

점진 스트립: 점진 전송을 위한 렌더링-효과적인 메쉬 표현 방법⁺

김병욱*, 양성봉*, 한탁돈*

*연세대학교 컴퓨터과학과

e-mail:kimbu@mythos.yonsei.ac.kr

Progressive Strips: A Rendering-Effective Mesh Representation for Progressive Transmission

Byung-Uck Kim*, Sung-Bong Yang*, Tack-Dong Han*

*Dept of Computer Science, Yonsei University

요 약

본 논문은 렌더링-효과적인 삼각형 메쉬의 점진 전송을 위한 새로운 메쉬 표현 데이터 구조(점진 스트립)를 제안한다. 점진 스트립은 기간 메쉬까지 단순화된 스트립과 이러한 스트립을 단계적으로 복원할 수 있는 일련의 상세 정보로 구성된다. 점진 스트립 기반의 메쉬 전송 전송에서 메쉬 연결 정보는 삼각형 스트립 상에서 직접적으로 복원되기 때문에 렌더링 과정에서 그래픽 가속기로 전송되는 데이터양을 크게 줄일 수 있다. 단순화된 스트립에서 발생하는 불필요한 과잉정보를 줄이기 위해 삼각형 스트립 표현의 간결함을 이용하여 스트립의 선축약 연산을 인코딩 하는 방법과 스트립의 정점 시퀀스를 재구성함으로써 불필요한 정점을 효과적으로 필터링하는 삼각형 스트립 필터링 알고리즘을 제안한다. 실험 결과 점진 전송동안 제안되었던 방법은 기존의 삼각형 기반 점진 전송 기법에 비해 메모리-그래픽스 프로세서 간 대역폭을 평균 40.5%~45.6% 줄인다.

1. 연구소개

3차원 모델의 원격지 접근에 대한 사용자 요구 증가에 따라 메쉬 전송과 그것의 인터랙티브 렌더링은 컴퓨터 그래픽스에서 중요한 문제가 되고 있다. 특히 네트워크 대역폭 또는 클라이언트의 렌더링 성능이 최대 해상도 모델을 전송하고 디스플레이하기에 불충분한 경우 메쉬의 대략적인 버전을 먼저 전송한 다음 일련의 상세 정보를 전송받아 메쉬에 상세함을 단계적으로 더하는 점진 메쉬(Progressive Meshes)와 그것을 기반으로 하는 점진 전송(Progressive Transmission)기법이 유용하다[5].

인터랙티브 시각화를 위해서는 클라이언트가 서버로부터 메쉬를 전송받는 속도뿐만 아니라 클라이언트가 전송받은 메쉬를 디스플레이하는 속도 또한 중요하다. 삼각형 스트립과 같은 효율적인 삼각형화 데이터는 효과적으로 그래픽 프로세서와 메모리사이의 데이터양을 크게 줄

일 수 있다[2], 주어진 메쉬에 대해 최적의 삼각형 스트립을 계산하는 것은 NP-complete 이며, 다양한 경험적 알고리즘에 제안되었다[2][10]. 하지만 그러한 알고리즘의 높은 계산 복잡성은 그것의 유용성을 오직 오프라인으로만 제한한다.

메쉬 위상 변화는 메쉬 스트립화의 변화를 요구한다[7]. 따라서 메쉬 위상 변화가 빈번하게 변하는 경우 메쉬 스트립화의 변화를 즉석에서 수행[6] 또는 수리[9]하거나, 직접적으로 삼각형 스트립을 갱신하는 방법[1]이 제안되었다. 특히 선축약 연산에 의한 메쉬 위상 변화는 스트립에서 정점을 반복함으로써 수행될 수 있다[8]. 이것은 전자의 두 방법이 메쉬 위상 변화에 따른 삼각형 인접 정보의 구축을 필수적으로 요구하는데 비해 후자의 경우 메쉬 연결성과 삼각형 스트립 사이의 연관성을 활용하기 때문에 그러한 추가적인 공간 및 연산 비용을 절약한다.

본 논문에서 우리는 삼각형 스트립 기반 점진 전송을

⁺ 본 연구는 국가지정 연구실 (NRL) 지원을 받아 수행되었음

위한 '점진 스트립(progressive strips)' 메쉬 표현을 제안한다. 기존의 점진 메쉬와 같이[5], 메쉬 해상도 변화는 선축약과 정점분할 변환에 의해 수행한다 하지만 이러한 연산은 삼각형 스트립 상에서 직접적으로 수행된다 삼각형 스트립 단순화를 위한 선축약은 [1][8]에서와 같이, 정점을 반복함으로써 수행한다 또한 선축약을 역으로 수행하기 위한 스트립-정점 분할을 위한 상세 정보를 계산하는 방법을 제안한다 스트립-선축약에 따른 불필요하게 반복되어지는 정점에 의한 과잉정보(redundancy)를 줄이기 위해, 단순화 스트립에 존재하는 퇴화 삼각형을 효과적으로 제거하는 필터링 알고리즘과 스트립의 표현 특성을 이용하여 동일 정점의 축척을 완화시킴으로써 복원에 필요한 상세 정보의 크기를 줄이는 스트립-선축약 연산의 인코딩 방법을 제안한다

본 논문의 실험 결과 점진 전송동안 제안되어진 기법은 삼각형을 개별적으로 전송하고 복원하는 기존의 기법에 비해 메모리-그래픽스 프로세서 간 데이터 전송량을 실험 데이터에 대해 평균 40.5%~45.6% 줄인다. 제안되어진 스트립-선축약 연산의 인코딩 기법을 통해 갱신되어야 하는 정점의 수를 평균 반 정도로 줄임으로써 상세 정보의 크기를 효과적으로 줄인다.

제 2절 점진 스트립

그림 1은 점진 전송의 메쉬 전이 연산자인 선축약과 정점분할의 예제를 나타낸다 기존의 점진 메쉬(그림 1.c)는 선축약에 의해 삼각형이 명확하게 제거된다 일반적으로 메쉬 M^i 에의 정점 p 가 q 로 축약되는 경우, 두 정점이 이루는 선(edge)에 인접한 삼각형 $\{(p,r,q)\}$ 와 $\{(p,q,s)\}$ 를 복원하기 위한 정점 분할을 위한 상세 정보는 다음과 같이 나타난다.

$$d_i = (v_q^i, (p^i, q^i, r^i, s^i)) \tag{1}$$

즉, 정점분할에 의해 추가되는 정점 q 에 대한 지형정보와 두 삼각형을 복원하기 위한 연결정보로 구성된다

이에 반해, 삼각형 스트립에서의 선축약은 정점을 반복함으로써 수행된다. 즉 메쉬 M^i 에의 정점 p 가 q 로 축약되는 경우, 스트립에 나타나는 모든 정점 q 를 p 로 대체하며 이것은 그림 1.d와 같이 퇴화삼각형을 이용한 암시적 제거를 수행한다. 역으로 정점 분할을 위한 상세 정보는 다음과 같이 표현된다.

$$d_i = (v_q^i, q^i, (o_1, \dots, o_r)) \tag{2}$$

여기에서 o_r 는 스트립에서 정점의 위치오프셋)를 나타낸다. 즉, 추가된 정점 q 에 대한 스트립을 복원하기 위해 갱

신되어야 하는 오프셋으로 구성된다

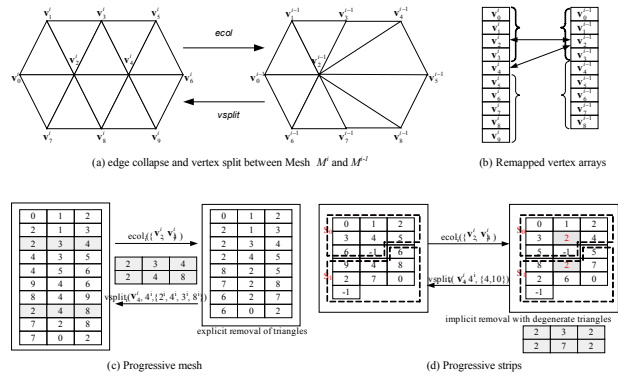


그림 1 선축약과 정점분할

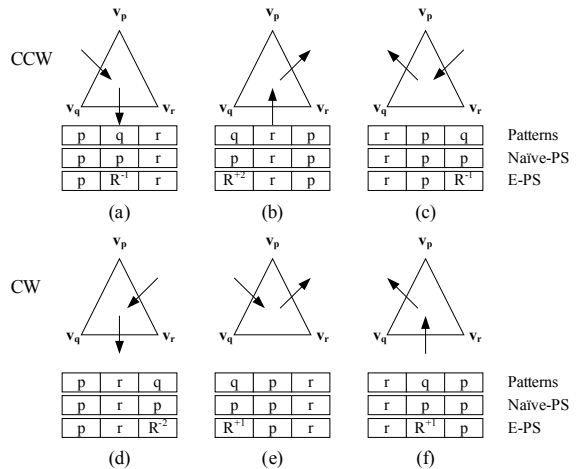


그림 2 스트립의 정점 시퀀스 패턴

하지만 정점 반복에 의한 스트립의 선축약은 스트립이 단순화될수록 동일 정점 축척에 의한 상세정보의 크기 증가와 퇴화삼각형을 통한 암묵적 제거(implicit removal)에 의한 렌더링에 불필요한 오버헤드가 발생한다

제 3절 스트립-선축약 인코딩

스트립 단순화에 따른 동일 정점의 축척은 역으로 메쉬 복원을 위해 갱신되어야 하는 오프셋의 수(식(2)의 r)를 증가시키며 이것은 상세 정보의 크기를 증가시킴으로써 메쉬 복원에 필요한 네트워크 대역폭을 증가시킨다

일반적으로 주어진 메쉬에서 선축약 되는 선은 스트립의 진행방향과 삼각형의 기술방향에 따라 그림 2와 같은 6가지의 경우 중 한 경우로 표현된다 따라서 그림과 같이 선축약 되는 두 정점의 오프셋 차이는 +2, +1, -1, -2 중 하나를 가지며, 4개의 기호($R^{+2}, R^{+1}, R^{-1}, R^{-2}$)로 인코딩될 수 있다. 그림 예제에서 인코딩 되지 않는 경우 (Naive-PS) 정점 q 가 p 로 축약되는 경우 2개의 p 가 축척되지만, 인코딩되는 경우(E-PS) 오직 하나의 p 만 존재한다. 따라서 다음 단계에서 p 가 r 로 축약되는 경우 2개의

정점을 갱신해야 하는 부담을 오직 하나의 정점만을 갱신함으로써 복원이 가능하도록 한다

제 4절 삼각형 스트립 필터링 알고리즘

전송되어진 점진 스트립은 일련의 상세 정보를 전송받음으로 오리지널 메쉬로 복원되며 동시에 그래픽스 시스템으로 전송되어 렌더링된다. 따라서 렌더링하면서 중간 단계 메쉬의 단순화 스트립에 존재하는 퇴화 삼각형을 효과적으로 줄임으로써 좀 더 디스플레이에 적합한 스트립으로 필터링하여 그래픽스 시스템으로 전송하는 기법이 필요하다. 제안되어진 필터링 알고리즘은 삼각형 스트립에서 연속하는 3개의 정점이 삼각형을 구성하며 그러한 정점들이 서로 다른 경우에만 오직 크기가 0 이상으로 최종 화면에 기여하는 삼각형을 생성한다는 사실을 기반으로 하여, 그래픽스 파이프라인으로 스트립을 전송하기 전에 정점 시퀀스에서 그러한 패턴만을 조사한다 이러한 퇴화 삼각형-프리 스캐닝(Degenerate Triangle-Free Scanning, DTFS)을 거쳐 살아남은 삼각형 패턴으로 그림 3과 같은 스트립 재구성 과정을 거친다 즉, 두 개의 연속하는 삼각형 패턴 (a b c)와 (d e f)의 관계를 다음과 같이 조사한다[11].

■ 정규 경우: 두 개의 연속하는 삼각형 패턴이 서로 다른 기술방향을 가지며 두 정점을 공유하면 (b=d 와 c=e), 두 삼각형은 삼각형 스트립 {a, b, c, f}로 재구성된다

■ 스왑 경우: 두 개의 연속하는 삼각형 패턴이 서로 같은 기술방향을 가지며 두 정점을 공유하면 (b=e 와 c=d), 두 삼각형은 삼각형 스트립 {a, b, c, b, f}로 표현된다. 여기에서 b는 스왑 연산을 위해 한번 반복된다

■ 분할 경우: 두 개의 연속하는 삼각형 패턴이 위의 두 경우에 해당되지 않는 경우, 하나의 스트립은 {a b c}로 끝나며, 새로운 스트립은 {d e f}로 시작된다

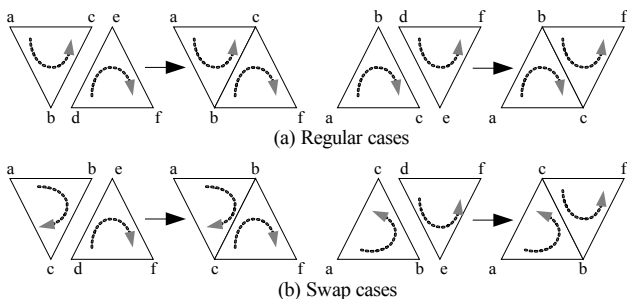


그림 3 스트립 재구성 방법

제 5절 실험결과

본 논문에서는 다양한 3차원 모델에 대해 점진 메쉬 기반 전송(PM)과 제안되어진 점진 스트립 기반 전송(PS)을 실험하였다. 표 2는 실험에서 사용된 3차원 모델

데이터 Cow, Skull, Dragon 그리고 Bunny의 메쉬 특성을 나타낸다. 기간 메쉬 복잡도는 오리지널 메쉬가 가지는 삼각형 수의 5%를 가지도록 하였으며, 표 2의 'Ecols'은 기간메쉬까지 수행된 선축약 연산의 수를 나타낸다. 실험에서 오리지널 메쉬는 STRIPE[3]를 사용하여 스트립화 되며, 단순화 오류 연산 기법으로 이차 오류 기준(Quadric Error Metric, QEM)[4] 방식을 사용하였다.

	Mesh characteristic				Ecols (or Vsplits)
	Original mesh (M^0)		Base mesh (M^1)		
	Vertices	Triangles	Vertices	Triangles	
Cow	2.9K	5.8K	129	254	2,722
Skull	11.0K	22.0K	495	1,104	10,412
Dragon	25.4K	50.7K	1,271	2,538	24,129
Bunny	35.9K	69.5K	1,756	3,471	33,077

표 2 실험 데이터

본 논문의 실험은 첫 번째 제안되어진 점진 스트립의 렌더링 효과성을 보이기 위해 점진 전송동안 요구되는 메모리-그래픽스 프로세서 간 데이터양을 실험한다 두 번째 제안되어진 선축약 연산의 인코딩 방법의 효과성을 설명하기 위해 갱신되는 정점의 빈도수와 네트워크 대역폭 감소율을 측정한다.

5.1 메모리-그래픽스 프로세서 간 데이터양

제안되어진 점진 스트립의 효율성은 점진 전송 동안 그래픽 프로세서와 메모리사이의 데이터양을 기존의 삼각형 기반 점진 메쉬에 비해 얼마나 줄이는 가로 나타내어진다. 또한 제안되어진 필터링의 우수성은 기존의 알고리즘으로 필터링 된 결과를 비교함으로써 측정되어진다 그림 4는 'Bunny' 모델을 전송하는 동안 측정된 결과이다. 다른 모델의 결과도 이와 거의 비슷한 결과를 보이기 때문에 생략한다.

그래프에서, 삼각형 기반 점진 메쉬의 경우 오리지널 메쉬로 복원할수록 필요한 대역폭이 선형적으로 증가한다 이는 구멍(hole)을 가지지 않는 2-manifold 메쉬의 경우 정점 분할 당 2개의 삼각형이 추가되기 때문이다. 필터링 되지 않은 점진 스트립(PS^{mf})은 전송 중간 단계에 있는 삼각형 스트립에서 퇴화 삼각형이 제거되지 않으므로 그러한 과잉정보를 가지는 정점 시퀀스가 그대로 그래픽스 파이프라인으로 보내어진다. 따라서 필요한 대역폭은 오리지널 스트립의 크기에 고정적이다 이러한 불필요한 대역폭은 삼각형 스트립 필터링 알고리즘을 통해 완화할 수 있다. 기존의 방법으로 필터링 된 결과(PS^{stss})[1] 보다 본 논문에서 제안되어진 알고리즘의 결과(PS^{dtfs})방법이 전체

적으로 우수한 성능을 나타낸다 특히 메쉬의 해상도가 낮은 경우 그 성능 차이는 더욱 커짐을 알 수 있다

결과적으로 본 논문에서 제안된 필터링을 통해 생성된 스트립은 점진 메쉬에 비해 실험 데이터 셋에 대해 40.5%~45.6%의 감소 효과를 얻을 수 있으며, 기존의 필터링 알고리즘에 비해 평균 21.3%~23.0%의 성능 향상이 있었다.

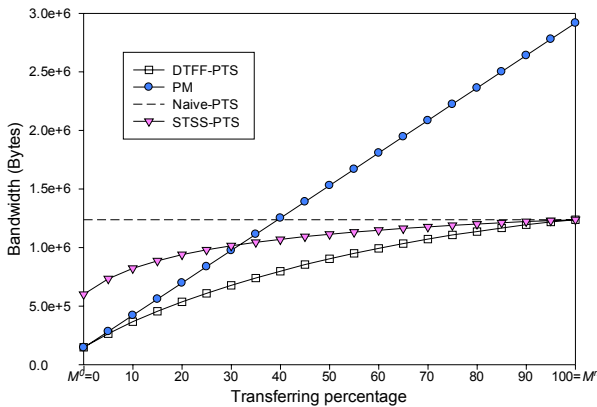


그림 4 메모리-그래픽 프로세서 간 데이터양

5.2 기호화의 효용성

단순화 연산으로 인한 동일 정점의 축척은 다음 단계의 상세함으로의 전이 시에 갱신되어야 하는 정점의 수를 증가시킨다. 이러한 동일 정점의 축척은 스트립선축약 인코딩에 의해 완화될 수 있다. 표 3에서 나타나듯이 49.9%~54.1%까지 그 빈도수를 줄이며, 식(2)에서 정점의 지형정보를 24바이트, 색인 및 오프셋을 나타내는 정수를 2바이트로 계산하는 경우 상세 정보의 크기를 평균 17.9% 줄인다.

	Cow	Skull	Dragon	Bunny
Naive-PS	6.6	7	6.7	5.9
E-PS	3	3.1	3.1	3
감소 비율	54.1%	55.2%	53.5%	49.9%

표 3 갱신되는 정점의 빈도수의 평균

제 6절 결론

본 논문의 주요 기여도는 메쉬의 위상이 연속적으로 변하는 점진 전송과 같은 그래픽스 시스템에서 삼각형 스트립과 같은 효율적인 삼각형화 데이터를 활용하는 것이다. 제안되어진 점진 스트립은 중간 단계 메쉬의 연결성을 삼각형 스트립 상에서 직접적으로 복원함으로써 둘 사이의 존재하는 연관성을 완전히 활용한다 이것은 클라이언트에서 지역적으로 메쉬의 복사본을 저장하거나 렌더링 단계에서 그래픽스 파이프라인으로 전송하는 데이터양

을 효과적으로 줄인다. 또한, 제안되어진 필터링 알고리즘과 선축약 인코딩 기법은 단순화된 스트립에 존재하는 과잉 정보를 효과적으로 줄인다.

우리의 연구는 원격지 컴퓨터가 복원에 필요한 상세 정보를 모두 다운받기 전에 중간 단계 메쉬의 렌더링을 요구하는 사용자 요구가 자주 발생하는 인터랙티브 그래픽스 시스템에 유용하다 또한 자원의 효율적인 활용은 지역 자원이 비교적 제약적인 모바일 환경에서 제한적인 렌더링 성능을 효율적으로 사용할 수 있게 해 줄 수 있다.

참고문헌

- [1] El-Sana J, Evans F, Skiena S, and Azanli E, "Efficiently Computing and Updating Triangle Strips for Real-Time Rendering", Computer Aided Design, 32(13), pp.753-772, 2000
- [2] Evans F, Skiena S, and Varshney A, "Optimizing Triangle Strips for Fast Rendering", IEEE visualization, pp.316-136, 1996.
- [3] Evans F, Azanli E, Skiena S, and Varshney A, Stripe Version 2.0 <http://www.cs.sunysb.edu/~stripe>
- [4] Garland M and Heckbert P, "Surface Simplification Using Quadratic Error Metrics", Computer Graphics (SIGGRAPH), pp.209-216, 1997
- [5] Hoppe H, "Progressive Meshes", Computer Graphics (SIGGRAPH), pp.99-108, 1996.
- [6] Hoppe H, "View-dependent refinement of progressive meshes", Computer Graphics (SIGGRAPH), pp.189-198, 1997.
- [7] Luebke D, Reddy M, Cohen J D, Varshney A, Watson B, and Huebner R, Level of Detail for 3D Graphics, Morgan Kaufmann, 2003.
- [8] SIGGRAPH '2000 Course Note, Advanced Issues in Level of Detail, 2000.
- [9] Stewart J A, "Tunneling for Triangle Strips in Continuous Level-of-Detail Meshes", Graphics Interface 2001, pp.91-100, 2001.
- [10] Xiang X, Held M, and Mitchell J S B, "Fast and effective stripification of polygonal surface models", In ACM Symposium on Interactive 3D Graphics, pp.71-78, 1999.
- [11] 김경화, 양성봉 "스킵스트립 렌더링을 위한 퇴화삼각형 제거필터", 정보처리학회 춘계학술발표 논문집(상) 제 10권 제1호, pp.398-396, 2003.