No-Zippering 기법을 이용한 가상객체 생성

○
김기호*, 권대현*, 유황빈**
*전자통신연구원,**광운대학교 전산과
*대전광역시 유성구 가정동 161번지
khkim@etri.re.kr, dkwon@etri.re.kr,

Virtual Character Generation using No-Zippering Method

○
Ki-Ho Kim*, Dae-Hyun Kwon*, Hwang-Bin Ryou
*Human Computer Interface Dept., ETRI
**Dept. of Computer Science Kwangwoon University
*161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
khkim@etri.re.kr, dkwon@etri.re.kr,

요 약

3차원 스캐너 또는 스테레오 비전 등으로부터 입력된 거리 정보(Range Data)를 실제 응용프로그램에 사용하기 위한 가상객체(Virtual Character)로 만들기 해서는 우선 메쉬(mesh) 형태로 변환하는 작업이 필수적이며, 기법들은 수십년 동안 활발히 연구되어 왔다.

많이 이용되는 방법은 초기 입력된 거리영상들의 각각의 점들이 연결관계 및 구조에 대한 정보를 입력시점에서 알 수 있으므로, 각 면에 대해서는 쉽게 메쉬를 생성하여, 2차원 측정 단면 중심의 부조 형태로 메쉬를 활용하고 있지만. 3차원 가상객체가 요구되는 경우에는 측정단면별로 형성된 결과들을 하나의 메쉬로 생성하기 위해서는 짜깁기(Zippering)와 같은 어려운 작업이 추가로 필요하다.

본 논문에서는 개선된 3차원 행진입방체 방법과 간략화 기법을 적용하여 짜깁기(Zippering)을 사용하지 않고 가상객체를 생성하는 방법을 소개하려 한다.

1. 서 론

인간의 오감(五感)을 컴퓨터에 의해서 재생하여 실세계와 동일한 가상공간(cyber space)을 만들어, 디지털 정보를 활용한 사회 및 경제활동을 하려는 시도가 높아지고 있다.

가상공간(cyber space)에서 활동하는 사이버 캐릭터(cyber character)들은 컴퓨터 처리속도를 여러 가지 제약조건에 의하여 단순한 형태로 모델링하여 사용하고 있었으나, 최근 컴퓨터 하드웨어의 급속한 발전에 힘입어 실물 기반 모델링이라는 기술이 발전하고 있다.

3차원 시각정보를 추출하고, 실물에서 추출된 정보를 근간으로 가상객체(virtual character)를 생성하는 기술은 실물기반 모델링이라고 한다.

이 기술의 첫 단계인 3차원 시각정보 추출단계에서는 측정영역에 해당되는 모든 점을 3차원 좌표(x,y,z)로 추출하게 되며, 이 추출된 좌표를 컴퓨터에서 사용하기 위해서는 다각형 형태인 메쉬로 만들어야 하며, 만들어진 메쉬를 근간으로 3차원 가상객체를 생성한다. 이때 측정 단면별로 메쉬를 만들고, 만들어진 메쉬들은 짜깁기하는 방법[1]을 사용하기도 하고, 측정된 여러 단면을 한번에 메쉬를 만들어서 짜깁기하는 과정을 생략하기 위한 방법들도 제안되고 있다.[2][3]

2. Zippering에 의한 Mesh 생성

여러 장의 거리영상(range data)을 조합하여 하나의 단일 메쉬(Mesh)를 생성하는 방법으로 널리 알려진 방법은 Greg Turk 와 Marc Levoy에 의해서 SIGGRAPH'94에서 발표된 , "Zippered Polygon Meshes from Range Images"으로 다음과 같은 4단계를 거쳐서 실행된다.

① Creating Triangle Meshes from Range Images
② Registration of Range Images
③ Integration : Mesh Zippering
④ Consensus Geometry

2.1 Creating Triangle Meshes from Range Images

각 range data는 2차원 배열로 구성되므로 인접하는 4점으로 구성된 사각형을 구할 수 있으며 이 사각형에서 최단 대각선을 이용하여 2개의 삼각형 Mesh를 구한다.

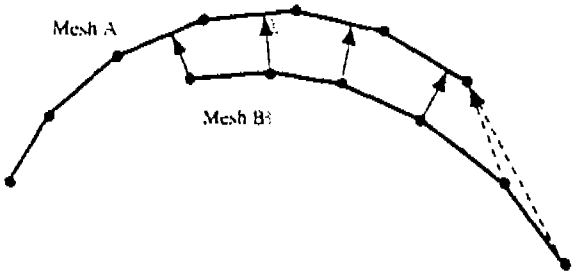
2.2 Registration of Range Images

다음의 단계를 거쳐서 두 중첩된 Mesh (mesh

A, mesh B) 에서 가장 가까운 점들을 찾는다.
- mesh B의 각 점들에 대하여 mesh A상의 가장 가까운 점들을 구한다.
- 거리가 일정 한계를 벗어나는 pair는 버린다.
- 두점이 mesh의 경계선에 있는 pair는 제거한다.
- 각 pair들 사이의 weighted least-squared distance를 최소화하는 rigid transformation을 찾는다. 다음과 같은 error를 최소화 하는 transformation vector T와 R을 찾으면 된다.

$$E=\sum_{i=1}^{n} w_i \|A_i - R(B_i - B_c) - T\|^2$$

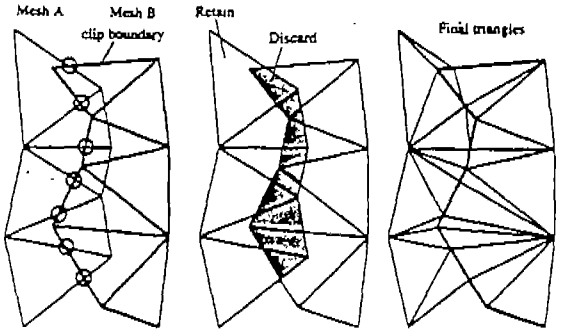
- 수렴할때까지 반복 수행한다.
- 좀더 세밀한 레벨에서 이 알고리즘을 적용한다.



[그림 1] Mesh A와 Mesh B의 연결점 찾기

2.3 Integration : Mesh Zippering

앞에서 계산된 정보를 이용하여 두 개의 Mesh를 붙여서 하나의 Mesh로 만든다. 이 과정을 반복 적용함으로써 입력된 다수의 Mesh로부터 하나의 3차원 모델을 생성할 수 있다. 다음과 같이 세 개의 단계로 구성된다.
- 각 mesh들에서 중첩되는 점들을 제거한다.
- 하나의 mesh에 대해 다른 mesh를 잘라낸다.
- 위의 과정에서 생긴 작은 삼각형들을 제거한다.



[그림 2] Mesh Zippering하는 과정

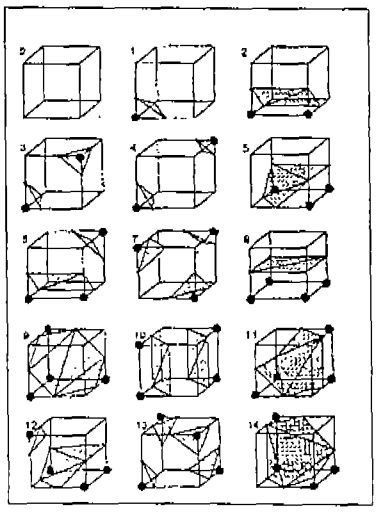
2.4 Consensus Geometry

모든 range data에 대해 위의 방법을 적용하여 구한 mesh는 측정된 원래 물체의 형태를 갖추게 된다. 대부분의 응용을 위해서는 이 정도면 충분하다. 그러나 표면의 세밀한 부분이 중요한 응용의 경우에는 미세한 조정이 필요하다. 이 논문에서는 다음과 같은 과정을 거쳐서 consensus surface를 찾는다.
- 각 surface의 normal에 대한 국부적 근사치를 찾는다.
- normal 방향으로 선을 그어 이 선과 원래의 물체(range data)가 만나는 점을 구한다.
- 만나는 두 점간의 가중 평균을 구한다.

3. No-Zippering에 의한 Mesh 생성

3.1 3차원 Marching Cube

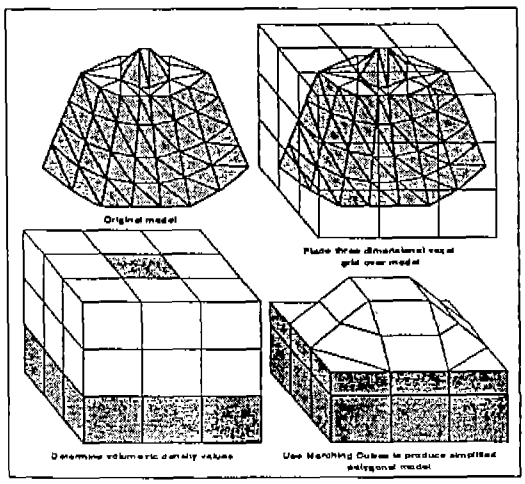
Marching Cube는 divide-and-conquer 방식을 사용하여 여덟 개의 꼭지점으로 구성된 cube를 물체의 표면에 위치하게 하여, 이것을 진행시켜서 polygon을 구하는 방법이다. cube의 configuration에 따라 15가지 조합을 만들어 낼 수가 있다. Marching Cube 방법에서는 3차원 물체가 포함된 공간을 격자 형태로 분할한다. 그렇게되면 공간은 다수의 육면체로 구성되는데 Marching Cube 방법에서는 이들 각각의 육면체들을 대상으로하여 Mesh를 생성한다. 즉 3차원 물체의 표면이 지나가는 cube에서는 cube가 분할되면서 새로운 면이 나타나는데 이 면들이 우리가 생성하고자 하는 Mesh가 되는 것이다. 따라서 Marching Cube를 이용하여 Mesh를 생성한다는 것은 각 cube에서 어떻게 분할이 일어날 것인가를 결정하는 문제로 귀착된다.[4]



[그림 3] 15종류의 3차원 Marching Cube

하나의 cube에는 8개의 꼭지점이 있으므로 각 꼭지점에 대하여 물체의 바깥쪽에 있는지 혹은 안쪽에 있는지를 점사하면 그 cube가 어떠한 형태로 분할될 지를 결정할 수 있다. 각 꼭지점이 가질 수 있는 상태가 안쪽 또는 바깥쪽의 두가지이고 꼭지점의 수가 8개이므로 가능한 상태의 수는 28 = 256가지가 된다. 그런데 cube의 형태는 대칭적이므로 이러한 대칭성을 고려하면 그림11에서와 같이 14가지의 상태만이 존재한다. 따라서 각 꼭지점이 안쪽에 있는지 바깥쪽에 있는지만 알아낼 수 있으면 table lookup에 의해서 쉽게 Mesh를 생성할 수 있다. cube를 분할할 때 모서리의 절단 위치는 각 꼭지점과 물체 표면과의 거리를 계산하여 결정하여야 하지만 일반적으로 cube의 크기가 매우 작으므로 모서리의 중간점을 분할점으로 하더라도 오차가 크지 않다.

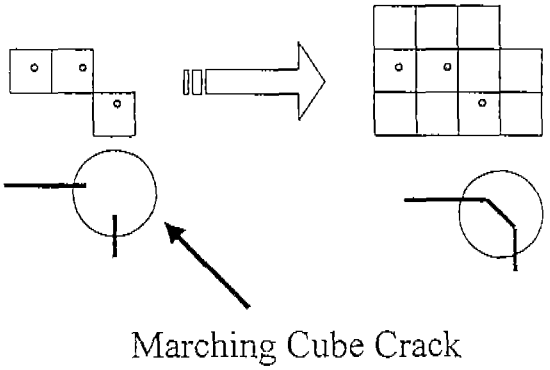
연결되지 않은 거리 정보들에 Marching Cube 기법을 적용하기 위해 각 거리 정보들에 대해 부호화된 거리 (signed distance)를 측정하였다. 이를 위해 각 점들에 대해 인접한 점들과의 위치 관계를 보아 접평면(tangent plane)을 찾아낸다. 이때 접평면의 가능한 방향은 서로 반대방향인 두가지가 가능하므로 이 중의 한가지로 선택하기 위하여 최적화 과정을 거쳐서 물체를 이루는 모든 점들에서 접평면의 방향이 일관되는 방향을 선택한다. 그리고 나서는 각 cube의 꼭지점들에 대해 가장 가까운 접평면을 찾고 그 꼭지점에서 접평면으로의 벡터와 접평면의 normal 벡터와의 내적으로써 부호화된 거리를 측정한다. 이 측정값의 부호에 따라 꼭지점의 상태가 결정되는 것이다.



[그림 4] Marching Cube에 의한 메쉬생성단계

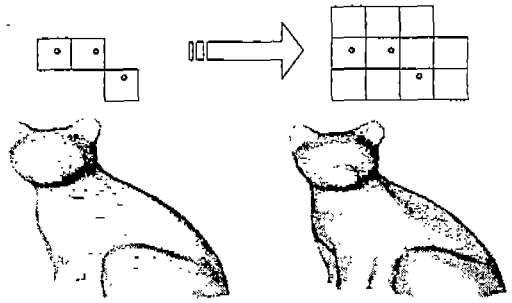
3.2 개선된 Marching Cube

3차원 Marching Cube는 그림5의 좌측 그림과 같이 cube가 서로 맞물려서 연결되는 경우에만 선으로 연결하고, 대각선으로 걸쳐서 연결되는 경우에는 다각형을 만들지 않기 때문에 선이 끊어지는 현상(Marching cube crack)이 생긴다. 이를 개선하기 위해서는 그림 5의 우측그림과 같이 끊어진 선을 연결하여야 한다.

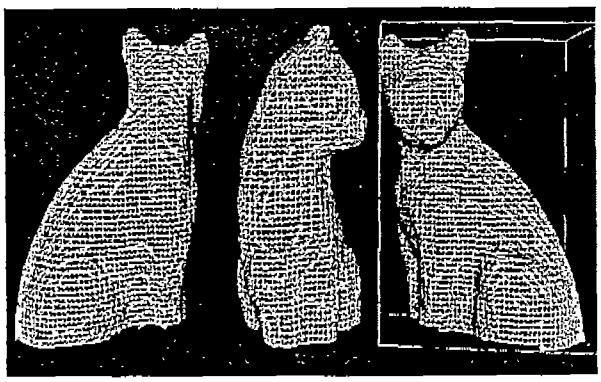


[그림 5] 개선된 Marching Cube 방법

그림6은 개선된 3차원 Marching Cube 방법에 의해서 메쉬를 생성한 결과로써 좌측 그림에 나타나는 구멍(crack)을 메울 수 있는 효과가 있다.



[그림 6] Marching Cube Crack 제거



[그림 7] 개선된 Marching Cube에 의한 결과

3.3 개선된 Decimation 알고리즘

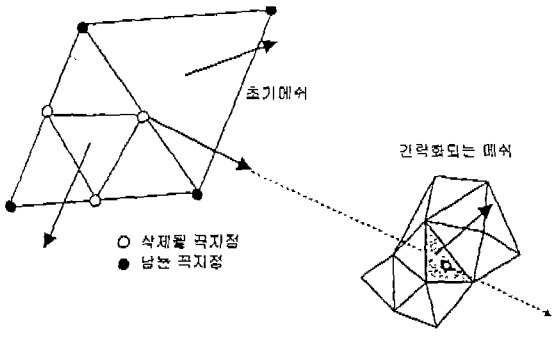
3차원 시각정보 추출시스템으로부터 초기에 입력된 거리영상 자료는 16Mega 바이트의 크기를 갖는다. 이 거리영상자료는 개선된 marching cube 방법에 의하여 초기 메쉬를 만들면 marching cube 크기에 따라 차이가 나지만 5~10%정도로 축소되어, 1 Mega 바이트 정도의 크기를 갖는다.

그러나, 이 초기메쉬를 그대로 사용하기에도 자료의 크기가 방대하기 때문에, 간략화(decimation) 알고리즘에 의해서 자료를 줄여야 한다.[5]

여러 종류의 간략화 알고리즘이 개발되어 활용되고 있으나, Marching Cube에 의해서 생성된 모델에 적당한 간략화 방법은 Schroeder가 SIGGRAPH'92에서 발표한 "Decimation of Triangle Meshes" 방법으로 현재 많이 활용되고 있다.[6]

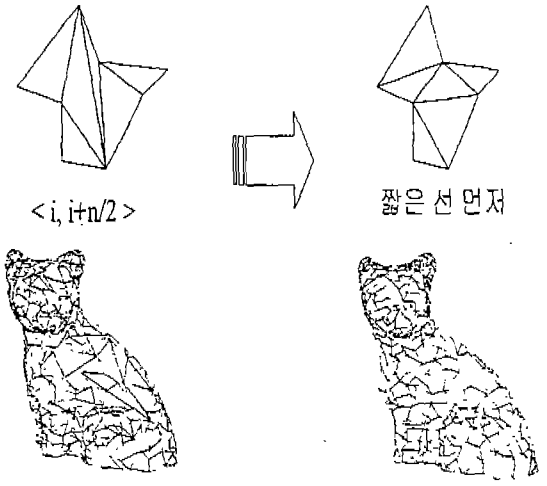
이 알고리즘은 사용자가 허용 가능한 거리 오차(error distance)를 지정해 주면, 각 꼭지점들마다 국부적 topology에 대한 거리를 측정하여 지정한 거리 오차 이내이면 해당하는 꼭지점을 지우게 된다. 이 방법도 원형의 꼭지점은 유지된다.

그런데 이 알고리즘에서 거리 오차를 계산할 때 바로 전 단계의 결과와 비교하게 되는데 이렇게 하면 오차가 누적이 될 수 있으므로 전 단계의 결과가 아닌 원형과 비교함으로써 거리 오차를 계산하는 방법이 요구됨에 따라 초기메쉬를 계속 참조함으로 원형과의 비교 되도록 개선하였다.



[그림 8] 초기메쉬 연속참조 Decimation 기법

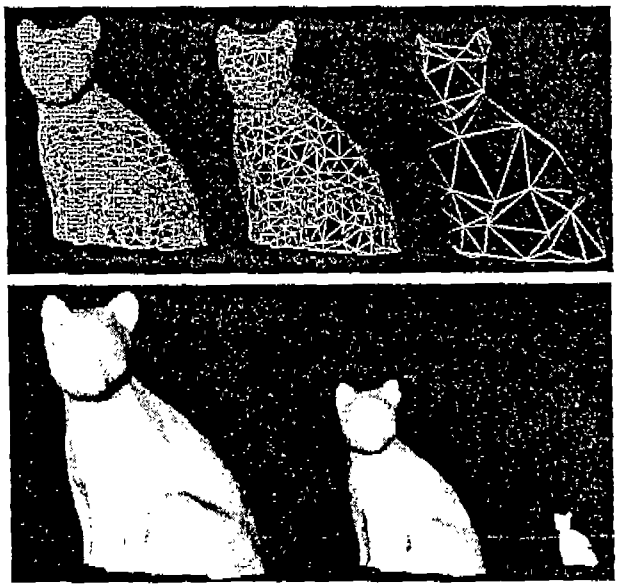
간략화 작업을 위해서는 제거될 꼭지점 선정하고 난 후에 새로운 삼각형으로 분할시키게 되는데, 이 단계에서는 처리 시간을 다소 더 요구되더라도 재귀적으로 주위에 있는 모든 꼭지점을 비교하여, 그림9과 같이 종횡비를 비교하여 연결될 꼭지점을 선정한다.



[그림 9] Recursive Looping Splitting 알고리즘

또한, 사용자들은 현재의 자료로부터 줄이고 싶은 압축률을 미리 정한 후 간략화 작업을 수행한다. 이와같은 작업을 Schroeder 알고리즘에 의해서는 처리하기 위해서는 여러번 수행(multiple pass)을 거쳐서, 거리 오차를 차츰차츰 줄여야 하는 사용상의 불편함이 있다.

이를 해결하기 위해서는 거리오차 크기로 정렬한 후 사용자가 정한 개수만큼 순서대로 제거하는 single pass 방법으로 사용자는 압축 정도를 초기에 결정해 주면 1회에 작업이 완료되도록 개선하였다.



[그림 10] 개선된 간략화 방법에 의한 결과

4. 결론

짜깁기(Zippering)작업을 통하지 않고 만들어진 3차원 객체 생성은 속도면에서 많은 향상을 보이고 있으며, 사용자 입장에서 개선된 방법들은 시스템 개발하고자 하는 사람들에게 많은 도움을 줄 것이다.

그러나, 다각형 메쉬로 생성된 3차원 가상객체를 움직이도록 하는 애니메이션 및 표정 변화들을 위한 작업에서 많은 수의 꼭지점을 움직여야하는 어려움을 내제하고 있다.

따라서, 꼭지점으로 이루워진 다각형 메쉬는 자유곡선으로 이루워진 면 메쉬로 변형되어 처리되는 기술의 연구가 요구된다

또한, 이 기술은 계속 발전되어 감성공학 측면에서 요구되는 디지털 인간 생성에 도움이 될 것을 기대하고 있다.

Acknowledgements

본 논문은 정보통신부 정보통신연구개발사업의 하나인 "3차원 시각정보 자동추출 및 실감표현 기술개발"사업의 연구개발임을 밝힌다.

참고문헌

[1] Greg Turk and Mark Levoy, Zippered Polygon Meshes from Range Images, Computer Graphics (SIGGRAPH 94 Proceedings), 1994.

[2] KIM Kiho, CHUNG Yunkoo, RYOU Hwangbin, 3D Level-of-Detail Object Constructed Based Decimation, KIPS(Korea Information Processing Society) 1998 Proceedings, April 1998.

[3] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle, Surface Reconstruction from Unorganized Points, Computer Graphics (SIGGRAPH 92 Proceedings), 26(2):71-78, July 1992.

[4] William E. Lorensen and Harvey E. Cline, Marching Cubess: A High Resolution 3D Surface Construction Algorithm, Computer Graphics (SIGGRAPH 87 Proceedings), 21(3):163-169, July 1987.

[5] Carl Erikson, Polygonal Simplification: An Overview, Technical Report TR96-016, Dept. of Computer Science, UNC, 1996.

[6] William J. Schroeder, Jonathan A. Zarge and William E. Lorensen, Decimation of Triangle Meshes, Computer Graphics (SIGGRAPH 92 Proceedings), 26(2):65-70, July 1992.