

벡터합을 이용한 위치 예측 기법

이현석, 양성봉
연세대학교 컴퓨터과학과

Prediction Method Using Weighted Vector Addition

Hyun-Suk Lee, Sung-Bong Yang
Dept. of Computer Science, Yonsei Univ.

요약

본 논문은 Geometry Compression 분야에서 다루지는 압축 기법 중 delta encoding 과정을 보완하여 좀 더 높은 압축률을 얻고자 하는 vertex position prediction 과정에 대한 내용으로 구성되어 있다. 이것은 triangle strip 형태의 입력을 받아서 그 vertex data 중 position 정보들간의 delta encoding 과정을 예측 기법을 이용한 encoding 과정으로 대체하여 Huffman encoding 과정에서의 symbol 개수를 줄여 압축률을 향상시키자는 개념에서 출발한다. triangle strip 생성 기법 중 greedy algorithm을 적용한 후, 기존의 parallelogram 방식과 이 논문에서 새로이 제안하는 방식을 비교하여 보다 나은 압축 방식을 제시하는 것이 이 논문의 목적이다. 이 논문에서 제시하는 방식을 실험한 결과, 기존의 예측 기법에 비해 2.4% 정도의 향상을 보여주고 있다.

1. 서론

최근 3차원 컴퓨터 그래픽 분야는 다방면에 걸쳐 응용되고 있으며 눈부신 발전을 보이고 있다. 그러나, 이러한 3차원 그래픽 분야에 있어서 여러 문제점들이 있는 것이 또한 현실이다. network상에서의 대용량 graphic data 전송 시에 일어나는 lag 및 graphic pipeline input bandwidth의 bottleneck 현상이 바로 그 문제점이라고 할 수 있는데, 이러한 문제점들을 해결하기 위해 여러 연구가 진행 중이고, 대표적인 것이 바로 이 논문에서 다루게 될 geometry compression 이다.

geometry compression은 자바 3D에서 표준 압축 format으로 사용되고 있는 Deering의 GTM(Generalized Triangle Mesh) 개념[1][6], 적절한 triangle strip 형성에 대한 정책을 제시하고 있는 Generalized Triangle Strip[2]에 관한 Evans의 논문에서부터, network 환경에 LOD 개념을 도입한 Hoppe의 PM(Progressive Mesh)[3], 그리고 최근의 Edgebreaker 방식[5]에 이르기까지 활발한 연구가 진행되고 있다.

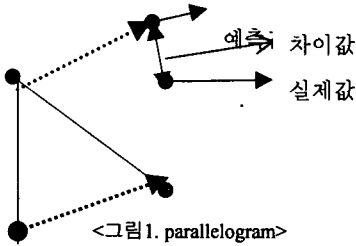
본 논문에서는 geometry compression 과정의 후반부에 해당하는 위치 예측 기법에 대해 다루게 된다. triangle strip 형태로 만들어진 데이터들 이 논문에서 제안하는 새로운 예측 기법을 이용하여, 기존의 parallelogram[4]의 단점을 보완하고자 한다. 본 논문은 4개의 장으로 구성된다. 2장에서는 parallelogram에 대해 간략히 설명하고, 3장에서는 새로운 방식을 제안하고, 그에 따른 실험 결과를 보인다. 마지막으로 4장에서 결론을 내린다.

2. 관련 연구

최근 geometry compression에 있어서의 encoding/decoding 방식은 대부분 Deering이 제안한 생각을 기본 골격으로 삼고 있다. 그 내용을 간단히 요약하자면, normal, color, position 등에 대한 양자화 작업을 거친 후, 압축된 triangle data들을 집어넣은 후, vertex간의 연결정보를 분석한다. 분석을 토대로 vertex traversal 순서와 mesh buffer 참조를 결정한다. 다음, 인접한 vertex 간의 위치, normal vector, color 값의 차를 구하여 delta encoding을 수

행한 후, 빈도수를 기반으로 Huffman tag를 구성한다. 마지막으로, binary output stream을 tag와 data를 이용해 구성한다. decompression은 위의 과정의 역이다.[1][6]

이에 반해, Touma와 Gotsman은 triangle strip 입력을 받아서 vertex position 정보들간의 delta encoding 과정을 parallelogram 예측 기법을 이용한 encoding 과정으로 바꾸어 Huffman table에서의 symbol 개수를 줄이고자 하는 생각을 제안하고 있다.[4] 그림 1에 묘사된 parallelogram 기법은 strip진행에 있어서 직전에 형성된 triangle을 구성하는 3개의 vertex를 이용하여 같은 chain에 있는 점을 base로 잡아 나머지 두 개의 vertex가 형성하는 선분의 기울기와 길이를 이용하여 다음 vertex의 위치를 미리 예측해주고, 실제 값과의 차이를 Huffman table로 넘겨주는 방식을 취하고 있다.



3. 알고리즘 및 실험 결과

2장에서 설명된 예측 기법인 parallelogram은 3D data 예측 기법 중 가장 일반적인 방법으로 알려져 있다. 이 방식이 널리 쓰이는 이유를 분석해 보면, 첫째로, 구현 자체가 무척 쉽다는 점을 들 수 있을 것이다. 두 번째 장점은, 3D data의 일반적인 성질을 간파하여 우수한 예측률을 보인다는 점을 들 수 있겠다.

결과적으로 이 방식은 단순한 알고리즘에도 불구하고, 적은 연산으로 큰 효율을 자랑하고 있다. 그렇지만, 이러한 parallelogram 방식에도 몇 가지 문제점이 있다고 볼 수 있다.

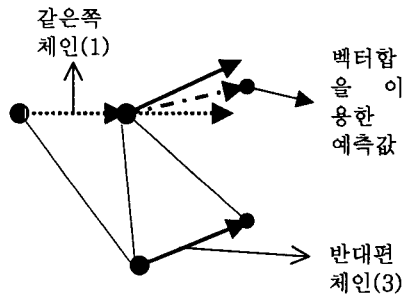
첫째는 모든 예측이 바로 직전의 triangle의 형태에 의해서 결정되므로, 규칙성을 가지고 증가한다거나, 감소하는 triangle에 대해 적절히 예측할 수 없다는 약점이 있다. 둘째는 평면적 상황 위주의 예측으로 입체적으로 급격하게 변하는 상황에 대해서도 적절히 대응하기 어렵다는 측면이 있다. 물론, 이러한 점은 parallelogram이 제안된 논문에서 각 삼각형 간의 crease angle을 고려하여 극복한

다는 말이 추상적으로 제시되어 있긴 하다. 그렇지만, 이러한 crease angle을 계산하여 다음 예측에서 rotation해주는 방식은 구체적으로 제시되어 있지도 않을 뿐더러, parallelogram의 장점인 구현이 쉽다는 점과 적은 연산을 요한다는 점을 잃게 되므로, 실질적으로 이 방법을 잘 사용하지 않는 것으로 알려져 있다. 이러한 측면들을 고려했을 때, 최근 data들의 성향을 적절히 적용할 수 있는 알고리즘, 그리고 입체적인 측면을 고려할 수 있는 예측 알고리즘이 위의 parallelogram의 예측률을 향상시킬 수 있을 것으로 보인다.

따라서, 우리는 여러 실험 및 이에 따른 결과를 통해 parallelogram의 단점을 극복할 수 있는 방법을 찾고자 연구하였다. 입체적인 변화량을 crease angle, 즉 삼각형 간의 변화된 각도만큼을 고려해주는 방식과 근접한 효과를 낼 수 있으면서도 단순한 방법, 그리고, 직전의 반대편 chain에서의 delta값만을 고려하지 않고, 보다 최근 변화량을 고려할 수 있는 방법으로 우리는 최종적으로 다음과 같은 방식을 실험해 보았다.

같은 쪽 chain의 값들도 어느 정도 일정한 규칙으로 다음 vertex값에 영향을 줄 수 있다는 것을 실험을 통해 알게 되었기 때문에, 우리는 같은 쪽 chain의 직전 delta 값도 인자로 집어 넣기로 했다. 즉, parallelogram에서 사용되었던 반대편 chain의 직전 delta 값에다가 같은 쪽 chain의 직전 delta 값을 더한 후, 이에 대한 평균으로 다음 vertex를 예측하는 방식을 취했다.

실험 결과는 parallelogram 보다 우수한 예측률을 보여주었으며, 거기에 우리는 각 chain의 delta값에 가중치를 주어서 test 해보았다. 실험적으로 반대편 chain과 같은 쪽 chain에 3:1로 가중치를 주었을 때, 가장 좋은 결과가 나왔다. 그림 2는 이 방식을 묘사하고 있다. 실험 결과는 표 1에 기술되어 있으며, 실험 data는 hoppe의 실험 data를 이용하였다.



<그림 2. 벡터합을 이용한 예측 기법>

Data	Data Size (Triangle Strip Input)	Parallel -ogram (byte)	가중치(1:3)를 적용한 벡터합 예측기법 (byte)
cactus	5771	4861	4738
club71	9459	7856	7671
engine	15164	12767	12543
fanisk2	5863	5003	4923
hypersheet	17296	14349	14057
mannequin	24560	20511	20057
monkey2	12104	10470	10131
monkey3	9567	8064	7982
nascar	8143	6780	6625
teapot	22088	18504	17750

<표 1. 예측 기법 비교>

표 1에 제시되어 있는 data를 토대로 본 논문에서 제안하는 방식은, parallelogram에 비해 평균적으로 2.4%정도의 향상을 보여주고 있다. 위에서 언급한 초기 vertex 3개의 delta encoding 은 parallelogram과 같은 조건을 주었고, 초기 4번째 vertex는 parallelogram을 사용하였다.

4. 결론

이번 실험을 통해, 일반적인 strip형성과정에서 본 논문에서 제시한 방식이 parallelogram보다 더 우수할 수 있다는 것을 확인할 수 있었고, 각각의 data별로 미세한 압축률의 차이를 볼 수 있었는데, 예측 기법에 의한 압축은 strip형성 방법과 data자체의 특성에 영향을 받는다는 사실을 알 수 있었다.

그리고, progressive mesh에서의 edge collapse가 적용된 data에 있어서, 더 나은 performance를 보여주는 성향이 드러났는데, 본 논문에서 제안한 방식이 그러한 data의 압축에 더 적합하다는 결론을 얻을 수 있었다. 원인으로는, parallelogram이 좀 더 밀도 있는 triangle들의 모임, 즉, 거의 평면에 가까운 변화율을 보이는 dataset들에 강한 면모를 보일 수 밖에 없기 때문인 것으로 보인다. 즉, 여기서 제시된 방식은 입체적인 변화율이 심한 data에서 보다 감점을 보이는 것으로 판명된다.

이러한 결론 하에, 결국 우리는 일반적인 예측 기법에 집착하는 것보다는 각각의 strip 기법에 대해 가장 잘 들어 맞는 예측 기법을 개별적으로 적용할 수 있으리라는 결론을 내리게 되었고, data 자

체의 특성에 따라 예측 기법도 각각 변형되어야 바람직하리라는 예상을 해본다. 따라서, 이러한 생각을 기반으로 하여, 기존의 여러 geometry compression 기법에 대하여 알맞은 예측 기법을 개발 적용한다면, 성능 향상을 꾀할 수 있을 것으로 보이므로, 이에 대한 연구 개발을 계속 진행해 나갈 계획이다.

References

- [1] M. Deering, "Geometry Compression", In *SIGGRAPH '95 Proc.* Pp. 13-20, Aug. 1995.
- [2] F. Evans, S. Skiena, and A. Varshney, "Optimizing triangle strips for fast rendering", In *IEEE Visualization '96 Proc.*, pp. 319-326, 1996.
- [3] H. Hoppe, "Progressive meshes," In *SIGGRAPH '96 Proc.*, pp. 99-108, Aug. 1996.
- [4] C. Touma and C. Gotsman, "Triangle mesh compression", In *Proc. of Graphics Interface '98*, pp. 26-34, 1998.
- [5] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No.1 pp.47-61, January-March, 1999.
- [6] Henry Sowizral, Kevin Rushforth, Michael Deering, "The Java 3D API Specification" pp. 365-398, 1998