

LOD(Level of Detail)를 지원하는 하이브리드 렌더링 모델

김학란*, 박화진**

요약

컴퓨터 그래픽의 다중 해상도를 지원하는 하이브리드 렌더링 방법을 제안한다. 기본적으로 단말기 환경의 성능과 사용자 요구조건에 따른 그래픽을 위한 다중해상도 방법은 메시지를 이용하거나 렌더링 부분에서 텍스처의 밍매핑이나 옥트리를 이용한 레이 트레이싱들의 적용 방법이 사용되었다. 본 연구에서는 제안한 하이브리드 렌더링 방법은 지역조명 모델에서 기존의 고로 음영과 평면 음영 라이브러리를 개선한 방법으로 하나의 물체를 이루는 여러 개의 폴리곤에 각각 다른 음영법을 적절하게 적용한다. 실시간 렌더링 시간을 줄일 수 있는 효율적인 대안 방법이 될 수 있으며 이러한 장점이 유비쿼터스 환경에서 다양한 단말기 환경의 그래픽 콘텐츠의 실시간 적용 서비스에 매우 적절하게 사용될 수 있다.

A Hybrid Rendering Model to support LOD(Level of Detail)

Hakran Kim*, Hwajin Park**

Abstract

We propose the Hybrid Rendering model to support multi-resolution for computer graphics. LOD method for computer graphics system considering performance of device environment and end-user preference usually adopts mesh resolution, mipmap in texture rendering, or oct-tree structure in ray tracing. The hybrid rendering model, as a local shading model combining Gouraud shading model and a flat shading model, applies a proper shading method to each of polygons consisting of an object. This method can be an effective alternative to reduce real-time rendering time so that it can be utilized in real time adaptive service of computer graphic contents among various device environments under ubiquitous environments.

Keyword : hybrid rendering model, LOD(Level of Detail), multi-resolution, adaptive service

1. 서론

최근 단말기에 서비스 되고 있는 콘텐츠 중에는 간단한 그래픽인 아바타나 스크린 세이버 등과 복잡한 계산을 요구하는 게임, 애니메이션, 광고, 가상현실 등의 컴퓨터 그래픽의 사용 및 응용이 폭발적인 인기를 누리고 있으며 향후 이러한 추세는 더욱 증가할 것으로 보인다. 따라서 그래픽 콘텐츠를 이용하고 있는 사용자 단말기의 환경 분석을 통한 실시간 그래픽 콘텐츠 서비스에 대한 연구 또한 더 많은 관심을 받게 될

것이다.

그래픽 콘텐츠는 프로그램 특성상 콘텐츠를 생성하는데 많은 계산 시간을 필요로 하므로 일반적으로 고사양의 시스템을 요구한다. 그러나 최근에는 다양한 모바일 기기의 등장으로 여러 가지 성능을 가지는 단말기의 종류가 출시되었다. 모바일 기기인 휴대폰이나 PDA 등에서도 그래픽 콘텐츠 서비스가 많이 사용되고 있는데 이러한 단말기의 성능은 데스크 탑이나 워크스테이션 등의 고 사양에 비하면 매우 낮다. 따라서 다양한 단말기 성능을 고려한 그래픽 콘텐츠 서비스를 위해서는 시간이 많이 걸리는 그래픽 콘텐츠를 경량화 시키거나 저 성능의 단말기에서 고성능의 단말기로 서비스가 이동되었을 경우 다시 질적으로 향상된 그래픽 콘텐츠를 제공하는 LOD 방법에 대한 연구가 필요하다.

과거 그래픽 분야에서의 LOD는 주로 동일한

※제일저자(First Author): 김학란
접수일자:2008년08월30일, 심사완료:2008년09월18일
*숙명여자대학교,
imhera@sm.ac.kr
**숙명여자대학교

단말기 환경에서 실시간 렌더링 시간을 줄이기 위해 제안된 방법들로 다중해상도를 지원하였다 [2,3,5]. 가장 보편적인 방법으로 메시의 수를 단계에 맞게 줄이는 기법이 연구되었고 렌더링 분야에서는 텍스처맵핑에서 밍매핑을 이용한 LOD를 적용하거나 실사와 같은 장면을 표현하기 위하여 너무 많은 계산시간을 필요로 하는 광선추적법과 같은 방법에서 옥트리나 BSP를 이용한 LOD를 사용하고 있다[1]. 하지만 메시지를 다단계로 표현하는 방법 이외에 렌더링 부분에서 기존의 제시한 방법들은 휴대폰 같은 저성능 단말기에는 적용하기 어려운 실정이다. 따라서 기존의 LOD개념을 확장하여 렌더링에서 지역조명모델의 음영법을 개선한 새로운 하이브리드 렌더링 방법을 제안하였다.

2. 기존연구

복잡한 대량의 자료를 빠른 시간 내에 처리하기 위한 기법 중의 하나인 LOD의 기본 개념은 모델의 모양을 최대한 유지하면서, 물체를 간략하게 표현하여 기하학, 위상정보를 줄이는 방법이다. 자료를 그 중요도에 따라 구분하여 중요도가 낮은 물체의 경우 낮은 정밀도로 처리함으로써 전체적인 시스템 자원이 갖는 제약을 만족시키고자 하는 접근이다. 모델링 중 메시의 수를 조절하는 방법과 렌더링 시 적용하는 방법으로 나눌 수 있다. 그래픽의 오브젝트 렌더링 속도를 향상시키기 위한 목적으로 주로 사용되던 기존의 LOD는 그 개념을 확장하여 다양한 영역에서 사용되고 있다. 예를 들어 게임에서 캐릭터의 움직임 계산할 때도 카메라에서 멀어지는 경우와 카메라 가까이 있는 경우 구체적이고 다양한 움직임의 정도를 레벨을 두고 계산하는 방식이다. LOD를 위한 기존의 방법은 다음과 같다.

2.1 다중 해상도 메시

복잡하고 정교한 모델을 위한 양질의 이미지 생성과 실시간이나 상호 작용하는 물체의 빠른 렌더링을 위한 메시의 상세도를 낮춘 다중 해상도 표현 연구는 LOD와 관련된 연구의 큰 흐름으로 대표될 수 있다. 다양한 단말기 성능을 가지는 사용자 환경일 경우 적용 레벨에 따라 적

정한 수준이 결정되면 단말기 성능에 맞는 수의 메시지를 사용하는 LOD개념을 사용한다. <표 1>은 메시지를 줄이는 방법에 대한 기존연구의 요약이다[4].

<표 1> 메시지를 사용한 다중 해상도 방법

방법	요약
Image Pyramids	현재 컴퓨터그래픽에서 레스터 이미지를 위해 가장 일반적으로 사용하고 있는 방법으로 가장 간단한 방법이다.
Volume Methods	Image Pyramid를 일반화한 방법으로 Volume Pyramids를 제안했다. 볼륨데이터를 받아서 간단한 볼륨을 렌더링할 때 효율적인 방법이지만 간단한 볼륨 데이터의 경우 렌더링하기 전에 폴리곤 형태로 전환되어 있어야 하는 단점이 있다.
Vertex Decimation	반복적인 표면 단순화 방법으로 인접한 정점을 모든 표면에서 제거해나가는 방법이다. 질적으로 우수하다. 위상정보를 그대로 보존한다. 다만 다면체 물체를 만들 대상으로 한다.
Vertex Clustering	입력된 기하에 대해 매우 빠른 가정을 만들어 내는 간단한 방법으로 객체를 바운딩 박스를 나누고 셀 안에 있는 모든 정점을 하나의 정점으로 간주하는 방법이다. 빠르고 일반적인 방법이지만 제어하기가 어렵고 질적으로 우수하지 못하다.
Edge Contraction	가장 많은 제안 논문이 나온 방법으로 하나의 에지를 선택하고 그 양 끝점 중 하나의 정점을 제거하는 방법이다. 부드러운 이동 처리가 장점이다. 질적으로 우수하지만 모델의 에지를 축소해 나가는 방법을 반복적으로 사용하게 되는데 어떤 에지를 선택하여 축소하는가 하는데 따라서 효율성이 달라진다.
Simplification Envelopes	원래의 결면과 안쪽 면의 오프셋을 이용해서 근사하는 방법이다. 전역적인 오류를 보증한다. 위상정보를 그대로 보존하지만 다면체 객체만 가능하고 구성(생성)하기

	가 어렵다
Wavelet Surfaces	wavelet 을 사용하여 표면을 다시 구성하는 것. 실질적인 다중해상도 방법으로 부드러운 다면체 물체만을 대상으로 하고 위상정보를 바꿀 수 없다.

2.2 렌더링

렌더링에서 이용되고 있는 기존의 LOD방법도 메시지를 이용한 방법만큼 다양하다.

렌더링의 기본은 물체의 사실적 표현 방법으로 주로 조명모델을 설정하고 조명이 물체에 미치는 영향을 계산하여 표면의 색상을 결정하게 된다. 하지만 실시간/비 실시간으로 렌더링 시간을 감소시키기 위한 방법으로 텍스처링 중에서 밍매핑을 사용한 방법과 전역 조명모델에서의 옥트리나 BSP등의 적용기법을 사용한 렌더링 방법이 LOD렌더링 방법으로 사용되고 있다. 켈링도 하나의 방법으로 제시되고 있다.

복잡한 환경의 빠른 렌더링을 위해 기여도 켈링 방법을 제안한 논문에서는 이미지 공간에서 객체의 바운딩 볼륨의 넓이가 매우 작은 것들은 장면에서 기여도가 떨어진다는 점을 이용하여 기여도가 낮은 객체를 켈링 함으로써 장면의 질을 보장하고 렌더링 속도를 향상시킬 수 있는 방법을 제안하였다[13].

거리에 따라서 텍스처의 해상도를 조절하는 방법인 밍매핑은 텍스처를 이용한 렌더링에서의 대표적인 LOD방법이다. 게임을 개발하기 위한 3D엔진 등에서 주로 사용하는 밍매핑 방법은 가장 높은 해상도가 제공되어지면 낮은 해상도의 텍스처는 길이 단위로 1/2씩 축소하는 방식으로 자동으로 만들어지는데 텍스처 크기의 결정은 영상의 질과 렌더링 성능사이에서 신중하게 저울질 해야 하는 최적화의 중요한 이슈로 다양한 접근 방법이 있다[8].

밍매핑을 이용한 텍스처 매핑에서의 LOD방법을 제안한 논문들을 보면 밍매핑 이미지 생성 부분에서의 새로운 방법을 제안하기 보다는 실시간 그래픽 렌더링 시간에 선택을 효율적으로 하기 위한 연구들이 많다. 기하 이미지 밍매핑을 위한 동적 LOD방법으로 원-패스 셰이더를 이용한 밍매핑 이미지 선택과 켈링을 실행하는 것을 제안한 방법도 그런 연구 중의 하나이다. 이 방법은 스캔 받은 이미지를 기하이미지 밍매핑으로 삼고 밍매핑

선택알고리즘을 이용하여 카메라 위치라든지 맵핑할 기하이미지를 생성시키는 방법으로 특별한 자료구조를 요구하지 않는 점이 부각되고 있다. 하지만 기존의 밍매핑의 개념을 그대로 두고 텍스처 이미지의 선택에 중점을 둔 제안 방법이다 [11].

또 다른 방법으로 텍스처 맵핑을 이용한 LOD 방법은 동적인 장면을 위한 렌더링 접근 에 기초한 하이브리드 LOD방법이다. 모든 객체의 다중해상도를 미리 생성시켜 놓은 후 실행시간에 적합하고 효율적이면 이미지를 찾아서 텍스처 맵핑을 적용시키고 아니면 다중해상도에 해당하는 객체를 원래대로 렌더링 하는 방법이다[12].

전역 조명 모델은 렌더링 과정에서 오브젝트 상호간의 관계도 모두 고려하기 때문에 오브젝트들 간의 상호 반사(interreflection), 굴절, 그림자 효과 등을 재현할 수 있어 훨씬 사실적인 이미지를 생성할 수 있으나 엄청난 렌더링 시간을 요구하기 때문에 실시간으로 계산할 수 없는 단점을 가지고 있다[7]. 렌더링 시간의 부담을 줄이기 위한 LOD기법의 하나로 BSP나 옥트리를 기본적으로 사용한다.

레이 트레이싱을 좀 더 효율적으로 계산하기 위해 공간상에 존재하는 물체를 파악하여 8개의 서브공간으로 나누어 나가는 공간색인 기법인 옥트리는 광선의 교차점의 수만큼만 공간을 다시 나누는 옥트리-R[9], 계층적 자료구조에서 볼륨 레이의 재방문의 단점을 개선하는[10] 등의 다양한 적용 방법들이 연구되어 왔다.

3. 하이브리드 렌더링

앞장에서 제시한 모델링이나 렌더링 방법에서의 LOD보다 좀 더 간단하며 최근에 많은 관심을 받고 있는 유비쿼터스 환경하에서 사용되고 있는 다양한 단말기 성능에서도 적용 가능한 형태의 렌더링 방법이 필요하다.

하드웨어성능이 뛰어나고 고사양의 그래픽 카드를 가진 고성능의 단말기에서의 렌더링은 풍음영법도 실시간으로 처리 가능하지만 아직까지는 모바일 단말기 환경 등에서는 그래픽 콘텐츠를 실시간으로 렌더링 하려고 할 때 폴리곤의 음영을 계산하기 위한 시간 때문에 실시간 애니

메이션이나 게임 같은 그래픽 콘텐츠의 경우 실행율이 떨어지게 되고 사용자를 만족시키지 못하는 서비스가 제공 될 수 밖에 없다. 이러한 문제점을 해결하기 위해서는 저 성능 사용자 환경에서는 렌더링 시간을 줄이는 방법이 필요하다. 기존의 그래픽에서의 LOD는 2장에서 언급한 것처럼 모델링이나 전역 조명모델의 렌더링에서는 여러 가지 방법이 적용되지만 지역조명 모델을 이용한 렌더링 분야에서는 현재 LOD방법이 시도되고 있지 않다. 따라서 지역조명모델의 LOD 렌더링 방법을 제안한다.

지역 조명 모델은 빛의 작용을 계산함에 있어 사용자가 관심을 갖는 특정 표면과 이 표면을 직접 비추는 광원만을 고려하는 방식으로 광원, 표면, 시점간의 관계를 고려해서 난반사 및 정반사되는 빛의 세기를 계산하여 물체 표면의 색상을 구하는 모델로 많은 모델들이 개발되어 있으나 실시간 렌더링을 위해서는 가장 간단한 플랫폼 음영법이나 플랫폼 음영법보다는 계산시간을 더 요구하지만 사실적 표현에 좀 더 적합한 고로 음영이나 뿔 음영법이 사용된다.

실시간 렌더링에서 주로 사용되는 평면 음영이나 고로 음영은 각각의 라이브러리를 하나의 물체나 장면 전체에 적용시키는 방법으로 물체를 이루는 폴리곤에 각각의 라이브러리를 혼합하여 적용할 수 없는 단점이 있다. 따라서 본 논문에서는 하나의 물체에 혼합 형태의 음영법을 적용하기 위한 방법을 제안한다.

지역조명모델 계산에서 렌더링 시간을 줄이는 가장 간단한 방법은 평면 음영을 사용하면 되지만 그래픽의 질이 떨어지게 되어 부드럽지 못한 그래픽 콘텐츠를 생성하게 된다. 객체의 질이 너무 떨어지는 경우에는 계산 시간을 더 요구하지만 고로 음영을 사용한 렌더링을 적용하면 물체의 표현질은 좋아진다. 따라서 제안한 하이브리드 렌더링은 렌더링 시간과 단말기 성능 사이를 고려하여 하나의 물체에서도 평면 음영을 적용하는 폴리곤과 고로 음영을 적용하는 폴리곤을 결정하여 표면의 색상 값을 결정하는 방법이다. 이런 방법을 적용하게 되면 저성능 단말기의 경우 모두 고로 음영을 사용하여 처리하는 폴리곤 수 보다 더 많은 수의 폴리곤을 처리 가능하게 된다. 즉, 고로 음영으로 처리할 최대 처리 폴리곤 수보다 적은 수의 폴리곤을 본 논문에서 제

시하는 선택방법에 의하여 비율을 결정하여 처리하고 나머지는 평면음영으로 처리하면 일반적으로 각 단말기 성능을 고려하여 처리할 수 있는 최적의 폴리곤 수 보다 더 많은 수의 폴리곤 처리가 가능하면 빠른 렌더링을 보장 할 수 있다.

단말기 성능에 맞는 최적화된 렌더링 폴리곤 수가 결정되어지면 기준에 따라서 두 가지의 렌더링 방법 중 하나를 선택하여 처리하기 위해서는 음영법을 선택하기 위한 조건이 필요하다. 적용 되어지는 음영법의 선택에서 고려해야 할 사항은 세 가지로 먼저 카메라와 객체의 위치에 따른 거리를 몇 개의 단계로 세분화 시킨다. 그다음은 폴리곤을 구성하는 각 정점에서의 인텐시티를 고려하여야 한다. 마지막으로 폴리곤의 크기를 고려하였다.

세가지 기준에 대한 연산은 다음과 같은 세부적인 방법을 사용한다.

- 카메라 위치와 객체와의 거리: 카메라에서 가장 멀리 떨어진 객체는 모든 폴리곤에 평면음영을 적용하며 가장 가까운 곳의 객체는 고로 음영을 적용한다. 중간 거리에 해당하는 객체는 카메라에서 가까운 곳의 구간일수록 고로 음영을 적용하는 폴리곤의 비율을 더 높이고 멀어질수록 평면 음영의 비율을 높인다. 중간거리에 해당하는 단계는 단말기 성능에 따라 많은 단계와 적은 단계를 적용시킬 수 있다.
- 하나의 객체를 구성하는 전체 폴리곤의 수를 n 이라고하면고로음영을적용할폴리곤의수 g 와평면음영을적용할폴리곤의수 f 는선형 1차 방정식 형태로 간단히 구해질 수 있다. 카메라에서 가까운 곳에서부터 먼 곳을 나타낼 값을 d 라고설정하면수식은다음과같다.

$$n=(1-d)g+d\cdot f \quad \text{단, } 0\leq d\leq 1$$

- 폴리곤의 크기: 폴리곤을 이루는 세 개의 에지의 길이의 합을 구한다. 즉, 장면에서 하나의 오브젝트를 구성하는 폴리곤의 크기를 구하기 위해서는 폴리곤의 면적은 에

지의 크기에 비례하므로 연산횟수를 줄이고 간단히 하기 위해 에지의 길이를 사용하여 폴리곤의 크기를 대신한다. 응용 프로그램에서 표현되는 물체의 크기에 따라서 폴리곤의 크기는 모두 다른 결과 값을 나타내게 되므로 각 물체의 폴리곤 크기를 모두 구한 결과를 분석하여서 물체의 크기별로 정렬을 한다. 폴리곤 크기를 정렬하는 이유는 하이브리드 렌더링에서 음영법을 선택할 때 기준 레벨에 따라서 몇 개의 폴리곤을 평면 음영으로 처리 할 지에 대한 기준으로 적용하기 위해서이다.

- 각 정점에서의 인텐시티: 광원은 하나만 존재 하는 것으로 간주하여 폴리곤의 법선 벡터에 대해서 각 정점에서의 풍 반사 모델을 적용하여 계산한다. 풍 반사 모델은 주변반사, 확산반사, 경면반사의 세기를 모두 합하여 인텐시티(intensity)를 구한다. 각 반사의 세기에 대한 수식은 아래와 같다.

각 반사모델에 대한 세기는 R, G, B에 대해서 각각 구해지며 이를 하나의 정점에 대한 인텐시티로 모두 합산한 결과 값을 구한다. 그 후 하나의 폴리곤에서 각 정점에서의 인텐시티의 차이를 각각 구한다. 세 정점의 인텐시티 차이를 비교하여 보았을 경우 차이가 미미하다면 라이팅 모델에 의한 색상차이가 크게 나지 않으므로 굳이 고로 음영을 사용할 필요가 없으며 평면 음영을 사용하여 처리 하여도 질적인 면에서 크게 떨어지지 않는다. 차이가 많이 나는 경우는 고로 음영을 사용하여 처리해 주는 편이 질적인 면에서 우수 하지만 차이가 많이 나는 경우에도 모두 고로 음영을 사용하지 않고 폴리곤의 크기를 비교하는 단계를 거치게 된다. 폴리곤의 크기가 작다면 평면 음영을 사용한 처리를 하고 폴리곤의 크기가 크다면 고로 음영을 사용하여 처리하도록 한다.

```
Algorithm Hybrid_Shader(Object meshes)
{
for(int i=0; i< n; i++) //about#ofpolygonn{

double Decide_ShadingRate(){
```

```
// decide # of flat shading and # of Gouraud shading
// d: distance between an object and a camera
//f= # of flat shading, g=# of Gouraud shading
g = (n - d * f)/(1-d);
f = n -(1-d)g/d;
}

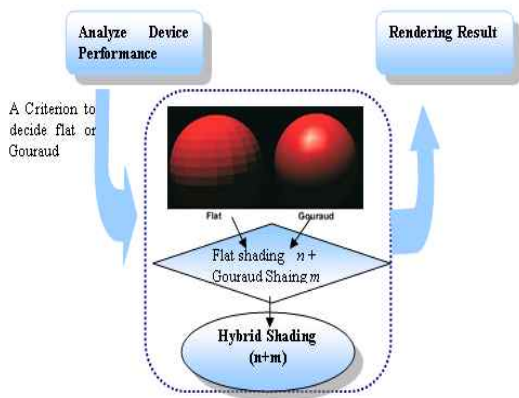
float Polygon_Size(floatv1,floatv2,floatv3){
// compute polygon size
poly_size = distance of v1 v2 + distance of v2v3 +
distance of v1v3 ;
return poly_size; }

float[] Sort_Polygon(){
// Sort polygon_size
return sort_array[]; }

float Vertex_Intensity(floatv1,floatv2,floatv3){
// compute intensity light sources
vertex_intensity = ambient_intensity +
diffuse_intensity + specular_intensity;
return vertex_intensity; }

float Difference_Intensity(*vertex_intensity){
// compute difference light sources' intensity
return distance_intensity; }

float first_base = polygon_average;
//decide first base
float second_base = c; //c:constant,decidesecondbase
// 폴리곤에서의 세 정점의 광원의 차이가 모두 두 번째
기준 값보다 작으면 평면 음영법 적용
// 폴리곤에서의 각 정점의 광원의 차이가 하나라도 기준
값보다 크면 폴리곤의 크기에 따라서 평면 음영과
고로 음영을 사용한 렌더링 실행
// 단, 평면 음영과 고로 음영적용 폴리곤수가 각각 g
와 f값보다 작을 동안 실행
if(distance_intensity1 <= second_base &&
distance_intensity2 <= second_base &&
distance_intensity3 <= second_base) then
while(# of flat shading < f) FLAT_SHADING();
else if(sort_array <= first_base ) then
while(# of flat shading < f) FLAT_SHADING();
else while(# of Gouraud shading < g)
GOURAUD_SHADING();
}
}
```

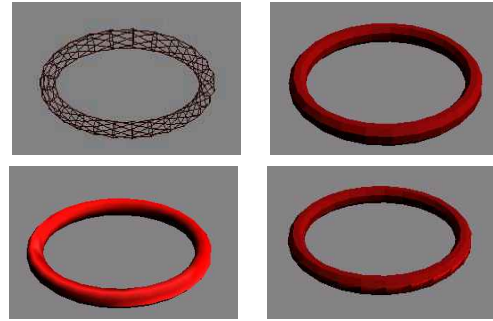


(그림 1) 하이브리드 렌더링 개념도

4. 구현 및 성능분석

그래픽 예제 프로그램을 통하여 하이브리드 렌더링 방법에 대한 성능 분석을 하였다. 임의의 레벨이 결정된 메시를 일정한 고로 음영법을 적용한 경우와 평면 음영법을 적용한 경우 그리고 평면 음영을 적용할 음의의 개수 n 과 고로 음영을 적용할 임의의 개수 m 을 결정하여 하이브리드 음영 방식을 적용하였을 경우의 예를 각각 실행해 보이고 각 방식을 적용 하였을 경우 응용프로그램에서 필요로 하는 연산시간을 비교한다.

성능 분석에 대한 실행환경은 OpenGL API로 만들었으며 프로그램에 사용된 모델링 예제는 234개의 정점을 사용하여 416개의 폴리곤으로 구성된 토루스이다. 데스크 탑 환경은 Window XP 펜티엄 4, CPU 2.8GHz, 512MB RAM으로 구성되어 있고, 모니터의 해상도는 1280 x 1024이며 그래픽 카드는 NVIDIA GeForce 4 MX 440이다. (그림 2)는 데스크 탑 환경에서 예제로 사용된 토루스를 평면 음영법과 고로 음영법을 사용하여 렌더링한 경우와 메시로 표현한 경우, 하이브리드 렌더링을 적용한 경우이다. 평면 음영과 고로 음영을 각각 적용할 폴리곤의 개수를 결정하기 위하여 카메라의 위치와 토루와의 거리를 임의로 설정하여 가장 가까운 거리에 해당하는 경우와 가장 멀리 있는 경우에 해당하는 경우를 구현하였다.



(그림 2) 토루스: (위 왼쪽부터 아래로) 메시, 평면 음영, 고로 음영법, 하이브리드 렌더링으로 표현

폴리곤의 갯수를 n 이라고 하면, 기준 값을 결정하기 위해서는 폴리곤의 크기를 구하기 위한 반복 실행문 n 과 각 정점에서의 광원의 인텐시티를 구하기 위한 처리 과정에서 주변반사, 난반사, 전반사에 대한 계산이 $8n$ 만큼 필요하며 전체적으로 반복 실행 횟수는 $9n$ 이다. 그리고 평면 음영에 걸리는 시간은 기준 값을 결정하는 시간 외에 각 폴리곤에 대한 법선 벡터를 구하기 위한 시간을 더 요구한다. 또한 고로 음영의 경우 기준 값을 결정하기 위한 시간 n 과 각 정점에 대한 평균 법선 벡터를 위한 연산시간 $4n$ 을 더 요구하게 되므로 토루스 모델 예제의 경우 필요한 연산 횟수는 아래 <표 2>와 같다.

이번에는 단말기 환경을 임의로 현재 평균적으로 사용되고 있는 데스크 탑과 휴대폰으로 가정하여 단말기 성능에 따른 하이브리드 렌더링을 적용한 경우를 구현하였다. 1GHz~2GHz의 처리속도를 가지는 데스크 탑과 64~88MHz의 처리속도를 가지는 휴대폰의 성능비를 기준으로 데스크 탑에서 100% 렌더링 성능을 발휘하는 그래픽 콘텐츠를 휴대폰에서 실행하고자 할 때 필요한 폴리곤 수를 계산하여 하이브리드 음영방법의 효율성을 검증하기 위한 성능분석을 하였다. 따라서 데스크 탑에서 415개의 폴리곤을 가지는 토루스를 모두 고로 음영을 사용하여 처리한다고 하면 이를 휴대폰에서 실행할 때 다중레벨 메시 모델 결정 방법에 의해서 바뀐 사용자 단말기 성능비는 약 0.028이 된다. 사용자 선호도에 따른 조건을 배제하고 단말기에서 표현할 수 있는 최대 폴리곤 수를 모두 사용한다고 하면 데스크 탑과 휴대폰의 성능비에 의해서 처리

할 수 있는 휴대폰에서의 적절한 고로 음영 처리 폴리곤 수는 데스크 탑의 415개에 비해 토루스의 경우는 약 8~9개가 선택되어야 한다. 따라서 하이브리드 음영을 사용한 렌더링 시간을 반복 횟수로 나타내면 다음과 같다.

- 정점에 대한 인텐시티 계산에 걸리는 시간: $9 \times 415 = 3735$
- 법선 벡터 처리 시간: $(1 \times 406) + (4 \times 9) = 442$
- 총 4177만큼의 반복횟수

<표 2> 415개의 폴리곤을 가지는 토루스 예제를 이용한 연산횟수 비교

연산 대상	연산횟수
기준값 결정시	$9 \times 415 = 3735$
평면 음영만 적용 시	$(9 \times 415) + 415 = 4150$
고로 음영만 적용 시	$(9 \times 415) + (4 \times 415) = 5408$
하이브리드 렌더링 기준 값 적용 시	$(9 \times 415) + (1 \times 406) + (4 \times 9) = 4177$

비교한 연산횟수를 보면 고로음영 적용시보다 하이브리드 렌더링을 적용한 경우 연산횟수가 줄어들어 렌더링 시간을 덜 요구함을 알 수 있다. 다만, 성능분석을 하기위해 사용된 현재의 그래픽 콘텐츠가 실행중인 단말기를 휴대폰으로 설정했고 예제 프로그램인 토루스 모델의 폴리곤 수가 많지 않으며 토루스 모델을 구성하고 있는 전체적인 폴리곤 크기가 많이 차이가 나지 않아서 고로 음영이 적용된 폴리곤의 수가 약 8~9개 정도에 불과하다. 따라서 하이브리드 렌더링을 적용한 구현 결과는 질적으로 평면 음영과 크게 차이가 나 보이지 않는다.

하지만 많은 수의 폴리곤을 가지는 복잡한 오브젝트를 다양한 단말기 성능을 고려하여 테스트할 경우 연산횟수와 표현의 질적 수준에 따른 단계를 조절할 수 있을 것으로 보인다. 즉, 제안한 하이브리드 렌더링 방법은 음영법 선택에 따라서 응용 프로그램에서의 계산 횟수가 차이가 난다. 선택 기준을 어떻게 정하느냐에 따라서 평면 음영법과 고로 음영법이 적용되는 폴리곤의

개수가 달라지게 되며 만약 빠른 렌더링을 필요로 하는 경우 선택 기준의 변경에 의해서 비슷한 단말기 환경에서도 3D 그래픽 콘텐츠의 질에 대한 사용자의 요구 사항을 반영할 수 있다. 물론 폴리곤의 크기나 각 정점에서의 인텐시티, 관찰자의 시점에 대한 적절한 기준은 단말기 성능에 따라서 좌우 되지만 임의의 선택 기준 변경으로 좀 더 적절한 적응 서비스를 제공할 수 있다.

5. 결론 및 향후 과제

컴퓨터 그래픽 분야에서의 실시간 렌더링 기술을 위하여 개발된 LOD 개념을 확대하여 렌더링 대상의 복잡도를 몇 개의 상세도를 가지는 단계로 나누어 서비스 하고자 하는 기술은 다양한 사용자 환경에 매우 부합하는 기술로 관심을 받고 있다. 따라서 기존의 모델링이나 광역 조명 계산 방법에 적용되던 LOD개념이 아닌 지역조명 모델에서의 하이브리드 렌더링을 이용한 LOD 방법을 제안하였다. 하이브리드 렌더링은 기존의 고로 음영과 평면 음영 라이브러리를 개선한 방법으로 하나의 물체를 이루는 여러 개의 폴리곤에 각각 다른 음영법을 적절하게 적용한다. 다양한 단말기 성능을 고려하여 테스트할 경우 렌더링에 걸리는 연산횟수와 표현의 질적 수준에 따른 단계를 고로 음영을 적용할 폴리곤 수와 평면 음영을 적용할 폴리곤 수를 임의로 조절할 수 있으므로 실시간 렌더링 시간을 줄일 수 있는 효율적인 대안 방법이 될 수 있다. 또한 단말기 사용자의 요구조건에 부합하는 콘텐츠를 선택할 수 있게 함으로써 그래픽 콘텐츠 이용자의 만족도를 높일 수 있다. 이러한 장점은 향후 유비쿼터스 환경에서 다양한 단말기 환경의 그래픽 콘텐츠의 실시간 적응 서비스에 매우 적절하게 사용될 수 있다.

향후 과제로는 그래픽 콘텐츠가 서비스되고 있는 단말기의 성능이나 사용자의 선호도, 콘텐츠의 크기 및 기타 네트워크 환경 등을 고려하여 물체나 장면에 적용할 적절한 수준을 결정하는 알고리즘에 대한 연구가 필요하다. 즉, 적절한 평면 음영의 수나 고로 음영의 수를 자동으로 계산하여 렌더링 할 수 있는 알고리즘으로

각 단말기 종류 별 성능에 대해 다양한 컴퓨터 그래픽 콘텐츠 실행율에 대한 실험 및 분석이 더 요구 된다. 정확한 단말기 성능 분석이 그래픽 콘텐츠의 적합한 상세도에 밀접한 영향을 미치기 때문이다. 그리고 애니메이션, 영화나 게임 등의 다양한 실질적인 실시간 렌더링 예제에 대한 실험을 통해서 하이브리드 렌더링의 효율성에 대한 검증이 더 필요하다.

참 고 문 헌

[1] Alan Watt, 3D Computer Graphics, Addison Wesley, 2000

[2] H. Kim, C. Joslin, T. Di Giacomo, S. Garchery, N. Magnenat-Thalmann : Multi-resolution Meshes for Multiple Target, Single Content Adaptation within the MPEG-21 Framework, IEEE ICME Conference, pp.1699-1702, 2004

[3] P. Gioia, A. Cotarmanac'h, K. Kamyab, P. Goulev, E. Mamdani, I. Wolf, A. Graffunder, G. Panis, A. Hutter, A. Difino, B. Negro, M. Kimiaei, C. Concolato, J. Dufourd, T. Di Giacomo, C. Joslin, N. Magnenat-Thalmann : ISIS: Intelligent Scalability for Interoperable Services. IEE CVMP pp. 295-304, 2004

[4] Survey of Multiresolution Modeling, <http://www.cs.cmu.edu/afs/cs/user/garland/www/multires/survey.html>

[5] Anthony Vetro, Christian Timmerer : Digital Item Adaptation: Overview of Standardization and Research Activities. IEEE Transactions on Multimedia, VOL. 7, NO. 3, pp. 418-426, 2005

[6] Eric Lengyel : Mathematics for 3D Game Programming and Computer Graphics 2ndEdition, CHRLERIVER MEDIA, INC.

[7] 장호욱, 이인호, 사실적 장면 표현을 위한 렌더링 기술 동향, 전자통신동향분석, 20권 6호 (통권 96), 2005

[8] 멀티 플랫폼 공용 온라인 게임 개발, 정보통신 산업 기술 개발 사업 연구 개발 결과 보고서, 2003

[9] Kyu-Young Whang Ju-Won Song, Ji-Woong Chang, Ji-Yun Kim, Wan-Sup Cho, Chong-Mok Park, Il-Yeol Song, Octree-R: An Adaptive Octree for Efficient Ray Tracing, pp. 343-349, Vol. 1, No. 4, IEEE transaction on Visualization and Computer Graphics, 1995

[10] Aaron Knoll, Ingo Wald, Steven Parker, Charles Hansen, Interactive Isosurface Ray Tracing of Large Octree Volumes, Interactive Ray Tracing 2006, IEEE Symposium on, pp.115-124, 2006

[11] Karsten Hilbert, Guido Brunnert, "A Hybrid LOD Based Rendering Approach for Dynamic Scenes," *cgip*, pp.274-277, ComputerGraphicsInternational2004(CGI'04), 2004

[12] Proceedings of the 4th international conference on Computer graphics and interactive techniques, pp. 157 - 163, 2006

[13] 이범중, 윤종현, 박종승, 균일한 렌더링 부하를 위한 영역기반의 기여도 쉐딩 한국정보과학회 학술발표논문집 Vol. 33, No. 2(A), pp. 148~152, 2006



김 학 란

2003년 : 숙명여자대학교 대학원 (통신학석사)
 2007년 : 숙명여자대학교 대학원 (이학박사-컴퓨터그래픽스)

2003년~2007년: 숙명여자대학교, 한성대학교 출강
 2004년~현재: 한성대학교 멀티미디어공학과 겸임교수
 관심분야: 컴퓨터그래픽스, 게임, 가상현실, 애니메이션, 유비쿼터스 컴퓨팅(AR) 등



박 화 진

1989년 : 숙명여자대학교 대학원 (이학석사)
 1997년 : Arizona State Uni. Computer Science (공학박사-컴퓨터그래픽)

1997년~1998년: 삼성 SDS연구소 전임 연구원
 1998년~2000년: 평택대학교 전임강사
 2006년~현재: 숙명여자대학교 멀티미디어학과 교수
 관심분야: 컴퓨터그래픽스, 게임, 3D모델링, 가상현실, 멀티미디어 등