효율적인 점 기반 렌더링을 위한 확장 이차 오류 척도 기반의 간략화 방법 개발

김덕봉°, 강의철, 이관행, Renato B. Pajarola
광주과학기술원, University of Zurich
{eorka333°, eckang, lee}@kyebek.gist.ac.kr, {pajarola@acm.org}

Efficient Data Reduction for Point-Based Rendering using Extended QEM

Duck-bong Kim°, Eui-chul Kang, Kwan H. Lee, Renato B. Pajarola
GIST, Dept. of Mechatronics, University of Zurich, Dept. of Computer Science

요 약

본 논문은 효율적인 점 기반 렌더링(Point-based Rendering)을 위해 확장 이차 오류 척도(Quadric Error Metrics) 기법을 이용하는 간략화 알고리즘을 제안한다. 점 기반 렌더링의 기본 개념은 자유곡면을 메쉬와 같은 연결정보 없이 직접 점들로 표현하고, 렌더링하는 것이다. 확장 이차 오류 척도 기법은 메쉬를 간략화 하는데 있어 기하 정보뿐만 아니라 색상, 텍스쳐 좌표 정보까지 고려하여 간략화 하는 알고리즘이다. 이 연구는 3차원 점 데이터로부터 복원한 폴리곤 메쉬 모델로부터 효율적인 점 기반 렌더링(Point-based Rendering)을 위해 기하 정보 및 색상 정보까지 고려하여 원본 점 데이터를 간략화 하는 저용량의 효율적인 점 기반 렌더링 알고리즘을 제안하고, GPU 기반 렌더링 결과를 보였다.

1. 서 론

점 기반 렌더링 기법은 최근에 컴퓨터 그래픽스의 새로운 렌더링 분야로 많은 연구들이 이루어지고 있다[2,3,4,5,6]. 특히, 대용량 점 데이터를 처리하는 경우에 있어서나, 매우 복잡한 형상을 표현하는데 있어서는 점 기반 렌더링 기법은 메쉬 기반의 렌더링 기법보다 품질면에서나 렌더링 속도면에서 더 좋은 결과를 보인다[7].

최근의 3D 스캐너로 얻어지는 점 데이터의 경우에는 측정 속도 및 정밀도가 매우 향상되었지만 측정시 획득되는 점 데이터 량도 크게 증가되어 점 데이터의 저장, 전송, 처리 및 렌더링 등의 작업에 많은 시간과 비용이 소요된다. 따라서 점 데이터 감소에 관한 연구가 필요하게 되었고, 점 데이터를 효과적으로 줄이기 위해 연구가 선행되었다[2]. 기존에는 대용량 형상 정보의 실시간 디스플레이를 위해서 LOD(level of detail) 기법이 주로 이용되는데, 이는 복잡한 3차원 형상으로부터 해상도에 따라 다단계의 간략화 모델을 생성하는 방법이다. 또한 최근 기하 정보뿐만 아니라 색상 정보까지 획득할 수 있는 고정밀 레인지 스캐닝 시스템과 같은 장치들의 출현으로 컴퓨터 그래픽스와 기하학적 모델링에서는 매우 복잡한 모델을 흔하게 볼 수 있고, 이를 간략화 하여 LOD(level of detail)에 사용할 수 있는 3D 메쉬 간략화에 관한 연구가 활발히 이루어 졌다[1].

대다수의 점 기반 렌더링(Point-based rendering) 간략화에 관한 연구는 곡면의 법선 벡터나 곡률 등의 기하 및 위상 정보만을 고려하여, 원래 형상의 특징을 최대한 유지하면서 메쉬 곡면을 간략화 하는 알고리즘이 대부분이다. 이에 반해 본 논문에서는 기하 정보 외에 색상 정보까지 고려해서 모델을 간략화 하여 적은 점 데이터를 가지고 원본 모델과 품질면에서 차이가 없고, 렌더링 속도를 높일 수 있는 알고리즘을 제안한다. 본 연구에서는 3차원 점 데이터 기반의 복원된 폴리곤 메쉬 모델로부터 효율적인 점 기반 렌더링(Point-based Rendering)을 위해 기하 정보 및 색상 정보까지 고려하여 원본 점 데이터로부터 간략화 하여 효율적인 렌더링 알고리즘을 개발하고자 한다.

2. 기본 알고리즘

3차원 스캐너로 획득된 기하 정보 및 색상 정보를 갖는 점 데이터를 이용해 폴리곤 메쉬로 모델링한 후 확장 이차 오류 척도(Quadric Error Metrics) 방법으로 점 데이터의 개수를 줄인 후, 점 기반 렌더링을 하기 위해 각점의 노말 값과 반지름 값을 구한 후 점 기반 렌더러로 렌더링을 한다. 다음은 본 연구의 기본적인 알고리즘은 나타낸다.

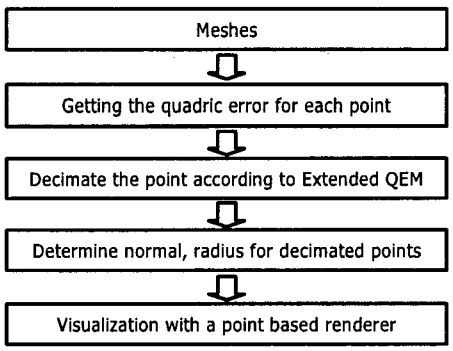


그림 1. Overall Algorithm

2.1. 확장 이차 오류 척도를 이용한 간략화

본 연구에서는 [1]의 알고리즘인 확장 이차 오류 척도를 이용하여 메쉬로부터 기하 정보뿐만 아니라 색상 정보도 고려하여 우수히 많은 점 데이터 량을 가진 모델을 효율적으로 간략화 하였다.

주어진 정점에 대해서 그것을 공유하는 삼각형의 집합이 있고, 각 삼각형은 자신을 포함하는 평면의 방정식을 가지고 있다. 정점을 옮기는 것에 대한 비용 함수는 각 평면과 새로운 위치에 대한 거리 제곱의 합이다. 아래의 식은 새로운 위치 v와 평면 m에 대한 비용함수이다. 여기서 n은 평면의 법선 벡터이고, d는 원점으로부터의 오프셋이다.

$$c(v) = \sum_{i=1}^{m} (n_i \cdot v + d_i)^2 \qquad (1)$$

식 (1)과 같이 기하 정보기반으로 간략화를 수행하는 이차 오차 척도에 색상 정보가 추가된 간단한 이차식으로 확장하여 표현할 수 있다.

2.2. Point-sampled geometry

간략화 된 점 샘플들을 점 기반 렌더링을 하기 위해서는 좌표, 색상, 노말, 반지름 정보를 가지고 있는 서펠(surfel)[5]을 만들어야 한다. 모델의 효과적인 라이팅을 위해 노말 정보가 필요하고, 반지름 정보는 오브젝트 공간에 있는 점들이 이미지 평면상에 splatting 되면서 생길 수 있는 구멍을 피하기 위해 필요하다. 간략화 된 점 샘플들의 노말 값과 반지름 정보를 구하기 위해 계층적 공간 분할(hierarchical space partitioning) 기법의 하나인 kd-tree를 이용하여 3D 공간상에 있는 점 데이터를 분할하였으며, 각 정점의 K-nearest 이웃점을 찾고, Covariance-based method를 이용해서 각 서펠(surfel)의 노말 값과 반지름 값을 구할 수 있다[6].

2.3. GPU 기반의 Surface Splatting

GPU를 기반으로 한 point splat의 렌더링에 있어서 몇 가지 고려해야 할 사항은 다음과 같다. 첫 번째 splat의 크기와 형상을 결정해야 한다. Splat의 크기는 이미지 평면상에서 홀(hole free)을 피하기 위한 것이다. Splat의 형상은 이미지 평면상에서 surfel이 얼마나 큰 타원형을 나타내는가에 따라 결정지어 진다.

더 좋은 품질을 생성하기 위해서는 splat간에 겹치는 부분을 가우스 필터를 이용해 블렌딩(blending) 해주어야 하는데, 현재의 GPU로 한번에 ε-depth test를 수행할 수가 없기 때문에 2-pass 렌더링을 한다. 첫 번째 pass에서는 이미지(scene)가 z-buffer에 저장되었다가, 두 번째 pass에서는 각 splat의 깊이정보 값이 ε보다 작게 되면 픽셀에 splat 값이 저장된다. 한 픽셀에 블렌딩된 splat의 색상 값들은 각 픽셀에 저장이 된다. 따라서 각 픽셀들은 $\sum_i w_i(rgb)_i$을 저장하게 된다. 노멀라이제이션 과정에서는 식 (2)와 같이 각 픽셀에 저장된 색상 값들을 weighted sum으로 나누어 준다.

$$\sum_i w_i(rgb)_i / \sum_i w_i \qquad (2)$$

3. Implementation

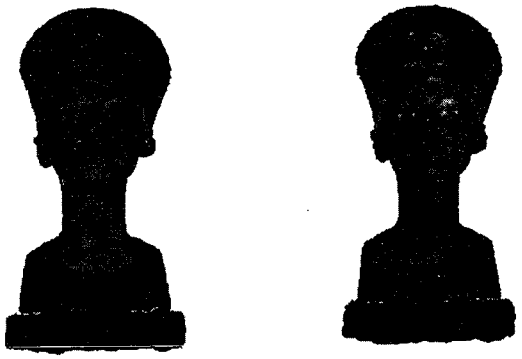
본 알고리즘은 Pentium IV, 2 GB RAM, nvidia GeForce 6800 윈도우환경 PC에서 Visual C++ 6.0과 OpenGL을 사용하여 구현하였다. 또한 효과적인 구현을 위해서 Programmable vertex shader 및 pixel shader를 이용하였으며, ARB_vertex_program 과 ARB_fragment_program extension을 이용하고, 플랫폼에 독립적인 Cg 언어를 사용하였다.

4. 실험결과

표 1. 렌더링 성능(frame per second)

Model	% of the Nefertiti	Point based rendering		
		1-pass	unfiltered	filtered
Nefertiti points: 445k	100	17	15	8
	75	31	31	13
	10	230	62	32
	3	530	70	42
	1	1700	700	58

표 1은 445k 개의 점을 가지고 있는 Nefertiti 모델을 점 기반 렌더링한 결과이다. 원본데이터의 점의 개수가 줄어들면서 렌더링 스피드는 빨라지는 것을 표.1을 통해 알 수 있다. 1-pass 렌더링의 경우에는 블렌딩 및 노멀라이제이션 과정 없이 렌더링이 이루어 졌기 때문에 렌더링 속도가 가장 빠른 것을 볼 수 있다. unfiltered의 경우에는 노멀라이제이션 과정이 없기 때문에 비교적 빠른 것을 보여주고, 이에 반해 filtered의 경우에는 블렌딩 및 노멀라이제이션 과정이 있기 때문에 가장 느린 것을 보여준다.



(a) Simplifying geometry (b) Simplifying geo. & color

그림 2. 3% data model rendered with a point splatting renderer.

그림 2는 확장 이차 오차 척도 기법을 이용하여 원본 모델의 데이터 크기를 약 3%까지 줄여 점 기반 렌더러로 시각화한 것이다. 기하 정보만을 고려하여 점을 줄인 모델의 렌더링 결과는 모델 (a)와 같이 눈 주위가 흐릿한 결과를 보여준 반면, 기하 정보 및 색상 정보까지 고려하여 점을 줄인 모델의 렌더링 결과는 모델 (b)와 같이 눈 주위가 뚜렷한 결과를 볼 수 있다.

그림 3과 그림 4는 확장 이차 오류 척도 기법을 이용해서 메쉬로부터 간략화 된 모델의 결과를 점 기반 렌더링한 결과를 보여준다. 원본 모델의 약 10% 까지는 원본의 모델과 형상이나 색상에 있어서 거의 차이가 없는 반면, 표 1에서 볼 수 있듯이 렌더링 속도에 있어서는 4배에서 약

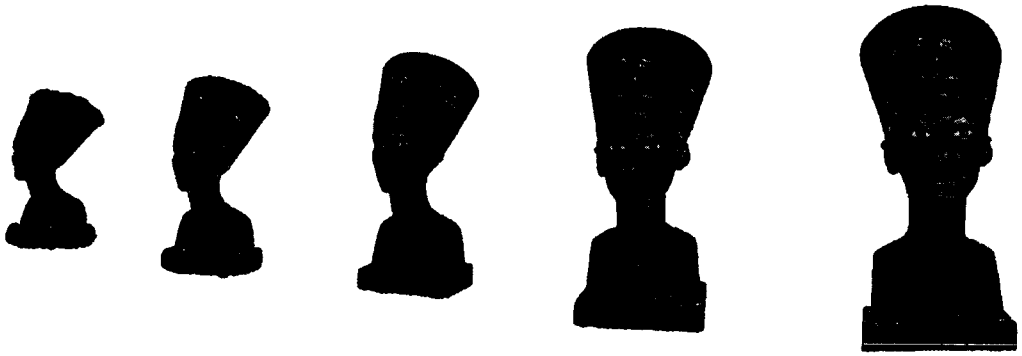


그림 3. Nefertiti at different levels-of-detail. From 1, 3 10, 25 and 100% points for original model, 448k points, rendered with a point splatting renderer, simplified by Extended QEM.

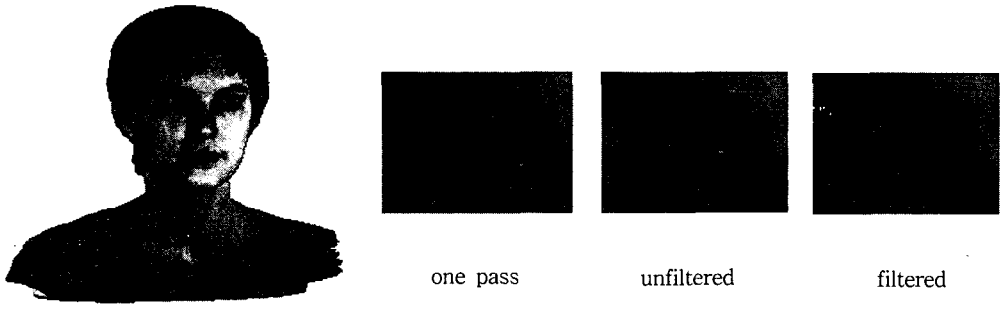


그림 4. female model simplified by geometry & color from 300k point, rendered with a point splatting renderer with 30k points.

15배까지 빨라진 것을 볼 수 있다. 하지만 원본 모델의 약 10%보다 적은 점의 수를 가지고 렌더링하면, 속도는 더욱 더 빨라지지만, 품질은 나빠지는 것을 볼 수 있다. 즉, 렌더링 속도와 품질은 서로 상충 되는 것을 볼 수 있다. 그림 4는 각각 1-pass 렌더링한 결과와 unfiltered 된 결과와 filtered된 결과를 시각적으로 비교할 수 있다. unfiltered된 경우에는 splat이 겹치는 부분에서 노멀라이제이션 과정이 없기 때문에 밝기가 달라지는 시각적 오류가 발생하는 것을 볼 수 있다.

5. 결론 및 향후과제

본 논문에서는 기하 정보뿐만 아니라 색상 정보까지 고려하는 확장 이차 오류 척도 기법을 사용하여 복잡한 모델을 간략화를 하였고, 점 기반 렌더러로 렌더링한 결과 적은 샘플을 가지고도 품질면에서나 속도면에서 향상된 렌더링 결과를 얻을 수 있었다. 향후 연구과제로는 연결정보가 없는 점 샘플로부터 간략화 할 수 있는 기법을 개발하려고 한다.

Acknowledgement

이 논문은 한국과학재단에서 지원한 해외공동연구과제 수행결과입니다. 본 연구는 광주과학기술원 실감방송 연구센터를 통한 정보통신부 대학 IT 연구센터(ITRC) 사업의 지원과 광주과학기술원 실감 콘텐츠 연구센터(ICRC)를 통한 과학기술부 특정 연구개발 사업의 지원에 의한 것입니다.

6. 참고문헌

1. M.Garland and P.Heckbert, Simplifying surfaces wit color and texture using quadric error metrics, IEEE Visualization '98, p263-269, October 1998
2. M.Pauly, M.Gross, L.P.Kobbelt, Efficient Simplification of Point-Sampled Surfaces, In Proceedings EuroGraphics Workshop on Rendering, 2002
3. M.Botsch and L.Kobbelt. High-quality point based rendering on modern GPUs. In Proceedings Pacific Graphics 2003, pages 335-343. IEEE, Computer Society Press, 2003.
4. R.B.Pajarola, M.Sainz, P.Guidotiti. Confetti Object-Space Point Blending and Splatting. IEEE Transaction on Visualization and Computer Graphics, 2004.
5. H.Pfister, M.Zwicker, J.Van Barr, and M.Gross. Surfels: Surface elements as rendering primitives. In Proceedings SIGGRAPH 2000, p 335-342. ACM SIGGRAPH, 2000
6. M.pauly, Point primitives for Interactive Modeling and Processing of 3D Geometry, Ph.D. dissertation, Federal Institute of Technology(ETH) of Zurich, Zurich, 2003
7. M.Levoy, K.Pully, B.Curless, S.Rusinkiewicz, D.Koller,, L.Pereira, M.Ginzton, S.Anderson, J.Davis, J.Ginsberg, J.Shade, and D.Fulk. The digital Michelangelo project: 3D scanning of large statues. In Proceedings SIGGRAPH 2000, pages 131-144. ACM SIGGRAPH, 2000