

정점의 법선벡터를 이용한 기하이미지의 최적화

박종래, 양성봉
연세대학교 컴퓨터 산업 공학과

jlpark@cs.yonsei.ac.kr
yang@cs.yonsei.ac.kr

Geometry Image Optimization using a Normal Vector

Jong-Lae Park, Sung-Bong Yang
Dept of Computer Science, Yonsei University

요 약

일반적으로 메쉬(mesh)는 비정규 연결 형태(irregular connectivity)로 되어 있다. 리메싱(remeshing)은 비정규 연결 형태의 메쉬를 정규 연결 형태(regular connectivity)로 바꾸어 주는 작업이다. 메쉬의 기하 정보가 2D 그리드에 저장되어 있는 기하이미지(geometry Images)는 비정규 연결 형태의 메쉬를 완전정규 형태(completely regular connectivity)로 리메싱하는 데 사용된다. 원본 메쉬를 기하 이미지로 생성하는 방법은 변형되는 크기를 최소화 하는 스트레치 매트릭(stretch metric)을 기반으로 이루어 졌다. 이 방법은 리메싱된 메쉬의 언더샘플링(undersampling)을 줄여 주게 된다. 하지만 리메싱 과정에서 생기는 오버샘플링(oversampling)은 줄여 주지 못한다. 본 논문에서는 정점(vertex)의 법선 벡터(normal vector)를 이용하여 기하이미지의 오버샘플링을 줄이는 방법을 제시한다.

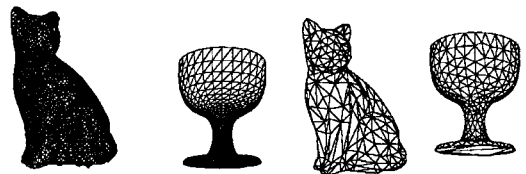
1. 서론

기하이미지(geometry images) [1][그림1,우측]는 비정규 연결 형태(irregular connectivity)의 메쉬(mesh)를 완전 정규 연결 형태(completely regular connectivity)로 리메싱(remeshing)하기 위해서 사용된다. 원본 메쉬[그림1,좌측]의 정점을 2D 그리드에 매핑을 시킨 다음[그림1,중앙] 생성된 기하이미지 [그림1, 우측]를 균일하게 샘플링(regular sampling) 하게 되면 완전 정규 연결 형태의 메쉬를 만들 수 있다. 매핑이 되지 않은 그리드 상의 점들은 보간법을 이용하여 구한다. 기하이미지로 표현된 메쉬는 LOD, 모핑(Morphing), 압축을 손쉽게 할 수 있는

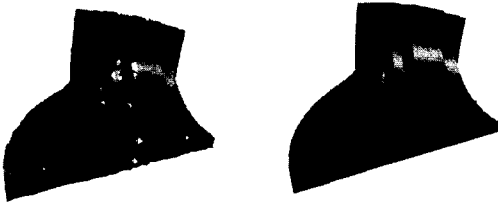
장점이 있다[2,5]. 하지만 평면으로의 매핑에 사용된 스트레치 매트릭(stretch metric)[1,2,3,4,5]은 리메싱된 메쉬에서 언더샘플링(undersampling)이 되는 것은 최소화[6] 하지만 오버샘플링(oversampling)이 되는 것은 줄이질 못한다. 정점은 기하 정보 이외에 법선 벡터를 가지고 있다. 인접 정점과의 법선 벡터 변화량이 작다는 것은 기하 정보의 변화가 작다는 의미이며 샘플링 비율(sampling rate)은 낮아야 된다. 우리는 법선 벡터의 변화량에 따라서 기하이미지의 샘플링 비율을 조절하는 방법을 제안한다.



[그림1] Geometry Images (Stanford Bunny) , [1]



[그림2] Mesh Optimization,[9]



[그림3] Feature Sensitive Remeshing,[10]

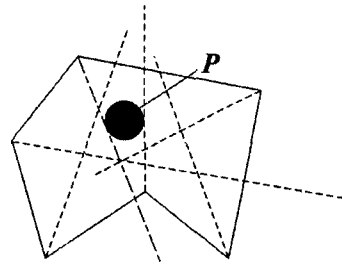
2. 관련 연구

기하이미지를 생성하는 방법은 Hoppe et al.가 제안하였다. 처음에는 원본 메쉬를 절단하여 원형 디스크에 매핑 시킨 다음 사각형 기하이미지를 방법[1]을 제시하였다. 이후에 메쉬를 프로그레시브메쉬(progressive mesh)[11]로 전환한 뒤 구면에 매핑을 하여 기하이미지를 생성하는 방법[2]과 차트를 이용하여 텍스처 아틀라스(texture atlas)[7,8]처럼 생긴 기하이미지를 생성하는 방법을 제시하였다.

메쉬최적화는 메쉬의 연결정보와 기하정보를 변형하여 메쉬가 대상이 되는 실세계의 사물을 보다 잘 표현하도록 한 리메싱 작업이다. Hoppe et al.은 [9]에서 에너지함수를 이용하여 간결하고 충실한(high fidelity) 메쉬를 생성하는 방법을 제시하였다[그림2]. Vorsatz et al.은 메쉬의 특성(feature)을 보다 잘 살릴 수 있는 방법[10]을 제시하였다. 정점을 곡률(hierarchical curvature)에 의해서 이동하여 간선(edge)을 일관되게 정렬하였다[그림3].

3. 기하이미지의 최적화

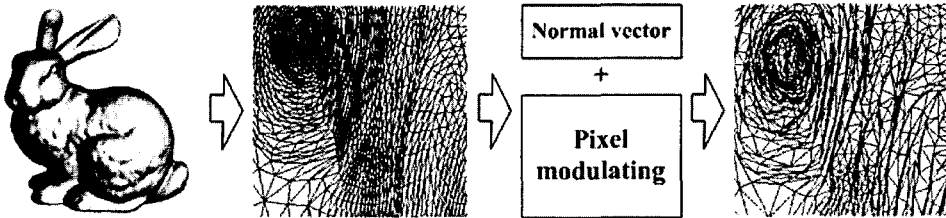
본 논문은 종수(genus)가 0인 모델을 단일 차트[1][2]로 생성한 기하이미지를 사용하였다. 2D 그리드에 매핑이 된 기하이미지는 인접 정점과 법선 벡터 차이를 구하여 이를 바탕으로 기하이미지 상의 픽셀은 이동이 된다. [그림4]는 기하이미지를 최적화 하는 과정을 나타내고 있다.



[그림5] 최소 이동 영역

3.1 픽셀의 최소 이동 영역 계산

이웃하는 정점과의 법선 벡터 차이가 클수록 기하이미지에서 넓은 영역을 차지해야 된다. 기하이미지는 균일하게 샘플링이 되므로 새로 생성되는 메쉬의 샘플링 비율은 기하이미지에서 차지하는 영역과 비례한다. 그러나 픽셀은 이동하면서 간선에서 교차되어서는 안 되므로 픽셀이 이동할 수 있는 영역을 우선적으로 한정시켜야 된다. 본 논문에서는 메쉬의 모든 정점에 해당하는 기하이미지 상의 픽셀은 동시에 이동하도록 하였다. 그래서 이웃하는 픽셀과 서로 겹치지 않도록 이동분하는 반직선을 사용하여 최소 영역을 구했다. 서로 겹치지 않는 최소 영역은 [그림5]로 나타낼 수 있다. 간선으로 연결되어 있는 세 개의 인접 픽셀 p_{i-1} , p_i , p_{i+1} 에서 p_{i-1} , p_i 와 p_i , p_{i+1} 의 사이각을 이등분하는 직선 h_p 그리고 이동시켜야 하는 원본 기하이미지의 픽셀을 P 라 할 때 이 픽셀이 이동할 수 있는 위치는 h_i 에 접하면서 P 를 중심으로 하는 원 중에서 그 반지름이 가장 작은 것의 내부가 된다. [그림5]에서 음영이 들어간 원 내부가 P 의 최소 이동 영역이 된다.



[그림4] 정점의 법선 벡터를 이용한 기하이미지의 최적화, 생성된 원본 기하이미지는 이미지상의 픽셀을 법선 벡터에 따라 그 위치를 이동하여 최적의 기하이미지를 생성한다.

3.2 픽셀의 이동(Pixel Modulating)

픽셀의 이동 방향과 이동 거리는 인접 픽셀과의 법선 벡터 차이에 의해서 결정 된다. P 와 인접 픽셀 p_i ($i=0\sim n-1$)간 법선 벡터의 차이를 n_i , 인접한 픽셀과의 거리를 d_i 라 하면 이동 거리는 n_i/d_i 값의 상대적인 크기에 의해서 결정된다. 이 값이 기준 값 보다 작게 되면, 법선 벡터 차이가 작지만 샘플링 영역은 크다는 의미이다. 그러므로 샘플링 영역을 작게 하기 위해서는 d_i 값이 작아져야 된다. 반대의 경우도 마찬가지다.

$$P' = \frac{\sum_{i=0}^{n-1} c_i}{n}, \quad [식1-1]$$

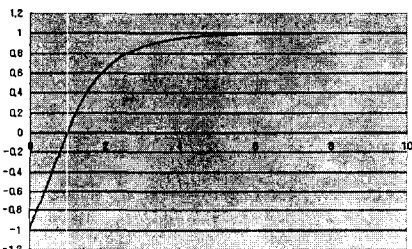
$$c_i = \frac{d_i P + r_i v_i}{d_i + r_i}, \quad [식1-2]$$

$$r_i = \begin{cases} R(t-1), & 0 < t < 1, \\ R(1-e^{-(1-t)}), & 1 < t, \end{cases} \quad [식1-3]$$

$$t = \frac{(n_i/d_i)}{(n_{std}/d_{std})} \quad \text{where } i=0\sim n-1. \quad [식1-4]$$

위 식에서 R 은 최소 이동 영역의 반지름이다. 그리고 n_{std} , d_{std} 는 기준으로 삼는 인접 픽셀과의 법선 벡터 차이와 거리가 된다. P' 는 p_i 과 P 의 내분점 혹은 외분점들의 평균점이 된다[식1-1]. 내분점 혹은 외분점 c_i 을 구하는 데[식1-2] 필요한 r_i 값은 기준이 되는 n_{std}/d_{std} 와의 비율로 r_i 값을 계산한다. [식1-3]은 c_i 가 t 에 상관없이 최소 영역 내에 존재하도록 만들어 준다. [그림6]은 r_i/R 값이 t 에 관계 없이 모두 -1과 1사이 존재함을 보여 준다.

[식1]에 의해서 P 는 인접 픽셀과 법선 벡터 차이가 적으면 인접 픽셀과의 거리를 줄임으로 샘플링 되는 영역을 줄이게 된다. 반대로 법선 벡터 차이가 크게 되면 인접 픽셀과의 거리를 늘임으로 샘플링



[그림6] 최소이동영역에 대한 픽셀의 이동거리 비율

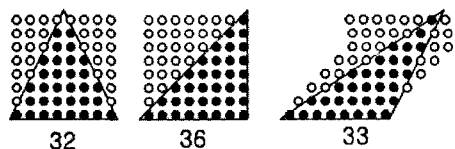
되는 영역을 늘리게 된다. 최소 이동 영역 내에서 P 는 이동하게 되므로 간선이 서로 겹치는 일은 없지만 그 이동거리는 제약을 받게 된다. 그래서 적절한 수준의 최적화된 기하이미지를 얻기 위해서는 반복적으로 위의 작업을 수행해야 된다.

3.3 종료 조건

최적화 작업이 수행되는 기하이미지는 작업이 종료되는 조건이 존재 한다. 최적화 작업이 무한히 수행된다면 법선 벡터 차이가 상대적으로 적은 픽셀들은 서로 같은 위치로 수렴을 하게 된다. 이런 현상을 방지하기 위해서는 픽셀사이의 최소 거리를 두어야 한다. 기하이미지는 디지털 형태의 데이터 이므로 그 해상도에 따라서 픽셀사이의 최소거리를 정할 수 있다. 그래서 최소거리 미만의 픽셀들은 서로 가까워지지 않도록 해야 된다. 이런 최소거리 미만인 픽셀들의 수가 사용자가 정한 한계치 이상이 되면 최적화 작업을 종료하게 된다.

3.3 기하이미지의 영역넓이와 샘플링 비율

기하이미지의 영역의 넓이는 샘플링 비율과 직접적으로 연관이 되어있다. 기하이미지에서 차지하는 영역이 넓은 것은 리메싱된 메쉬의 샘플링 비율이 높다는 의미이다. 영역의 넓이 이외에 삼각형의 형태에 따라서 샘플링 비율이 달라 질 수 있다. [그림7]은 동일한 넓이의 삼각형이 다른 샘플링 비율을 가지는 것을 보여주고 있다. 기하이미지상에서 영역의 넓이 이외에 픽셀이 이루는 삼각형의 형태에 대해서도 고려를 해 주어야 한다. 본 논문에서는 휴리스틱한 방법을 사용하여 세 개의 면 중에서 하나라도 기하이미지의 테두리와 평행한 것이 존재하면 가중치를 주었다.



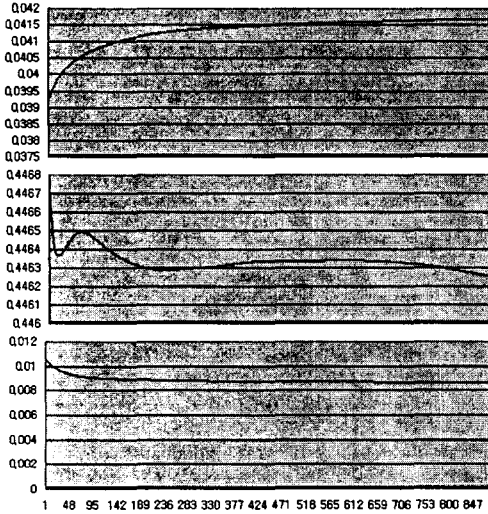
[그림7] 형태에 따른 샘플링 비율의 차이

4. 결과

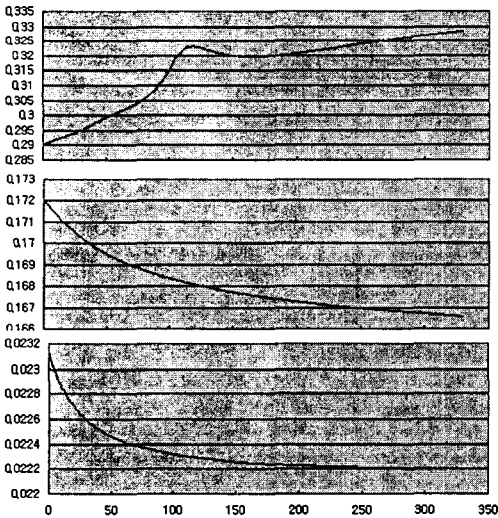
구면 매핑을 통해 생성된 기하이미지[2]를 사용하여 기하이미지 최적화 작업을 하였다. 정점이 34,818개 있는 Stanford Bunny와 11610개 있는 Cow 모델에 대해서 실험을 하였다. Pentium 4,

2.4GHz CPU, 512M RAM, Radeon 9000 Pro에 대해서 한 번의 최적화 작업은 각각 3초와 1초가 걸렸다. 최적화 작업의 횟수는 877번과 322번이 이루어졌으며 임의의 픽셀과 인접 픽셀로 이루어진 영역의 넓이를 측정하였다.

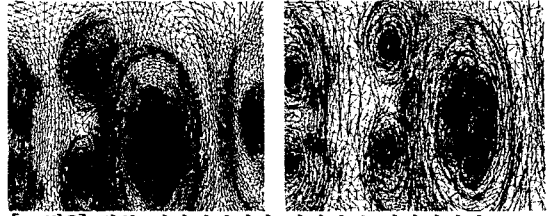
[차트1] Stanford Bunny



[차트2] Cow



차트는 법선 벡터 차이가 큰 영역, 중간 영역, 작은 영역의 넓이를 나타낸다. 영역의 구분은 법선 벡터 차이의 표준 편차와 평균을 기준으로 하였다. 차트에서 볼 수 있듯이 최적화 작업이 수행 될수록 법선 벡터 차이가 큰 영역은 기하이미지에서 차지하는 넓이가 커지게 된다. 반대로 차이가 작은 영역은



[그림8] 원본 기하이미지와 최적화된 기하이미지

넓이가 작아지게 된다. 결국 리메싱 되는 메쉬에서 샘플링 비율 또한 작아지게 되어 오버샘플링을 줄일 수 있다.

참고문헌

[1] X. Gu, S. J. Gortler H. Hoppe, "Geometry Images", ACM SIGGRAPH 2002.
 [2] E. Praun, H. Hoppe, "Spherical Parametrization and Remeshing", ACM SIGGRAPH 2003.
 [3] P.V. Sander, Z.J. Wood, S. J. Gortler, J. Snyder, H. Hoppe, "Multi-Chart Geometry Images", Eurographics Symposium on Geometry Processing 2003.
 [4] F. Losasso, H. Hoppe, S. Schaefer, J. Warren, "Smooth Geometry Images", Eurographics Symposium on Geometry Processing 2003.
 [5] H. Hoppe, E. Praun, "Shape Compression using Spherical Geometry Images", Multiresolution in Geometry Modeling 2003.
 [6] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, M. Halle, "Conformal Surface Parametrization for Texture Mapping", IEEE Transactions on Visualization and Computer Graphics 2000.
 [7] J. Maillot, H. Yahia, A. Verroust, "Interactive Texture Mapping", ACM SIGGRAPH 2001.
 [8] P. Cignoni, C. Montani, C. Rocchini, R. Scopigno, "A General Method for Recovering Attribute Values on Simplified Meshes", IEEE Visualization 1998.
 [9] H. Hoppe, T. DeRose, T. Duchamp, "Mesh Optimization", ACM SIGGRAPH1993.
 [10] J. Vorsatz, C. Rossl, L. P. Kobbelt, H. P. Seidel, "Feature Sensitive Remeshing", EUROGRAPHICS 2001.
 [11] H. Hoppe, "Progressive Meshes", ACM SIGGRAPH 1996.