

게임 응용을 위한 다면체 모델의 연속 상세정도 표현

최정주*

1. 개요

최근 컴퓨터 하드웨어의 발달에 힘입어 현실감 있는 게임 화면을 얻기 위하여 점차 많은 수의 점, 선, 면으로 이루어진 복잡한 다면체 모델이 컴퓨터 게임에 빈번히 등장하고 있다. 복잡한 다면체 모델은 필연적으로 렌더링에 많은 계산 시간을 필요로 하고, 이러한 상황은 게임과 같은 실시간 대화형 응용 분야에서는 큰 장애 요소가 될 것이다. 본 기고에서는 복잡한 다면체 모델을 실시간 응용에서 다루기 위한 연구분야인 다면체 모델의 상세정도(Level of Detail, 이하 LoD) 표현을 위한 다양한 방법을 게임 응용의 관점에서 살펴보고, 현실적인 게임 응용을 위해서 필요한 기술적 내용을 소개한다.

1.1 LoD 연구의 역사

LoD의 연구는 1976년 James Clark이 화면에 등장하는 물체를 여러 단계의 해상도로 표현할 경우 얻을 수 있는 이점을 언급하면서 컴퓨터 그래픽스의 모델링 분야에서 많은 주목을 받아 왔다 [1]. 1990년대부터 오늘에 이르기까지 매우 많은 관련 논문들이 발표되었으며 현재까지 이어지는 LoD와 관련된 연구는 물체의 다중 해상도 표현

(multiresolution representation) 연구로 대표될 수 있다.

LoD는 물체를 화면에 표시함에 있어서 성능과 품질이라는 서로 상반된 목표를 동시에 달성하기 위한 해법의 하나로 연구되기 시작했으며, 오늘날 하드웨어의 발달이 가속화되는 시점에서 현실적으로 성능과 품질을 동시에 만족시킬 수 있는 거의 유일한 해법으로 받아들여지고 있다. 물체를 화면에 표시하여 좋은 품질을 얻기 위해서는 많은 수의 다면체를 사용하는 것이 가장 일반적인 방법이다. 하지만 많은 수의 다면체를 하드웨어 혹은 소프트웨어적으로 처리하기 위해서는 많은 시간이 소요되는데 이는 실시간 대화형 그래픽스(real-time interactive graphics)를 추구하는 시대의 흐름에 반하는 결과를 낳게 된다. 따라서 현재 시점에서 화면 품질을 크게 손상하지 않으면서 충분한 성능을 얻을 수 있는 정도(level)의 상세함(detail)을 얻을 수 있는 방법이 있다면 품질과 성능의 목표를 동시에 이룩할 수 있게 될 것이다.

초기의 LoD 표현은 물체에 대한 다단계의 기하 정보를 직접 수작업으로 입력하여 얻는 것이 일반적이었다. 당시에는 디자이너들이 한 물체에 대해 해상도가 다른 다수의 물체를 디자인 하는 것은 컴퓨터를 사용하는 비용에 비하면 미미한 수준이었기 때문에 이러한 방법이 선호되었다. 그러나 1990년대 이후로 컴퓨터를 사용한 계산비용은 점

* 아주대학교 정보통신대학 미디어학부 교수

차 줄어들고 물체를 여러 번 디자인하는 비용은 급속히 증가하면서 기존의 LoD 표현을 자동화하는 방법이 모색되게 되었으며, 다면체를 구성하는 점집합의 개수를 줄이는 정점 소거방법(vertex decimation), 정점 군집방법(vertex clustering) 등과 같은 다양한 방법이 소개되기 시작하였다 [2,3].

기존의 LoD 연구가 물체 표현에 있어서 이산적(discrete)인 성격을 갖는다면 1996년 Hoppe에 의해서 등장한 메쉬구조의 점진적 표현방법(progressive mesh representation)은 화면에 등장하는 물체의 표현이 연속적(continuous)으로 다루어진다는 점에서 LoD 연구의 큰 전기를 마련하였다[4]. 이후 약 2~3년간 시점 종속 LoD(view dependent LoD), 2차 오차 측정법(quadric error metrics), 경계 보존(boundary preserving), 텍스처 왜곡 최소화(texture distortion minimization), 위상 단순화(topology simplification) 관련 연구 등 심도있는 다양한 주제들이 다루어졌다[5-9].

1.2 Level of Detail 응용 분야

LoD의 개념이 소개된 후로 최초의 응용분야는 비행 모의실험(flight simulation)이라고 보아야 할 것이다. 비행 모의실험 시 화면에 보여지는 대개의 정보는 지형정보라고 할 수 있는데, 이 지형 정보를 계층적 화면 그래프로 나타내고 비행 속도나 조종사의 시점 등 다양한 기준에 따라서 지형 정보를 다단계로 표시할 수 있다. 이런 LoD 표현을 통해서 사실성과 함께 실시간성이 중요한 비행 모의실험에서 좋은 결과를 얻을 수 있었다.

최근 LoD가 가장 널리 쓰이는 응용분야로 실시간성과 함께 화면의 품질이 급속히 향상되고 있는 게임 분야를 들 수 있다. 게임 응용에서 실시간성은 타협의 여지가 없는 요소이기 때문에 고전적인

게임들에서는 하드웨어의 제약으로 인해 복잡한 물체(캐릭터)를 게임에 등장시킬 수 없었다. 기본적인 3차원 물체에 대한 처리 기능을 갖춘 하드웨어의 등장과 함께 3차원 게임들이 등장하고 사용자의 화면 품질에 대한 기대수준이 높아지면서 게임에서도 고해상도 물체를 사용하게 되었다. 하지만 이러한 하드웨어 기술의 비약적인 발달에도 불구하고 여전히 대규모의 캐릭터가 등장하는 게임에서는 3차원 물체의 처리는 LoD의 기술을 필요로 하며, 현재까지 개발된 거의 모든 게임은 등장 캐릭터 별로 몇 개의 모델셋을 사용하는 전통적인 이산 LoD(Discrete LoD, 이하 DLoD) 기술이 많이 사용되고 있다. DLoD는 캐릭터를 구성하는 다면체 모델이 디자이너에 의하여 수작업으로 만들어지거나, 혹은 자동화된 기법에 의하여 만들어지거나의 여부에 상관없이 화면의 품질이 급격히 변하는 현상(popping effect)을 피할 수 없다. 본 기고에서는 DLoD가 가진 이상현상을 극복할 수 있으면서 향후 게임 제작의 모델링 과정에 도입될 가능성이 있는 연속 LoD(Continuous LoD, 이하 CLoD) 기술을 중심으로 기술적 발전과 연구동향, 향후 연구 전개방향 등을 소개한다.

2. 최근의 기술적 동향

앞서 소개한 바와 같이 LoD 분야에서 많은 연구 주제들이 다루어져 왔으며, 그 동안 관련 연구의 큰 흐름은 한마디로 점 데이터의 최소화 혹은 간략화라고 할 수 있다.

오늘날 보편적인 하드웨어의 도움을 받는 그래픽스 파이프라인(혹은 렌더링 파이프라인)의 처리 단위는 점과 화소라고 할 수 있는데, 하드웨어적 처리의 최초 과정은 점 데이터의 처리에서부터 시작된다. 이때 하드웨어로 전달되는 점 데이터의 양이 하드웨어의 실시간 처리한계를 넘어서면 용

용 프로그램의 실시간성은 상실된다. 따라서 가능한 적은 수의 점 데이터를 하드웨어로 처리하기 위해서 고전적으로 view frustum의 이용이나 back face culling과 같은 방법들이 많이 사용되어 왔다. 이러한 방법은 동시에 화면에 등장하는 물체가 많은 경우에는 한계가 분명한 방법으로, 결국 가장 손쉬운 해결 방법은 적은 수의 점으로 표현되는 물체를 디자인 하고 사용하는 방법이다. 그러나 적은 점으로 표현되는 물체는 필연적으로 화면 품질의 저하로 연결된다. 일반적으로 실시간 처리 가능성과 화면의 품질은 동시에 달성되기 어려운 주제이지만, LoD는 이 두가지 상충되는 목표를 달성하기 위한 역할을 충실히 수행하고 있다.

그동안 LoD 연구에 있어서 커다란 질문은 “단지 몇 개의 화소에 표현될 가능성이 있는 물체를 많은 수의 점(혹은 면)으로 상세하게 표현하는 것이 바람직한가?”에 대한 답을 제시하기 위하여 진행되어 왔다. 적은 수의 화소에 표현되는 물체는 적은 수의 점(혹은 면)으로 표현하는 것이 바람직하다는 전제하에서 연구가 진행되어 왔다. 이런 커다란 전제에는 분명한 한계가 있는데, 예를 들면 적은 수의 다면체로 물체를 표현할 때 외관의 심하게 변형되는 경우, 빛에 대한 반사 방향이 변하는 경우, 매핑되는 텍스처 좌표의 왜곡에 의해서 원하지 않는 결과를 얻게 되는 경우가 발생할 수 있다. 최근 몇 년간 LoD의 주제는 적은 개수의 점 데이터 표현과 함께 시각적으로 화면 품질의 저하가 발생하지 않는 표현법을 얻는데 많은 연구가 이루어지고 있다.

일반적으로 사용자는 물체를 외곽선(silhouette)의 정보에 의존하여 인지하는 경향이 있기 때문에 사용자의 시점에서 물체의 외곽선을 추출하여 외곽선 주변은 자세하게 보여주고 외곽선에서 떨어진 내부는 간략하게 보여주는 방법이 이러

한 관찰로부터 하나의 답을 제시할 수 있다. 또한 물체의 빛에 대한 반사 방향은 물체의 법선 벡터(normal vector)에 의하여 결정되는데 가장 상세한 모델에서 얻는 법선 벡터의 지도(map)를 간략화된 물체에 매핑하여 물체의 빛에 대한 반응 결과를 그대로 유지하는 방법도 연구되어 왔다. 또한 텍스처 매핑을 균일하게 하여 화면의 품질을 높이는 방법이 많은 주목을 끌고 있다. 이러한 LoD 연구 방향의 변화는 결국 연구의 초점이 점 데이터의 간략화에서 외양의 보존(appearance preserving) 가능성이라는 곳으로 변화하고 있음을 알 수 있다.

3. LoD 연구의 다양한 주제

3.1 이산성과 연속성

DLoD는 대개 전처리 단계에서 동일한 물체에 대한 여러 단계의 표현을 수작업 혹은 자동화 작업에 의하여 미리 만들어 놓고, 시점에서 물체 사이의 거리 혹은 물체가 화면에서 차지하는 면적 등의 특정한 기준에 의하여 이미 만들어 놓은 여러 단계의 물체 표현 중 하나를 선택하여 사용하는 것이다(그림 1 참조).

DLoD 표현은 시점으로부터의 거리와 같이 매우 간단한 기준에 의하여 적절한 표현만 선택하면 되기 때문에 계산량이 매우 적어서 실시간성이 중요한 게임과 같은 응용분야에 적합하다. DLoD를 실제 응용에 적용할 때 주의해야 하는 점은 시점과 물체의 거리가 LoD 수준을 변경하는 경계에서 자주 움직이는 경우에 LoD를 자주 바꾸어야 하는 문제가 발생할 수 있다. 이 경우 이른바 게으른 처리(lazy processing) 방법을 사용하여 물체가 특정한 경계의 범위를 벗어날 경우 LoD를 변경하는 방법을 사용하면 좋은 결과를 얻을 수 있다. DLoD 표현은 지형 데이터와 같이 물체가 매

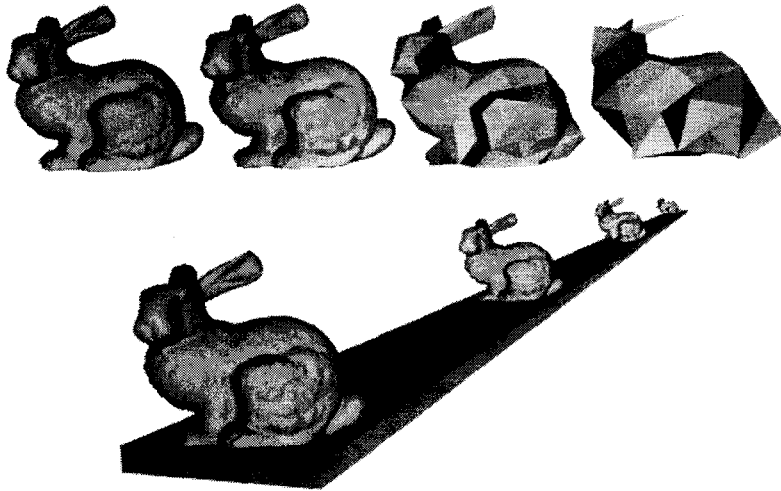


그림 1. 물체의 DLod 표현과 거리를 기준으로 한 응용의 예

우 큰 범위를 차지하고 있는 경우에는 시점으로부터 모델까지의 거리의 편차가 지나치게 크기 때문에 적절한 LoD를 미리 디자인하기가 불가능하고 전처리 단계에서 매우 세밀한 주의를 기울이지 않으면 화면의 품질이 급격히 변하는 단점(popping effect)을 가지고 있다.

이러한 문제를 해결하기 위해서 등장한 CLoD는 LoD를 단순히 실행 시간에 해결하는 것뿐만이 아니라 실행시간에 적절한 수의 다면체(혹은 점)를 필요한 수요만큼 추출할 수 있도록 물체의 표현법을 얻는데 더 큰 목적이 있다. 결과적으로 하드웨어 자원을 효과적으로 사용하게 되고, 화면이 급격히 변하는 단점이 극복되며, 더불어 물체를 인터넷과 같은 환경에서 점진적으로 전송할 수 있게 되고, 추가적으로 물체에 대한 점진적 압축 표현을 얻을 수 있는 장점이 있다. 일반적으로 CLoD 표현에서는 시점을 항상 고려하고 있기 때문에 대개의 CLoD는 시점 종속성을 가지고 있다고 할 수 있다.

3.2 위상 정보

LoD의 커다란 주제 중 다른 하나는 물체의 위

상정보 (topology)와 관련된 것이다. 흔히 LoD는 물체를 표현하는 점, 선, 면과 같은 기하정보의 수를 줄이는 문제로 인식되지만, 이 과정에서 물체를 표현하는 기하정보의 연결관계를 표현하는 위상정보가 보존되는 것이 일반적으로 화면의 품질이 우수하다. 대개의 경우 위상정보는 물체의 구멍(hole, tunnel 혹은 cavity)의 수로 표현되는데 이러한 위상의 성질이 유지되면 비교적 우수한 품질의 화면을 얻을 수 있다(그림 2 참조).

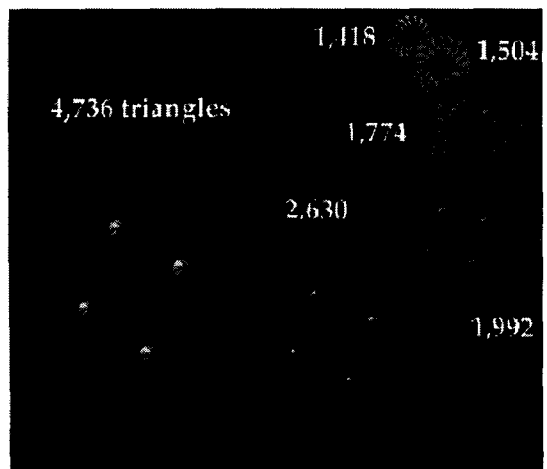


그림 2. 위상을 유지하는 물체의 LoD 표현

3.3 다면체 모델의 점진적 표현 방법

1996년 Hoppe에 의해서 제안된 다면체의 점진적 표현 방법(progressive mesh representation)은 간단한 두 개의 연산자를 바탕으로 표현되는 CLoD 표현 방법이다. 주어진 물체에서 한 개의 선분을 선택하여 선분의 양 끝 점을 하나의 점으로 합병하는 선분 합병(edge collapse) 연산자와 합병된 하나의 점을 두개의 점으로 분할하는 점 분할(vertex split) 연산자로 물체의 연속적인 LoD를 표현하고 있다(그림 3 참조).

최초에 주어진 매우 복잡한 모델로부터 선분

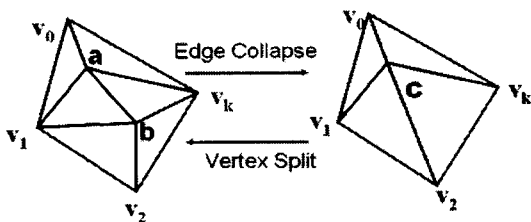


그림 3. 점진적 표현법을 구성하는 선분합합 및 점분할 연산자

병합연산을 연속적으로 수행하면 종국에는 단지 하나의 면(혹은 점)으로 이루어진 매우 단순한 형태의 모델을 얻을 수 있으며, 이러한 선분 병합연산을 모두 모으면 CLoD 표현을 얻을 수 있게 된다(그림 4 참조). 이때 어느 순간에 병합의 대상이 되는 선분(candidate edge)를 선택하고 병합 후 점의 위치를 결정하는 방법이 유일하게 결정되면 물체에 대한 유일한 CLoD 표현이 가능하게 된다. 선분 병합 대상 선분을 결정하는 다양한 방법은 대개 오차 측정법에 의해서 결정되며 다음 3.6절에서 좀 더 자세히 다루도록 한다.

3.4 시점 종속성

대개의 CLoD 표현은 시점을 고려하기 때문에 시점 종속성을 가지고 있다. 일반적으로 시점 종속성은 시점에서 거리에 따라 자동으로 단순화할 수준을 결정한다(그림 5 참조). 이때 지형 데이터와 같이 단순히 거리가 멀어짐에 따라서 보다 단순화된 데이터를 사용하는 방법이 많이 사용되고

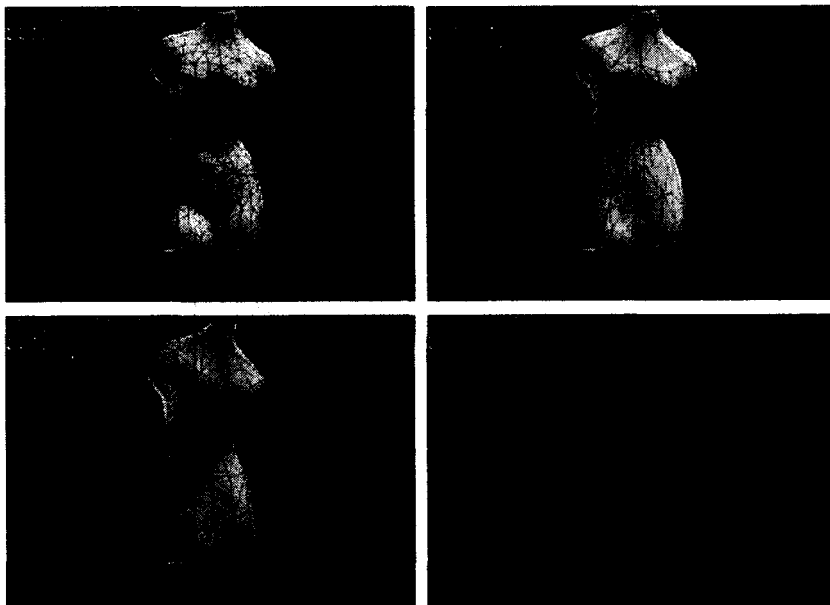


그림 4. 점진적 모델 표현의 진행 과정

있다(그림 5(a) 참조). 사람은 대개의 경우 물체를 외곽선(silhouette)의 모양에 따라서 물체의 모양을 인식하는 경향이 두드러지기 때문에 시점에서 보이는 외곽선을 경계로 단순화 정도를 조절하면 화면의 품질을 유지할 가능성이 높아진다(그림 5(b) 참조).

3.5 외양의 보존

최근 LoD 연구의 주된 흐름 중에 하나는 단순히 물체를 표현하는 점, 선, 면 등 기하 정보를 단순화할 경우 발생하는 외양(appearance)의 변형을 최소화하는 방향의 연구가 많이 이루어지고 있다는 점이다.

예를 들어 그림 6 (a)와 같은 물체가 주어졌을 때, 물체의 점, 선, 면 등 기하정보를 무작정 단순

화 하면 렌더링 후 같은 품질의 화면을 얻을 수 없게 된다. 그림 6 (a)와 같은 화면에서 사용자에게 인식되는 중요한 부분은 광원의 색으로 반짝이는 하이라이트 부분인데 이는 시점의 위치와 더불어 광원의 위치가 큰 역할을 한다. 이때 기하정보를 아무런 고려 없이 단순화하면 하이라이트가 존재하는 부분의 기하정보가 변경되고 하이라이트의 모양이 심하게 변형되는 결과를 얻게 된다. 한가지 간단한 해결책은 그림 6 (b)와 같이 하이라이트 부분은 좀 더 상세히 표현하면 LoD 표현으로도 충분히 고품질의 영상을 얻을 가능성이 생긴다. 위와 같은 방법은 한가지 문제점을 가지고 있는데, 광원의 위치가 변하면 LoD 표현도 변해야 한다는 점이다.

LoD 표현에 따라서 렌더링 품질이 달라지는 원

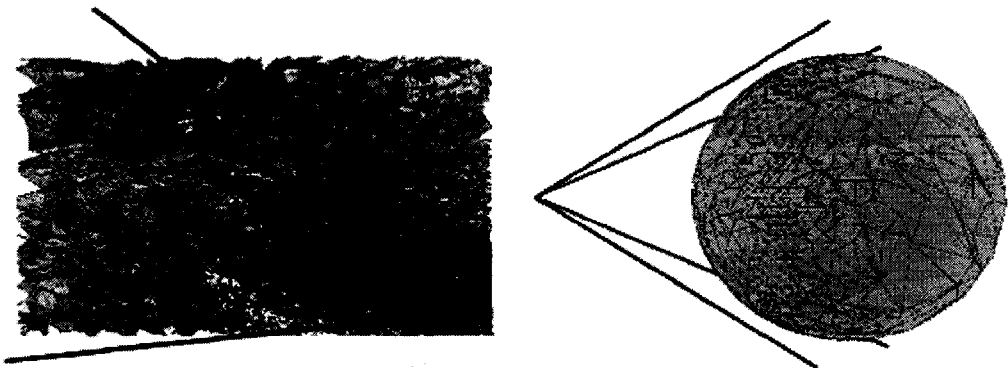


그림 5. 시점 종속적 LoD 표현 방법



(a)

(b)

그림 6. 광원에 의한 렌더링 효과를 고려한 LoD 표현

인을 살펴보면 LoD 표현에 의해서 물체가 단순화 되면 물체의 외양을 이루는 면의 개수가 줄어들게 되고, 따라서 단순화된 물체는 복잡한 물체에 비해서 한 개의 면이 차지하는 평균 면적이 커지게 된다. 이에 따라 개별 화소를 표현하기 위한 상대적인 정보의 양이 줄어들게 된다. 즉 렌더링에 사용되는 중요한 정보인 면의 법선벡터(normal vector)가 원래 복잡한 물체보다 적은 정보만을 표현하게 되는 상황(normal under-sampling)에 처하게 된다. 만일 원래 복잡한 물체에서의 법선 정보를 유지할 방법이 있고 이를 간략화된 물체의 렌더링 시간에 적용할 수 있다면 복잡한 물체와 단순화된 물체 사이의 렌더링 품질 차이가 적어지게 된다 (그림 7 참조). 그림 7 (a)와 같이 주어진 물체에 대해서 물체를 단순히 1/10 정도로 기하 정보만을 단순화 한 후에 렌더링 작업을 한 그림 7 (b)의 결과에 비해서 그림 7 (a)에서 샘플링된 법선 정보를 단순화 후 렌더링 작업에 반영한 그림 7 (c)는 매우 큰 차이를 보여주고 있으며, 이 결과는 초기에 주어진 복잡한 모델의 렌더링 결과와 큰 차이점을 발견할 수 없다.

3.6 오차 측정

CLoD 표현에서는 실행 시간에 LoD 수준을 결

정하기 때문에 어떤 점, 선, 면을 선택하여 단순화 하는가를 결정하는 것이 매우 중요한 연구 주제이다. 일반적으로 좋은 선택이 좋은 단순화로 연결되며, 이때 선택의 기준이 되는 것이 오차 측정법(error metrics)이다. 오차 측정법은 크게 기하 오차(geometric error)와 속성 오차(attribute error) 측정으로 나눌 수 있다.

기하오차는 두 정점(vertex)이 하나로 병합될 경우 병합된 정점과 원래 두 정점간의 거리의 최대값을 기준으로 오차를 측정하는 가장 간단한 방법(vertex-vertex error metrics)과, 정점과 다면체의 거리를 기준으로 오차를 측정하는 방법(vertex-plane error metrics), 면과 면사이의 거리를 기준으로 오차를 측정하는 방법(surface-surface error metrics) 등이 있다. 1996년 정점과 다면체의 거리 측정을 기초로 하는 Garland와 Heckbert의 2차 오차 측정(quadric error metrics) 방법은 속도와 품질 면에서 일반적으로 우수한 결과를 보이고 있다[7]. CLoD의 연산자로서 선분 교환(edge swap)을 사용하는 경우에는 두 정점간의 거리를 오류측정 기준으로 하면 정점에서의 오차는 없으나 면의 오차가 심하게 발생하는 경우가 발생할 수 있으므로 LoD 표현 방법에 따라서 적절한 오차 측정방법을 선택하는 것이 중요하다. 속성이란 정점에 귀속된 색상(color), 법선 벡

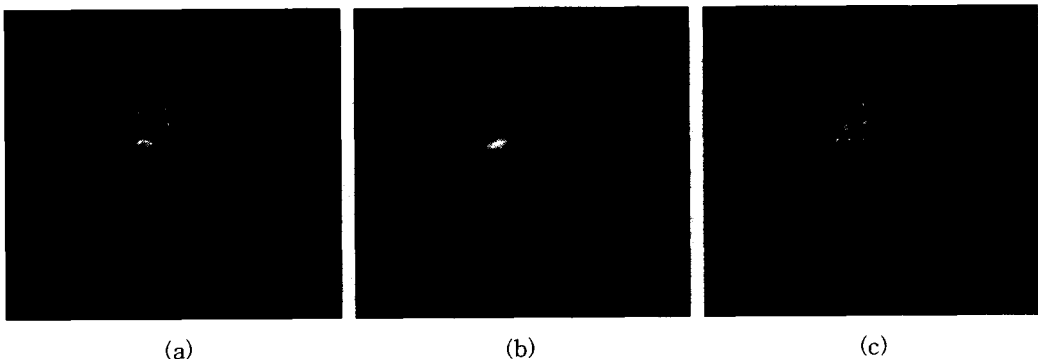


그림 7. 법선맵을 고려한 외양을 보존하는 LoD 표현

터(normal vector), 텍스처 좌표(texture coordinates) 등과 같은 것을 통칭하는 것으로, 속성 오차는 일반적으로 렌더링된 화면의 품질에 영향을 미치는 중요한 요소이다. 속성 오차측정은 대개 렌더링 시에 고려하는 모든 속성을 아우르는 고차원 공간에서의 기하 오차 측정과 동일한 기법을 그대로 적용하는 방법이 많이 사용되고 있다.

3.7 인지적 요소

LoD의 수준 선택 문제는 인간의 시각적 인지 능력과 관련이 깊다. 인지적 관점에서 잘 선택된 LoD의 수준은 시각적으로 영상이 급격히 변화하는 현상(popping effect)을 줄여 결과적으로 영상의 품질 향상으로 이어진다. LoD 수준 선택에서 인지적 관점은 물체와 시점 사이의 거리, 물체가 화면에서 차지하는 면적, 시각 초점과의 각도, 초점거리와의 차이, 물체의 각속도 변화량 등에 따라서 영향을 받는다.

시점으로부터 멀리 떨어진 물체는 간략히 표현해도 인지적으로 차이가 없다는 점은 명백하며 LoD 수준 선택의 가장 기본적인 선택 기준이라고 할 수 있다. 단순한 3차원 거리 측정만을 사용하므로 매우 빠른 LoD 수준 선택을 할 수 있다. 그러나 이 방법은 동일한 물체라도 확대/축소와 화면의 해상도에 따른 종속성을 가지고 있는 단점이 있다. 이에 대한 간단한 해결책으로는 물체가 2D 화면에 투영되었을 때의 면적을 기준으로 LoD 선택을 하는 방법이 있다. 이때 투영 계산은 단순한 거리측정 보다 많은 연산을 필요로 하기 때문에 속도의 문제가 있을 수 있으나, 물체를 대표하는 6면체, 구, 타원체 등의 기하입체(bounding volume)를 대신 사용하면 비교적 좋은 결과를 얻을 수 있다.

인지적 관점에서 더 중요한 요소는 실제로 사

람이 가진 시각의 특수성을 이용하는 방법이다. 기본적으로 사람은 초점(eccentricity)에서 벗어난 물체는 정확히 인지하지 못하는 경향이 있기 때문에 2D 화면의 중심에서 벗어난 물체는 더 간략히 표현해도 화면 품질에 큰 영향을 미치지 않는다. 또한 시각은 빨리 움직이는 물체를 제대로 인지하지 못하기 때문에 빠른 물체는 더 간략히 표현할 수 있는 가능성이 있다. 이때 물체의 속도는 선형속도(linear velocity)가 아닌 화면에서 시점을 기준으로 표현되는 각속도(angular velocity)가 기준이 된다. 대개의 경우 같은 선형속도를 가지고 있어도 시점에서 가까운 물체의 각속도가 더 크기 때문에 단순히 시점으로부터의 거리가 가까운 물체라고 해도 움직임이 있는 경우는 더 단순화된 다면체 모델을 사용할 수 있게 되는 것이다. 시각과 관련된 또 하나의 특징은 인간의 눈은 초점 거리를 가지고 있기 때문에 초점 거리에서 벗어난 물체는 흐리게 보이는 경향이 있다. 따라서 화면의 중앙에 표시되는 물체라도 초점 거리를 벗어나서 멀리 혹은 가깝게 있다면 더 간단한 다면체 모델을 사용할 수 있다.

3.8 정보의 압축과 임의 접근성

물체 정보의 압축은 LoD연구의 파생분야로 인식될 수 있으나, 최근에는 대용량의 자료가 보편화 되고 통신망을 통한 전달이 요구되면서 중요한 연구분야로 인식되고 있다. 3차원 스캐너를 이용한 물체 정보의 입력이 늘어나면서 물체를 표현하는 자료의 양이 상상을 초월하는 수준으로 증가하고 있다. 특히 물체 전체가 컴퓨터의 주기억장치(main memory)에 모두 적재 되지 않는 경우의 압축(out-of-core compression) 방법이 특별한 관심을 불러일으키고 있다.

메쉬 형태의 다면체 모델 데이터의 압축은 메

쉬가 가지고 있는 여러 가지 특별한 성질을 이용하여 압축을 수행하는데, 위치 정보를 표현하는 비트의 수를 적은 오차 범위 내에서 양자화(bit quantization)하는 방법에서 부터, 메쉬를 삼각형 스트립으로 표현하여 연결정보를 줄이는 방법 등이 사용되고 있다. 1998년 Touma와 Gotsman에 의하여 발표된 방법은 점에 연결된 선분의 개수(valence number)를 이용하여 연결정보를 함축적으로 표현하고 이를 이용하여 압축을 수행하는데, 압축효율이 대단히 높은 방법으로 알려져 있다[10]. 메쉬가 다중해상도로 표현되어 있는 경우에는 메쉬의 계층구조를 이용하여 압축을 수행하는 효율적인 방법이 Alliez에 의하여 소개되었다[11].

최근 주목을 받는 분야 중에 하나는 메쉬 구조가 컴퓨터의 주기억장치에 적재 되지 않는 경우의 압축방법이다. 적게는 수십 기가 바이트에서 수백 기가 바이트에 이르는 데이터를 적당히 분할하고 분할된 패치(patch)를 독립적으로 압축하여 병합[12]하는 방법과 메모리 외부에서 압축을 수행하는 방법[13] 등이 소개되어 있다. 이러한 외부기억장치 압축의 문제는 필요한 부분만 압축을 해제하는 방법이 취약하다는 단점이 있다. 화면에 필요한 특정 부분을 얻고자 할 경우에 대용량의 데이터를 모두 압축 해제하고 필요한 부분을 얻어야

하는 어려움이 있다. 압축된 다면체 모델의 원하는 부분에 바로 접근하는 임의 접근성은 선택적 전송, 선택적 압축 및 압축 해제 등을 비교적 저사양의 컴퓨터에서도 다룰 수 있는 길을 제시하고 있다(그림 8 참조).

4. 결론 및 향후 연구과제

LoD와 관련된 연구 분야는 1990년대 후반에 매우 많은 주목을 끌어 비약적인 발전을 이룩했다. 최근 몇 년 사이에는 관련 연구들이 학계에 발표되지 않는 등 연구가 일정한 수준에 도달해 새로운 주제를 찾기가 어려운 상황에 있다고 판단할 수 있을 것이다. 다만 LoD 연구와 간접적으로 관련된 메쉬 재구성(remeshing) 및 메쉬 매개화(parametrization)의 연구는 꾸준히 진행되고 있는 상황이다. LoD 연구의 커다란 흐름 중에 하나는 이미 언급한 인지적 관점의 도입과 외양의 일관성을 유지하기 위해 기하 오류 및 속성 오류를 동시에 최소화하는 결합된 방법의 연구가 좀 더 진행될 수 있다.

LoD의 응용 분야로서 게임은 매우 중요한 위치를 차지하고 있으며 앞으로도 당분간은 그러한 위치에 크게 변화가 없을 것이라고 판단할 수 있다. 게임 응용분야는 그 동안 여러 가지 이유로

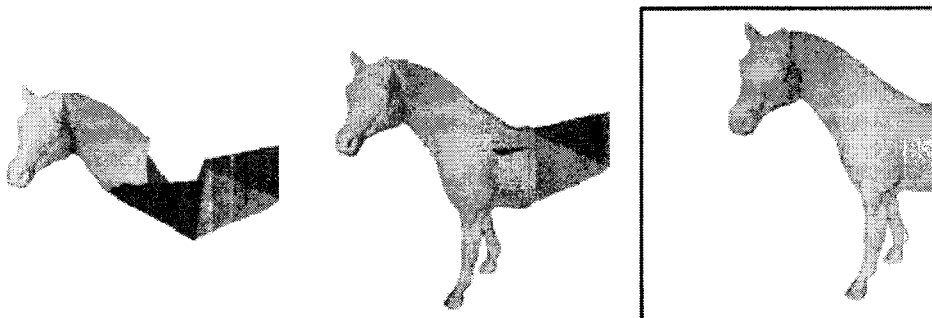


그림 8. 임의 접근 메쉬를 사용한 압축 해제의 예

DLoD를 주로 사용해 왔고, 최근 하드웨어의 고속 성장에 따라 CLoD를 적극적으로 도입하기 위한 움직임이 나타나고 있다. 게임에서 CLoD를 이룩하기 위한 커다란 두 개의 흐름은 progressive mesh와 subdivision surface의 도입 움직임으로 볼 수 있다. 많은 게임 프로그래머의 의견은 subdivision surface 쪽이 현실적으로 가능성이 많다는 의견이 지배적이다. Subdivision surface는 수학적으로 잘 정의된 모델과 빠르게 다면체의 개수를 줄여갈 수 있다는 점, 임의의 위상으로 물체의 표현이 가능하다는 점 등 다양한 장점이 있어 게임 응용에 적합한 측면이 있다.

Progressive mesh로 대표되는 다중해상도 메쉬를 이용한 CLoD의 처리는 게임에서 subdivision surface에 비해서 상대적으로 중요한 한 가지 장점을 가질 수 있는데, 그것은 기존 DLoD로 만들어진 게임을 새로이 프로그래밍하지 않고도 CLoD화 할 수 있는 가능성이다. DLoD의 단점 중 하나인 화면의 급격한 변화(popping effect)를 방지할 수단으로 CLoD를 고려하는데, 이때 이미 만들어진 DLoD 모델 사이를 점진적으로 변화하는 progressive mesh 구조가 있다면 아주 적은 코딩 만으로도 기존 DLoD를 지원하는 게임에 CLoD를 도입할 수 있는 가능성이 생긴다. Progressive mesh는 선분 병합을 하는 단계에서 병합될 선분을 선택하는 방법만으로는 중간 결과들의 예측이 완전히 불가능하기 때문에 원하는 결과를 얻기가 힘든 단점이 있다. 현재까지의 연구결과에 의하면 특별한 몇가지 조건이 성립하는 경우 임의의 두 주어진 다면체 모델 사이에서 부드럽게 변화해 가는 progressive mesh 모델의 표현을 구할 수 있음이 알려져 있는 정도이다. 앞으로 이 문제의 해결이 progressive mesh를 게임 분야에서 이용하기 위한 문제점으로 남아 있다.

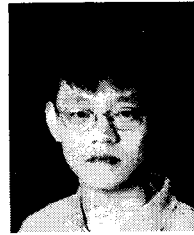
참 고 문 헌

- [1] J. Clark, Hierarchical Geometric Models for Visible Surface Algorithms, Communications of the ACM, 19(10), pages 547-554, 1976.
- [2] J. Rossignac and P. Borrel, Multi-Resolution 3D Approximation for Rendering Complex Scenes, Modeling in Computer Graphics: Methods and Applications, Springer-Verlag, pages 445-465, 1993.
- [3] W. Schroeder and et. al., Decimation of Triangle Meshes, Proc. of SIGGRAPH 92, pages 65-70, 1992.
- [4] H. Hoppe, Progressive Meshes, Proc. Of SIGGRAPH 96, pages 99-108, 1996.
- [5] H. Hoppe, View-Dependent Refinement of Progressive Meshes, Proc. of SIGGRAPH 97, pages 189-198, 1997.
- [6] D. Luebke and C. Erikson, View-Dependent Simplification of Arbitrary Polygonal Environment, Proc. of SIGGRAPH 97, pages 199-208, 1997.
- [7] M. Garland and P. Heckbert, Surface Simplification Using Quadric Error Metrics, Proc. of SIGGRAPH 97, pages 209-216, 1997.
- [8] J. Cohen and et. al., Appearance-Preserving Simplification, Proc. of SIGGRAPH 98, pages 115-122, 1998.
- [9] T. He and et. al., Controlled Topology Simplification, IEEE Tr. On Visualizations and Computer Graphics, 2(2), pages 171-184, 1996.
- [10] C. Touma and C. Gotsman, Triangle Mesh Compression, Proc. of Graphics Interface, 1998.
- [11] P. Alliez and M. Desbrun, Progressive Compression for Lossless Transmission of Triangle Meshes, Proc. of SIGGRAPH 2001, pages 195-202, 2001.
- [12] J. Ho and et. al., Compressing Large Polygonal Models, IEEE Visualization, pages 357-362, 2001.
- [13] M. Isenburg and S. Gumhold, Out-of-Core

Compression for Gigantic Polygon Meshes,
Proc. of SIGGRAPH 2003, pages 935-942, 2003.

[14] D. Luebke and et. al., Level of Detail For 3D
Graphics, Morgan Kaufmann Pub. 2003.

[15] D. Luebke and et. al., Advanced Issues in Level
of Detail, SIGGRAPH 2002 Course Notes 14,
2002.



최 정 주

- 1990년 한국과학기술대학 전산학과(학사)
- 1992년 포항공과대학교 컴퓨터공학과(석사)
- 1997년 포항공과대학교 컴퓨터공학과(박사)
- 1997~2000년 한국전자통신연구원 선임연구원
- 2000~2002년 (주)디지털아리아 선임연구원
- 2002~현재 아주대학교 미디어학부 조교수
- 관심분야 : 컴퓨터그래픽스, 기하모델링, 애니메이션