GUI에도 신경을 쓴다. 이런 것들은 칩 하나가 하기에는 많은 기능들이다.

게임 가속기가 칩 하나로 이 모든 작업을 하는 반면, 더 고가의 제품들은 각각의 작업들에 대해 별개의 칩들을 사용한다고 Reddick은 이야기한다. 그것은 게임에 사용되는 그래픽스는 정교한 텍스처 매핑을 이용해 삼각형 primitives의 조잡함을 없애주기 때문이다. CAD 시스템들은 정밀성을 높이기 위해서 그래픽스 프로세서의 부하를 엄청나게 증가시키는 상당히 좋은 먼 근사방식을 사용한다.

Matrox의 Millennium을 포함한 게임 기반 카드들은 2차원의 대화식 그래픽스 및 3차원 와이어 프레임과 잘 작동한다. 그리고 미리 저장된 3차원 이미지들의 재생을 많이 하는 3차원 응용들에 적절하다. 몇몇 게임 기반 어댑터들은 간단한 솔리드 모델들에 적절히 이용될 수 있다.

◎ 3차원 시장의 분할

게임 시장은 별도로 하고, 3차원 그래픽스 어댑터

들은 네개의 큰 범주로 구분된다.

고속 래스터라이저 : 이 종류의 가속기는 솔리드 모델링을 위한 시작점이다. 그것은 IBM의 GXT 500 계열, SGI의 Indy Modeler, Sun의 Creator graphics, 그리고 Diamond의 Fire GL, Elsa의 Gloria와 AccelGraphics의 AG300같은 3D Labs의 Glint 300SX를 위한 어댑터들이 여기에 포함된다.

셋업 프로세서들 : 이것들 중에서 가장 눈에 띄는 것인 3D Labs의 Delta 셋업칩 기반 어댑터들이다. 300SX, Dynamic Pictures의 Oxygen 102, 그리고 Intergraph의 Intense3D와 RealizM 가속기들 등이 이 셋업칩을 이용하고 있다. Diamond의 Fire GL 2000과 3000, Elsa와 Accel 등도 이 범주에 속한다.

형상 프로세서들 : 형상 프로세서로 잘 알려져 있는 것들로는 SGI의 XZ, Extreme과 Solid IMPACT 가속기들이 있고, HP의 Visualize-8과 -24어댑터들, IBM의 G4xi와 새로운 GXT 1000 계열과 마찬가지로 Superand hyperSPARC 컴퓨터를 위한 Sun의 구형 ZX 어댑터들도 이에 속한다. Windows NT 시스템에 대해서는 Intergraph만이 형상 가속기를 제공하고 있다.

텍스처 매핑 프로세서들 : 매우 정교한 그래픽스 시스템들은 나무, 도자기, 금속 곡면 등의 미리 정의된 질감을 물체에 매핑하는 계산 강도가 높은 일의 속도를 향상시키는 회로를 제공한다. SGI의 Reality Engine 제품들과 SGI의 High and Maximum Impact devices, HP의 Visualize-48, 그리고 몇몇 Intergraph 가속기들이 여기에 속한다.

◎ 병목현상의 조사

일반적으로, 응용에 적절한 수행도를 갖는 가장 싼 그래픽스 프로세서를 사야만 한다. 그러나 만약 하드웨어가 특정 작업에 충분히 빠르지 못하다면, 수행도에 장애가 되는 프로세스를 구체화하는 것이 문제 해결에 도움이 된다.

만약 컴퓨터가 말을 할 수 있다면 문제점에 대해 이렇게 대답할지도 모른다. "모델이 너무 많은 웨이딩 삼각형들로 구성되어 있어 변환할 수 없습니다. 제발 형상 가속기를 달아 주십시오." 혹은 "입력 형상을 처리할 수는 있지만 화소들이 너무 많습니다. 더 좋은 래스터 라이저를 달아 주십시오."

그러나 컴퓨터는 말을 하지 못한다. 그러면 어디

서부터 시작해야 할까? 만약 메뉴를 드래깅하는데 문제가 있으며, 더 빠른 GUI 프로세서가 필요한 것이다.

메뉴를 드래깅하는 것이 문제가 되지 않는다면, 그래픽스 파이프라인의 후미에서부터 문제를 분석해보자. 그 이유는 점진적으로 파이프라인의 끝을 향해 가다 보면, 각 단계에서 모델이 개별적 계산을 요하는 많은 수의 작은 단위로 분해되게 된다.

일반적으로 래스터라이징은 가장 많은 계산을 요하므로 속도 저하의 기본적인 원인이 된다. Kilgard는 병목현상이 래스터라이징 프로세서에서 발생하는지를 알아보기 위한 하나의 방법은 작업을 하면서 윈도우의 크기를 줄여 보는 것이라고 말한다. 만약 이것이 렌더링 속도를 높인다면, 래스터라이징 성능이 더 좋은 가속기가 수행도를 높일 수 있을 것이다.

만약 윈도우의 크기를 조정하는 것이 도움이 안된다면 형상 처리에 문제를 가지고 있을 수 있다. 또다른 가능성은 응용 프로그램 그 자체가 중앙처리장치를 마비시키고 있을 수도 있다는 것이다. 모델을 생성하는데 너무 많은 시간이 걸려서 마이크로프로세서가 그래픽 명령이 늦어질 수도 있다.

만약 병목현상이 메인 프로세서와 래스터라이징 어디에서도 없다면 다른 선택이 있을 수도 있다. 몇몇 컴퓨터에 있어서, 당신은 조건을 제대로 갖춘 형상 가속기 대신 셋업 프로세서를 후보로 할 수 있다.

하드웨어 구입처에 셋업 프로세서 보드를 갖춘 워크스테이션과 그렇지 않은 워크스테이션에서 당신의 응용 소프트웨어를 실행시켜 보도록 요청하라. Windows NT 시스템들에서는 Glint 300SX 기반 어댑터를 사용하는 Delta 셋업칩이 있는 시스템과 없는 시스템을 간단히 비교해 볼 수도 있다.

값싼 형상 가속기 중의 하나는 Intergraph가 Intense3D 어댑터들을 위해 \$2,790짜리 추가모듈이다 (\$2,499의 래스터라이징 보드 값에 추가된다). Intergraph의 최상급 RealizM 어댑터에 추가되는 형상 가속기 모듈은 \$5,000이다.

그래픽스 시스템이 갖는 능력이 월등해짐에 따라 SGI와 HP같은 판매 업체는 가격이 훨씬 비싼 복합형상 가속기를 개발하게 되었다. 추가적인 형상 가속기들은 더 많은 래스터라이징 회로를 필요로 한다. 이런 점에서는 그래픽스 부시스템이 관료 체제와 같다. : 파이프라인에서 처음의 소량의 작업은 마지막

에서 대량 작업을 창출한다. 사용자는 별도의 형상 엔진과 래스터라이징 엔진들의 구매에 돈을 들어야 할뿐 아니라 개개의 형상 엔진들에게 일을 분배하고 모아진 결과를 응집된 이미지로 재구성하는 작업을 모두 관리하는 칩에 대해서도 비용을 든다.

◎ CPU의 선택

일반적으로 말하자면, 호스트 시스템의 CPU 파워와 시스템이 하드웨어 가속기에 그래픽스 작업을 맡기는 정도에는 상관 관계가 있다. Digital Equipment는 강력한 알파 마이크로 프로세서로 하여금 front-end 부분의 그래픽스 부하를 처리하도록 하고 있다. DEC의 새로운 PowerStorm 가속기들은 래스터라이징 단계까지는 그래픽스 파이프라인을 보강하지 않는다.

역으로, 인텔의 펜티엄 프로의 형상 변환 작업을 덜기 위한 디자인된 가속기들의 수가 증가하고 있다. 펜티엄 프로는 강력한 정수 연산 능력이 있지만 부동 소수점 연산이 느리기 때문에 셋업 가속기로서는 덜 이상적인 것 같다. 심지어 우리는 인텔이 펜티엄 프로를 보완하기 위하여 셋업 가속기를 개발하고 있는 것으로 믿고 있다.

많은 양의 CAD 작업에서, 빠른 부동 소수점 연산이 필요한 유일한 시간은 3차원 형상을 변환하고, 솔

리드 모델을 셰이딩할 때이다. 이때는, CPU쪽의 상류가 아니라 그래픽스 하류부분에 추가적인 실수 연산 능력을 배치하여야 한다.

그래픽스 작업을 위한 CPU들을 평가할 때 우리는 다음의 사항들을 제시한다. 우선 가능하면 정수 연산이 강력한 것을 택하라. 왜냐하면 대부분의 작업은 빠른 정수 연산을 요한다. 그 다음으로 CPU의 부동 소수점 연산 능력을 찾아라. 필요하면 그때 가서 부동 소수점 프로세서를 갖는 가속기를 추가하면 된다. 물론, 응력해석이나 아날로그 회로 시뮬레이션과 같이 부동 소수점 연산이 많이 필요한 응용에 있어서는 예외이다.

※ 다음 호에는 본 기사의 2편으로 그래픽스에 필요한 메모리의 양과 텍스처 매핑 하드웨어에 관한 내용을 기재할 예정이다.

«CAD Report Vol. 16, No. 9, September 1996»

본 기사는 경상대학교의 전차수 편집위원이 "CAD Report"에서 발췌하였으며 출판사인 CAD/CAM Publishing Inc.의 연락처는 다음과 같다.

- Fax: 1-619-488-6052
- e-mail: Cadcire@aol.com