

PC 클러스터상에서의 병렬 LOD생성 알고리즘

황주영, 이종현, 김경호, 임상석, 박규호
한국과학기술원 전기 및 전자공학과 컴퓨터공학연구소
Email: jyhwang@computer.org

A Parallel Level-of-Detail Generation Algorithm on PC Clusters

Joo Young Hwang, Jong Hyun Lee, Kyung Ho Kim, Sang Seok Lim, Kyu Ho Park
Computer Engineering Laboratory, Electrical Engineering Department
Korea Advanced Institute of Science and Technology

요 약

그래픽스에서 많이 사용되는 메쉬의 복잡도는 점점 증가하여 실시간에 렌더링하기 불가능하기 때문에 Level-of-detail(LOD)를 사용한다. 기존 LOD 생성알고리즘들은 수메가 폴리곤의 메쉬를 렌더링하는 데 수십시간이 걸리는 한계가 있다. 따라서 본 논문에서는 병렬 LOD 생성 알고리즘을 제안한다. 이것은 메쉬단순화과정에서 병목이 되는 에지의 비용계산부분을 여러노드에 분산하여 계산하는 방법으로서, MPI를 사용하여 PC 8대의 PC 클러스터상에서 구현하여 단일노드에서보다 3~4 배 속도향상을 얻었으며, 노드를 더 추가하면 10 배까지 속도향상을 얻을 수 있다.

1. 서 론

3 차원 그래픽스에서 사용되는 메쉬모델은 날로 그 복잡도가 증가하고있다. 이러한 데이터를 실시간으로 렌더링하는 것은 도전적인 일이다. Level-of-detail(LOD)을 사용하면 렌더링이미지의 질을 크게 떨어뜨리지 않으면서 실시간으로 렌더링할 수 있기 때문에 대부분의 렌더링시스템에서 채택하고 있다.

복잡한 메쉬데이터로부터 LOD를 생성하는 것은 상당한 시간이 걸리기 때문에 전단계처리(preprocessing)를 하고 있다. 그러나 기존에 수 메가 폴리곤을 갖는 메쉬의 LOD를 생성하는 것은 수십시간이 걸리는 단점이 있다. 최근 3D scanner를 사용한 geometry reconstruction은 수십초 정도면 수 메가 폴리곤을 갖는 모델을 얻을 수 있다. 또 위성사진으로부터 얻는 지형모델은 그 복잡도가 수십 메가 폴리곤에 해당하는 방대한 양이다. 따라서 LOD 생성을 가속화하는 것은 점점 더 중요해지고 있다.

본 논문에서는 수 메가 폴리곤을 갖는 복잡한 모델의 LOD를 빠르게 생성할 수 있는 병렬 LOD 생성 알고리즘을 제안한다. 제안한 알고리즘은 PC cluster 상에서 MPI를 사용하여 구현되어, scalable한 speed up을 보인다.

본 논문의 내용은 다음과 같이 구성되어 있다. 2 절에서는 병렬 LOD 생성 알고리즘을 설명하고, 3 절

에서는 구현 및 실험결과를 제시하고 토의를 한다. 그리고 4 절에서는 결론을 내리고 더 할일을 제시한다.

2. 병렬 메쉬 단순화 방법

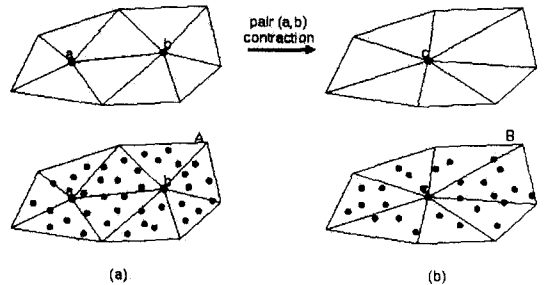


그림 1. Hausdorff distance의 계산

기존의 순차적인 메쉬 단순화 알고리즘들([1][2])은 복잡한 메쉬로부터 가장 적은 비용을 갖는 에지부터 하나씩 제거해감으로써 LOD를 생성한다. 에지의 비용은 그 에지를 제거함으로써 발생하는 근사화에러이다. 한 에지가 제거되면 그 에지 주변의 에지들의 비용은 다시 계산해야만 한다. 메쉬 단순화 알고리즘에서는 에지의 에러를 계산하는 시간이 90%이상이므로 이 부분을 병렬화하는 것이 절실하다.

우리는 에지의 비용을 원래의 초기 메쉬와 단순화 중

인 메쉬간의 Hausdorff 거리로 정의한다. Quadric error metric[2]는 빠른 반면 정확하지 않기 때문에 LOD의 품질이 나쁘다. Hausdorff 거리는 고품질의 LOD를 생성할 수 있다는 장점을 갖지만 Hausdorff 거리를 정확하게 구한다는 것은 많은 point sample을 취해야만 가능하고, 따라서 상당히 느리다. 그림 1에서와 같이 에지 (a,b)의 비용은 에지 (a,b)가 제거될 때 제거된 후의 메쉬와 초기 메쉬간의 hausdorff 거리를 근사화여러로 정의한다. A는 (a)의 메쉬에 해당되는 원래의 메쉬상에서 조밀하게 점을 샘플링한 점 집합이고, B는 (b)에서 조밀하게 샘플링한 점 집합이라고 하면 에지 (a,b)의 비용은 A의 점들로부터 (b)의 메쉬로의 거리의 최대값과 B의 점들로부터 (a)의 메쉬로의 거리의 최대값중에서 최대값이다.

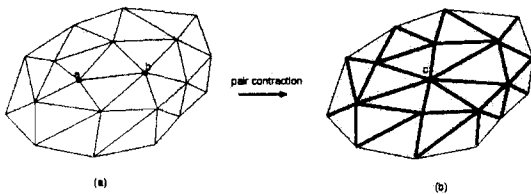


그림 2. 비용을 다시 계산해야 할 에지

비용계산에 드는 계산량을 병렬화하기 위해서 한 에지 e가 제거된 뒤 에지 주변의 에지들의 비용을 여러 노드에서 병렬로 계산한다. 비용을 다시 계산해야 할 에지들 Q는 e의 양 끝점에 연결된 에지들의 집합을 P라 할 때, P의 모든 에지의 양 끝점에 연결된 에지들의 집합이다. 예를 들면 그림 1의 경우에 대한 Q는 그림 2의 굵게 표시된 에지들이다.

병렬화 방법은 다음과 같다. 루트노드만이 전체메쉬를 갖고 있고 다른 노드들로 Q를 분산시켜주는 데, 이 때 전달하는 에지와 인접한 triangle들을 같이 넘겨준다. 이것을 받은 노드는 앞서 설명한 바와 같이 에지 비용을 계산하여 루트노드로 전달한다. 부하를 균등하게 하기 위해서 Q의 에지들을 라운드로빈방식으로 각 노드들에 분산시킨다. 취합하는 것도 분산시킬 때의 순서로 취합한다. Heterogeneous computing 환경에서는 보다 적응적으로 부하를 조절하는 방식이 필요할 것이다.

3.구현 및 실험결과

알고리즘은 C++로 구현되었으며, 병렬화를 위해서 MPI를 사용하고 있다. 사용한 MPI package는 MPICH 1.1.2 (<http://www-unix.mcs.anl.gov/mpi/mpich/>)이다. 현재까지 Pentium PC, Linux환경을 위한 실행프로그램을 release하였다.(<http://www-core.kaist.ac.kr/~jyhwang>)

실험환경은 Pentium PC 8노드로 구성되어 있다. 단일노드는 150MHz Intel Pentium processor, 32MB 메모리를 갖고 있으며, PC 간의 통신은 일반적인 Ethernet card를 통해서 이루어지고 있다. 한 대는 루트노드가 되어 전체적인 메쉬를 관리하고, 다른 7개의 processor들로 에지의 비용 계산을 분산하고 취합한다.

실험한 모델은 triceratops model과 sphere model이다. triceratops는 4906개의 triangle로 구성되어 있고, sphere는 5780개의 triangle로 구성된 구이다. 현재의 실험환경상에서는 수만개, 수백만개이상의 triangle을 갖는 모델을 메모리에 로드할 수 없어서 실험할 수 없었다. 루트노드를 제외한 다른 노드의 수를 1,2,3,...으로 바꿔 가면서 실험해본 결과를 표 1에 정리한다. 사용한 processor 수가 증가함에 따라 속도가 증가하는 것을 볼 수 있다. 그러나 정확히 프로세서수에 비례하여 스피드업이 증가하는 것이 아니라 포화하는 것을 알 수 있다. 이것은 메쉬단순화의 전체 계산량중에서 90%를 차지하는 에지비용계산부분이 병렬화가 되고, 10%인 노드간 통신시간은 병렬화되지 않았기 때문이다. 왜냐하면 메쉬가 루트노드에만 존재하기 때문에 통신을 병렬화할 수 없기 때문이다. 그림 3를 보면 실제로 에지비용계산부분은 노드 수 증가에 비례하여 속도가 증가한다는 것을 알 수 있다.

노드수	1	2	3	4	5	6	7
Triceratops	109.0	60.5	47.0	38.0	32.5	30.0	28.0
Sphere	130.5	72.7	58.0	45.2	39.7	37.0	34.5

표 1. 수행시간 (sec)

위와 같은 한계를 극복할 방법은 여러가지로 가능하다. 첫째 각 노드가 메쉬를 로컬메모리에 캐싱하고, 에지가 제거될 때마다 루트노드가 브로드캐스팅하여 주는 방법이다. 이 방법은 메모리가 각 노드마다 충분히 커야 하며, 메쉬를 각 노드로 복사하는 시간이 오버헤드가 된다. 둘째는 각 노드마다 메쉬를 분산하여 주고, 각 노드는 자신이 갖고 있는 메쉬에 대해서만 단순화를 수행하는 것이다. 이 방법은 [3]에서 구현되었으며, 좋은 scalability를 보여주고 있다. 그러나 노드간 경계부분은 단순화되지 못하기 때문에 LOD의 질은 크게 떨어진다. 셋째는 각 노드에 메쉬를 분산시키고 병렬로 에지제거를 수행하되 노드간 경계에 놓인 에지를 제거해야 하는 경우 노드간 통신을 하는 것이다. 이 방법은 노드간 통신이 병렬화 가능하다. 이 세번째 방법을 앞으로 구현할 계획에 있으며, 이를 통해서 scalable하면서 고품질의 LOD를 생성할

수 있을 것이다.

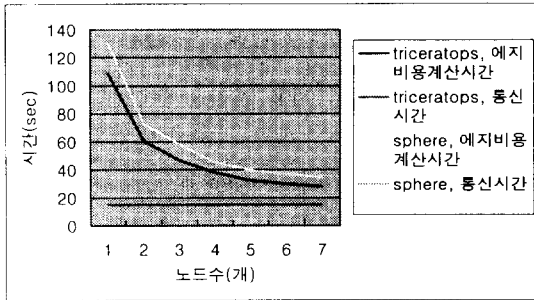


그림 3. 수행시간의 분석

4. 결론 및 추후과제

본 논문에서 제안한 알고리즘은 메쉬단순화에서 병목이 되는 에지비용계산부분을 병렬화하는 데 성공하여 8개의 노드를 사용하여 기존 LOD 생성알고리즘보다 3-4 배의 속도향상을 얻었다. 더 많은 수의 노드를 사용하게 되면 기존 알고리즘의 10 배까지의 속도를

낼 수 있게 된다. 또한 이러한 한계점을 극복할 수 있는 메쉬 분산화 알고리즘을 제안하였다. 앞으로 남은 일은 메쉬 분산화 알고리즘을 구현하는 것과, 수메가폴리곤 데이터에 대한 실험을 하는 것이다. 그리고 Heterogeneous 컴퓨팅환경에서 부하균등화방법을 도입하는 것이 필요하다.

참고문헌

- [1] Hughes Hoppe, "Progressive Meshes", Computer Graphics (SIGGRAPH '96 Proceedings),: 99-108 August 1996
- [2] Michael Garland and Paul S. Heckbert, "Surface simplification using quadric error metrics", Computer Graphics, Proceedings of SIGGRAPH '97, pages 209-216, ACM, 1997
- [3] J. El-Sana and A. Varshney, "Parallel Construction and Navigation of View-Dependent Virtual Environments", in SPIE '99 Conference on Visual Data Exploration and Analysis, Jan 23 - 29, 1999, San Jose, CA.

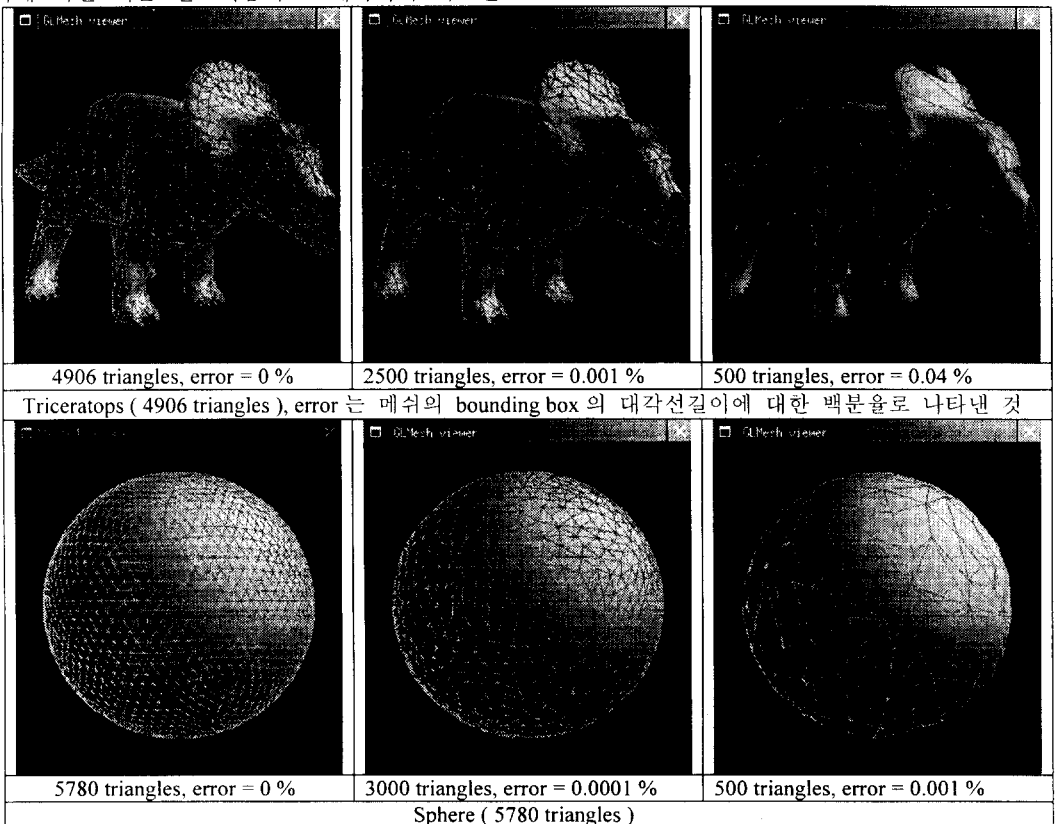


그림 4. 생성된 LOD