

점 위치 추적 기반 기하데이터 압축 알고리즘

박 수진 한 준희 이 혜영
홍익대학교 컴퓨터공학과
{parks, hanj, leeh}@cs.hongik.ac.kr

Vertex Chasing for Geometry Compression of Triangle Meshes

Sujin Park Joonhee Han Haeyoung Lee
Department of Computer Science and Engineering, Hongik University

요 약

본 연구는 3차원 삼각 메쉬의 기하 데이터 압축을 위해 지역 좌표계를 이용한 알고리즘을 소개한다. 기존의 지역 좌표계를 이용한 기하 데이터 처리 방식은 전역 좌표계를 이용한 방식에 비해 높은 압축률을 보여준다. 하지만, 각 메쉬에 적합한 지역 좌표계 3개의 축에 대한 양자화 수(quantization number)를 결정하는 어려움 때문에 실용성 측면에서 문제가 있다. 본 연구에서 새로 제안하는 지역 좌표계를 이용한 방법은 지역 좌표계 안에서 점의 위치를 기술할 때에 오직 분할 깊이(subdivision level)만을 필요로 한다. 분할 깊이가 깊을수록 점의 위치를 정확하게 기술할 수 있다. 본 연구는 보다 정밀한 점 위치의 기술을 위해 16(2x2x4)개의 subcell을 갖는 균일하지 않은 분할을 가지고 실험을 하였다. 본 연구의 새로운 압축/압축해제기는 기존 기하 데이터 압축과 비슷한 효율성을 유지 하면서 실용성 면에서 중요한 압축률과 왜곡간의 제어를 직관적이고 단순하게 할 수 있도록 한다.

1. 서론

센싱과 스캐닝 기술의 놀라운 비약으로 우리는 복잡한 구조의 3차원 모델을 손쉽게 모델링 할 수 있게 되었다. 이러한 복잡한 3차원 모델의 방대한 데이터 크기는 3차원 메쉬 압축이라는 분야에 활기를 불어 넣었고, 결과적으로 단순 압축(Single-rate) [1-6]과 점진적 압축(Progressive Transmission) [7-12] 등 많은 기법들이 소개되었다. 지금까지는 압축 알고리즘의 주요 척도로 주로 압축률이 사용되었으나, 초고속망이 빠른 속도로 확산되고 있는 현실에서 압축률 보다는 빠른 처리 및 사용의 용이성, 그리고 사용 중 에러가 적은 신뢰성 있는 알고리즘이 선호될 것으로 예측된다.

3차원 메쉬의 압축은 일반적으로 연결 데이터 압축과 기하 데이터 압축으로 나뉘게 된다. 연결 데이터 압축 분야는 이미 연구가 성숙된 상태이고, 압축률의 90%를

기하 데이터가 결정하기 때문에 본 연구에서는 기하 데이터 압축의 새로운 접근 방식을 제안한다.

Angle-Analyzer[6]에서 선보인 지역 좌표계를 이용한 기하 데이터 압축 기술은 기존의 Touma-Gotsman [1]의 전역 좌표계를 이용한 압축 기술에 비해 월등한 압축률을 보여준다. 하지만 지역 좌표계를 실제로 구현하기엔 일정한 규칙이 없는 좌표축 양자화(quantization)의 문제가 걸림돌이 된다. 그래서 우리는 지역 좌표계를 이용하여 압축률도 유지하면서 단순성과 신뢰성, 그리고 효율성을 겸비한 기하 데이터 압축을 제안하고자 한다. 본 연구의 점 추적 기법(Vertex Chasing)은 다른 연결 데이터 압축 기술과도 같이 사용될 수 있지만, 현재 최고의 압축률을 보여준 Angle-Analyzer의 연결 데이터 압축과 연동하여 설명하고자 한다.

1.1. 용어 정의

(다음 용어 정의에 대해 그림 1 참조)

게이트 : 방향이 있는 두 점 사이의 선분

앞면 : 현재 게이트에서 인코딩 할 점을 포함하고 있는 아직 처리되지 않은 면

뒷면 : 인코딩 과정에서 처리된 면

앞면의 점 : 앞면에 위치한 인코딩 될 점

게이트 리스트 : 인코딩/디코딩 과정에서 생성된 게이트들을 순서대로 넣은 리스트

게이트 리스트 스택 : 인코딩/디코딩 과정에서 생성되는 새로운 게이트 리스트를 저장할 공간

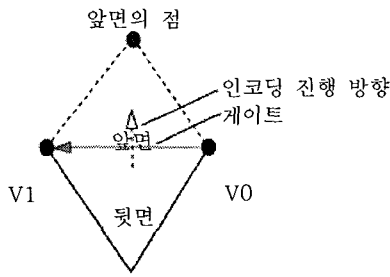


그림 1. 용어 설명 (삼각 메쉬)

1.2. 관련 연구

지금까지 연구된 바에 의하면 연결성 압축 분야는 이론적으로 최적의 압축률이 증명되었을 뿐 아니라, 실제로 좋은 압축 알고리즘이 많이 소개되었다. 반면, 기하 데이터의 압축에 관한 연구는 상대적으로 저조한 편이었다.

기하 데이터 압축에서는 기하예측(geometry prediction)을 이용한 전역 좌표계(global coordinates)[1]가 가장 널리 사용되었다. 이 방법은 3차원 메쉬를 에워싸는 정육면체를 만들어 메쉬의 점을 양자화(quantization)하여 인코딩한다. 사실 이 방식이 널리 사용된 것은 압축 효율이 높을 뿐만 아니라, 알고리즘의 구현이 단순하기 때문이었다.

한편, 지역 좌표계(local coordinates)를 이용한 압축 기법[6,12,10,13]의 등장으로 전역 좌표계에 비해 높은 압축률을 보여주었지만, 각 지역 좌표계 3개의 축에 대한 양자화 수(quantization number)를 결정하는데 어려움이 있었다.

1.3. 점 위치 추적 기반 기하데이터 압축 개요

본 연구의 새로운 기하데이터 압축 알고리즘의 핵심은 지역 좌표계의 조밀도(quantization number)를 점의 위치에 따라 다르게 하는 것이다. 전체적으로 볼 때 지역 좌표계 상에 나타나는 점들의 분포는 특정 층에 집중되어 있기 때문에 이 부분의 조밀도를 높게 하고, 나머지 부분을 낮게 하여 압축하게 된다. 사용자는 점의 위치를 추적할 때에 분할 깊이(subdivision level)만 결정하면 되기 때문에 기존 방법의 각 지역 좌표계 세 개의 축에 대한 양자화 수(quantization number)를 알아내야 하는 어려움이 사라지게 된다. 단순히 분할 깊이를 변경하여 보면 되기 때문에 압축률과 오차 사이의 제어가 매우 직관적이고 신뢰성 있다.

그러면 먼저 Angle-Analyzer[6] 알고리즘을 2절에서 소개하고 이를 개선한 새로운 알고리즘인 점 위치 추적 기반 압축 기법을 3절에서 자세히 소개한다.

2. 각도-분석 기반 압축 알고리즘 (Angle-Analyzer, AA)

일반적으로 메쉬 데이터의 압축은 메쉬간의 연결성을 압축하는 연결 데이터 압축과 실제 메쉬를 구성하는 점들의 위치를 압축하는 기하 데이터 압축으로 나뉘어서 생각한다.

본 연구에서 소개하는 각도-분석 기법[6]에서는 게이트와 앞면의 점 사이의 관계 코드들, 그리고 엔트로피를 최대한으로 줄일 수 있는 정복순서를 이용하여 연결 데이터 압축률을 높이는 방법과, 기하 데이터 압축을 위해 점의 위치를 지역 좌표계를 이용하여 표현할 수 있는 방법을 소개한다.

기본적으로 각도-분석 기법은 여러 개의 컴포넌트들을 압축할 수 있을 뿐 아니라, 삼각, 사각 메쉬로 이루어진 형태의 컴포넌트 처리가 가능하다. 본 연구에서는 같은 알고리즘으로 진행되는 삼각, 사각의 메쉬 압축중 삼각메쉬의 경우만 살펴보기로 한다.

아래의 수도코드는 컴포넌트를 인코딩하는 순서를 보여주고 있다. 그 중에서 최적의 게이트를 선택하는 부분이 각도-분석 기법의 핵심이다. 본 연구가 제안하는 적응형 탐색(adaptive traversal)은 현재 게이트 리스트에서 연속하는 두 게이트가 이루는 각도가 최소인 게이트

를 찾아냄으로써, 압축 효율을 저하시키고 처리시간이 많이 소요되는 코드의 발생을 줄이도록 한다. 각도를 고려하지 않고 탐색하는 고정형 탐색(deterministic traversal)에서는 게이트 리스트의 분할(split) 및 병합(merge)을 많이 발생시키기 때문에 압축률이 떨어지게 된다.

본 연구의 알고리즘은 연결 데이터 압축과 기하 데이터 압축이 서로 연동하여 더 이상 처리할 게이트가 없을 때까지 진행된다. 전체적인 압축 알고리즘의 적용 단계는 다음과 같다.

의사 코드 (Pseudo Code)

반복

초기화

압축되지 않은 다음 컴포넌트를 선택한다.

1. 시작 면을 정한다.
2. 시작 면의 게이트들을 게이트 리스트에 추가한다.
3. 생성된 게이트 리스트를 스택에 저장한다.

메쉬 정복 (Mesh Traversal)

반복

스택에서 한 개의 게이트 리스트를 꺼낸다.

반복

1. 게이트 리스트에서 다음 게이트와의 연결 각도 값이 최소인 게이트를 선택한다.
2. 만약 앞면이 처리되지 않았으면,
 - 앞면을 처리한다.
 - 처리 후 발생한 게이트들을 게이트 리스트에 추가한다.
 - 만약 새로운 점을 포함한 앞면이면 점의 위치를 인코딩 하는 기하 데이터 압축을 수행한다. (*)
3. 처리된 현재 게이트를 게이트 리스트에서 제거한다.

리스트가 비어 있을 때까지

스택이 비어 있을 때까지

더 이상 연결된 컴포넌트가 없을 때까지

위의 의사 코드는 본 연구의 점 위치 추적(Vertex Chasing) 기반의 기하 데이터 압축 알고리즘에서도 (*) 부분을 제외한 행들이 동일하게 적용된다.

2.1. AA의 연결 데이터 압축 알고리즘

연결 데이터 압축은 게이트 기반의 적응형 접근방법(adaptive traversal)을 이용한다. 이는 기존의 방식들이 정해진 순서로 메쉬의 면들을 정복하는 것과 달리, 메쉬의 내재적인 특성을 이용하여 정복 순서를 달리 하는 접근 방법이다.

연결 데이터 압축을 위해 현재 선택된 게이트와 앞면의 점과의 관계를 표현하기 위해 삼각 메쉬의 경우 5개의 관계 코드를 사용하고, 사각 메쉬의 경우 8개의 관계 코드를 사용한다. 이 코드들은 압축/압축해제 된 메쉬의 현재 게이트 상의 두 점과 앞 점을 연결하여 앞면을 복원하는데 이용된다. 게이트간의 각도 값 계산을 위해 게이트들은 순서대로 게이트 리스트에 추가되어야 한다. 또한 생성된 여러 개의 게이트 리스트들을 저장하기 위해 스택이 필요하다.

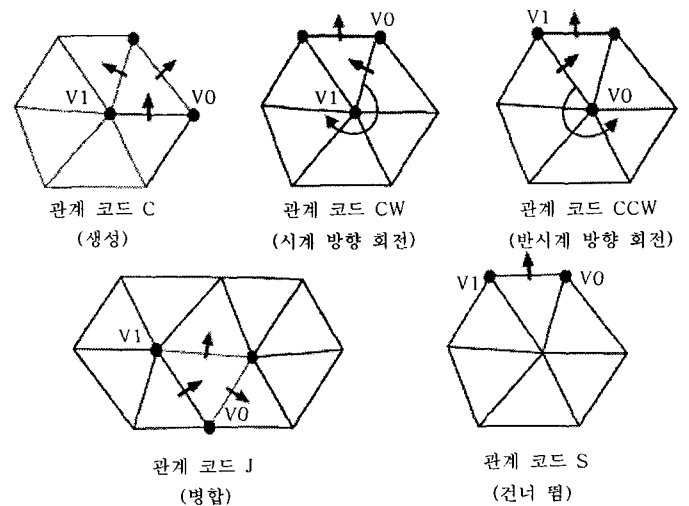


그림 2. 삼각 메쉬에서의 5가지 관계 코드. 검은 화살표는 현재 게이트의 위치와 정복 방향을 나타내며, 빨간 화살표는 새로 생성되는 게이트의 위치와 이 게이트가 정복하려는 면의 방향을 나타낸다. 빨간 선분은 게이트 리스트에 추가될 새로 생성된 게이트이다. 또한, 초록 면은 인코딩/디코딩 된 면을 나타낸다.

연결 데이터의 부호화(Encoding)을 위한 5가지 관계 코드는 다음과 같다. (그림 2 참조) C(생성), CW(메쉬 시계 방향 회전), CCW(메쉬 반시계 방향 회전), S(건너뛴), J(병합). 또한, 그림 2에서의 기호는 다음과 같은 의미를 갖는다.

연결 데이터 압축 (Connectivity Encoding) (그림 2 참조)

만약 앞면의 점이 이전까지 정복되지 않은 새로운 점이라면 C(Create)가 생성된다. 그리고 새로운 두개의 게이트들이 게이트 리스트안의 현재의 게이트를 대체한다. 만약 앞면의 점이 이전에 정복되었었다면, 우리는 V1을 기준으로 시계방향으로 돌거나, V0를 기준으로 반시계방향으로 돌면서 앞 점의 위치를 찾아낼 수 있다. 시계 방향으로 돌아 위치를 찾았다면 CW(Clockwise)코드가 생성되고, 반시계 방향으로 돌아 위치를 찾았다면 CCW(Counter Clockwise)코드가 생성된다. 새로운 게이트는 현재 게이트와 게이트 리스트 안의 현재 게이트와 연속한 다음 게이트를 대체할 것이다. 만약 현재 게이트가 메쉬의 가장자리에 위치하고 있다면 앞면이 없는 상태이기 때문에 S(Skip)가 생성된다. 만약 앞점이 이미 정복된 점이지만 CW나 CCW로 찾아 낼 수 없다면 J(Join)코드가 생성되고 오프셋(offset) 정보가 추가된다. 이 경우, 오프셋은 현재 게이트 리스트안의 모든 점들을 V1으로부터 유클리디안 거리(Euclidean Distance)가 짧은 순으로 정렬한 배열의 인덱스(index) 값이 된다.

J모드에서 만약 앞면의 점이 현재의 게이트 리스트에 속해 있다면, 현재 게이트 리스트는 앞면의 점을 기점으로 2개의 게이트 리스트들로 분할되어야 한다. 만약 앞면의 점이 스택안의 다른 게이트 리스트에 속한다면, 이 두 개의 리스트들은 하나로 병합된다. 이러한 J모드의 처리 과정 때문에 J코드가 발생하면 처리 시간이 길어지고, 오프셋 추가로 압축률이 떨어진다.

연결 데이터 압축 해제 (Connectivity Decoding)

헤더파일에 저장된 3개의 시작점을 연결하여 시작 면으로 사용한다. 3개의 시작 게이트들이 게이트 리스트

에 추가된다. 그리고 메쉬 복원 절차가 인코딩과 항상 같은 규칙을 사용해 일어난다. 인코딩 시와 동일하게 연결된 두 게이트들 간의 최소의 각도를 찾는 규칙을 적용하여 최적의 게이트가 선정되고, 압축 해제된 코드에 따라 앞면이 복원된다. 즉, C 코드의 경우 기하 데이터 압축 해제와 함께 새로운 앞면의 점이 복원된다. 그리고 두 개의 새로운 게이트들이 생성되어 현재 게이트 리스트에 추가된다. CW, CCW 코드들의 경우, 적당한 게이트 점 (V1 또는 V0)을 기준으로 회전하여 앞면의 점을 찾아낸다. 그러면 새로운 앞면이 생성되고, 새로운 게이트가 현재 게이트와 현재 게이트에 연결된 다음 게이트를 대체한다. S 코드의 경우, 현재의 게이트 리스트에서 현재 게이트만 삭제된다. J 코드의 경우, 오프셋을 정렬된 점들의 배열에 대한 인덱스 값으로 사용하여 인코딩 시와 마찬가지로 게이트 리스트의 병합 또는 분할을 수행한 후 앞면의 점을 복원한다.

2.2. AA의 기하 데이터 압축 기법

일반적으로 기하 데이터 압축이 전체 압축 알고리즘의 효율을 결정하는 중요 요소이다. AA에서는 기하데이터 압축에 있어서 점의 위치를 전역 좌표계를 사용하여 나타내는 대신 지역 좌표계로 변환하여 압축한다.

지역 좌표계 사용시 Two-Pass 알고리즘을 사용하는데, 전처리(First-Pass)시에 메쉬를 구성하고 있는 각 점들을 지역 좌표계로 변환한 후, 3개의 축에서 갖는 최소값과 최대값을 알아내어 양자화를 위한 범위를 구한다. 후처리(Second-Pass)시에는 점의 위치를 지역 좌표계로 변환한 후, 전처리 과정에서 결정된 범위안의 값으로 양자화 하여 압축한다.

그림 3에서 보이는 것과 같이 지역 좌표계 상에서 현재 게이트를 지역 x-좌표축으로 설정한다. V0는 원점으로 간주되고, 지역 x-좌표축에서의 단위 벡터(Unit Vector)를 뒷면의 법선 벡터(Normal Vector)를 중심으로 $-\pi/2$ 만큼 회전시켜 지역 y-좌표축으로 설정한다. 지역 z-좌표축은 x와 y의 외적으로 구하거나, 뒷면의 법선 벡터의 반대방향 벡터로 구할 수 있다. 각 지역 좌표계에서 V0와 앞면 점 사이의 거리는 각 면에 따라 달라진다. 모든 메쉬의 점들이 3개의 각 지역 좌표축

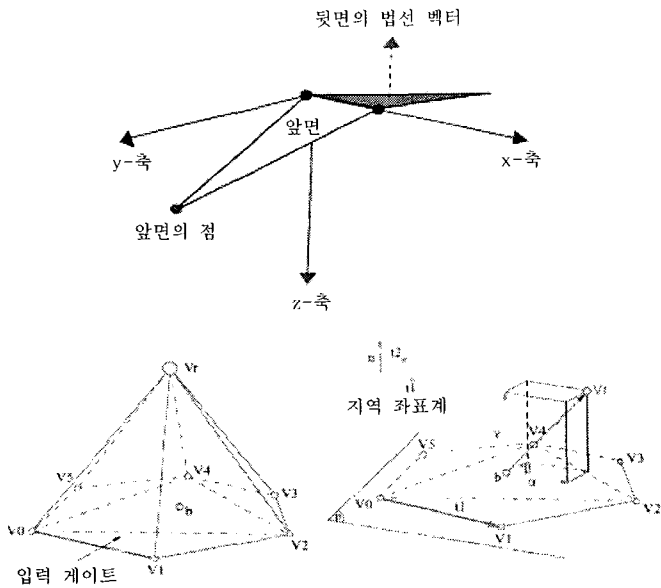


그림 3. 현재의 게이트와 뒷면을 이용한 지역 좌표계 정의 방법 (위), Alliez-Desbrun[5]의 지역 좌표계 (아래)

안에서 표현될 때 필요한 범위를 설정하기 위해 First-Pass 시에 범위의 최소 값과 최대 값을 구한다. Second-Pass 시에는 $(V_0 - \text{앞면의 점})$ 벡터가 각 지역 좌표축의 단위벡터에 투사되고, 각 지역 좌표축은 선형(linear)적으로 양자화 되어 투사된 값에 가장 가까운 점으로 인코딩 된다.

디코딩 과정에서 생기게 되는 오차를 누적시키지 않기 위해 인코딩시에 디코딩 시뮬레이션을 수행한다. 실제 점의 위치를 이용하지 않고, 디코딩 된 앞면의 점의 위치를 다음 인코딩 과정에서 쓰게 됨으로써 누적 오차가 생기지 않게 된다. 결국 시뮬레이션을 이용함으로써 교정벡터가 필요치 않게 되었고, 압축률을 높이는 근간이 되었다.

기하데이터 압축 해제

압축해제 과정은 단순히 압축의 반대 과정이다. 각 좌표축의 단위 벡터를 알아 낸 후에 디코딩 된 좌표는 전역 좌표계상의 좌표로 변환되어 앞면의 점으로 복원된다.

2.3. 기하 압축 기법상의 문제점

지역 좌표계를 이용한 방법은 전역 좌표계에 비해 압축률이 월등히 높다. 하지만, 메쉬의 성격에 따라 지역 좌표계의 각 축을 얼마만큼 양자화 시킬 것인가에 대한 확고한 규칙이 없기 때문에 시행착오가 불가피하다. 즉, 각 지역 좌표축의 양자화 정도를 달리하여 복원된 메쉬와 원형 메쉬간의 왜곡률(Distortion rate)을 비교하여야만 각 메쉬에 적합한 양자화 값(quantization number)을 알아낼 수 있다.

3. 점의 위치 추적 기반 기하데이터 압축 알고리즘

초고속망의 보급이 증가함에 따라, 압축률의 중요성이 상대적으로 낮아지고 있다. 구현하기 어렵고, 사용하기 어려운 알고리즘을 사용하여 데이터 크기를 약간 줄이기 보다는 처리가 빠르고 사용하기 쉬우며 사용 중 에러 발생률이 적은 알고리즘도 application에 따라 필요할 것으로 예측된다. 그래서 우리는 최고의 압축률을 가지고 있지만, 실제로 사용할 때에 여러 가지 어려움이 있는 Angle-Analyzer[6] 방법을 사용성과 신뢰성의 향상에 초점을 맞춰 개선하였다. 압축률과 왜곡 간의 균형을 직관적으로 조절할 수 있게 하고, 알고리즘의 구현 및 사용을 용이하게 하는데 초점을 맞추었다.

Angle-Analyzer[6]는 전처리와 후처리로 나누어진다. 전처리에서는 양자화를 위한 지역 좌표계의 점의 범위를 알아내고, 후처리에서는 실제적인 압축을 한다. 전체적인 처리과정은 기존의 Angle-Analyzer[6]와 동일하므로, 3쪽에 있는 의사코드를 참고하기 바란다.



그림 4. cell을 분할 깊이 4로 세분화한 모습

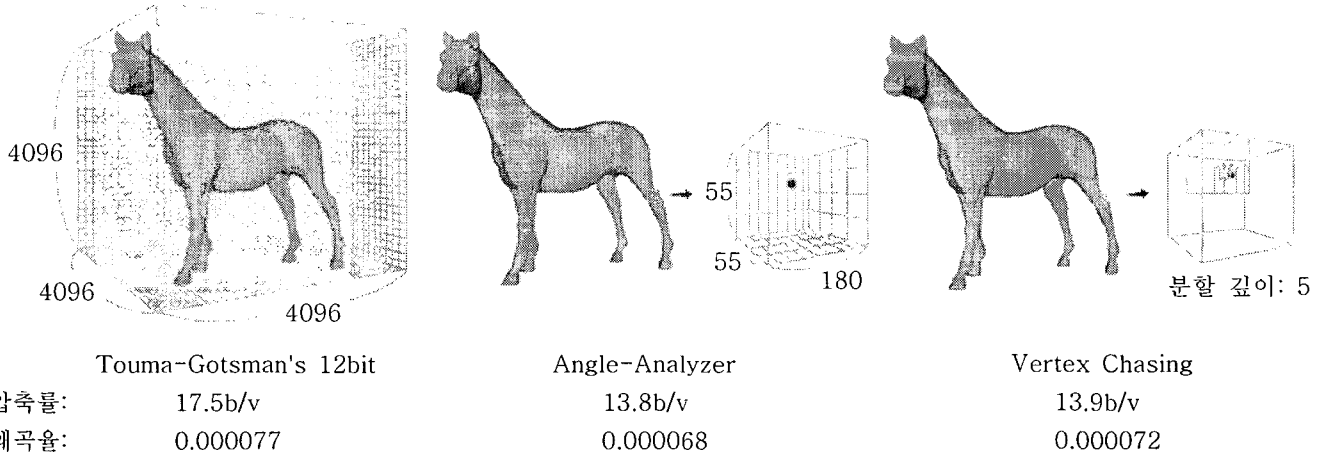


그림 5. horse의 압축방법에 따른 압축률과 왜곡율 (Metro[15]에 의해 측정되었음)

3.1 점 추적 기법

대부분의 지역 기하 데이터 압축 방법은 그림 3에 보이는 것처럼 한 면 위에 지역 좌표계를 설정한다. 그들의 효율은 이미 증명되었고, 특히 [6,10,13]은 가장 좋은 압축률을 보여주었다. 하지만, 그 방법들은 실제로 왜곡이 적으면서 좋은 압축률을 갖게 되는 비균일적 양자화 수를 찾는 데 많은 시간을 소비해야 했다. 위치 추적 기반의 기하 데이터 압축은 비균일적 양자화 대신에 세분화를 한다. 그림 4,6에 보이는 것처럼 cell을 세분화하여 점의 위치를 추적하는 방법으로 압축을 한다.

Angle-Analyzer가 압축 시에 x축, y축, z축 각각에 대해 양자화 수 3개를 요구한 것에 비해 본 기하 데이터 압축은 세분화 정도를 나타내는 분할 깊이 하나만을 입력으로 요구한다. 그림 4,6에 보이는 것처럼, 왜곡을 적게 하려면, 분할 깊이값을 크게 주어야 한다.

subcell을 세분화 할 때는 [1]처럼 균일한 방법을 쓰지 않고, 그림 6에서 보이는 것처럼 비균일한 방법을 사용했다. X,Y평면(tangential plane)보다 평면의 법선 벡터방향인 Z축방향이 양자화에 더 민감하므로, Z축을 X,Y축의 2배로 세분화한 2×2×4의 방법을 사용했다. cell을 반복적으로 세분화하며, 점을 포함하는 subcell의 번호(0~15)를 압축한다. 3개의 이전 레벨(parent-level)의 subcell 번호를 문맥(context)으로 사용하여 압축률을 더 높일 수 있었다.

위치 추적 기반 방법을 적용함으로써, 그림 7에서 보이는 것처럼, 왜곡과 압축률 간의 조정이 더 간단하고 신뢰할 수 있게 되었으며, 표 1에 보이는 것처럼 기존의 압축률과 비슷하게 유지할 수 있었다. 위치 추적 기반의 기하 압축은 Angle-Analyzer외에 valence-based 같은 다른 연결데이터 압축 방법들과도 결합하여 사용할 수 있다.

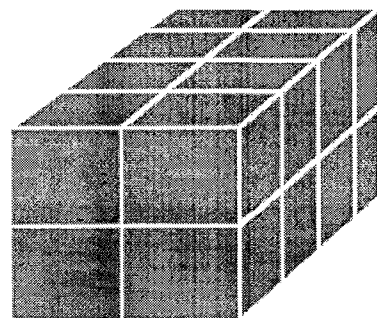
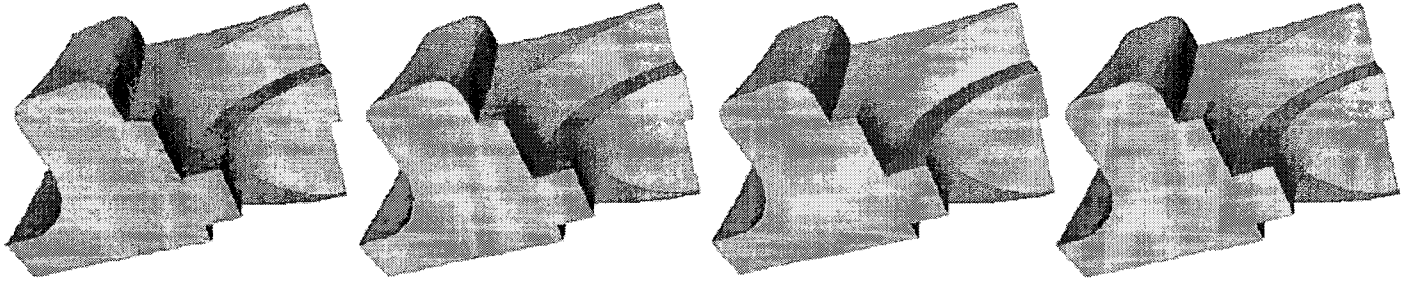


그림 6. cell을 2×2×4로 세분화한 모습



분할 깊이:	4	5	6	7
압축률:	8.46b/v	11.56b/v	14.27b/v	17.8b/v
왜곡율:	0.000079	0.000039	0.000016	0.000009

그림 7. Fandisk를 위치 추적 기반의 기하 압축 방법으로 압축했을 때 분할 깊이에 따라 압축률과 왜곡율이 달라짐을 알 수 있다. 분할 깊이를 크게 하면 원래 모델에 가깝게 복원됨을 알 수 있다. (Metro[15]에 의해 측정되었음)

3.2 구현 및 실험 결과

위치 추적 기반의 기하 압축은 현재 최고의 기하 압축/해제기인 Angle-Analyzer와 비슷한 기하 압축률을 만들어냈다. 비록 압축률은 비슷하지만, 그림 7에서 보이는 것처럼 왜곡과 압축률 사이의 제어를 매우 쉽게 할 수 있게 되었고, 구현하기가 쉽게 되어 알고리즘의 신뢰성이 증가하였다.

4. 결론 및 향후 연구과제

본 논문은 새로운 위치 추적 기반의 기하 데이터 압축방법을 소개하였다. 위치 추적 기반 방법의 연결 데이터 압축은 다양한 특성의 모델을 처리할 수 있는 기존 Angle-Analyzer를 사용했다. 위치 추적 기반 방법은 Angle-Analyzer외에 다른 연결 데이터 압축 방법에도 사용할 수 있다. 우리는 Angle-Analyzer의 비균일한 양자화 수를 찾는 어려움을 쉽고 간단한 분할 깊이를 정해주는 것으로 바꾸었다. 그래서 왜곡과 압축률 사이의 균형을 맞추는 일은 매우 직관적으로 바뀌었으며, 구현의 용이성이 증가하여 신뢰성도 높아졌다. 향후연구는 본 기하 데이터 압축 방법을 점진적(progressivity)으로 압축하는 알고리즘으로의 확장할 생각이다. 이미 위치 추적 기반의 압축 방법은 셀들을 한 깊이씩 세분화하여 원래 점에 가까워지게 되는 점진적인 성질을 가지고 있

다. 우리는 또한 위치 추적 기반의 방법이 사면체(tetrahedral) 메쉬도 다룰 수 있도록 개선할 생각이다.

Model	점의 갯수	AA기하 압축률 (b/v)	AA 왜곡률	VC기하 압축률 (b/v)	VC 왜곡률
fandisk	6475	11.9	0.000041	11.5	0.000039
bunny	35947	12.3	0.000007	12.4	0.000005
horse	19851	12.4	0.000067	12.5	0.000072
maxplanck	25445	13.8	0.000060	15.5	0.000065
tf2	14169	11.5	0.000052	12.4	0.000056
dinosaur	14070	17.0	0.004540	14.9	0.009890
feline	19864	13.8	0.000032	14.8	0.000033
venus	8268	19.6	0.00044	15.2	0.000458

표 1: Angle-Analyzer[6]와 본 논문의 새로운 방법인 VC를 비교한 결과 비슷한 압축률을 유지하는 것을 볼 수 있다. VC는 AA를 개선한 방법으로 압축률은 유지하면서 구현의 용이성과 사용성이 개선된 새로운 기하 압축 방법이다. (Metro[15]에 의해 측정되었음)

감사의 글

본 연구는 과학기술부 과학재단 목적기초연구
(R01-2005-000-10120-0) 지원으로 수행되었습니다.

참고문헌

- [1] Touma C., Gotsman C.: Triangle Mesh Compression. *Graphics Interface 98 Conference Proceedings* (1998), 26-34.
- [2] Rossignac J.: EdgeBreaker : Connectivity Compression for Triangle Meshes. *IEEE Transactions on Visualization and Computer Graphics* 5, 1 (1999), 47-61.
- [3] Gumhold S.: Improved cut-border machine for triangle mesh compression. *Erlangen Workshop'99 on Vision, Modeling and Visualization* (November 19-99).
- [4] Isenburg M.: Triangle Strip Compression. *Proceedings of Graphics Interface 2000* (2000), 197-204.
- [5] Alliez P., Desbrun M.: Valence-driven connectivity encoding of 3d meshes. *Computer Graphics Forum* 17, 3 (2001), 480-489. (Proc. Eurographics'98).
- [6] Lee H., Alliez P., Desbrun M.: Angle-analyzer: A triangle-quad mesh codec. *Computer Graphics Forum* 22, 3 (2002), 383-391. (Proc. Eurographics '98).
- [7] Hoppe H.: Progressive Meshes. *Siggraph 96 Conference Proceedings* (1996), 99-108.
- [8] Taubin G., Gueziec A., Horn W., Lazarus F.: Progressive forest split compression. In *Proc. of SIGGRAPH* (1998), pp. 123-132.
- [9] Bajaj C. L., Pascucci V., Zhuang G.: Progressive compression and transmission of arbitrary triangular meshes. In *Proc. of IEEE Visualization* (1999), pp. 307-316.
- [10] Alliez P., Desbrun M.: Progressive Encoding for Lossless Transmission of 3D Meshes. *ACM Siggraph Conference Proceedings* (2001), 198-205.
- [11] Gandoin P.-M., Devillers O.: Progressive lossless compression of arbitrary simplicial complexes. *ACM Trans. on Graphics* 21, 3 (2002), 372-379.
- [12] Khodakovsky A., Schröder P., Sweldens W.: Progressive Geometry Compression. *ACM Siggraph Conference Proceedings* (2000), 271-278.
- [13] Lee H., Desbrun M., Schröder P.: Progressive encoding of complex isosurfaces. *ACM Transactions on Graphics* 21, 3 (2003), 471-476. Proc. Siggraphics' 03.
- [14] Käerer F., Polthier K., Reitebuch U., Wardetzky M.: Connectivity compression of triangle meshes using geometric information. *ZIB Preprint* (2004).
- [15] Cignoni P., Rocchini C., Scopigno R.: Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum* 17(2) (1998), 167-174.