3D 표면데이터의 실시간 렌더링

(Real-time Rendering of 3D Surface Meshes)

고명철* · 최윤철*

1. 서 론

3D 컴퓨터 그래픽스 분야의 연구는 사실적 영상의 표현을 필요로 하는 다양한 응용분야의 요구에 부응하여 점차 그 중요성이 증가하고 있다. 이와 더불어, 최근 그래픽스 하드웨어 분야의 많은 발전은 과거에는 상상할 수 없을 정도의 렌더링 속도로 다양한 응용분야에서 요구하는 복잡한 영상들을 매우 사실적으로 표현할 수 있게 해주었다. 그러나, 역 공학(Reverse Engineering)[1] 개념이 3D 모델링 분야에 도입되면서[1], 3D 스캐너와 같은 정밀한 장비를 이용하여 생성된 객체의 복잡한 정도는 현재의 그래픽스 하드웨어의 처리수준을 벗어나고 있다. 즉, 그래픽스 하드웨어의 발달과 더불어 고해상도의 영상을 컴퓨터 상에 모델링 할 수 있게 됨에 따라 이에 따른 사용자의 요구사항도 점점 더 실 객체에 가까운 수준의 영상의 질을 요구하게 되어 하드웨어의 개발과는 별도로 수준 높은 렌더링 알고리즘에 관한 지속적인 연구가 필요하다. 이러한 알고리즘은 복잡한 영상의 렌더링에 드는 시간적인 지연을 개선하는 것은 물론, 영상을 구성하는 많은 양의 데이터를 효과적으로 변형, 조작, 관리하는 방안들에 대해서도 제시하여야 한다. 특히, 3D 컴퓨터 그래픽스를 기반으로 하는 가상현실 분야의 경우 매우 방대한 양의 데이터를 기반으로 하는데, 항해(Navigation) 시 일정 속도 이상의 렌더링 성능을 보장하기 위해서는 영상의 사실감 정도를 다소 떨어뜨리거나 상황에 따라 영상의 해상도를 다양하게 조절할 수 있게 하는 방법론이 필요하다.

본 논문에서는 3D 객체의 복잡한 표면을 나타내기 위한 대용량의 기하데이터를 효과적으로 단순화시키는 방법과 실시간 렌더링 성능의 극대화를 위한 다양한 기술적인 접근 방법들에 대해서 현재 진행되고 있는 연구들을 중심으로 그 특징과 개선 방안들에 대해서 설명한다.

2. 실시간 3D 컴퓨터 그래픽스

실시간 3D 컴퓨터 그래픽스 기술은 사용자가 자신의 반응에 따라 동적으로 만들어지는 3D 영상의 내부로 들어가 실세계와 유사한 다양한 체험을 가능하게 하는, 기존의 3D 컴퓨터 그래픽스 보다 진일보 된 개념으로서 가상현실을 위한 핵심 기술중의 하나이다.

3D 영상의 실시간 렌더링을 위한 알고리즘들은 영상을 구성하는 객체의 내부적인 구성방법에 의존적이므로 3D 객체를 모델링 하는 방법과 이를 기반으로 하는 실시간 렌더링의 주된 요소

*연세대학교 컴퓨터과학과

[1] 원래는 소프트웨어 공학의 한 분야로서 이미 만들어진 시스템을 역으로 추적, 해독하여 설계단계에서의 다양한 자료들을 얻어 내는 연구분야.

(factor)에 대해서 설명한다.

2.1 3D 객체의 모델링

3D 객체를 모델링 하는 방법은 크게 솔리드 기반(solid-based) 모델링과 표면기반(surface-based) 모델링으로 구분할 수 있다[10,11].

솔리드 모델링은 3D 객체 내부의 기하학적인 정보가 필요한 경우, 육면체나 구, 실린더 등과 같이 볼륨(volume)을 갖는 기본 객체들을 조합하여 객체를 모델링 하는 방법으로서 객체의 내, 외부를 수학적으로 명확히 구분, 정의할 수 있을 때 많이 사용하는 모델링 방법이다(그림 1). 따라서, 매우 정교한 모델링을 가능하게 하며 객체가 갖는 물리적인 특성을 분석할 필요가 있을 때 큰 위력을 발휘하게 된다.

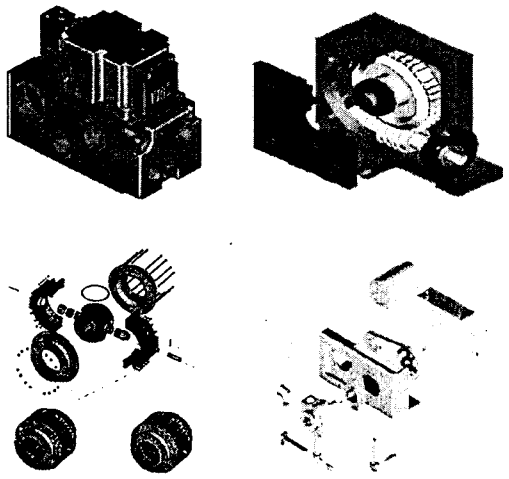


그림 1. 기계부품 설계분야의 솔리드 모델링

표면 모델링은 3D 객체의 외부 형상만을 모델링 하고자 할 때 사용하는 방법으로서 객체의 외부 표면을 수많은 다각형 조각(polygonal patch)들로 나누고 각각의 면을 구성하는 정점과 연결선의 좌표로 객체를 표현하는 방법이다(그림 2).

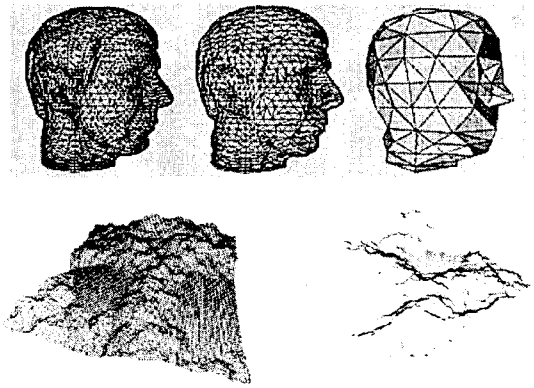


그림 2. 인체(얼굴) 및 지형 표현 분야의 표면 모델링

객체의 형상이 복잡하거나 표면이 일정치 않은 자유곡면 형태를 갖는 경우 이를 수학적으로 정의하기가 불가능하므로 연속된 다각형 조각을 이용하여 실 객체의 표면에 근사 시켜 표현하는 이러한 방법을 사용하게 된다.

표면 모델링 방법에서는 다각형 표면을 직접적으로 조작할 수 있기 때문에 3D 스캐너와 같은 장비를 이용해 얻은 다량의 데이터를 응용분야의 다양한 시스템 환경에 맞게 감소시켜 사용할 수 있다. 따라서 어느 정도 시각적인 사실감을 손해보더라도 상대적으로 렌더링 성능을 우선 시하는 응용분야에서 매우 효과적으로 이용될 수 있는 모델링 방법이다.

본 논문은 이러한 표면 모델링 방법에 의해 생성된 3D 객체를 가정하여 설명한다.

2.2 실시간 렌더링을 위한 요소: 정확성(Visual Accuracy) vs. 속도(Rendering Speed)

실시간 3D 컴퓨터 그래픽스 분야에서 영상의 표현을 위한 모든 연구의 궁극적인 목표는 시각적인 정확성과 이의 렌더링 속도간의 균형 점을 찾는 일이다. 정확성이라는 용어는 응용분야마다 그 문맥적인 해석이 다를 수 있다. 예를 들어 시뮬레이션이나 가상현실과 같이 사용자의 움직임에 따

라서 영상이 계속적으로 갱신되어야 하는 분야에서는 시각적인 사실감을 다소 떨어뜨리더라도 렌더링 성능을 높이는 것이 중요하다(그림 3)(그림 4). 즉, 모델링 된 객체의 시각적인 유사도를 지정하는 일정한 임계값을 두어 현재의 영상이 이 값을 만족시킬 경우 시각적으로 정확하다고 보는 것이다. 이러한 임계값은 응용분야 뿐만이 아니라 다양한 시스템 환경에 따라서도 유동적일 수 있다.

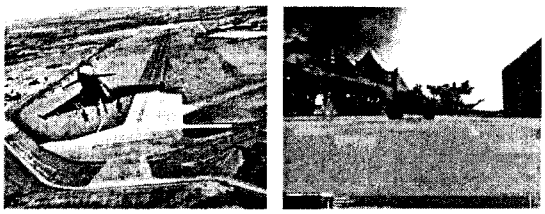


그림 3. 시뮬레이션(군사) 분야 영상의 사실감 정도

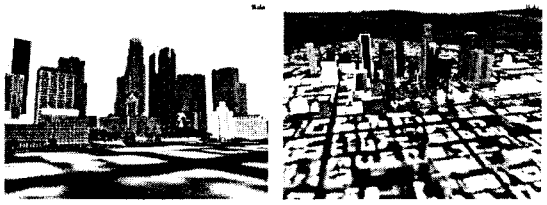


그림 4. 가상현실 분야 영상의 사실감 정도(Virtual L.A.)

최근에는 인터넷과 같은 네트워크 환경에서 3D 가상환경을 구축하려는 시도들이 늘고 있는데 이러한 분야에서는 대용량의 데이터를 실시간에 전송하여 영상을 렌더링 하는 것이 불가능하다. 따라서 영상의 해상도를 어느 정도 떨어뜨려 데이터의 양을 줄임으로써 전송시간이나 렌더링 성능을 높이는 것이 바람직하다.

세밀하게 모델링 된 3D 객체의 표면데이터를 지정된 임계 치 내의 해상도를 유지하면서 줄이는 방안이나, 같은 영상 내의 객체라도 시점(Viewpoint)과의 거리에 따라서 해상도에 차등을 두어 표현하는 기법들에 대한 다양한 연구들이 현재 진행 되고 있다. 이들 각각의 접근 방법들에 대해 다음 장에서 설명한다.

3. 메쉬 간략화(Mesh Simplification)

3D 스캐닝 시스템과 같은 모델링 툴의 발달과 더불어 복잡한 형태를 갖는 객체도 쉽게 3D 모델로 표현이 가능해짐에 따라 이들 대용량의 데이터를 메모리 관리나 전송, 렌더링의 효율을 높이기 위해 줄이는 방법이 필요하다.

메쉬란 연속적인 다각형 조각들로 구성된 선형 표면(linear surface)으로서 이를 이용하면 곡면 형태의 복잡한 3D 객체의 표면을 근사 시켜 표현할 수 있다. 메쉬를 구성하는 다각형은 보통 삼각형을 사용하는 것이 일반적이다.

메쉬 간략화는 이러한 삼각형들을 직접 조작하여 실 객체와의 근사도를 조절함으로써 다양한 해상도의 영상을 상대적으로 적은 양의 데이터를 이용하여 표현할 수 있게 하는 개념이다(그림 5).

3.1 메쉬 근사(Mesh approximation)

메쉬를 간략화 하는데 있어서 원본 메쉬와의

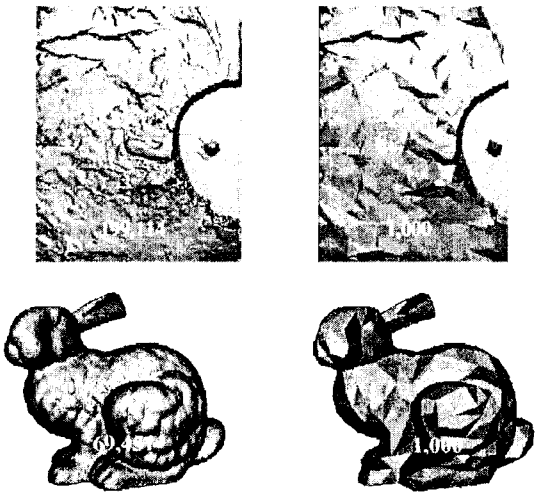


그림 5. 메쉬 간략화의 개념(숫자는 사용된 삼각형의 수)

시각적인 근사도를 높이기 위해서는 다음 두 가지 사항을 고려하여야 한다.

- Selection problem : 제거할 대상객체를 어떠한 기준으로 선택할 것인가?
- Positioning problem : 새롭게 생성된 정점을 어느 위치에 둘 것인가?

(그림 6)은 에지 기반 간략화 방법을 예로 들어 위 두 가지 문제를 설명하고 있다.

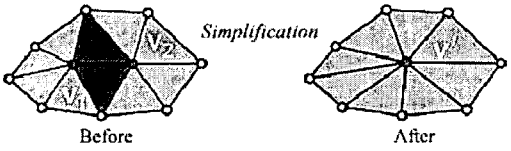


그림 6. 간략화 시 고려사항 : 제거 대상객체($\overline{V_1V_2}$)의 선택 및 새로운 정점(v')의 위치선정 문제

먼저, 간략화 시 제거할 대상을 어떻게 선택하는가 하는 문제는, 단순히 각 객체간에 의존도가 가장 작은 것들을 순서대로 지정한 수만큼 뽑아내는 방법을 생각해 볼 수 있다. 즉, 메쉬의 전 지역에 걸쳐 삼각형 망(triangular network)을 구성하는 정점이나 에지가 직접적으로 연결되어 있지 않으면서 지역적으로도 떨어져 있는 객체를 대상객체로서 선택하는 것이다. 이는 직관적이고 구현도 비교적 용이하지만 메쉬의 기하학적인 특징정보를 이용하지 못하는 단점이 있다. 예를 들어, 성긴(sparse) 지역의 정점이나 에지들은 지역적으로 인접해 있거나 삼각형 망이 연결되어 있더라도 이를 제거하였을 경우 전체적인 해상도에 영향을 주지 않기 때문에 제거하는 것이 바람직하다.

[3]에서는 동적인 우선순위 큐(priority queue)를 이용하여 제거할 객체를 선택한다. 큐에는 제거할 우선순위가 높은 에지들이 순차적으로 저장이 되는데 우선순위를 부여하는 기준은 다음과 같다. 메쉬 M을 다음과 같이 나타냈을 때,

$$M = \{K, V, D, S\} \quad (1)$$

(where, K: simplicial complex(mesh topology); V: set of vertex geometry; D: discrete attribute; S: scalar attribute)

M→M'으로 간략화 하는 과정에서 M'이 갖는 정확도를 다음과 같은 에너지 함수를 이용하여 표현한다.

$$E(M') = E_{dist}(M') + E_{spring}(M') + E_{scalar}(M') + E_{disc}(M') \quad (2)$$

(where, dist: distance between M and M'; spring: fairness(smoothness); disc: geometric accuracy of its discontinuity curves).

여기서 각각의 에지 축약(edge collapse) K→K'에 대해, 이 때의 계산 비용 $\Delta E = | E_{K'} - E_K |$를 다음의 식 (3)을 이용하여 계산하고 이 값이 작을수록 해당 에지가 인접한 지역의 해상도에 미치는 영향이 작다고 보아 높은 우선순위를 부여하는 것이다.

$$E_{K'} = \min E_{dist}(V) + E_{spring}(V) + E_{scalar}(V,S) + E_{disc}(V) \quad (3)$$

제거된 에지에 인접한 에지들의 에너지는 매번 새로 계산되어 동적으로 큐를 갱신 시킨다.

두 번째로, 간략화 시 새로운 정점의 위치선정 문제는 간략화 에러(Simplification error)를 최소화시키는 문제이다. 간략화 에러를 구하는 방법은 에러 메트릭(Error Metric) $E(M_{ref}, M_{cur})$를 정의하여 현재 메쉬(M_{cur})와 비교할 대상 메쉬(M_{ref})를, 원본 메쉬(M_{org})로 하느냐[9] 혹은 간략화의 각 단계에서 생성되는 이전 메쉬(M_{prev})로 하느냐[3,12]에 따라 메모리방식(memorial approach)과 비 메모리(memoryless approach) 방식이 있다

(그림 7). 이들 각각을 전역(global) 및 지역방식(local approach) 등으로 표현하기도 한다.

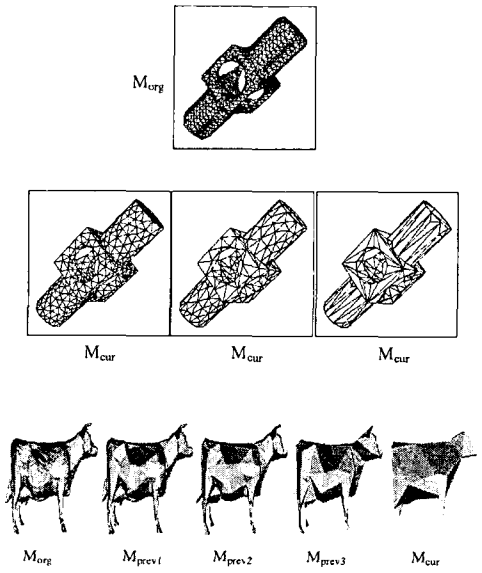


그림 7. 간략화 에러의 측정(Memorial(위) vs. Memoryless(아래) approaches)

메모리방식은 새로운 정점의 위치를 최적화하기 위해 원본 메쉬 상의 정점들을 참조하기 때문에 간략화 에러를 최소화시킬 수 있다. 그러나 많은 수의 정점들을 참조하여야 하므로 에러 계산을 위한 시간비용이 크다는 단점이 있다.

비 메모리 방식의 경우, 각 간략화 단계에서 생성되는 바로 이전 메쉬를 참조하기 때문에 상대적으로 적은 양의 정점들과 비교하게 되어 빠른 시간 내에 에러를 계산할 수 있다. 그러나 근사를 위한 명확한 에러 메트릭을 정의하지 못할 경우 간략화를 수행할수록 에러가 누적되어 원본 메쉬와의 근사도가 떨어질 수 있다.

[3]에서는 새로운 정점의 위치를 계산하기 위해 샘플링에 기반 한 에러 메트릭 정의방법을 사용한다. 이는 식 (3)의 첫 번째 에너지 항목(min $E_{dist}(V)$)을 다음의 식 (4)와 같이 정의하였을 때 최적의 위치 v_s를 찾는 것을 의미한다.

$$Dist(M_{cur}, v_i) = \min \| M_{cur} - v_i \|^2$$
$$E(M_{ref}, M_{cur}) = \sum Dist(M_{cur}, v_i)^2$$
$$v_s = \min_v E(M_{ref}, M_{cur}) \quad (4)$$

즉, M_{ref}로부터 샘플링 된 v_i(where, i=1, ..., n) 각각에 대해 M_{cur}와의 평방거리(square distance)의 합이 최소가 되게 하는 v_s를 찾는다. 이 방법은 최적의 위치를 찾을 수 있는 방법으로 알려져 있지만 구현이 어려운 단점을 가지고 있다.

3.2 간략화 알고리즘(Simplification algorithms)

현재 발표되고 있는 많은 메쉬 간략화 알고리즘들은 기본적으로 다음 네 가지 접근방법을 사용한다.

3.2.1 정점 제거 및 재삼각형화(Vertex removal/ re-triangulation)

적절한 평가함수를 정의하고 메쉬 상의 모든 정점들에 대해 각각의 에너지를 계산한다. 정점의 제거는 지정된 간략화 임계 치에 도달할 때까지 우선순위가 높은 것부터 차례로 수행한다. 정점을 제거하고 나면 해당 정점이 있던 자리에 다각형 구멍(polygonal hole)이 생기는데 적절한 삼각형 구성 알고리즘을 이용하여 이를 재구성한다(그림 8).

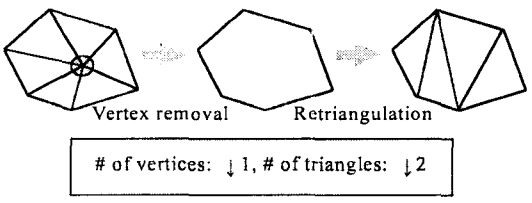
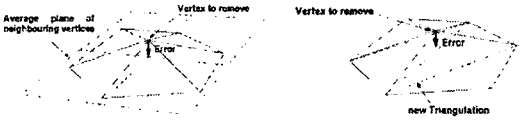


그림 8. 정점 기반 간략화 방법

[13]에서는 정점이 위치한 지역의 기하학적인 특징에 따라 정점들을 분류하고, 각 정점의 간략

화 에러를 (그림 9)와 같이 세 가지 방법으로 계산한 후 우선순위에 따라서 리스트에 유지시키는 방법을 이용한다.



$$curv(v) = \max_{i=(i,k)} \frac{\|n_i - n\|_1}{\|x_i - x\|_1}$$

$$\|y\|_1 = \sum_{i=1}^{k} |y^{(i)}|$$

그림 9. 간략화 에러의 측정(Vertex to average plane, Vertex to simplified mesh, Curvature approximation)

3.2.2 에지 축약(Edge collapse)

간략화 과정은 정점 축약방법과 비슷하나 제거되는 대상이 정점이 아닌 에지라는 것이 다르다. 또한, 정점 축약방법에서는 정점이 제거된 후 재삼각형화 과정이 필요했지만 여기서는 새로운 정점의 최적화 된 위치를 계산해 내는 정점 재배치 과정이 필요하다(그림 10).

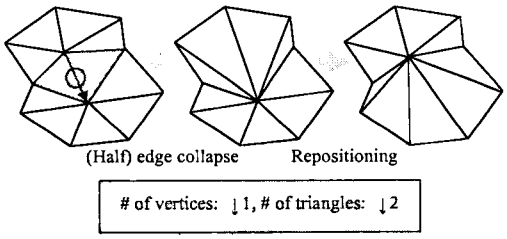


그림 10. 에지 기반 간략화 방법

일반적으로 에지 축약방법을 쓰지만 새로운 정점의 위치를 찾는 과정에서 간략화 에러계산의 효율을 위해 반-에지 축약(Half-edge collapse)방법을 쓰기도 한다.

앞서 3.1절에서 언급한 [3]은 에지 기반 간략화 방법의 대표적인 예이다.

3.2.3 삼각형 축약(Triangle collapse)

이 방법은 삼각형 단위로 간략화를 수행하기 때문에 한번에 제거되는 정보의 양이 많다. 따라서, 지정된 임계 치에 도달하는 간략화 속도가 다른 방법보다 빠른 장점이 있지만 제거 대상객체를 잘못 선택했을 경우 근사 에러가 커질 수 있다(그림 11).

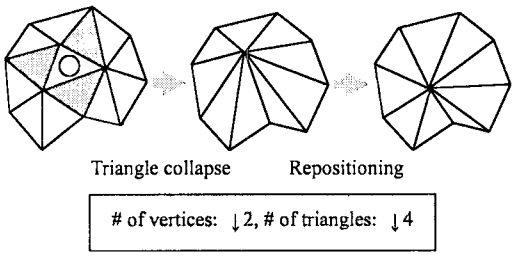


그림 11. 삼각형 기반 간략화 방법

[2]에서는 메쉬 표면의 곡률(Curvature)을 이용하여 제거할 삼각형들을 결정한다. 즉, 곡률이 작다는 것은 해당 지역의 기하학적인 복잡도가 작다는 의미이므로 이 지역의 삼각형들을 제거해도 원본 메쉬와의 근사도에 미치는 영향이 작다고 볼 수 있다.

3.2.4 정점 클러스터링(Vertex clustering)

이 방법의 기본적인 아이디어는 3D 메쉬 표면을 2D 평면에 투영(Projection)시켰을 때 충분히 작은 지역에 밀집해 있는 정점들은 그룹화 시켜 한 개의 대표정점으로 표현하는 것이다(그림 12). [5]에서는 이렇게 그룹핑 된 정점들을 각각 클러스터(Cluster)로 표현하고, 계속적으로 클러스터와 클러스터를 다시 그룹핑 하여 상위 클러스터를 생성하는 반복적인 처리를 통하여 계층구조 형태의 클러스터 트리(Cluster Tree)를 생성한다. 이 자료구조는 차후에 LOD(Levels Of Detail) 기반

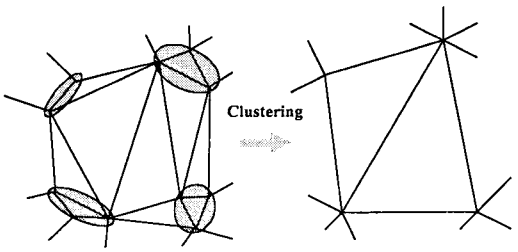


그림 12. 정점 클러스터링

렌더링이나 네트워크 상에서의 점진적인(progressive) 데이터 전송 등에 이용될 수 있다.

정점 클러스터링 방법은 비교적 단순한 알고리즘을 사용하면서도 간략화 성능이 좋은 것으로 평가된다. 그러나 2D 평면상에 투영시킨 결과만을 가지고 그룹핑을 수행하기 때문에 3D 메쉬의 깊이(z) 정보를 이용하지 못하는 단점이 있다. 또한 그룹핑의 규모를 결정하는 클러스터의 크기를 결정하는 문제도 대상 메쉬의 기하학적인 특성에 따라서 가변적일 수 있다.

3.3 삼각형의 재구성(Re-Triangulation)

간략화로 인해 정점, 에지 혹은 삼각형이 제거되었을 때 해당 객체의 주변에는 다각형 구멍이 생기게 된다. 따라서, 이 지역을 삼각형 메쉬로 재구성해 주는 작업이 필요하다. 3D 모델링 분야에서 사용하는 대표적인 재삼각형화 알고리즘으로서 거리 순차법(Distance ordering)과 델로니 삼각법(Delaunay triangulation) 등이 있다.

3.3.1 거리 순차법(Distance ordering)

다음 세 가지 과정을 반복한다.

- 모든 정점간에 거리를 구하고 가장 가까운 정점끼리 선으로 연결한다.
- 다음 번 가까운 정점끼리 선으로 연결하되 이전의 선과 교차할 경우는 제외한다.
- 모든 정점이 연결된 삼각형 망이 완성될 때까지 이 과정을 반복한다.

이 알고리즘은 개념적으로 단순하고 구현도 용이하지만 길쭉한(skinny) 삼각형들을 많이 만들어 내는 단점이 있다.

3.3.2 델로니 삼각법(Delaunay triangulation)

기하학 분야에 이미 잘 알려져 있는 델로니 삼각법을 모델링 분야에 이용한 것으로서 각 정점을 포함하는 보로노이(Voronoi) 다각형을 먼저 생성한 후에 이를 이용하여 델로니 삼각형을 정의에 맞게 생성한다.

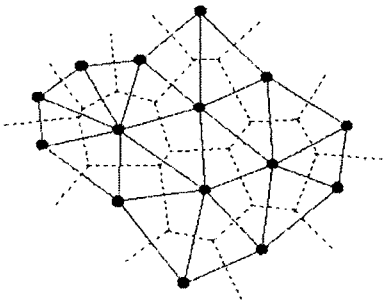


그림 13. Voronoi/Delaunay tessellation(점선/실선)

이 방법은 평평한(fat) 삼각형들을 많이 만들어 냄으로써 모델링의 관점에서 바람직하나 다음과 같은 두 가지 단점을 갖는다. 첫 번째는 델로니 삼각법을 만족시키는 삼각형간의 연결관계만을 중요시하기 때문에 근사하고자 하는 원본 메쉬의 기하학적인 특징 정보를 이용하지 않는다. 두 번째는 계층구조를 형성하지 못하므로 인해 LOD 기반 렌더링 분야 등에 적용이 어렵다.

따라서 현재 발표되고 있는 알고리즘들은 위 두 가지 방법 각각의 단점을 보완하기 위해 부분적으로 알고리즘을 수정하여 적용한다.

4. 다중해상도 메쉬표현(Multiresolution Mesh Representation)

원본 메쉬를 간략화의 정도에 따라 차등을 두어 여러 단계의 해상도를 갖는 메쉬로서 표현하는 기법을 의미한다. 이러한 기법은 응용분야가 갖는 특성 혹은, 다양한 시스템환경에 따라 매우 융통성 있는 렌더링 기법을 제시해 줄 수 있다. 일반적으로 LOD에 기반 한 렌더링 기법들이 다중해상도 표현방식으로서 많이 알려져 있다. 최근에는 단순 LOD 방식이 갖는 단점인 각 메쉬 간의 비연속성을 보완하기 위한 기법으로서 점진적 메쉬(Progressive Mesh)가 제안되었으며, 다시 점진적 메쉬 방식이 갖는 지역성(locality) 결여 문제를 해결하기 위한 선택적 정제(Selective Refinement) 기법들이 차례로 제안되었다. 이들 각각에 대해서 설명한다.

4.1 LOD(Levels Of Detail)

LOD를 기반으로 하는 대부분의 연구들은 (그림 14)와 같이 거리(distance)를 주된 인자로서 사용하고 있다. 그러나 효율적인 실시간 렌더링을 위해서는 좀 더 많은 영상의 구성 요소들을 LOD에 기반 하여 처리해 줄 필요가 있다. LOD 처리를 필요로 하는 주된 요소들은 다음과 같이 모두 다섯 가지로 구분할 수 있다[14].

- 거리기반 LOD(Distance LOD)
- 크기기반 LOD(Size LOD)
- 속도기반 LOD(Velocity LOD)
- 편심률기반 LOD(Eccentricity LOD)
- 깊이영역 기반 LOD(Depth of Field LOD)

이들 각각의 의미는 다음과 같다.

4.1.1 거리기반 LOD(Distance LOD)

시점과 영상 내 특정 객체간의 거리(Euclidian distance)에 따라서 가까이 있는 객체(d1)는 고해상도의 메쉬로서 표현하고 멀리 떨어져 있는 객체(d2)에 대해서는 저해상도의 메쉬로 표현한다(그림 14).

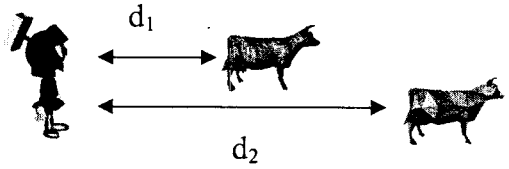


그림 14. Distance LOD

4.1.2 크기기반 LOD(Size LOD)

멀리 떨어져 있는 객체는 렌더링 시 2D 스크린에 투영되는 크기도 작고 저해상도로서 표현하고, 가까이 있는 객체는 크고 고해상도로서 표현하는 기법이다(그림 15(왼쪽)).

4.1.3 속도기반 LOD(Velocity LOD)

시야 각 내에서 이동 속도가 빠른 객체는 저해상도로서 표현하고 느린 객체에 대해서는 고해상도로 표현하는 기법이다(그림 15(오른쪽)).

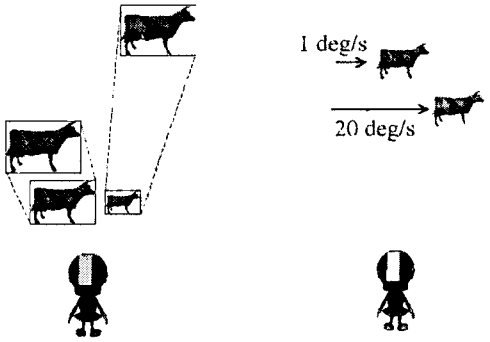


그림 15. Size LOD(왼쪽)와 Velocity LOD(오른쪽)

4.1.4 편심률기반 LOD(Eccentricity LOD)

시선의 정 중앙에 가깝게 위치한 객체는 고해상도로서 표현하고 이와 떨어져 있을수록 저해상도로 표현하는 기법이다(그림 16 (왼쪽)).

그림 16. Eccentricity LOD(왼쪽)와 Depth of Field LOD(오른쪽)

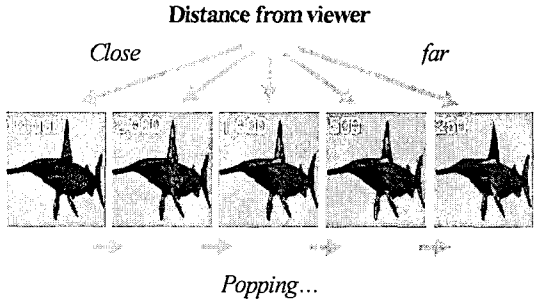


그림 17. 단순 LOD 기법의 문제점

4.1.5 깊이영역 기반 LOD(Depth of Field LOD)

인간의 시각시스템이 갖는 특징을 반영하기 위한 LOD 기법으로서, 객체를 바라볼 때 눈의 초점을 맞춰 망막에 정확한 상이 맺힐 수 있는 지역의 객체들은 고해상도로 표현하고 그 이내의 지역에 있어 초점을 정확히 맞출 수 없는 객체에 대해서는 저해상도로서 표현하는 기법이다(그림 16(오른쪽)).

4.2 점진적 메쉬(Progressive Mesh)

(그림 7)에서와 같이 간략화 수행의 결과로서 생성되는 메쉬 M_{cur}과 참조 메쉬M_{ref}는 각각 독립적인 객체로서 서로 직접적인 연관성(coherence)이 적다. 즉, 실시간 렌더링 분야에 이를 그대로 이용하였을 경우, $M_{ref}\rightarrow M_{cur}$로 화면 전환(switching) 시에 상당한 깜박거림(popping) 현상이 발생한다. 또한 역으로, $M_{cur}\rightarrow M_{ref}$ 전환이 가능하게 하기 위해서는 정제(Refinement)를 위한 추가적인 알고리즘과 자료구조가 마련되어야 한다. 단순 LOD 기법의 경우도 각 레벨로의 스위칭 시에 이러한 깜박거림 현상이 발생한다(그림 17). 따라서 $M_{ref}\leftrightarrow M_{cur}$으로 간략화 및 복원(Restoration) 하는 과정에서 메쉬 간의 연관성을 높여 부드러운 화면전환이 가능하도록 해야 한다.

점진적 메쉬[3]는 연속적인 LOD(continuous LOD)를 지원하는 기법으로서 다단계 메쉬 간의 부드러운 화면전환을 가능하게 한다. 식 (1)과 같이 정의 된 메쉬 상에서, 에지를 기본으로 각 에지를 축약했을 때의 에너지를 식 (2)와 같이 표현하고 축약 및 복원을 위한 두개의 연산자를 다음의 식 (5)와 같이 정의한다.

$$ecol\ (s_i,\ t_i)$$
$$vspl\ (s_i,\ l_i,\ r_i,\ A_i) \qquad (5)$$

(where, *ecol*: edge collapse operator; *vspl*: vertex split operator; s_i, t_i: two adjacent vertices constructing an edge; s_i: target vertex; l_i; r_i: one of vertex each of two triangle adjacent to restored edge; A_i: attributes).

(그림 18)은 점진적 메쉬의 전반적인 수행 개념을 나타낸 것이다.

점진적 메쉬를 위한 알고리즘은 다음 두 부분으로 구성된다.

- 분석 단계(Analysis phase(off-line)) : 메쉬 간략화를 수행하는 단계로서 각 간략화 단계에서의 상세한 기하 및 연결 정보들을 저장한다.
- 합성 단계(Synthesis phase(on-line)) : *ecol*/*vsptl* 연산자를 이용하여 실시간에 시점의 이동에 따라 영상의 해상도를 조절한다.

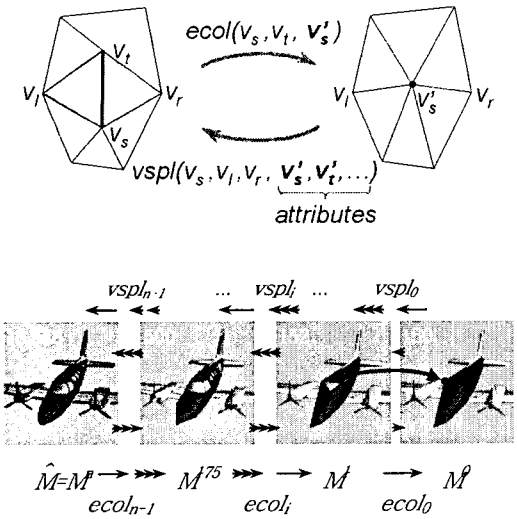


그림 18. 점진적 메쉬의 개념

렌더링 외에 점진적 메쉬가 갖는 또 다른 장점은 네트워크와 같은 환경에서 점진적인 메쉬데이터 전송이 가능하다는 것이다.

다량의 3D 데이터를 네트워크 환경에서 전송하는 것은 시간적인 부담이 크다. 점진적 메쉬를 이용하여 저해상도의 메쉬에서부터 차례로 전송하였을 경우($M^0 \rightarrow M^i \rightarrow ...$) 수신 측에서는 기다림 없이 바로 렌더링 할 수 있다(그림 19).

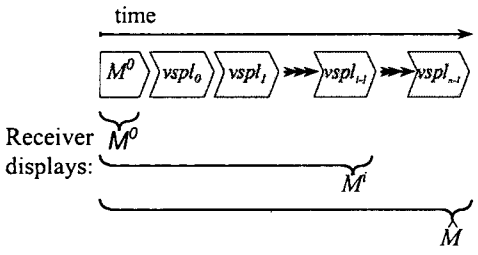


그림 19. 점진적 메쉬데이터의 전송

최근에는 기본적으로 점진적 메쉬 방법을 사용하면서 간략화 된 메쉬의 정확도나 렌더링 성능을 높이기 위해 식 (2)에서의 각 인자들을 수정, 보완하는 형식의 연구들이 많이 진행되고 있다[4,6,7,8].

4.3 뷰-기반/선택적 메쉬 복원(View-dependent/Selective Mesh Refinement)

점진적 메쉬나 연속적인 LOD 기법들은 각 메쉬 단계간의 부드러운 화면전환을 가능하게 하지만 메쉬 상의 모든 지역에 있어 동일한 해상도를 부여하기 때문에 여전히 개선의 여지가 남아있다.

뷰-기반 및 선택적 메쉬복원 기법[6,7,8]은 점진적 메쉬의 이러한 지역성 결여문제를 해결해 줄 수 있다. 이 기법은 메쉬의 각 지역에 대해 차별적인 해상도를 부여하는 방법으로서, 현재 스크린 공간상에 보여지는 지역에 대해서는 고해상도의 메쉬로 표현하고 그렇지 않은 지역에 대해서는 저해상도의 메쉬로 표현함으로써 전체적인 렌더링 성능을 높일 수 있게 한다(그림 20).

(그림 21)은 점진적 메쉬에 선택적 메쉬 복원 기술을 도입하여 각각의 정점들에 대해 선택적으로 *vspl*연산을 적용시키는 예를 보인 것이다. 활동 메쉬(*Active mesh*)[8]는 현재 스크린 공간상

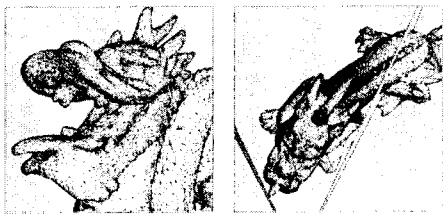


그림 20. 뷰-기반 메쉬 복원

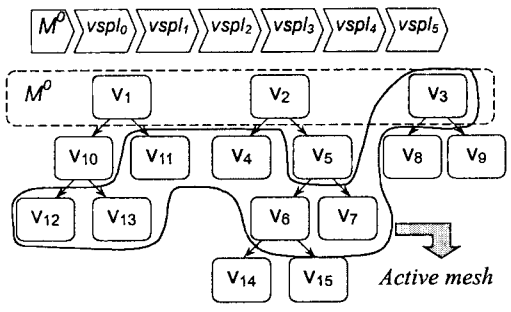


그림 21. 선택적 메쉬 복원

에 보이는 지역들을 유지하기 위한 자료구조로서 시점이 변할 때 마다 동적으로 갱신된다. 그림에서, $vspl_2$연산의 경우 v_3 정점을 v_8과 v_9로 분할(split)하는 연산으로서 현재 v_8과 v_9가 활동 메쉬가 아니므로 이 연산을 수행하지 않게 된다.

5. 결 론

3D 스캐너와 같은 장비를 이용하여 생성된 3D 모델은 매우 많은 표면 메쉬들로 구성되어 있다. 이러한 대용량의 데이터는 실시간 렌더링 분야에 사용할 경우 렌더링 성능, 저장공간, 데이터의 전송 등에 있어 많은 문제들을 야기 시킨다. 따라서, 시각적인 정확성(visual fidelity)을 잃지 않으면서 효과적으로 데이터의 양을 줄일 수 있는 방법론이 필요하다.

본 논문에서는 3D 객체의 복잡한 표면을 나타내기 위한 대용량의 기하데이터를 효과적으로 단순화시키는 기법과 실시간 렌더링 성능의 극대화를 위한 다양한 기술적인 접근 방법들에 대해서 알아보았다.

지금까지 살펴본 내용들은 (그림 22)와 같이 요약, 정리해 볼 수 있다.

현재 그래픽 하드웨어 분야의 많은 발전으로 인해 하드웨어 수준에서 제공하는 실시간 렌더링 성능이 과거에 비해 많이 향상되었지만 렌더링의 대상이 되는 3D 모델의 복잡도 역시 이에 비례하여 증가하고 있는 추세이다. 따라서 향후 하드웨어 성능이 계속적으로 향상되더라도 소프트웨어 차원에서 렌더링 알고리즘의 지속적인 개선은 불가피할 것으로 보인다.

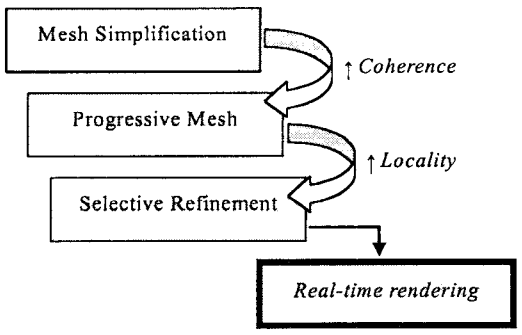


그림 22. 3D 객체의 실시간 렌더링을 위한 다양한 기술들

참 고 문 헌

[1] Varady, T, Martin, R. R. and Cox, J., Reverse engineering of geometric models-an introduction, *Computer-Aided Design*, Vol. 29, No. 4, pp. 255-268, 1997.

[2] Gieng, T.S., Hamann, B., Joy, K.I., Schussman, G.L. and Trotts I. J., Constructing hierarchies for triangle meshes, IEEE Transactions on *Visualization and Computer Graphics*, Vol. 4, No. 2, pp. 145-161, 1998.

[3] H. Hoppe, Progressive meshes, Proc. of SIGGRAPH '96, pp. 99-108, 1996.

[4] Swen Campagna, Hans Peter Seidel, Generating and Displaying Progressive Meshes, Proc. of 3D Image Analysis and Synthesis '97, pp. 35-42, 1997.

[5] Dieter Schmalstieg, Gernot Schaufler, Smooth Levels of Detail, Proc. of VRAIS '97, 1997.

[6] Xia, Julie and Amitabh Varshney, Dynamic View-Dependent Simplification for Polygonal Models, Proc. of IEEE Visualization '96, pp. 327-334, 1996.

[7] H. Hoppe, View-dependent refinement of progressive meshes, Proc. of SIGGRAPH '97, 1997.

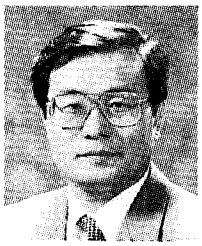
[8] H. Hoppe, Smooth View-Dependent Level-Of-Detail Control And Its Application To Terrain Rendering, Proc. of IEEE Visualization '98, 1998.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Mesh optimization, Proc. of SIGGRAPH '93, pp. 19-26, 1993.

[10] Tomas Moller, Eric Haines, Ream-Time Rendering, *A K Peters*, pp. 281-287, 1999.
[11] Alan Watt, 3D Computer Graphics, *Addison Wesley*, pp. 23-55, 1993.
[12] Michael Garland, Paul Heckbert, Surface Simplification Using Quadric Error Metrics, Proc. of SIGGRAPH '97, 1997.
[13] W.J. Schroeder, J.A. Zarge, and W.E. Lorensen. Decimation of Triangle Meshes, *Computer Graphics*, Vol. 26, No. 2, pp. 65-70, 1992.
[14] D. Luebke, J. Cohen, B. Watson, M. Reddy and A. Varshney, Advanced Issues in Level of Detail, Course #41, SIGGRAPH 2000, 2000.

고 명 철

• 1994년 제주대학교 정보공학과(공학사)
• 1997년 연세대학교 컴퓨터과학과(공학석사)
• 1997년~현재 연세대학교 컴퓨터과학과 박사과정
• 관심분야 : 3D 컴퓨터그래픽스, 실시간 렌더링, 가상현실



최 윤 철

• 1973년 서울대학교 전자공학과(공학사)
• 1975년 Univ. of Pittsburgh(공학석사)
• 1979년 Univ. of California Berkeley Dept. of IE & OR (공학박사)
• 1979년~1982년 Lockheed사 및 Rockwell International 사 책임연구원
• 1982년~1984년 Univ. of Washington 전산학과 박사과정
• 1990년~1992년 Univ. of Massachusetts 연구교수
• 1984년~현재 연세대학교 컴퓨터과학과 교수
• 관심분야 : 멀티미디어 문서처리(SGML/XML), 컴퓨터 그래픽스, 가상환경, Web Based instruction