

볼륨과 경계보존을 고려한 메쉬 간략화

조진화^{*0} 김성수^{**} 김영택^{***}

^{*} 경성대학교 컴퓨터 교육학과
^{**} 어비스

^{***} 경성대학교 정보과학부 컴퓨터과학 전공

Mesh Simplification for Volume and Boundary Preservation

Jin-Hwa Cho^{*0} Sung-Soo Kim^{**} Young-Taek Kim^{***}

^{*} Dept. Of Computer Education, KyungSung University

^{**} AVISS(Advanced Visualization System) Co.

^{***} Dept. Of Computer Science, KyungSung University

{jhcho, ytkim}@love.kyungSung.ac.kr, sskim@aviss.com

요약

대부분의 3차원 모델을 가시화해 주기 위한 시스템들은 삼각 메쉬를 기반으로 한 모델 표현법을 사용하고 있다. 이러한 시스템들은 복잡한 3차원 모델을 원격 액세스할 때에는 모델을 가공하지 않으면 전송 및 렌더링 시에 많은 제약이 따른다. 이러한 전송 시간과 렌더링 시간, 저장공간을 줄이기 위해 모델을 간략화하거나 압축하는 많은 기법들이 소개되어져 왔다. 본 논문에서는 메쉬 간략화를 위해 소개 되어진 많은 다른 방법에서 사용한 에지 축약(Edge Collapse) 기반의 간략화 알고리즘을 제시한다. 기본적인 접근 방법으로는 기하학적인 메쉬 구성요소들 중에서 에지의 특성에 따라 제거 기준을 설정한 후 간략화해 나간다. 에지 축약을 위한 우선순위와 축약 이후 새로운 정점의 위치는 원본 모델을 참조하지 않고 간략화 단계의 모델에서 삼각형의 연결성(connectivity)과 정점의 위치만을 이용하여 계산하였다. 실제 이 방법은 방대한 다각형 메쉬를 간략화할 경우 메모리를 효율적으로 이용할 수 있으며, 처리 속도를 개선할 수 있다. 또한, 간략화 이후에도 원본 모델의 위상(topology)을 유지할 수 있는 기법을 제시하고 있다. 새가지 모델에 대한 실험 결과에서 기하학적 에러(geometric error)를 최소화하였고, Heckbert가 제안한 방법(QSlim V2.0)과의 비교에서는 처리 속도가 개선되었다.

1. 서론

복잡한 삼각 메쉬는 컴퓨터 그래픽스와 가시화 등의 여러 분야에서 현실감을 더하기 위해 요구된다. 자동화된 모델 간략화는 이런 복잡한 삼각 메쉬를 원본 모델의 기하학적 구성요소를 적게 가지면서 유사한 모양으로 생성해주는 새로운 표현 방법이다. 이런 간략화 기술들은 의학용 데이터(CT, MRI), 레이저 스캐너, 컴퓨터 비전 알고리즘, 인공 위성 데이터 등으로부터 생성되어진 방대한 데이터를 처리하는 데 있어서 그 중요성이 높아지고 있고 데이터 저장, 전송, 계산에 드는 비용과 효율적으로 디스플레이 할 수 있다는 점에서 유용하다. 결과적으로 현재의 대부분의 연구는 이런 방대한 데이터 집합을 어떻게 관리할 것인가 초점을 두고 있다. 가장 잘 알려진 두가지 접근 방법으로는 메쉬 간략화와 기하학적인 압축기법이 있다[7].

메쉬 간략화 문제의 특성은 입력 데이터의 위상(topology)과 기하학적 속성, 간략화된 후 출력된 정점(vertices)의 범위, 출력의 삼각화(triangulation)의 구조, 구성요소의 근사화, 여러 축정법 등이 있다[3]. 백만개 이상의 삼각형으로 구성된 메쉬가 최근에 점차적으로 일반화 되어가고 있다. 따라서, 본 논문에서는 방대한 데이터 모델에 대한 간략화 처리에 초점을 두고 있다.

2. 관련 연구

메쉬 간략화의 목표는 적은 정점으로 면을 나타내더라도 원본 모델의 위상을 잘 유지하도록 좋은 근사(approximation)를 제공하는 것이다. 메쉬 간략화의 장점은 대용량 모델의 저장공간을 줄일 수 있고, 자료 구조 구축시간을 줄이며, 빠른 속도로 모델을 가시화할 수 있다는 것이다. 지금까지 상세도 제어를 위한 메쉬를 간략화하는 많은 알고리즘들이 연구되었다.

초기 간략화 방법들은 연속된 정점 제거를 통해 모델을 간략화해 나가는 알고리즘을 사용하였다. Schroedor는 메쉬의 지역적인 부분만을 고려하여 에러가 가장 적은 지역 정점을 제거한 후 재삼각화(re-triangulation)하는 방식의 정점 제거(Vertex decimation) 방법을 제안하였다[9]. 여기서, 정점에 대한 에러는 정점을 둘러싸고 있는 간략화 후의 근사된 면(average plane)과 정점사이의 거리가 된다. 이 방법은 간략화 과정에서 이전 정보(history)를 저장하지 않고, 메쉬 간략화를 위한 처리속도는 빠르지만, 이전의 간략화되어진 지역 모델을 사용하므로 에러가 누적될 가능성과 메모리 요구량이 크다.

반복적인 에지축약(Edge Collapse)을 이용하여 메쉬를 간략화하는 많은 방법들이 소개되었다. Hoppe는 간략화 모델을 생성하기 위해 에지 교체(edge swap)와 에지 분할(edge split)을 가진 에지축약 방법을 사용한 최적화(optimization) 기반의 알고리즘을 개발하였다[6]. 이 방법에서 메쉬 간략화를 원본 모델에서 샘플된 정점의 집합과의 거리를 전역적인 에너지 측정값으로 하여 에러를 최소화하는 에지축약 이후의 새로운 정점의 위치를 최적화 기술을 이용하여 해결한 방법이다. 이 방법은 좋은 간략화 결과를 제공하지만, 원본 모델을 참조해야 하므로 처리 속도면에서 느리다는 단점이 있다. 이 방법의 확장으로 간략화 과정에서 이전 정보를 저장하므로써 원래의 메쉬를 단순한 기본 메쉬(M^0)와 원본 메쉬로 되돌릴 수 있는 에지축약의 역연산인 정점분할(Vertex Split)의 시퀀스(sequence)들로 구성되는 Progressive mesh라는 메쉬구조를 제안하였다[4].

Garland와 Heckbert는 모델에서 연결되지 않은 영역들의 join(join)을 하고, 반복적인 에지 축약(contraction)에 기반한 Quadratic error metric을 이용한 면 간략화 알고리즘을 제시했다[2][8]. 그러나, 이러한 방법들은 계산이 복잡하고, 처리속도가 느리기 때문에 실제 대용량의 모델에 대해 적용했을때 부적합하다.

본 논문에서는 이전에 많은 연구들에서 사용한 에지 축약 방법을 사용하여 메쉬 볼륨과 경계 보존을 고려하여 메모리를 효율적으로 이용하여 처리 속도를 개선한 메쉬 간략화 기법을 제시한다. 에지 축약을 위한 우선순위 결정시에는 Hoppe가 제안한 최적화 프로시저에 의해 선택하는 것이 아니라 삼각형당 볼륨과 영역(area) 차이를 계산하여 결정함으로써 처리속도를 개선하였다.

3. 간략화 알고리즘

3.1 알고리즘 개요

모델을 구성하는 기하학적인 구성요소로는 정점과 에지, 그리고 삼각형이 있다. 본 논문에서는 에지축약을 이용하여 메쉬를 간략화한다. 에지축약은 메쉬에서 에지를 선택한 다음 하나의 정점으로 바꾸는 방법이다. 에지 축약 $ecol(v_0^e, v_1^e)$ 는 새로운 정점 위치 v 를 추가한 뒤 정점 v_0^e 와 v_1^e 를 제거한다. 제거 후 v_0^e 와 v_1^e 에 부사되는 모든 에지들을 v 에 연결한다. 에지축약은 새로운 정점을 위치시키므로써 원본 메쉬의 모양을 유지하는데 유리하고 별도의 삼각화(triangulation) 처리를 하지 않아도 된다는 특징이 있다. 그림 1에서 볼수 있듯이 에지 축약은 하나의 정점과 세개의 에지, 두개의 삼각형을 제거한다.

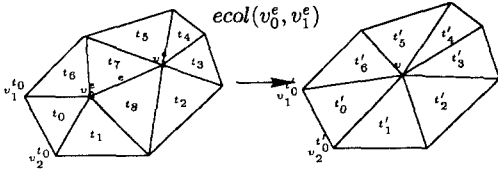


그림 1: 에지축약(Edge Collapse) 연산

메쉬 간략화를 수행하기 위해 다음과 같은 두 가지 주요한 과정이 있다.

- 에지축약 우선순위 결정
- 에지축약 이후 새로운 정점 위치선정

본 논문에서는 위 두가지 사항을 결정하는 데 있어 볼륨(volume)과 면 영역 정보를 이용하여 볼륨과 경계를 보존할수 있는 방법을 이용하였다.

3.2 정점 위치선정

본 논문은 에지 축약 이후 새로운 정점 v 의 위치를 결정하는데 있어서 메쉬 볼륨과 영역등의 기하학적 속성 변화를 최소화하는 정점의 위치를 선택하였다.

3.2.1 볼륨 보존

본 논문에서 다루는 삼각 메쉬의 속성은 매니폴드면(manifold surface)이라고 가정한다. 그림 1에서 삼각형 $t = (v_0^e, v_1^e, v_2^e)$ 는 에지 축약 후 새로운 삼각형 $t' = (v, v_1^e, v_2^e)$ 으로 구성되어진다. 먼저 모델의 볼륨을 변화를 최소화하는 에지를 결정하고 모델의 볼륨을 보존하기 위해 정점의 위치를 식 (1)과 같이 설정하였다[8].

$$\sum_i V(v, v_0^e, v_1^e, v_2^e) = \sum_i \frac{1}{6} \begin{vmatrix} v_x & v_0^e_x & v_1^e_x & v_2^e_x \\ v_y & v_0^e_y & v_1^e_y & v_2^e_y \\ v_z & v_0^e_z & v_1^e_z & v_2^e_z \\ 1 & 1 & 1 & 1 \end{vmatrix} \quad (1)$$

위 식을 v 에 대해 풀면, 다음과 같은 볼륨 보존 비용함수를 얻을 수 있다.

$$F_{vol}(e, v) = \sum_i (v_0^e \times v_1^e + v_1^e \times v_2^e + v_2^e \times v_0^e)^T v \\ = \sum_i n_i^T v = \sum_i [v_0^e, v_1^e, v_2^e]$$

여기서, n_i 는 삼각형 t 의 법선 벡터가 된다. 새로운 정점 선택은 에지 축약 이후 삭제될 두 삼각형 위의 랜덤한 후 두 정점들을 선택하여 각 정점을 위 식 (1)에 적용했을 때 가장 볼륨 변화를 최소화 하는 정점을 선택한다. 실험에서 새로운 정점을 제거 될 에지의 중점(mid-point)으로 선택하는 것 보다 기하학적 에러를 줄일 수 있었다. 모든 삼각형에 대해 위 방식으로 에지 축약을 해나가면 지역적인 볼륨 보존이 되므로 전체적인 볼륨 또한 보존할 수 있다.

3.2.2 경계 보존

모델이 지닌 경계(boundary)의 모양을 보존하기 위해 볼륨 보존에서 사용했던 방법과 유사한 알고리즘을 이용한다. 여기서는 에지 축약 이후 경계의 영역 변화를 최소로 하는 정점을 찾을 수 있다. 따라서, 평면 경계의 영역(area) 변화를 평가하는 비용함수 F_{boun} 를 설정할 수 있다.

$$F_{boun}(e, v) = \sum_i A(v, v_1^e, v_2^e) =$$

$$\left\| \sum_i \frac{1}{2} (v \times v_0^e + v_0^e \times v_1^e + v_1^e \times v) \right\|$$

여기서, 각 항은 경계 평면과 수직인 벡터가 된다. 각 벡터의 방향은 변화된 경계 영역의 부호를 결정한다. 그림 2는 에지 $e = e_1$ 이 v 로 축약될때 경계 e_0, e_1, e_2 의 영역 변화를 보여준다. 그림 2에서 수직선으로 채워진 부분은 에지 축약 이후 경계 영역이 감소한 부분이고, 수평선으로 채워진 부분은 경계 영역이 증가한 부분이다. 경계 영역의 변화를 최소화하므로써 경계 보존을 할 수 있다.

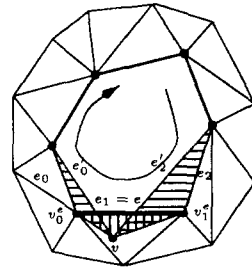


그림 2: 경계 에지 e 축약

3.3 에지축약 우선순위

메쉬 간략화를 위해 에지 축약의 우선 순위 결정은 중요한 과정이다. 원래의 메쉬에서 에지를 제거하기 위한 제거기준으로는 에지의 길이 $L(e)$, 두 정점의 차수(degree)의 합 $d(v)$ 가 있다. $d(v)$ 에 관한 내용은 하나의 정점이 제거 되어지면서 영향을 미치는 정점의 수를 최소화 하는 정점 차수법(Degree method)을 적용한 것이다. 이 기준들에 따라 각 에지에 해당하는 값을 구하는 함수를 비용함수(cost function)로 정의한다. 비용함수 $L(e)$ 에 대해서만 적용할 경우의 간략화된 결과는 전체 메쉬에 대해 에지가 골고루 제거가 되어 균등한 삼각형을 이루지만, 실제 간략화 레벨을 높이면 모델에서의 볼륨과 경계의 특성을 유지하기가 힘들어진다. 따라서, 에지의 주변 정보들을 전체적으로 고려해야 할 필요가 있다. 그러므로, 이전 섹션에서 언급했던 볼륨과 경계를 보존할수 있는 정보를 우선 순위 선정시 포함 시켜야 한다. 에지 우선 순위를 결정하기 위한 비용함수 F_C 를 최소로 하는 에지를 우선으로 한다. F_C 는 각각의 비용함수를 선형조합한 것이다.

$$F_C = \lambda F_{vol} \cdot d(v) + (1 - \lambda)L(e)^2 \cdot F_{boun} \quad (2)$$

여기서, λ 는 임계치 값(threshold value)으로 실험에서는 $\lambda = \frac{1}{3}$ 로 설정하였다.

4. 실험결과

본 논문의 실험은 Pentium-III-333MHz(M.M: 192MB)에서 OpenGL 라이브러리를 사용하여 C++로 구현하였다. 데이터는 Standford Graphics Lab.과 CMU에서 획득하여 실험하였다. 그림 3에서는 실험에 사용된 간략화 이전의 원본 토끼 모델을 보여주고 있다. 그림 4는 원본 토끼 모델에 대해 90% 간략화된 모델을 보여 주고 있다. 그림 3의 (b)와 그림 4의 (b)의 결과에서 메쉬 경계가 잘 보존되고 있음을 볼 수 있다. 그림 5는 1,087,716 개의 삼각형으로 이루어진 원본 모델(a)에 대해 90% 간략화된 모델(b)와 20,000개 삼각형으로 구성된 모델(c)를 보여주고 있다. 그림 7은 원본 드래곤 모델에 대해 90% 간략화된 모델을 보여 주고 있다. 처리 속도 비교를 위해 Heckbert가 제안한 기법[2](Qslim V2.0)으로 속도 비교를 수행하였다(표 1). 본 논문에서 객관적인 비교를 수행하기 위해서 간략화된 모델의 기하학적 에러

측정을 위해 Metro를 이용하였다[1]. Metro는 원본 메쉬와 간략화된 메쉬를 비교하여 기하학적 최대에러, 평균에러를 계산해 준다. 그림 6의 Qstim V2.0과의 최대 에러 비교결과는 토끼 모델에 대해 같은 제거율로 측정하였다.

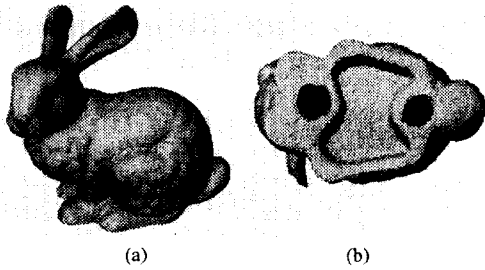


그림 3: 원본 토끼(bunny) 모델 : 35,947 vertices, 69,451 faces

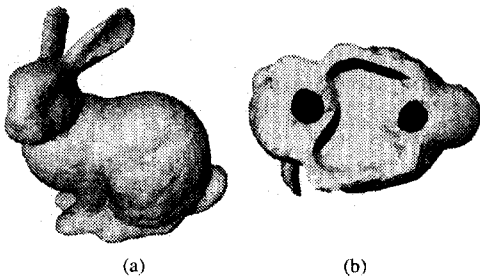


그림 4: 간략화된 토끼(bunny) 모델 : 4,620 vertices, 6,945 faces

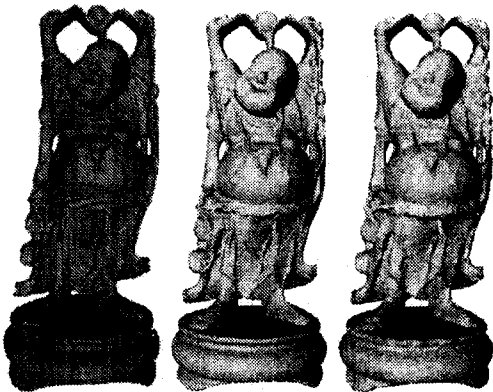


그림 5: 행복한 부처(happy-buddha) 모델 간략화

삼각형수	VBPSlim	QSlim
41670 개	8.13 sec	10.08 sec
27780 개	9.14 sec	10.64 sec
10325 개	10.02 sec	13.05 sec
6945 개	10.03 sec	13.13 sec
694 개	10.13 sec	14.16 sec

표 1. 간략화 수행시간 비교(VBPSlim: 본 논문의 알고리즘)

Maximum Geometric Errors

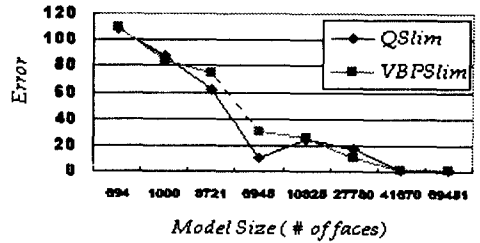
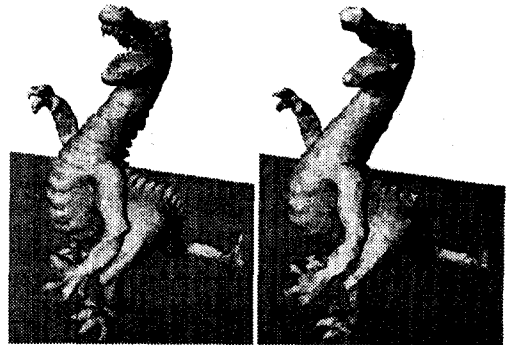


그림 6: 토끼 모델에 대한 최대 에러비교



(a) 25,418 vertices, 50,761 faces (b) 2,538 vertices, 5,000 faces

그림 7: 드래곤(dragon) 모델 간략화

은 메쉬 간략화 수행시 원본 모델이 아닌 간략화된 모델을 참조해 나가면서 모델의 볼륨과 경계를 최대한으로 보존할 수 있으며, 처리 속도 측면에서도 기존에 연구되었던 다른 방법들보다 개선된 성능을 보인다.

향후 연구 과제로는 네트워크 전송시 간략화된 모델을 삼각형 스트립(strip)으로 구성하고 정점 참조시 별도의 캐쉬를 사용하였을 때 캐쉬 적중률을 높일수 있는 삼각형 스트립 생성에 관한 연구가 이루어져야 할 것이다[5].

참고 문헌

- [1] R. C. Cignoni P. and S. R. Metro: Measuring error on simplified surfaces. Technical report, Istituto I. E. I.-C. N. R, Jan. 1996.
- [2] P. S. Heckbert and M. Garland. Surface simplification using quadric error metrics. In SIGGRAPH '97 Proc., Aug. 1997.
- [3] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. In SIGGRAPH '97 Course Notes., May. 1997.
- [4] H. Hoppe. Progressive meshes. In SIGGRAPH '96 Proc., pages 99-108, Aug. 1996.
- [5] H. Hoppe. Optimization of mesh locality for transparent vertex caching. In SIGGRAPH '99 Proc., Aug. 1999.
- [6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In SIGGRAPH '93 Proc., pages 19-26, Aug. 1993.
- [7] S. S. Kim, Y. S. Kim, M. K. Cho, and H. G. Cho. A geometric compression algorithm for massive terrain data using delaunay triangulation. In WSCG '99 Proc., pages 124-131, Feb. 1999.
- [8] L. P. and T. G. Fast and memory efficient polygonal simplification. In Visualization '98 Proc., pages 279-286, 1998.
- [9] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. Computer Graphics (SIGGRAPH '92 Proc.), 26(2):65-70, July 1992.

5. 결론 및 향후 과제

본 연구는 대용량의 다면체 삼각형 메쉬 모델을 간략화하기 위한 효율적인 메쉬 간략화 알고리즘을 제안하고 구현하였다. 본 알고리즘