

## 그래픽스 하드웨어 구매 가이드(2/2)

지난번('96. 12월호)에서는 컴퓨터 그래픽스 가속기들의 기본적인 기능들에 대해 살펴봤었다. 이런 기능들은 "그래픽스 파이프라인"이라는 용어로 표현되기도 한다. 가속하고자 하는 파이프라인의 기능들을 간략히 명사해 보는 것만으로는 그래픽스 부시스템을 구매하는데 있어 충분치 않다. 또한 각 기능들을 지원하는데 적절한 양의 메모리를 구입필요가 있다. 이번에는 그래픽스 메모리에 대해 이야기 하고 필요한 양을 결정하는 방법을 설명한다. 또한 텍스처 매핑 프로세스들, 그래픽스 하드웨어를 구매하기 전에 응용에 대해서 알아야 할 것들, 그리고 경쟁관계에 있는 시스템들의 성능을 비교하기 위해 벤치마크들을 어떻게 사용할 것인지를 살펴본다.

### ◎ 그래픽스 메모리

그래픽스 가속기의 회로기판에 놓이는 그래픽스 메모리는 일반적으로 컴퓨터의 메인 메모리와는 물리적으로 다르다. 그래픽스 메모리는 여러 개의 버퍼들로 나누어지는데 이 버퍼들은 그래픽스 파이프라인을 통해 이동하는 데이터의 저장소 역할을 한다. 메모리의 양과 위치는 그래픽스 가속기의 비용과 능력에 많은 영향을 준다.

### ◎ 메모리의 종류

디스플레이 부시스템은 대화식 작업에서 리드로우를 빠르게 하기 위해서는 많은 일을 고속으로 처리할 수 있어야만 한다. 따라서 메인 메모리에 사용되는 표준 동적램(DRAM)칩보다 더 빠른 다양한 종류의 메모리가 사용된다.

이제까지는 주로 비디오램(VRAM) 칩들이 프레임 버퍼들을 위해 선택되었다. VRAM은 더블 포트이기 때문에 데이터를 보내면서 입력도 받을 수 있다. 그러나 지난 몇 달 전부터 제조자들은 동기식 DRAM(SDRAM)이라 불리는 제품에 관심을 가지기 시작했다. SDRAM은 VRAM 보다 더 빠른 클럭 속

도로 작동하기 때문에 더 많은 데이터를 관리할 수 있다. 하지만 SDRAM은 싱글 포트이기 때문에 VRAM보다 싸다. 동기식 그래픽스 RAM(SGRAM)은 SDRAM과 유사하지만 조금더 빠르다. 미쯔시비사는 3DRAM이라는 새로운 종류의 메모리를 개발했다. 3DRAM은 동기식이며 더블 포트를 갖는다. 3DRAM은 SUN사의 Creator 가속기에 사용된다. 그래픽스 메모리의 또다른 타입으로서 Windows RAM이 있다. 이 Windows RAM은 더블 포트이고 Matrox사의 Millennium 가속기에 사용된다. 실리콘 그래픽스는 IMPACT 가속기에 RAMBUS DRAM(RDRAM)을 사용한다. RDRAM은 500 MHz 이상의 엄청난 빠른 클럭 속도로 작동한다.

일반적으로 그래픽스 램의 종류는 카드 설계자들만의 관심사다. 예를 들면, RAMBUS DRAM을 갖는 가속기는 더 높은 클럭 속도로 작동하지만 다른 타입의 메모리를 갖는 시스템보다 더 좁은 버스를 이용한다. 결국 서로 다른 형태의 메모리를 사용하는 두 시스템은 유사한 데이터 처리 속도를 가질 수 있다. 시스템 구매자가 그래픽스 메모리의 종류에 관심을 가져야할 경우는 같은 구조를 갖고 동일한 기능에 대해 여러 종류의 메모리가 이용될 수 있는 경우이다. 이때 더 빠른 종류의 메모리를 찾거나 또는 더 저가의 것을 알아봐야 한다.

### ◎ 3차원 메모리 수요

2차원 그래픽스에서는 프레임 버퍼 이상을 필요로 하지 않지만, 3차원 그래픽스에서는 다르다. 셰이딩된 이미지를 만들기 위해서는 상당히 많은 양의 추가적인 데이터가 요구되므로 많은 보조 버퍼들이 사용된다. 그것들 중에는 Z-버퍼(Z-buffer), 더블 버퍼(double buffer), 알파 버퍼(alpha buffer), 오버레이 버퍼(overlay buffer), 스텐실 버퍼(stencil buffer), 집적 버퍼(accumulation buffer) 등이 있다. 이들 중 CAD에 있어서 가장 중요한 것은 double buffer와 Z-buff-

er 순이다. 만약 솔리드 모델과 부드러운 대화 작업이 필요하다면 이런 기능들을 모두 만족시키는 그래픽스 메모리를 구입해야 한다.

### ◎ 메모리와 움직임

만화 영화와 같이 컴퓨터 화면에서 CAD 모델의 움직임은 실제로 정지 프레임들이 빠른 속도로 그려지는 것이다. 초당 30개의 프레임의 빠른 속도를 지원하는 어댑터가 컴퓨터 애니메이션 전문가를 위하여 출현하였다. 그러나 Intergraph 사의 Steve Pesto에 의하면 많은 응용에서는 15개면 충분하고, 영화에서는 초당 24개의 정지 이미지들이 지나간다고 한다.

모니터의 리프레시 속도를 설정할 때는 그래픽스 가속기가 작동하는 프레임 속도를 고려할 필요가 있다. 리프레시 속도는 프레임 속도의 배수이어야 한다. 이렇게 설정하는 것은 이미지가 실제로 반으로 찢어지는 현상을 방지한다. 만약 60 hz의 리프레시율로 작업한다면 그래픽스 부시스템이 정상적일 때 화면 제어기는 초당 15개나 30개의 프레임을 허용할 것이다. 리프레시율이 72 hz인 경우에는 초당 24개의 프레임이 좋고, 초당 12개의 프레임일 경우에는 물체의 움직임이 부드럽지 않을 것이다.

### ◎ 더블 버퍼링

대부분의 컴퓨터 시스템들을 아주 빠른 프레임 속도로 3차원 그래픽스 작업을 할 수 있게 한다는 것은 어렵다. 이러한 이유 때문에 더블 버퍼는 유용하다. 더블 버퍼가 없다면 반응이 느릴 뿐 아니라 이미지가 화면에서 움직임에 따라 깜빡거리게 될 것이다. 더블 버퍼는 간단히 첫 번째 프레임 버퍼와 교대로 동작하는 두 번째 프레임 버퍼이다. 임의의 주어진 순간에, 두 개의 프레임 버퍼중의 하나가 화면을 조종하는 앞버퍼 역할을 하고, 이미지는 또 다른 버퍼(뒷버퍼)에서 지워지고 다시 그려진다. 이런 기술은 화면상에서의 물체의 움직임을 더 부드럽고 자연스럽게 한다.

### ◎ Z 버퍼

Z 버퍼는 컴퓨터가 서로 겹치는 물체의 정확한 모습을 투영하는 계산을 하는데 이용된다. 컴퓨터는 화면상의 이미지에서 각 화소들의 깊이 혹은 Z축 값을 비교한다. 컴퓨터 사용자에게 가장 가까운 화소들은 나타나고 다른 곡면에 의해 가려지는 화소들은

감춰진다. 이 효과는 물체들을 솔리드처럼 보이도록 한다. 저가의 시스템에서는 Z 버퍼링이 메인 메모리에서 이루어지지만, 전용 Z 버퍼는 컴퓨터 화면에서 세이딩된 모델의 움직임을 더 쉽게 한다.

### ◎ 오버레이 버퍼

오버레이 버퍼들은 모델 위에 겹쳐지는 메뉴나 주석 그리고 다른 이미지들을 표현하기 위한 화소 데이터를 저장한다. 오버레이 버퍼가 없다면 그런 이미지들은 아래에 놓이는 모델의 일부분을 지우게 된다. 이 방법을 사용하면 팝업 메뉴에 의해 감춰졌던 모델의 일부분이 나타나기를 기다리거나 메뉴가 닫힐 때 재생성될 필요가 없다.

### ◎ 알파 버퍼

알파 버퍼는 모델의 각 곡면이 어느 정도의 투명도를 가져야 하는지를 래스터라이저에게 알려주는 정보를 저장한다. 그 데이터를 가지고 래스터라이저는 겹치는 화소들의 색을 얼마만큼 혼합해야 하는지를 알게 된다. CAD 응용이 자동차의 전면 유리같은 것을 렌더링할 수 있게 하려면, 메모리 계산시 알파 버퍼를 위해 약간의 RAM을 포함시킨다는 것은 좋은 생각이다. 때때로 "버퍼" 대신에 "알파 채널"이라는 용어가 투명도 데이터를 유지하는 메모리를 지칭하기 위해 사용된다.

### ◎ 기타 버퍼들

일부 시스템들은 움직임을 흐릿하게 하거나 다른 특별한 효과를 갖도록 하기위해 여러 개의 프레임들을 저장하는 집적 버퍼를 갖는다. 집적 버퍼는 많은 데이터를 저장하기 때문에 커지게 된다. 화면의 일부에만 그림이 그려지도록 하는 스텔스 버퍼도 있다. 집적 버퍼나 스텔스 버퍼 모두 대부분의 CAD 응용에서는 필요치 않지만 애니메이션이나 특수효과를 나타내고자 할 경우에는 유용하다.

마지막으로, 많은 어댑터들은 현재 텍스처 메모리를 위해 별도의 슬롯들을 갖고 있다. 일반적으로 텍스처 메모리는 다른 버퍼들을 위한 메모리와 분리되어 있다.

### ◎ 필요한 그래픽스 메모리의 양은?

메모리의 양은 세가지 요소에 의해 정해질 것이다.

- 이미지를 나타내고자 하는 해상도
- 각 화소를 정의하기 위한 데이터의 양
- 더블 버퍼의 필요 여부(3차원에 대해서는 필요하다)

### ◎ 해상도

복잡한 배선도나 비행기 날개의 곡면 모델, 혹은 피스톤 엔진의 솔리드 모델을 살펴보면, 고해상도의 이미지가 더 많은 정보를 주고 일반적으로는 보는 즐거움을 더해 준다. 워크스테이션급의 기계에서는 1,280×1,024의 해상도가 표준으로 등장했다. Windows 환경의 PC에서는 여전히 많은 작업들이 1,024×768의 해상도에서 이루어진다. 더 낮은 해상도에서 작업을 할 수 없는 것은 아니지만 해상도가 높을수록 더 좋은 결과를 얻는 것이다.

### ◎ 깊이와 정밀도

화소를 정의하기 위해 이용되는 정보의 양을 종종 화소의 깊이라고 부른다. 가장 일반적인 깊이의 크기는 8비트, 16비트, 24비트이다. 깊이는 또한 플레인 인의 수로서 표현되기도 한다. 그것들을 비트나 플레인 중 어느 것으로 부르든지 그 수가 증가함에 따라 컴퓨터로 표현되는 이미지의 정밀성이 높아진다.

8개의 플레인이라면 화면상의 각 화소를 정의하는데 2<sup>8</sup>(256)개의 다른 값을 사용할 수 있다. 마찬가지로 16비트라면 2<sup>16</sup>(65,536)개를, 24비트라면 1670만 개 이상의 다른 값을 가질 수 있다.

필요로 하는 깊이는 얼마나 될까? 그것은 이용하고자 하는 대상에 따라 다르다.

- “Alpha” 채널에서는 총 8비트만으로도 충분하다.
- Overlay buffer에서 풀다운 메뉴를 렌더링하려면 8비트면 적당하다.
- 색복사를 위한 메모리의 양은 응용과 사용자의 개인적 기호에 따라 달라진다.
- Z-buffer에 있어서는 최소 16비트를 추천한다. 24비트면 대부분의 환경에서 안전하다.

### ◎ 색

화소당 24비트의 색깊이에서, 아티스트는 16,777,216개의 색상 중에서 임의로 선택할 수 있다. 그는 이런 팔레트를 가지고서 진실한 사랑을 담은 얼굴을 정교하게 표현할 수 있을 것이다. 그러나 8비

트를 가지고서는 단지 256가지의 다른 색을 가질 수 있다. 문제는 우리가 수천가지 이상이 색을 구별할 수 있느냐 하는 것이다. 관찰에 의하면 65,536가지의 다른 색을 표현할 수 있는 16비트의 색이면 충분하다.

수천 혹은 수백만 색을 나타내기 위해서, 컴퓨터는 소위 RGB값으로 불리는 빨강, 녹색, 파랑의 원색들을 혼합한다. 24비트의 색에서, 컴퓨터는 각 화소의 RGB 요소 각각에 대해서 8비트의 데이터를 할당한다. 즉 주어진 화소에서 빨강, 녹색, 파랑이 각각 갖는 2<sup>8</sup>(256)가지의 다른 강도의 색들을 혼합한다. 16비트의 색은 RGB 요소 각각에 5비트를 할당할 수 있다. 즉 빨강, 녹색, 파랑 각각이 2<sup>5</sup>(32)가지의 명암들을 갖는 것이다. 인간의 눈은 32종의 명암을 쉽게 식별할 수 있다. 각 원색을 이용해 32가지의 명암만으로 표현된 사과에는 줄무늬나 얼룩이 나타날 수도 있다. 8비트나 16비트의 색이 기계적인 구조를 표현하기에 충분한 반면 포토리얼리즘은 24비트의 색보다 작아서는 안된다.

트루칼라보다 적은 색으로 표현할 때 나타나는 다른 문제로는 이미지에 왜곡이 생길 수 있다는 것이다. 컴퓨터는 완전한 무지개색을 선택할 수 없을 경우 color lookup table(CLUT)에 의존할 것이다. 즉, CLUT에서 응용에 요구되는 실제 색과 가장 가까운 색을 찾는다. 더 많은 색들은 “dithering”이라는 기법을 통해 표현될 수 있다. Dithering된 이미지는 컴퓨터가 서로 이웃하는 화소에 다른 색을 배치하면 눈으로 볼 때 화면에 실제 나타나지 않은 색으로 서로 섞여서 보인다는 것이다. Hewlett-Packard는 dithering 알고리즘과 압축회로를 결합한 “색 복구 기법”을 이용하여 8비트의 프레임 버퍼 내에서 8백만가지의 색을 충실하게 재표현할 수 있다고 한다.

Intergraph 사의 Steve Pesto에 의하면 dithering은 색의 해상도를 위해 공간 해상도를 사용한다고 한다. 단일 화소가 컴퓨터의 팔레트에 없는 색을 가져야 한다고 가정해보자. 적절한 색을 표현하기 위해 dithering 디스플레이 제어기는 인접 화소의 색을 변경시킨다. 결국 dithering은 한 화소의 색이 하나 이상의 화소에 의해서 표현되기 때문에 디스플레이에 있어 효과적인 해상도를 나타낸다. 이러한 트릭은 실제에 가까운 색을 사용하는 것보다 덜 깔끔한 이미지를 만든다.

결국 화면에 표현하고자하는 색의 수는 주관적 선

택의 문제이다. 기계부품을 디자인하거나 과학데이터를 분석하는 것은 8비트의 색이면 충분하고 데미무어의 모습을 랜더링하려면 24비트 정도는 되어야 한다.

### ◎ Z 버퍼의 정밀도

색이 종종 기호의 문제인 반면 Z 버퍼의 정밀도의 필요성은 훨씬 명확하다. Z축이 충분히 정밀하지 못하다면 모델은 깨져보일 수도 있다.

넓은 범위의 두께를 갖는 모델을 작업하고 있다고 상상해보자. 얇은 부분에서 정밀도가 낮은 Z-buffer를 가지고서는 화소의 앞의 것과 뒤의 것이 동일한 Z축 값을 갖는다. 그때 꼭면 뒷부분의 화소들 중 몇 개가 비칠수도 있다. 최악의 경우에는 모델이 깨져보일 수도 있을 것이다.

일부 응용들은 24비트 이상의 정밀도를 요구한다. 만약 얇은 용기들과 몇 에이커에 걸쳐 펼쳐진 파이프들로 이루어진 거대한 공장 배치를 랜더링하고자 한다면 거의 43억가지를 구별할 수 있는 32비트 Z-buffer가 필요할 것이다. 물론 이런 능력은 대부분의 CAD 사용자에게는 불필요하게 많은 것이다. 하지만 32비트의 Z-buffer를 필요로 한다면 구입한 그래픽스 카드가 그것을 달성할 수 있는지 확인해야만 한다. 모든 가속기들이 32비트 Z-buffer를 사용할 수 있는 메모리 제어를 갖춘 것은 아니다.

### ◎ 필요한 메모리의 계산

공통의 화소 길이는 바이트 크기로 된 데이터의 배수를 나타낸다. 8비트는 1바이트이고 16비트는 2바이트, 그리고 24비트는 3바이트이다. 색이나 Z축의 정밀도 등 요구되는 각 특징들에 대해서 필요한 화소당 길이를 바이트 단위로 더한 후 이 값과 화소의 개수를 곱한다. 메가바이트 단위로 답을 얻기 위해서  $1,048,576(2^{20})$ 으로 나누어주면 메가바이트 단위로 된 정확한 바이트수가 된다.

8비트의 색으로 고해상도의 이미지를 디스플레이 하기 위해 필요한 메모리 양을 얻기 위해서는 다음과 같은 계산을 한다.  $1 \text{ byte} \times 1,280 \text{ pixels} \times 1,024 \text{ pixels} = 1,310,720 \text{ byte} = 1.25 \text{ MB}$ . 16비트 색의 프레임 버퍼에 있어서는 두배의 메모리가 필요하고, 24비트의 고해상도 칼라 이미지를 만들기 위해서는 3.75 MB 이상이 필요하다.

Z 버퍼를 위해 필요한 여분의 메모리를 계산하는데도 동일한 식을 이용한다. 즉, 화소의 수에 바이트 단위의 버퍼 깊이를 곱한다.  $1,280 \times 1,024 \text{ pixels}$ 의 해상도를 얻기 위해서는 8비트 Z 버퍼로는 1.25 MB, 16비트의 Z 버퍼로는 2.5 MB 그리고 24비트의 Z 버퍼로는 3.75 MB가 필요하다.

불행하게도 버퍼들 각각의 모든 메모리 요구량을 단순히 더하여 필요한 총량을 산출할 수 없다. 몇몇 버퍼들은 더블 버퍼링을 위해 두배의 메모리를 필요로 하기 때문이다.

디스플레이 제거에 의해 임혀지는 버퍼들은 3차원 그래픽스에서 두배가 필요하다. 단지 이미지를 구성하는데 사용되는 버퍼들은 두배로 할 필요는 없다. 구체적으로 말하자면 프레임 버퍼 그 자체는 두배이어야 하고, 오버레이 버퍼도 마찬가지다. 반면 Z나 알파 버퍼는 오직 이미지를 구성하는데만 이용되므로 두배일 필요는 없다. 두배로된 버퍼들은 동일한 해상도와 색깊이에서 동작해야 한다. 그러나 이미지 구성에만 이용되는 버퍼들은 화소 길이를 임의의 다른 메모리 영역에서 이용되는 것들과 매치시킬 필요는 없다.

만약 더블 버퍼를 살 여유가 없다면 색깊이를 반으로 함으로써 부드러운 움직임 볼 수 있다. 예를 들면, 4 MB의 그래픽스 메모리는 24비트 칼라 1,280×1,024 해상도를 갖는 단일 프레임 버퍼나 12칼라의 플레인을 가진 동일한 해상도의 더블 버퍼에 충분하다. 물론 이는 그래픽스 카드가 이러한 종류의 버퍼와 해상도를 걸충한다고 가정한 것이다. 모든 카드가 12비트의 더블 버퍼 모드에서 동작할 수 있는 것은 아니다.

제품을 검토할 때 하고자하는 모든 것을 동시에 할 수 있는지를 확인해야 한다. 종종 광고를 통해서 여러 제품들이 트루칼라의 고해상도 이미지들을 랜더링하고 회전할 수 있다는 것을 볼 수 있을 것이다. 그러나 광고에서 본 가격만으로는 고정밀도의 Z 버퍼, 더블 버퍼, 트루칼라 깊이를 위한 충분한 메모리를 구입 못할 수도 있다. 한번에 하고자하는 모든 것을 위한 충분한 여유 메모리를 살 수 없을 경우에는, 요구사항들을 줄이거나, 더 고가의 그래픽스 가속기를 위해 예산을 늘려야 할 것이다.

### ◎ 텍스처 매핑 하드웨어

텍스처 맵은 CAD 모델이 더욱 현실적으로 보이도록 나뭇결, 금속광택, 나뭇잎 등과 같은 재료의 질감을 나타내는데 이용된다. 텍스처 맵은 모델의 적절한 화소 위에 매핑되는 화소크기의 텍스처 요소들("texels")로 구성된다.

SGI 사는 텍스처 매핑을 위하여 가장 강력히 추진하고 있다. 지난해 SGI 사는 desktop Indigo IMPACT 워크스테이션으로 CAD에 텍스처 매핑을 도입하려는 시도를 했다. HP, Intergraph 그리고 Digital Equipment 사도 SGI 사를 따라 텍스처 매핑을 추진하고 있다. 그러나 텍스처 매핑은 아직 워크스테이션의 주요 시장을 장악하지 못했다.

만약 텍스처 매핑에 투자하기로 결심한다면 다음 사항들을 행해야 한다.

첫째, CAD 응용이 그것을 지원하는지 확인하라. 그렇지 못하다면 텍스처 매핑을 지원할 수 있는 추가 소프트웨어와 그것의 비용을 알아봐야 한다.

둘째, 그래픽스 카드에 꽂을 수 있는 텍스처 메모리를 구입할 계획을 하라.

### ◎ 응용

그래픽스 관련 제품들을 비교하기 전에 여러분이 가지고 있는 CAD 소프트웨어의 그래픽적인 능력을 완전히 이해해야만 한다. 예를 들어보자. 3차원 그래픽스에서 최근 대유행하는 것이 모델들을 실물처럼 보이게 하는 텍스처 맵이다. 그러나 일부 CAD 응용들은 텍스처를 적용하도록 그래픽스 가속기에 지시할 수 없다. 이러한 경우에는 특별한 텍스처 프로세싱 하드웨어에 투자하는 것이 의미 없는 일이다.

여러분의 응용이 그래픽스 시스템과 상호 작용하는 방법은 고려해야 할 또 다른 문제이다. 응용프로그램들을 개발할 때 프로그래머들이 그래픽스 API(application programming interface)를 선택한다. API는 응용이 컴퓨터에게 여러 가지 그래픽스 작업들을 수행하도록 요구할 수 있게 하는 루틴들의 라이브러리로 간주할 수 있다. 그래픽스 하드웨어 구매자들은 자신들의 응용이 어떤 그래픽스 라이브러리를 사용하는지를 알아야 한다. 가장 보편적으로 이용되는 것들로는 PHIGS, PEX, Starbase, Sun사의 XGL, 실리콘 그래픽스사의 IRIS GL, Ithaca software사의 HOOPS와 HOOPS를 토대로 한 Heidi, OpenGL 등이 있다. SGI사의 OpenGL은 IRIS GL로부터 유래하여

현재 솔리드 모델링의 표준 API로서 대두되고 있다. 또 다른 인기 있는 3차원 API는 마이크로소프트사의 3D-DDI로서 이는 주로 게임 소프트웨어에 이용되는 그래픽스 라이브러리이다. 라이벌로서 RenderWare가 있다.

마이크로소프트사가 OpenGL을 Windows NT 운영체제의 일부로써 포함하고 있기 때문에 OpenGL primitives의 geometry 변환을 위한 많은 좋은 그래픽스 어댑터들이 현재 개발중에 있다. 그러나 이러한 사실이 그런 어댑터들 중 하나를 구입해서 사용해야 한다는 것을 의미하는 것은 아니다. 만약 응용이 OpenGL을 이용하지 않는다면 그런 하드웨어는 그래픽스를 가속화시키지 않을 것이다.

Autodesk의 소프트웨어는 HOOPS API나 Heidi를 사용한다. 이러한 프로그램들은 geometry 가속기를 최대한으로 이용할 수 없다. 따라서 Autodesk 그래픽스를 가속화시키기를 원한다면 보다 더 나은 레스터라이징 하드웨어나 더 빠른 메인 프로세서에 관심을 가져야 한다. 또한 그래픽스 가속기가 Heidi 드라이브와 잘 맞는지도 확인해야 한다. 3D Studio Max는 멀티프로세싱을 하기 때문에 마이크로프로세서를 더 추가시킴으로써 속도를 높일 수 있다. Autodesk사가 가장 최근에 발표한 Mechanical Desktop은 OpenGL 그래픽스 프로세서를 이용함으로써 세이딩 기능이 가속화될 수 있도록 한다.

PTC사의 Pro/Engineer와 같은 일부 소프트웨어들은 한 개 이상의 그래픽스 API를 사용할 수 있다. 따라서 응용이 여러 종류의 그래픽스 하드웨어를 이용할 수 있다. 다른 것들은 한개 내지 두개의 API를 사용한다. SolidWorks사는 처음에는 소프트웨어가 OpenGL과 RenderWare 두개 모두를 사용할 수 있게 했지만, 현재는 OpenGL만 사용한다. 이 회사에 따르면 Renderware 사의 모델 관리 시스템은 API로 하여금 하드웨어이 도움없이 빠른 솔리드 렌더링을 제공해 주도록 하지만 어셈블리 모델링을 하는 동안 메인 메모리를 너무 많이 차지한다고 한다. 만약 SolidWorks사의 소프트웨어를 구입하기 원한다면 좋은 OpenGL 가속기의 구입을 고려해야 한다.

### ◎ 드라이버

이 글에서 우리의 관심은 하드웨어에 있지만, 모든 그래픽스 부시스템들은 어떤 점에서는 소프트웨어

어 중속적이라는 것을 기억해야 한다. 심지어 그래픽스 프로세싱을 가속화시키는 시스템이라 할지라도 여전히 CPU와의 통신을 제어하기 위해서는 소프트웨어 드라이버에 의존해야 한다. 이것은 단지 학문적인 관점이 아니다. 드라이버에 쓰는 기술에 따라 성능에 큰 차이가 있을 수 있다. 예를 들어, Windows NT 시장에서 매우 성능이 좋은 OpenGL 드라이버는 Pro/Engineer, SolidWorks, 혹은 Solid Edge와 같은 응용에서 가속기를 가능한 최고 속도를 낼 수 있게 할 것이다. 가장 정교한 드라이버를 마이크로소프트 사에서는 "installable client drivers"라고 한다.

단지 효율성만으로 소프트웨어 장치 드라이버를 판단할 수 없다. Windows와 Unix 환경 모두에서 시스템에 손상을 주거나 성능을 저하시키는 드라이버들에 대한 보고서를 보는 것이 특별한 것은 아니다. Windows NT 플랫폼의 그래픽스 카드를 파는 사람은 다른 업체의 드라이버가 심한 손상을 주기 때문에 그래픽 보드의 설치 후 운영체제를 재 설치해야 할 것이라고 말한다. 여러분들은 제품을 사기전에 다른 사용자에게 사고자하는 드라이버가 가진 문제점의 유무를 물어보거나, 환불 보증서를 받아두어야 한다.

### ◎ 그래픽스 벤치마크

어떤 컴퓨터 하드웨어에 사용할 수 있는 가장 좋은 벤치마크는 사용하는 응용에 기초를 두고서 직접 개발하는 것이다. 불행하게도 직접 벤치마크를 디자인하고 운영하는데는 시간이 걸리는 일이다. 게다가 여러분이 수십 대의 워크스테이션을 사지 않는다면 하드웨어 판매인들은 벤치마크때문에 시간을 낭비하기를 원하지 않는다. 만약 워크스테이션을 몇 대만 살 계획이거나 직접 테스트하기 전에 테스트의 범위를 줄이기를 원한다면 기존의 벤치마크 자료가 도움이 될 것이다. 가장 믿을 만한 결과를 가진 테스트들은 표준기관에 의해 감독되어지는데 지금까지는 Standard Performance Evaluation Corporation(SPEC)의 것이 가장 좋다.

SPEC 레벨은 여러 CPU들의 성능을 비교하는 일련의 벤치마크들과 가장 많이 관련이 있다. 가장 잘 알려져 있는 것이 SPECint92와 SPECfp92 그리고 최근 버전인 SPECint95와 SPECfp95이다. SPEC은 Graphics Performance Evaluation Characterization Committee(GPC)를 흡수하면서 CPU 벤치마크뿐만

다른 여러 가지 벤치마크에도 관여하게 됐다. SPEC은 수많은 유용한 그래픽스 벤치마크를 상속했는데 Picture Level Benchmarks(PLBs), OpenGL Viewport Benchmarks 그리고 XMark X-windows system benchmark 들이 그것들이다.

이 모든 벤치마크의 결과는 한벌에 테스트에 대한 성능의 가중 기하 평균으로 보고되고 있다. 응용에 있어서, 가중치 요인들은 어떤 그래픽스 부시스템에 매우 중요할 수도 있고 다른 것들에는 하찮은 것일 수도 있다. 표준화된 벤치마크를 최대한 이용하기 위해서 SPEC사는 기하 평균에 연연하지 말고 개개의 테스트 결과를 주시하라고 추천하고 있다. 데이터에 압도되지 않기 위해 자신의 응용이 갖는 특징을 더 잘 반영하는 가중치 시스템을 직접 개발할 수도 있다.

SPEC의 결과에 덧붙여서 일부 컴퓨터 잡지들은 자체적인 벤치마크들을 만들어 내고 있다. 이들의 가장 큰 결점은 SPEC 테스트만큼 보편적으로 활용할 수 없다는 것이다. 게다가 하나의 응용을 이용한 테스트가 다른 응용으로 작업하는 하드웨어 성능의 척도로서 항상 좋은 것은 아니다. 예를 들면 AutoCAD 테스트 결과의 비교는 Microstation, CADAM, Cadra, 혹은 다른 제도 프로그램의 사용자에게 적당할 수도 있고 그렇지 못할 수도 있다.

산업표준이나 특정 응용에 대한 테스트의 부재로 인해 초당 클리핑 혹은 변환되는 벡터의 수나 1초에 세이딩 되는 삼각형의 수 등과 같은 primitives 벤치마크에 의존해야 할 것이다. 여기에서 우리는 주의를 기울일 필요가 있다. Primitives 벤치마크 결과를 만들어내는 방법 때문에 실제의 응용으로 작업하는 그래픽스 부시스템의 능력을 모의실험하기에 좋지 않다. 컴퓨터 하드웨어 판매업자들은 전형적으로 단지 벡터나 삼각형과 같은 primitives를 생성하기 위해 디자인된 프로그램을 실행시킬 것이다. Intergraph사의 Peter Karnazes는 primitives는 주어진 그래픽 엔진의 최고 속도에 관한 정보를 주지만 실제 응용으로 작업하는 실제의 컴퓨터 시스템에서 어떻게 작용할지는 거의 나타내지 못한다고 한다.

Primitives에 있어 또 다른 문제는 모든 판매인들이 primitives의 완전한 자료를 발표하는 것은 아니라는 것이다(예로서 IBM은 3차원 primitives 데이터를 발표하려하지 않는다). 또한 공개되는 자료들이

동일한 단위가 아니라는 문제도 있다. 예를 들어 대부분 Unix 판매업자들은 그들의 3차원 하드웨어가 1초에 셰이딩할 수 있는 50 pixels의 삼각형 수를 말하지만 최근에 SGI는 25 pixels의 삼각형 수로 속도를 나타내기 시작했다. 매우 많은 계산이 필요한 Gourand 셰이딩된 삼각형 대신에 plat으로 삼각형을 셰이딩 하는 속도를 보고하는 비열한 그래픽스 카드 생산자를 경계해야 한다.

### ◎ 2차원 벤치마크

2차원 그래픽스에 대해서는 PLB2d93으로 알려진 Picture Level Benchmark가 사용되었지만 불행히도 Picture Level Benchmark는 작년에 중단되었다. Windows와 Windows 95 시스템에서 작동하는 것들을 위해 판매인들은 일반적으로 PC Week의 출판자인 Ziff-Davis가 만든 WinMarks를 이용한다. 다행히도 오래 전에 많은 하드웨어 생산자들에 의해 2차원 그래픽스의 문제점들이 밝혀졌기 때문에 2차원 그래픽스 구매 시 에러의 가능성이 줄어들었고 비용도 저렴하다.

### ◎ 와이어프레임 벤치마크

Unix 워크스테이션에서 작동하는 것들에 대해서 와이어프레임 성능 평가를 하려고 한다면 아마도 PLBwire93(the picture level wireframe benchmark)이 최고의 척도일 것이다. Windows 시장에서는 초당 3차원 벡터들의 수를 이용해야 한다. 만약 이러한 자료를 사용할 수 없다면 2차원 벡터의 성능을 살펴봐야 할 것이다. 여기서는 GUI 가속기로서 어댑터의 능력이 중요하기 때문에 WinMark와 다른 PC 시장 벤치마크들이 성능의 척도로서 적당하다.

### ◎ 3차원 벤치마크

Digital Equipment, Hewlett-Packard, Sun Microsystems, IBM의 유닉스 시스템들을 비교한 자료로서는 SPEC의 PLBsurf93이 가장 뛰어난 3차원 벤치마크이다. 그러나 만약 SGI 기계나 Windows NT 플랫폼과 이 시스템들을 비교하려면 PLBs는 부적당하다.

SGI는 PLB 결과들을 사용하지 않는다. SGI에 따르면 자신들의 그래픽스가 OpenGL 기반인 반면 PLBs는 PHIGS 그래픽스 API에 기반을 두고 있기 때문이라고 한다. PHIGS는 OpenGL과 많이 다르다고 SGI의 대변인은 주장한다. SPEC은 최근 PLB 테

스트의 OpenGL 버전을 완성했다고 발표했지만 SGI는 여전히 그 테스트는 별로 의미가 없다고 주장한다. SGI는 최근 세계에서 가장 좋은 그래픽스 워크스테이션을 만든다는 명성에 즐거워한다. 따라서 SGI 워크스테이션을 배제하고서 그래픽스들을 비교하는 것은 별로 도움이 되지 않는다.

PLBs가 가지고 있는 다른 문제점은 단지 유닉스 벤치마크라는 것이다. 빠른 3차원 그래픽스를 갖춘 NT 시스템을 생산하는 업체들이 증가하는 현실에서 워크스테이션 구매자들은 Windows NT 플랫폼과 유닉스 컴퓨터 시스템들을 비교해보기를 원한다.

Viewperf라는 3차원 렌더링을 위한 크로스 플랫폼 벤치마크들의 새로운 출현 때문에 OpenGL 기반의 응용프로그램을 가지고 있는 구매자들은 운이 좋은 편이다. 우리의 관점에서 보면 Viewperf의 최대 장점은 PC 기반의 하드웨어를 만드는 업체에서부터 UNIX 업체까지 심지어 실리콘그래픽스까지도 포함한 많은 업체들이 이용한다는 것이다. HP는 Viewperf에 대해 약간의 불평을 하기도 하지만 자신들의 그래픽스를 OpenGL과 호환 가능하게 만들것이라는데 동의함으로써 동일한 테스트를 이용하여 주요 워크스테이션 업체들을 비교한 결과를 볼 수 있게 될 것이다.

현재, Viewperf 결과는 다섯 가지의 분리된 벤치마크 혹은 다섯 개의 다른 응용에 기초를 둔 "Viewset"으로써 나타난다.

- Advanced Visualizer, SGI의 Alias/Wavefront division의 3차원 애니메이션 제품
- CDRS, Evans & Sutherland가 만들고 Parametric Technology가 판매하는 모델링 및 렌더링 소프트웨어
- Data Explorer, 과학 데이터의 시각화와 분석을 위한 IBM의 패키지
- Design Review, Intergraph사의 제품으로 3차원 공장 배치와 시각화를 위해 디자인된 프로그램
- The Lightscape Visualization System, Lightscape Technologies, Inc., 전체 조명 기법을 이용

응용프로그램들에 대해서 정확한 viewset을 매치시켜야 한다. 솔리드 모델링 프로그램을 가지고 작업하는 기계 엔지니어를 위해서는 CDRS Viewset이 가장 적당하다고 믿는다. 그러나 건축용 CAD 사용자뿐만 아니라 훨씬 복잡한 어셈블리를 디자인하는 사람은 Design Review와 Lightscape viewset의 성능

을 관찰해 봐야 한다.

CDRS viewset은 삼각메쉬를 이용해 만든 뒤 세이딩하고 다시 텍스처 매핑을 하는 과정을 포함한 3차원 모델을 렌더링하고 회전하는데 포함된 7가지 작업으로 구성된다. 각 작업의 결과는 그래픽스 속도의 기본 척도인 컴퓨터가 생성할 수 있는 초당 프레임의 수로 나타난다.

Viewperf에 대한 비평도 있다. 그래픽스 어댑터 업체인 Dynamic Pictures의 Dinesh Sharma는 CDRS viewset이 품질 때문에 적절하지 못하다고 평가한다. 한가지 예가 안티앨리어싱(anti-aliasing)을 처리하는 방식이다. 안티앨리어싱은 인접한 픽셀들의 중간색을 보간해서 지그재그로 된 에지(edge)들이 배경에 잘 섞이도록 한다. Sharmas는 CDRS viewset에서 OpenGL을 사용하는 것은 수행되고 있는 안티앨리어싱의 품질에 대해 응용이 고려하지 않는다는 것을 의미한다고 한다. 이를 아는 프로그래머들은 더욱 거친 선을 디스플레이 하는 간단한 하드웨어적 반응을 통해 Viewset의 작동들을 최적화할 것이라고 한다. 이와 같은 비평은 표준화된 벤치마크들 중 최고의 것이 시작점 역할을 해야한다는 점을 강조한다.

Pro/Engineer 사용자들은 Texas Instruments의 디자인 엔지니어인 Paul Gilliland와 Steve Levin이 구성한 새로운 벤치마크를 원할 수도 있다. 수차례의 벤치마킹 결과가 Pro/E: The Magazine에 실렸다. 그 벤치마크 자체가 개인 작업환경을 더 좋게 맞추기 위해 웨이팅 시스템을 수정할 수 있는 엔지니어에게 유용하다. 그 벤치마크 파일을 웹사이트 <http://www.proe.com>에서 다운로드 받을 수 있다.

### ◎ 3차원 Primitives

그래픽스 부시스템이 일초에 생성할 수 있는 삼각형의 수나 벡터의 수 같은 primitives 데이터를 가지고서 평가를 보충해보려고 할 수도 있다. 이름에 관계없이 "3차원 벡터"가 솔리드 그래픽스에 있어서 그렇게 좋은 척도가 아니라는 것을 명심해야 한다. 그 이유는 솔리드 모델링의 그래픽스가 다각형이기 때문이다. 세이딩된 솔리드에 있어서, 그래픽스 부시스템은 벡터나 선이 아닌 삼각형이나 사각형을 이용해 고려한다.

3개의 primitives가 3차원의 세이딩된 그래픽스 하드웨어의 성능 척도로서 공동적으로 이용된다. 한

메쉬로 연결된 25-pixel 삼각형, 한 메쉬로 연결된 50-pixel 삼각형, 독립적인 50-pixel 사각형.

중요한 것은 동일한 것을 비교해야 한다는 것이다. Gouraud 세이딩된 삼각형으로 된 그림을 측정하는 가속기의 성능과 주어진 수의 플랫폼 세이딩된 삼각형을 만들 수 있는 가속기의 성능을 비교하는 것은 곤란하다. 왜냐하면 Gouraud 세이딩은 플랫폼 세이딩보다 훨씬 많은 계산을 필요로 하기 때문이다. 독립 다각형은 메쉬에 얽힌 다각형보다 생성에 많은 시간이 걸린다. 각 독립적인 다각형에 대해 컴퓨터는 각 꼭지점과 변을 위치시키고, 빛을 주고, 세이딩하고, 그려야 한다. 그러나 메쉬에 있는 다각형들은 꼭지점들과 변들을 공유한다.

3차원 데이터가 필요할 때 Z 버퍼링, 더블 버러링되고 랜덤하게 빛이 주어진 primitives을 이용해 보라. 이 방법은 컴퓨터가 화면을 변환하고 라이팅, 세이딩, 렌더링하는 모든 작업들을 다루는데 더 좋은 아이디어를 얻을 수 있을 것이다.

만약 모델이 실물처럼 보이도록 텍스처를 적용하기 위해 그래픽스 부시스템을 이용하고자 한다면 컴퓨터가 초당 생성할 수 있는 텍스처링된 삼각형의 수와 시스템이 나타낼 수 있는 텍스처링된 화소의 수를 알 수 있으면 좋다. 그러한 데이터를 모을 때 그 화소들이 동일한 색 깊이를 갖는지 삼각형의 경우에 면적이 전부 동일한지를 확인하려고 노력해야 한다.

Xmark, PLB, Viewperf 데이터를 얻기 위해서는 SPEC의 Graphics Performance Committee와 연락해야 한다. Standard Performance Evaluation Corporation의 WWW 홈페이지(<http://www.specbench.org>)를 통하면 된다. 웹을 이용하기를 원치 않는 사람은 다음으로 연락할 수 있다. SPEC at 10754 Ambassador Drive, Suite 201, Mnassas, Virginia 22110. Telephone: (703) 331-0180, Fax: (703) 331-0181

-----  
«CAD Report Vol. 16, No. 10, October 1996»  
«CAD Report Vol. 16, No. 12, December 1996»  
.....

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

- Fax: 1-619-488-6052
- E-mail: Cadcirc@aol.com