

외곽선으로부터 3차원 기하모형을 생성하는

효율적인 삼각형화 알고리즘

노성^o 신병석

인하대학교 전자계산공학과

g2021332@inhavision.inha.ac.kr^o, bsshin@inha.ac.kr

Efficient Triangulation Algorithm

for Reconstructing 3D Models from Contour Lines

Sung Roh^o Byeong-Seok Shin

Dept. of Computer Science & Engineering, Inha University

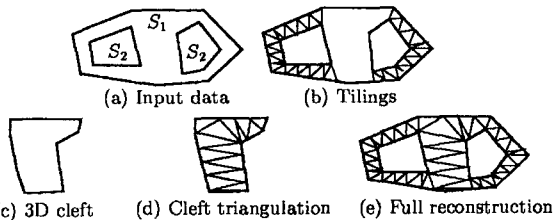
요 약

2차원 의료 영상으로부터 3차원 모델을 재구성 하면 여러 의학 분야에 효과적으로 활용할 수 있다. 컬러 영상이나 MRI의 영상은 표면을 자동으로 식별하기 어렵기 때문에 구역화한 결과로 나오는 외곽선으로부터 표면을 재구성해야 한다. 표면 재구성을 위한 기존의 삼각형화 알고리즘은 모델이 복잡할 경우 수행속도가 저하되는 단점이 있다. 본 논문에서는 정합되는 부분과 나머지 부분을 따로 처리하는 기존 방법 대신에, 알맞은 위치에 연결간선들만 결정하여 연결한 후, 그 사이의 정점들을 연결하는 간단한 삼각형화 알고리즘으로 속도를 향상시키는 방법을 제안한다.

1. 서 론

2차원의 의료 영상으로부터 3차원 모델을 재구성 하면 여러 의학 분야에 효과적으로 활용할 수 있다. 컬러 영상과 같이 자동 구역화가 어려운 경우나 MRI 영상과 같이 잡음이 많은 경우에는, 구역화한 결과로 나오는 외곽선으로부터 3차원 모델을 재구성해야 한다.

슬라이스별 외곽선 정보를 이용한 대표적인 모델 재구성 알고리즘으로는 Barequet과 Sharir의 알고리즘[1-3]이 있다.



[그림 1] Barequet과 Sharir의 삼각형화 알고리즘

(a)에서 연속된 두 슬라이스(S_1, S_2) 상의 외곽선을 입력으로 받는다. (b)에서 두 슬라이스의 각 정점들 중에서 가까운 부분을 찾아낸다. 즉, 두 슬라이스의 일부 연속되는 정점들의 리스트 (p_i, \dots, p_{i+l-1})과 (q_j, \dots, q_{j+l-1}) 사이에서, 파라미터 $\epsilon > 0$ 에 대하여, 인덱스 k 가 0에서 $(l-1)$ 로 증가할 때, 두 정점간의 거리 $\|p_{i+k} - q_{j+k}\|$ 가 ϵ 보다 작거나 같으면, 이 정점들의 리스트는 정합되는 부분으로 간주한다. 이렇게 찾은 부분을 먼저 타일화 한다. (c)에서는 두 슬라이스를 투영한 결과에서 (b)에서 삼각형화 된 부분을 제거한다. (d)에서는 남은 부분(클래프트)의 정점들에 대해서 Klineck의 동적계획법[4]을 적용하여, 간선 길이의 총합을 최소화 하는 방법으로 삼각형화를 수행한다. (e)에서 (b)와 (d)에서 구한 삼각형 메쉬들을 합치면 모델이 완성된다. 이 방법의 복잡도는 $O((n + \xi) \log(n + \xi) + \xi h^3)$

이다. ξ 는 클래프트의 개수, n 은 외곽선을 이루는 정점의 개수, h 는 가장 큰 클래프트에 속하는 정점의 개수이다. 여기서 $O((n + \xi) \log(n + \xi))$ 는 외곽선이 증첩되어 있을 경우 계층 구조를 찾는 데 드는 비용이고, $O(\xi h^3)$ 는 클래프트를 삼각형화 할 때 드는 비용이다. 따라서 이 알고리즘은 클래프트를 처리할 때 복잡도가 크게 증가한다. 이러한 문제를 해결하기 위해서 본 논문에서는 정합되는 부분과 클래프트들을 따로 잘라내어 처리하는 대신, 알맞은 위치에 연결간선들만 결정하여 연결한 후, 그 사이의 정점들을 연결하는 간단한 삼각형화 알고리즘을 제안한다.

본 논문은 다음과 같이 구성된다. 2절에서는 본 논문에서 제안한 방법을 살펴본다. 3절에서는 실험 결과를 보이고, 4절에서는 결론을 제시한다.

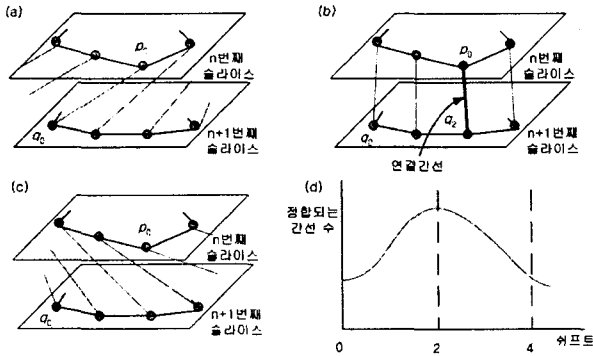
2. 효율적인 삼각형화 알고리즘

두 슬라이스(S_n, S_{n+1}) 상에 외곽선쌍(C_n, C_{n+1})이 있다면, 두 외곽선쌍에 포함되는 정점들을 연결하여 삼각형 띠를 만들어 낸다. 기본 원리는 두 개의 외곽선 중 임의의 한 정점에서 출발하여 이웃한 슬라이스의 외곽선에서 가장 가까운 정점을 차례로 연결하여 삼각형들을 생성하는 것이다. 그러나 실제로 인접한 두 슬라이스 상의 외곽선 모양이 다르고 외곽선을 구성하는 정점들의 개수가 다르기 때문에, 이러한 방법만으로는 올바른 삼각형 띠를 만들기 어렵다. 따라서 연결간선을 결정하여 연결하고, 그 사이의 정점들은 최근접 이웃 정점과 두 외곽선의 경계 정점 개수 비율을 적용해서 연결하는 방법을 사용한다. 또 분기가 있는 경우는 그에 적합한 처리를 해야 한다.

2.1 삼각형화의 연결간선 결정

외곽선쌍을 구성하는 각 정점들은 공간적으로 아무런 연관관계가 없고 있지 않으므로 알맞은 위치에 연결간선들을 결정하고 난 후, 그 사이의 정점들을 연결해야 올바른 삼각형 띠를 만들 수 있다. 연결간선을 결정하는 방법은 다음과 같다. 외곽선쌍의 각각의 정점들의 리스트를 (p_0, \dots, p_{l-1}), (q_0, \dots, q_{l-1})라

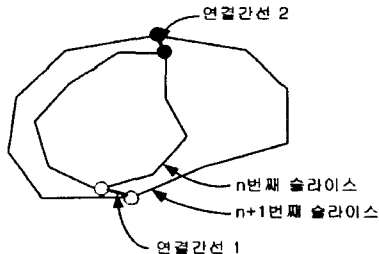
고 하자. 여기서 l 은 외곽선을 구성하는 정점들의 개수이고, 간선의 길이가 ϵ 이하이면 정합되는 간선으로 간주한다. 인덱스 k 가 0에서 $(l-1)$ 로 증가할 때 간선 $e(p_k, q_k)$ 는 [그림 2]의 (a)와 같이 잘못 연결되어 정합되는 간선 수가 적다. 그래서 C_{n+1} 의 첫 정점의 인덱스 j 를 0에서 $i-1$ 까지 증가시키면서 (p_0, \dots, p_{i-1}) 와 (q_j, \dots, q_{j+i-1}) 사이에서 간선 $e(p_k, q_{j+k})$ 중 정합되는 간선 수가 극대값이 되도록 하는 j 를 찾는다. 이 때 j 는 두 리스트에서의 상대적인 이동치이며 **쉬프트**라고 부른다. 정합되는 간선 수를 극대값으로 만드는 쉬프트를 적용했을 때의 간선들 중 가장 짧은 길이의 간선을 연결간선으로 정한다. 즉, 이 예에서는 쉬프트가 2일 때 정합되는 간선 수가 극대값을 갖게 된다. 따라서 연결간선으로는 쉬프트로 2를 적용했을 때의 간선 중 가장 짧은 간선인 $e(p_0, q_2)$ 가 결정된다.



[그림 2] 연결 간선 결정

(a) 쉬프트 = 0 (b) 쉬프트 = 2 (c) 쉬프트 = 4
(d) 쉬프트에 따른 정합되는 간선의 개수의 변화

극대값이 두 개 이상이 될 수도 있다. 이 경우에는 각각의 극대값에 대해서 해당 쉬프트를 적용했을 때 가장 짧은 간선을 구해서, 이를 연결간선으로 결정한다.

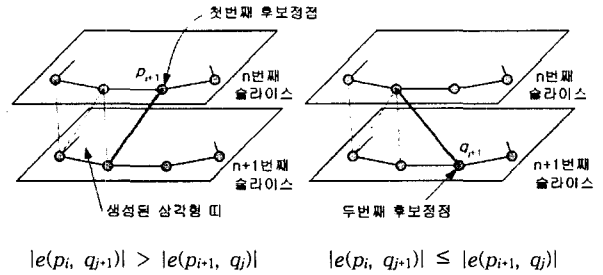


[그림 3] 복수의 연결간선을 가지는 외곽선쌍

2.2 최근접 이웃 정점의 계산

결정된 연결간선들 사이의 정점들을 연결하는 데에는, 최근접 이웃 정점과 두 외곽선의 경계정점의 개수비율을 사용한다. 이 중에서 최근접 이웃 정점을 계산하는 방법은 다음과 같다. 생성된 삼각형 띠가 시각적으로 양호한 형태를 가지도록 하려면 삼각형 띠를 구성하는 각각의 삼각형이 정삼각형에 가깝도록 해야 한다. 이를 위해서는 삼각형 띠에 새로운 정점을 삽입할 때 새로 생성된 간선의 길이가 최소화 되도록 하면 된다. 생성 중인 삼각형 띠의 마지막 삼각형을 구성하는 간선이 $e(p_i, q_j)$ 일 때 새로 삽입될 정점은 p_{i+1} 과 q_{j+1} 중에 하나이다. 따라서 이 두

후보정점을 연결한 후에 각 간선의 길이를 비교하여 짧은 쪽의 정점을 삽입한다.



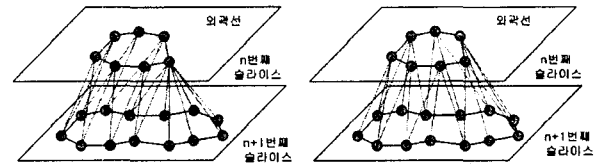
[그림 4] 최근접 정점의 선택

2.3 두 외곽선의 경계 정점 개수가 다를 경우의 처리

인체 장기의 복잡성을 고려할 때 인접한 두 슬라이스상의 외곽선이 동일한 개수의 정점으로 이루어져 있을 가능성은 매우 희박하다. 첫 번째 외곽선 C_n 이 l 개의 정점으로 이루어져 있고, 다음 외곽선 C_{n+1} 이 m 개의 정점으로 이루어져 있다고 하자. ($m > l$) 전술한 방식대로 차례대로 정점들을 추가하면 C_{n+1} 외곽선에서 $m-l$ 개의 정점들이 남게 되고 이로 인해 [그림 5]의 왼쪽 그림과 같이 한쪽으로 비틀린 삼각형 띠가 만들어진다. 이 문제를 해결하기 위해서 한 외곽선의 각 정점에 대응하는 상대편 외곽선의 정점개수를 균일하게 분배해 준다. 앞에서 예로든 경우, C_n 외곽선의 한 정점에는 C_{n+1} 외곽선의 $1 + (m-l)/l$ 개의 정점들이 대응되도록 해야 한다. 따라서 C_n 상의 정점 p_i 에 대응되는 C_{n+1} 의 정점의 인덱스는 다음과 같다.

$$j = \lfloor (1 + (m-l)/l)i + 0.5 \rfloor \quad (\text{식 1})$$

이렇게 하면 [그림 5]의 오른쪽 그림과 같이 올바른 삼각형 띠를 생성할 수 있다.



[그림 5] 외곽선의 정점개수가 다를 경우의 삼각형 띠(왼쪽), 외곽선의 정점대용관계를 균일하게 한 삼각형 띠(오른쪽)

2.4 분기가 있는 외곽선의 삼각형화

슬라이스에서 외곽선끼리의 연결 관계는, 1:1이 아닌, N:1 또는 1:N($N>1$)의 외곽선 개수로 연결될 수 있다. 이렇게 분기가 있는 경우는 각각의 외곽선에 연결될 부분을 정해주고 나서 전술한 방법으로 삼각형화 한다.

n 번째 슬라이스의 두 외곽선 C_1, C_2 가 $n+1$ 번째 슬라이스의 C_3 와 연결이 되는 경우, 외곽선을 삼각형화 하는 방법은 [그림 6]과 같다.

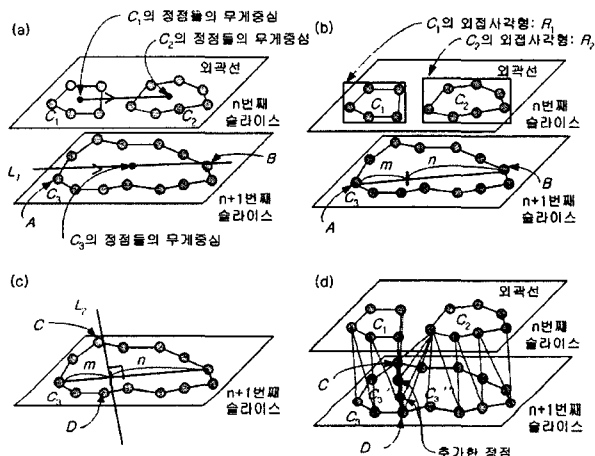
(a)와 같이 C_1 와 C_2 에 속하는 정점들로부터 두 무게중심을 구한다. 이 두 점을 잇는 직선과 기울기가 같으면서 C_3 에 속하는 정점들의 무게중심을 지나는 직선 L_1 을 구한다. C_3 에 속하는 정점들 중 직선 L_1 과 가장 가까운 양 끝점 A, B 를 구한다.

(b)에서 선분 AB 를 C_1 과 C_2 의 외접사각형 R_1, R_2 의 넓이의 비율 $m : n$ 으로 분할한다.

(c)에서 선분 \overline{AB} 를 분할한 중심점을 지나면서, 선분 \overline{AB} 과 직각을 이루는 직선 L_2 를 구한다. 다음으로 C_3 에 속하는 정점들 중 직선 L_2 에서 가장 가까운 양 끝점 C, D 를 구한다.

(d)에서 선분 \overline{CD} 위에 외곽선의 정점들의 정의에 맞추어 정점을 추가한다. 추가한 정점들을 중첩하여 외곽선 C_3 를 C_3' 와 C_3'' 로 분할한다. 마지막으로 C_1 의 정점들은 C_3' 의 정점들과, C_2 의 정점들은 C_3'' 의 정점들과 2.1-2.3절에서 설명한 방법으로 삼각형화 한다.

N:1 또는 1:N의 외곽선에서 N이 3 이상일 경우에는 위에서 설명한 알고리즘을 확장해서 적용한다.



[그림 6] 분기가 있는 외곽선의 삼각형화

2.5 비용 분석

본 논문에서 설명한 방법의 비용은 외곽선을 이루는 정점들의 개수에 대한 함수로 나타낼 수 있다.

수행과정별 비용은 다음과 같다. 먼저, 슬라이스 상의 외곽선이 중첩된 구조로 되어 있을 경우에, 외곽선의 정점들의 방향을 결정하는 데에는 $O(n \log(n))$ 의 비용이 소요된다. 그 다음으로 연결간선을 결정하기 위해 슈프리트별 정합되는 간선 수를 계산하는 데 $O(n)$ 의 비용이 소요된다. 마지막으로 연결간선 사이를 삼각형화 하는 데 $O(n)$ 의 비용이 소요된다. 모든 과정에 대한 비용 함수 $f(n)$ 은 다음과 같다.

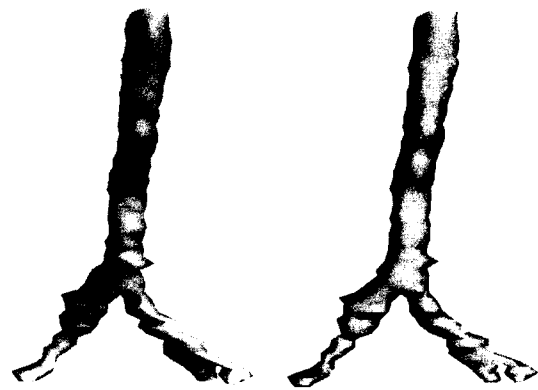
$$f(n) = kn \log(n) + k'n + k''n \quad (k, k', k'' \text{은 상수})$$

따라서 이 방법의 복잡도는 $O(n \log(n))$ 가 되고, 이는 기존 방법의 복잡도 $O((n + \xi) \log(n + \xi) + \xi h^3)$ 보다 낮다.

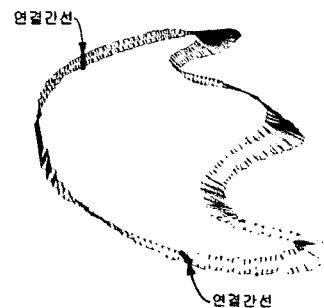
3. 실험결과

실험은 펜티엄IV 2.4GHz CPU와 1GB Main Memory, GeForce 4 Ti 4600 그래픽카드를 장착한 PC에서 하였다. 실험 데이터로는 MRI를 이용하여 성인 남성의 인체를 512*274*1239 해상도로 단층 촬영한 후, 다양한 해부구조물들을 구역화한 영상을 사용하였다.

[그림 7]은 3DS max에서 외곽선 정점들을 일일이 수작업으로 연결한 방법으로 3차원 영상을 모델링 한 결과(왼쪽)와, 본 논문에서 제안한 알고리즘으로 삼각형화를 해서 3차원 영상을 모델링 한 결과를 렌더링 한 것이다. 모델의 형태에 거의 차이가 없음을 알 수 있다. [그림 8]은 본 논문의 알고리즘을 이용하여 간의 일부를 3차원 재구성한 삼각형 띠이다.



[그림 7] 수동으로 만든 기관(왼쪽), 본 논문의 알고리즘을 이용한 기관(오른쪽)



[그림 8] 본 논문의 알고리즘을 이용한 간의 일부

4. 결론

본 논문에서는 기존의 연속되는 슬라이스 사이의 삼각형화 알고리즘의 효율성을 높이는 변형된 삼각형화 알고리즘을 제안하고 이를 실험하였다. 정합되는 부분과 이를 제외한 클래프트 부분으로 나누던 기존 방법과 달리, 알맞은 위치에 연결간선들만 결정하여 연결한 후, 그 사이의 정점들을 연결하는 간단한 삼각형화 알고리즘을 사용한다. 본 논문에서 제안한 방법은 기존 방법에서 비해서 복잡도가 낮아서 수행속도가 빠르고 효율적이다.

참고문헌

- [1] G. Barequet, D. Shapiro, and A. Tal, "Multilevel sensitive reconstruction of polyhedral surfaces from parallel slices", *The Visual Computer*, vol. 16 (2), pp. 116-133, 2000.
- [2] G. Barequet and M. Sharir, "Piecewise-linear interpolation between polygonal slices", *Computer Vision and Image Understanding*, vol. 63 (2), pp. 251-272, 1996.
- [3] G. Barequet, D. Shapiro, and A. Tal, "History consideration in reconstructing polyhedral surfaces from parallel slices", *Proc. IEEE Visualization, San-Francisco, CA*, pp. 149-156, 1996.
- [4] G. T. Klincsek, "Minimal triangulations of polygonal domains", *Annals of Discrete Mathematics*, vol. 9, pp. 121-123, 1980.