

# 외곽선을 이용한 효율적인 기하모델 재구성 기법

## (Efficient Geometric Model Reconstruction using Contour Lines)

정희상<sup>†</sup>      권구주<sup>†</sup>      신병석<sup>\*\*</sup>  
 (Hoe Sang Jung)    (Koo Joo Kwon)    (Byeong-Seok Shin)

**요약** 표면 재구성은 2차원 기하정보로부터 3차원 물체의 형상을 복원하는 방법이다. 기존의 표면 재구성 알고리즘 중 많이 사용되는 Barequet의 방법은 정합되는 부분을 먼저 타일화 하고, 정합되지 않는 부분인 클레프트는 동적 계획법으로 타일화 한다. 그러나 이 방법은 클레프트를 처리하는 시간이 오래 걸리기 때문에 모델이 복잡할 경우 수행속도가 저하되는 단점이 있다. 본 논문에서는 분기가 없는 단순영역을 한 번의 연산으로 외곽선의 최단거리 정점을 따라 타일화 하고, 분기가 있는 경우에는 정합이 잘되는 기본영역과 정합이 되지 않는 클레프트로 나눈다. 클레프트는 최단거리 정점들의 중점을 이용하여 간단하고 신속하게 타일화 하는 방법을 제안한다. 실험결과 기존 방법보다 mesh 재구성 속도와 정확도가 높아진 것을 확인할 수 있었다.

**키워드** : 외곽선, 표면 재구성, 타일화, 분기, 클레프트, 중앙축

**Abstract** 3D surface reconstruction is to make the original geometry of 3D objects from 2D geometric information. Barequet's algorithm is well known and most widely used in surface reconstruction. It tiles matched regions first, then triangulates clefts using dynamic programming. However it takes considerably long processing time while manipulating complex model. Our method tiles a simple region that does not have branches along minimally distant vertex pairs at once. When there are branches, our method divides contour lines into a simple region and clefts. We propose a fast and simple method that calculates medial axes using a minimum distance in cleft region. Experimental results show that our method can produce accurate models than the previous method within short time.

**Key words** : contour line, surface reconstruction, tiling, branch, cleft, medial axis

### 1. 서론

3차원 형상을 직접 모델링하거나 표시하기 위해서는 많은 비용이 소요되기 때문에 과거에는 연속된 단면 영상을 이용하여 단순히 표현하는 방법이 많이 사용되었다. 등고선만으로 지표면의 형태를 표현하거나 CT, MR 영상을 이용하여 인체구조를 표현하는 것이 좋은 예이다. 이러한 방법은 기억공간이나 처리 속도면에서 매우 효율적이지만 3차원 정보의 상당부분이 소실되어 있으므로 비숙련자가 단면으로부터 본래의 3차원 형상을 유추해내는 것은 쉽지 않다. 저렴한 고성능 컴퓨터의

등장으로 3차원 형상을 직접 저장하고 표현하는 것이 어렵지 않게 되었다. 이에 따라서 단면 영상으로부터 3차원 형상을 재구성하는 방법이 필요하게 되었다.

단면 영상으로부터 3차원 영상을 만드는 방법에는 볼륨 기반 방법과 표면 기반 방법이 있다. 볼륨 기반 방법은 2차원 단면 영상들을 쌓아서 얻은 볼륨 데이터에 직접 볼륨 렌더링 기법[1,2]을 적용하여 영상을 생성한다. 이 방법은 전처리 과정에서 3차원 모델을 만들지 않고도 영상을 생성할 수 있으나 렌더링 시간이 많이 걸리는 문제가 있다. 표면 기반 방법은 단면 영상에서 대상 물체의 외곽선을 수동 또는 자동화된 방법으로 추출한 후 외곽선들을 연결하여 본래의 3차원 형상을 복원하는 방법이다. 이 방법은 볼륨 기반의 방법에 비해 기억공간이 적게 소요되고 영상 생성 시간이 짧다는 장점이 있다. 또한 생성된 모델을 단순한 영상 생성뿐만 아니라 애니메이션, 시뮬레이션, 변형(deformation) 등에 활용할 수 있다.

표면 기반 방법에는 Marching Cubes[3]와 같이 3차

· 본 연구는 정보통신부 대학 IT 연구센터 육성, 지원사업의 연구결과로 수행되었습니다.

† 비회원 : 인하대학교 컴퓨터정보공학과  
 g2031372@inhavision.inha.ac.kr  
 g2012154@inhavision.inha.ac.kr

\*\* 종신회원 : 인하대학교 컴퓨터정보공학과 교수  
 bsshin@inha.ac.kr

논문접수 : 2004년 11월 29일

심사완료 : 2005년 6월 2일

원 블록 데이터에서 기하모델을 추출하는 방법과 단면 영상에서 외곽선을 추출한 후 이들을 타일화(tiling)하여 메쉬모델을 만드는 방법이 있다[4,5]. 본 논문에서는 평행한 2차원 영상위의 외곽선으로부터 3차원 형상을 자동으로 복원하는 3차원 모델 재구성 방법을 다룬다. 먼저 인접한 두 단면 영상에 있는 외곽선들의 대응하는 정점을 식별하고 그 정점들을 연결하는 간선을 삽입하여 영상 촬영 과정에서 손실된 기하정보를 복원한다. 기존의 표면 재구성 알고리즘 중에서 많이 사용되는 Barequet의 알고리즘은 인접한 외곽선 쌍을 검사하여 단순 정합이 가능한 부분과 클레프트 부분으로 나눈다[4,5]. 단순 정합부분에는 간단한 타일화 방법을 적용하여 삼각형 메쉬를 생성하고, 클레프트 부분의 정점들에 대해서는 동적계획법을 적용하여 삼각형화 한다[6]. 이 방법은 비교적 정확한 모델을 생성해내는 장점이 있지만 클레프트가 많은 복잡한 모델의 경우는 비효율적이다. 예를 들어 인체 해부구조물의 경우 대부분 복잡한 형태를 가지고 있기 때문에 이 방법을 적용하면 처리시간이 오래 걸리게 된다.

본 논문에서는 단면 영상의 외곽선으로부터 신속하게 3차원 모델을 생성할 수 있는 방법을 제안한다. 외곽선들 간의 대응관계를 결정하는 단계에서 분기(branch)의 유무를 쉽게 검출할 수 있다. 분기가 없는 경우는 정합 여부와 무관하게 한 번의 연산으로 외곽선의 최단거리 정점을 따라 타일화 함으로써 계산시간을 단축시킨다. 분기가 있는 경우는 정합이 잘되는 기본영역과 정합이 되지 않는 클레프트(cleft)로 나눈다. 클레프트를 타일화할 때는 먼저 중점을 계산하고 중점과 최외곽 정점들을 타일화 하는 방법으로 복잡도를 감소시킨다.

논문의 구성은 다음과 같다. 2장에서는 표면 재구성 방법과 관련 연구를 살펴보고, 3장에서는 본 논문에서 제안하는 방법으로 모델 재구성 알고리즘을 살펴본다. 4장에서 실험 결과를 보이고, 5장에서는 결론을 맺는다.

## 2. 관련 연구

표면 재구성 방법에서 고려할 것은 대응 관계 결정(correspondence determination), 타일화(tiling), 분기 처리(branch processing)로 나눌 수 있다[7]. 대응 문제는 임의 개수의 외곽선들을 가지는 연속된 두 슬라이스가 있을 때, 한 슬라이스의 특정 외곽선이 다른 슬라이스의 어떤 외곽선과 연결될지를 결정하는 것이다. 타일화는 대응되는 두 슬라이스 상에서, 임의의 모양을 가지는 두 외곽선들의 정점들을 연결하여 삼각형 띠(triangle strip)를 만드는 것이다. 분기 처리는 두 슬라이스에서  $M$ 개의 외곽선이  $N$ 개의 외곽선에 대응 될 때, 어떻게 타일화를 해야 하는지 결정하는 것이다. 그림 1은 대

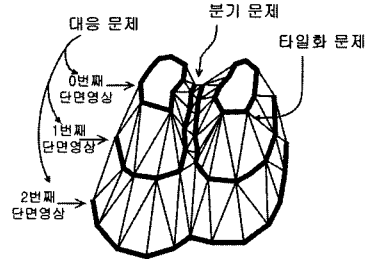


그림 1 인접한 단면영상에서의 대응 관계 결정, 타일화, 분기 처리의 예

응 관계 결정, 타일화, 분기 처리의 예를 보여준다.

대응 관계 결정은 외곽선들 간의 연결성을 찾는 것이다. 즉, 대응되는 외곽선 사이를 면으로 보간 하기 위해 인접한 외곽선들이 어떤 대응관계를 가지는지 찾아야 한다. 한 외곽선에 대응되는 외곽선이 한 개뿐인 경우는 이들을 타일화 하면 되지만 대응되는 외곽선의 개수가 두개 이상 된다면 올바른 모델을 생성하기 위해서 적당한 대응관계를 결정해야 한다. 대부분의 표면 재구성 알고리즘에서는 한 슬라이스의 외곽선들에 대응되는 인접 슬라이스의 외곽선들을 수직 방향으로 투영했을 때 서로 겹친다는 가정을 이용한다[8]. Wang은 대응관계를 결정하기 위해 서로 다른 외곽선의 영역이 겹치는지 검사하는 방법을 제안했다[9].

타일화는 한 쌍의 인접한 외곽선들로 부터 표면을 형성하는 삼각형 띠(triangle strip)를 만드는 것이다. 보통 인접 관계가 결정된 외곽선의 정점을 따라 삼각형화 한다. 그림 2에서 보는 것과 같이 타일화 되는 삼각형은 정확하게 2개의 슬라이스 현(slice chord)과 한 개의 외곽선의 부분으로 구성된다. Barequet과 Sharir은 외곽선들의 정합되는 부분을 위해 부분 곡선 정합 방법(partial curve matching technique)을 이용하고, 정합되지 않는 부분을 해결하기 위해서는 3차원 다각형의 최소면적 삼각형화(minimum-area triangulation) 방법을 사용했다[4,5]. 또한 간단하게 연결되지 않는 부분 사이의 보간을 위해서는 최소 신장 트리(minimum spanning tree) 휴리스틱을 사용했다. Bajaj는 재구성되는 표면에 세 가지 제약 조건을 정의하고 이 제약들로부터 정확한 대응규칙과 타일화 규칙을 얻었다. 이 제약 조건은 타일화된 지역이 실제의 구조를 따르고, 자연스러운 형태를 가지도록 한다[10]. Bajaj는 단단계 타일화 알고리즘을 사용했다. 이 알고리즘에서 복잡한 지역은 나머지 대부분의 영역이 타일화된 후 마지막에 타일화 된다.

분기 문제는 한 슬라이스에  $M$ 개의 외곽선이 있고, 인접한 슬라이스에  $N$ 개의 외곽선이 있을 경우에 발생

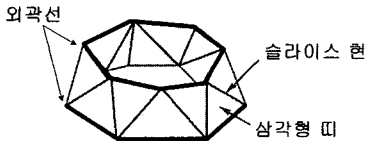


그림 2 두개의 외곽선으로부터, 삼각형 띠를 생성하는 예. 한 외곽선의 간선과 외곽선들을 연결하는 슬라이스현 두 개로 하나의 삼각형이 생성된다.

한다( $M \neq N, M, N > 0$ ). 그림 3(a)와 같이  $n$ 번째 슬라이스  $S^n$ 의 외곽선  $C_1$ 과  $C_2$ 를 투영하면  $n+1$ 번째 슬라이스  $S^{n+1}$ 의 외곽선  $C_3$ 에 포함된다. 그러면  $S_n$ 의 외곽선  $C_1$ 과  $C_2$ 의 사이에는 틈(hole)이 생긴다. 이런 틈은  $S_n$ 과  $S_{n+1}$  외곽선들의 표면 재구성을 어렵게 한다. 그림 3(b)~(e)는 분기로 인한 틈이 발생한 경우 여러 가지 해석방법들을 보여준다. 그림 3(b)는  $S_n$ 과  $S_{n+1}$  사이 공간에 간선  $L$ 을 보간하여  $C_1$ 과  $C_2$ 를 좀 더 부드럽게 타일화 한다[10]. 그림 3(c)는 외곽선  $C_3$ 를 대응되는 외곽선  $C_1$ 과  $C_2$ 의 넓이 비율에 맞춰 분할 한 후 분할된 각각의 외곽선을  $C_1$ 과  $C_2$ 와 타일화 하는 방법이다. 그림 3(d)에서는 외곽선  $C_1$ 과  $C_2$ 를 간선  $L$ 을 이용하여 하나의 외곽선으로 연결한 후 대응하는 외곽선  $C_3$ 와 타일화 하는 방법이다. 그림 3(e)에서는  $S_n$  슬라이스 상에서  $C_1$ 과  $C_2$ 를 타일화 함으로써 틈을 없앤다[4,5]. Bajaj나 Barequet는 분기 문제에서 유사한 부분을 먼저 타일화

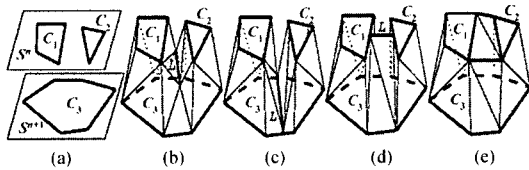


그림 3 분기 부분을 타일화 하는 방법 (a) 인접한 슬라이스의 외곽선 개수가 다른 경우 (b)  $C_1$ 과  $C_2$  사이의 공간에 간선  $L$ 을 보간하는 방법 (c)  $C_3$ 을  $C_1$ 과  $C_2$ 의 넓이 비에 맞춰 분할한 후 타일화 하는 방법 (d)  $C_1$ 과  $C_2$ 를 연결한 후 타일화 하는 방법 (e)  $C_1$ 과  $C_2$ 의 위치는 고려하지 않고 타일화 하는 방법

하고 나머지 부분은 따로 정점을 보간하여 확장 타일화 하는 방법을 제안했다[4,5,10].

### 3. 3차원 기하 모델 재구성 알고리즘

3차원 기하 모델을 재구성 하는 과정은 외곽선들 중에 서로 연결될 외곽선을 결정하는 대응 관계 결정 과정과 대응 관계가 결정된 후 두 개의 외곽선 사이의 면을 타일화 하는 과정으로 이루어진다.

#### 3.1 대응 관계 결정

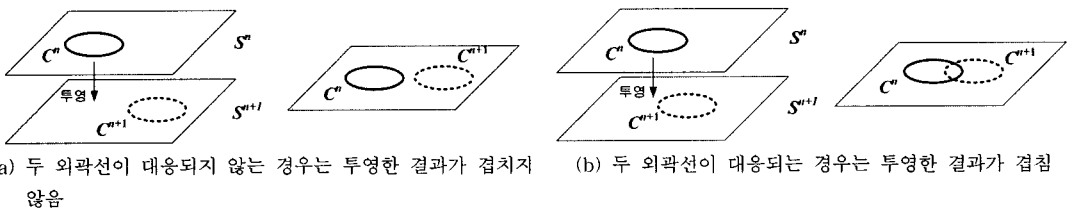
먼저 인접한 슬라이스에서 외곽선들 사이에 대응 관계를 결정해야 한다. 이 과정에서 서로 대응하는 외곽선들은 인접 슬라이스에 수직 방향으로 투영했을 때 외곽선이 겹치게 된다는 성질을 이용한다. 그림 4(a)는 대응되지 않는 두 외곽선의 예를 보여준다.  $S^n$ 의 외곽선  $C^n$ 과  $S^{n+1}$ 의 외곽선  $C^{n+1}$ 는 투영한 결과가 겹치지 않으므로 외곽선  $C^n$ 과  $C^{n+1}$ 는 서로 대응하지 않는다. 그림 4(b)는 대응되는 두 외곽선의 예를 보여준다.  $C^n$ 을 투영하면  $C^{n+1}$ 과 겹치므로 외곽선은 서로 대응된다.

#### 3.2 타일화

대응되는 두 외곽선이 정해지면 외곽선을 이루는 정점들을 연결하는 삼각형 띠를 만든다. 삼각형 띠를 만드는 기본 원리는, 외곽선 위의 임의의 한 정점에서 출발하여 이웃한 슬라이스의 외곽선에서 가장 가까운 정점을 차례로 연결하여 삼각형들을 생성하는 것이다.

##### 3.2.1 일반 타일화

$n$ 번째 슬라이스의  $i$ 번째 외곽선을  $C_i^n$ 이라하자.  $C_i^n$ 의 정점들은 대응되는  $C_i^{n+1}$  정점들을 따라 타일화 하게 된다. 그림 5(a)에서  $C_i^n$ 의 정점  $u_k$ 와 가장 가까운  $C_i^{n+1}$ 의 정점은  $u_k'$ 이고,  $C_i^n$ 의 정점  $u_{k-1}$ 과 가장 가까운  $C_i^{n+1}$ 의 정점은  $u_{k-1}'$ 이다. 최단거리 정점들을 구한 후에 이들을 차례로 연결하여 타일화 한다. 생성된 삼각형 띠가 보기 좋게 하려면 삼각형 띠를 구성하는 각각의 삼각형이 정삼각형에 가깝도록 해야 한다. 이를 위해서는 삼각형 띠에 새로운 정점을 삽입할 때 새로 생성된 간선의 길이가 최소화 되도록 해야 한다. 그림 5(b)에서 생성 중인 삼각형 띠의 마지막 삼각형을 구성하는 간선이  $e(u_k, u_{k-1})$ 일 때 새로 삽입될 정점은  $u_{k+1}$ 과  $u_{k-1}'$ 중에서 하나이



(a) 두 외곽선이 대응되지 않는 경우는 투영한 결과가 겹치지 않음 (b) 두 외곽선이 대응되는 경우는 투영한 결과가 겹침

그림 4 두 외곽선의 대응관계

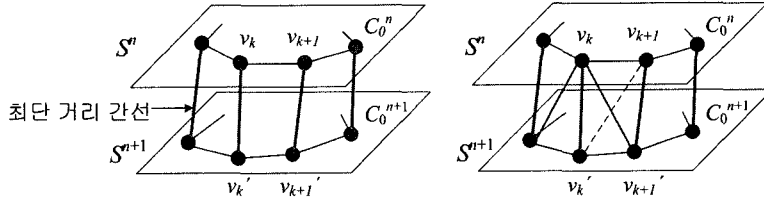


그림 5 최단접 정점을 연결하여 타일화 하는 예. 현재 간선이  $e(v_k, v_{k+1})$ 일 때 두 개의 후보정점  $v_{k+1}, v_{k+1}'$ 을 선택하고 이들을 연결한 간선  $e(v_k, v_{k+1}')$ 과  $e(v_{k+1}, v_k')$ 중에 짧은 쪽을 연결한다.

다. 따라서 이 두 후보정점을 연결하는 간선의 길이를 비교하여 짧은 쪽의 정점을 삽입한다. 그림 5(b)에서  $|e(v_k, v_{k+1}')| < |e(v_k', v_{k+1})|$  이므로  $v_{k+1}'$ 이 선택된다.

3.2 서로 모양이 다른 외곽선의 타일화

서로 모양이 다른 외곽선을 일반적인 방법으로 타일화 하면 표면 다각형 간에 교차가 발생하여 타일화가 잘못 될 수 있다. 따라서 모양이 다른 외곽선에 대해서는 확장된 타일화 방법을 사용해야 한다. 확장된 타일화는 대응관계가 결정된 외곽선에서 그림 6(a)와 같이 먼저  $C_0^n$ 와  $C_0^{n+1}$ 의 최단 거리 정점들을 찾는다. 앞서 설명한 기본 타일화를 진행하는 과정에서  $C_0^n$ 의 연속된 정점  $v_i$ 와  $v_{i+1}$ 에 대응하는 최단거리 정점인  $C_0^{n+1}$ 의 정점  $v_0$ 와  $v_k$ 사이의 정점 개수를 검사한다. 이 값이 정해진 임계치를 넘으면 외곽선의 모양이 다른 것으로 판별한다. 모양이 다른 경우는 인접 슬라이스의 외곽선과 타일화 하지 않고  $C_0^{n+1}$ 의  $v_0$ 와  $v_k$ 사이에 있는  $k+1$ 개의 정점들을 따로 모아서 타일화 한다. 이때 그림 6(b)와 같이 먼저  $v_0$ 와  $v_k$  정점쌍부터 시작하여  $(v_1, v_{k-1}), (v_2, v_{k-2}), \dots, (v_{k/2}, v_{k/2+1})$ 의 순서로 중점을 구한 후 타일화 한다. 그림 7(a)는 모양이 다른 외곽선에 기본 타일화 방법을 적용한 예이고 (b)는 확장된 타일화 방법을 적용한 경우이다.

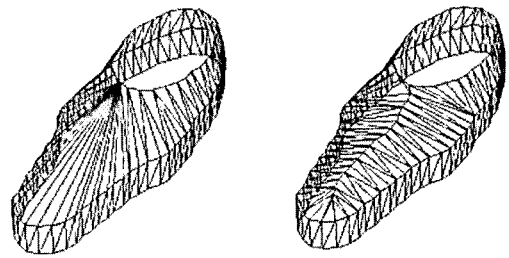
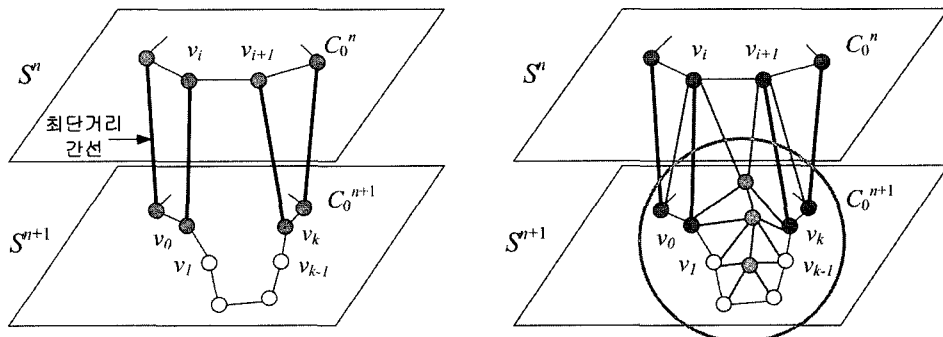


그림 7 두 외곽선의 모양이 다른 경우 타일화한 예 (a) 기본 타일화 방법을 적용한 결과 (b) 확장된 타일화 방법을 적용한 결과

3.3 분기 처리

하나의 외곽선은 인접 슬라이스의 여러 외곽선과 대응될 수 있다. 다시 말해 외곽선의 연결 관계는 1:1이 아닌, N:1 ( $N > 1$ ) 또는 N:M ( $N, M > 1$ )이 되는 것이다. 이것을 분기라 한다. 분기가 생기는 외곽선은 정합이 잘 되는 부분과 그렇지 않은 부분으로 나뉜다. 여기서 정합이 잘 되는 부분을 기본영역이라 정하고, 남은 영역을 클레프트라 한다. 기본영역은 3.2절에서 언급한 방법으로 타일화 한다. 클레프트 부분은 다시 최단거리를 이용하여 중앙



(a) 형태의 유사성 판별

(b) 동일한 슬라이스의 외곽선 정점들끼리 타일화 함

그림 6 서로 모양이 다른 외곽선들을 타일화 하기위한 과정

축(medial axis)을 구하고 정점을 보간 하여 타일화 함으로써 분기 부분의 3차원 모델을 재구성 할 수 있다.

3.3.1 1:N 분기

1:N 분기는 한 외곽선에 2개 이상의 외곽선이 대응되는 경우이다. 대응되는 외곽선이 정해지면 먼저 기본영역과 클레프트를 찾는다. 기본영역과 클레프트를 구분하는 과정은 그림 8과 같다.

먼저  $v_i^n(k)$ 을  $C_i^n$ 의  $k$  번째 정점이라 하고  $v_j^{n+1}(k')$ 을

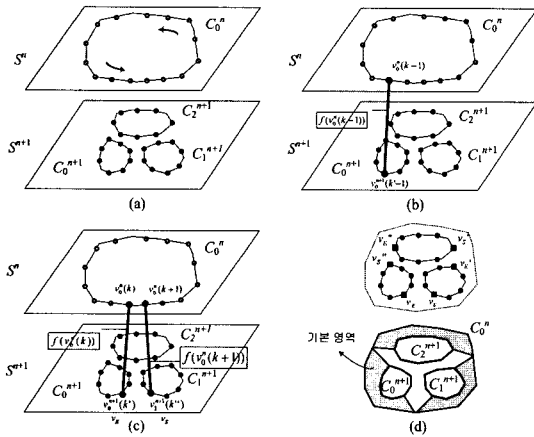


그림 8 최단 거리를 이용한 기본영역 식별 (a) 1:3분기가 검출된 경우 (b) 최단거리 정점 검색 (c) 인접한 최단거리 정점이 서로 다른 외곽선에 포함되는 지를 검사 (d) 검사된 정점들을 이용하여 기본영역 구분

$C_j^{n+1}$ 의  $k'$ 번째 정점이라 하자.  $S^{n+1}$ 에는  $Q$ 개의 외곽선이 있고  $C_j^{n+1}$ 는  $T$ 개의 정점으로 되어있다고 하자.  $f(v_i^n(k))$

은 인접한 슬라이스의 모든 정점들과 거리를 계산하여  $v_i^n(k)$ 와의 최단거리 정점을 계산하는 함수이다. 그림 8(b)는  $C_0^n$ 을 기준으로 하여  $C_0^{n+1}$ ,  $C_1^{n+1}$ ,  $C_2^{n+1}$ 상의 최단 거리 정점들을 찾는 모습을 보여준다.

$$f(v_i^n(k)) = \min_{p=0, q=0}^{Q-1, T-1} (|v_i^n(k), v_p^{n+1}(q)|) \quad (1)$$

$C_0^n$ 의 모든 정점에 대해서 대응하는 최단거리 정점들이 모두 동일한 외곽선에 포함된다면 분기가 없다는 것을 의미한다. 하지만 그림 8(c)와 같이 일부 정점들이 다른 외곽선의 정점들과 대응되면 분기가 발생한 것이다.

$C_0^n$ 의 모든 정점들에 대응하는 최근접 정점들을 찾는 과정에서 그 정점이 포함되는 외곽선이 변경되면 외곽선이 바뀌는 위치의 정점을  $v_E$ (end-of-cleft)와  $v_S$ (start-of-cleft)로 정한다(그림 8(c)).  $v_E$ 정점들과  $v_S$ 정점들이 정해지면 기본 영역과 클레프트 영역으로 나눌 수 있다. 그림 8(d)와 같이  $C^n$ 의 모든 정점들을 포함하고  $C_0^{n+1}$ 의  $v_S$ 정점부터  $v_E$ 정점까지( $C_0^{n+1}(v_S, v_E)$ ),  $C_1^{n+1}$ 의  $v_S'$ 정점부터  $v_E'$ 정점까지( $C_1^{n+1}(v_S', v_E')$ ),  $C_2^{n+1}$ 의  $v_S''$ 정점부터  $v_E''$ 정점까지( $C_2^{n+1}(v_S'', v_E'')$ ) 음영으로 표시된 부분을 기본영역으로 정한다. 나머지 영역은 클레프트가 된다. 영역 구분이 되면 기본영역을 먼저 타일화 한다. 이 때 3.2절에서 설명한 기본 타일화 방법을 사용한다.

클레프트 타일화 하는 과정은 다음과 같다.

- ①  $C_i^n(v_S, v_E)$ 의 부분영역인  $V_i^n(k)$ 에 대하여 클레프트의 중앙축을 만들기 위해 정점쌍  $\{V_i^n(k), f(V_i^n(k))\}$ 의 중점들을 계산한다(그림 9(a)-(c)).
- ② 클레프트 중앙에 틈을 제거하기 위해 모든 중앙축에 연결되는 중심점  $v^*$ 를 삽입한다.  $v^*$ 는 모든 중앙축의

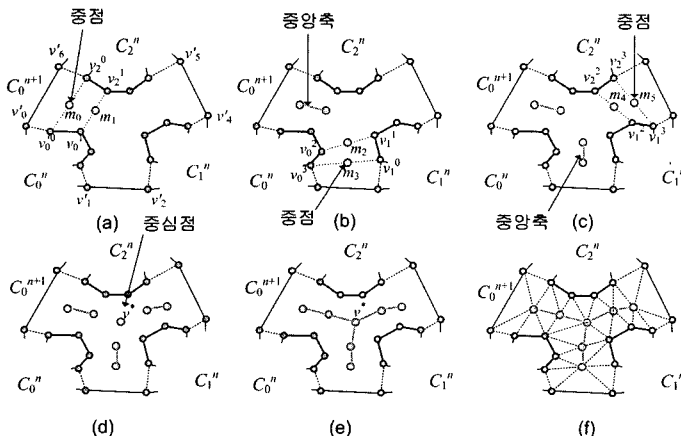


그림 9 클레프트를 타일화하는 과정 (a)-(c) 클레프트 영역의 중앙축 계산 (d)-(e) 중심점을 이용한 중앙축 연결 (f) 생성된 삼각형 메쉬

가장 가까운 끝점들로 이루어진 영역의 무게 중심점으로 계산한다(그림 9(d)).

- ③ 각각의 중앙축들을 정점  $v^*$ 를 통과하는 하나의 축으로 병합한다(그림 9(e)).
- ④ 클레프트 영역과 중앙축을 일반 타일화 방법을 사용하여 삼각형 메쉬를 생성한다. 클레프트의 외곽선은 정점  $\{v_0^0, v_0^1, v_0^2, v_0^3, v_1^1, v_2^1, v_1^0, v_1^1, v_1^2, v_1^3, v_3^1, v_4^1, v_2^0, v_2^1, v_2^2, v_2^3, v_5^1\}$ 로 구성되며, 중앙축은 정점  $\{m_0, m_1, v^*, m_2, m_3, m_2, v^*, m_4, m_5, m_4, v^*, m_1\}$ 로 이루어진다. 결과는 그림 9(f)와 같다.

그림 10은 1:3의 분기 외곽선 타일화 과정이다. 그림 10(a)는 기본영역만을 타일화 한 결과이고, 그림 10(b)는 최종 분기 외곽선의 클레프트까지 타일화한 결과이다.

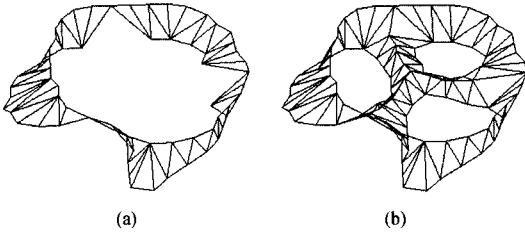


그림 10 1:3 분기 영역의 타일화 과정 (a) 기본영역만 타일화 한 결과 (b) 클레프트까지 타일화 한 결과

3.3.2 M:N 분기

그림 11(a)는 3:2로 분기가 발생한 모습을 보여준다. 이 경우  $S^n$  슬라이스와  $S^{n+1}$  슬라이스 외곽선만으로는 기본영역과 클레프트로 구분하여 타일화 하기가 어렵다. 그래서  $S^n$  슬라이스와  $S^{n+1}$  슬라이스 사이에 하나의 중간 외곽선을 삽입한다.

중간 외곽선을 삽입하여 타일화 하는 과정은 다음과 같다.

- ①  $S^n$  과  $S^{n+1}$  사이에 중간 슬라이스를 정하고  $S^n, S^{n+1}$ 의 외곽선을 그 위에 투영한다(그림 11(b)).
- ② 투영된 모든 외곽선들에서 최외곽에 있는 정점들만

추출하여 하나의 중간 외곽선  $S'$ 을 만든다.

- ③  $S^n : S'$  는 3:1인 분기이므로 N:1 분기영역 처리방법으로 기하정보를 생성한다(그림 11(c)).
- ④  $S' : S^{n+1}$ 는 1:2인 분기이므로 1:N 분기영역 처리방법으로 기하정보를 생성한다(그림 11(d)).

그림 12는 예시한 외곽선들을 타일화한 최종 결과이다.

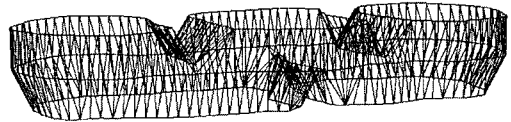


그림 12 M:N 분기가 있는 외곽선들을 타일화 한 결과

4. 실험 및 결과

인체의 단면 영상에 본 논문에서 제안한 알고리즘을 적용하여 표면을 재구성 하였다. 실험을 위해 펜티엄 IV 2.0GHz CPU와 768MB Main Memory, ATI Radeon 9600 PRO 그래픽 카드를 사용 하였다. 실험데이터는 성인 남성의 전신을 512×274해상도로 MRI를 촬영한 후, 다양한 해부구조물들을 구역화한 영상을 사용 하였다. 또한 지도의 등고선 데이터에 대해서도 동일한 실험을 수행하였다.

인체 단면 영상으로 만든 모델의 정확성을 알아보기 위해서 수작업으로 만든 3차원 모델과 비교하였다. 수작업으로 모델링 하는 방법으로는 3ds max™ 소프트웨어를 이용하여 테두리 구역화 영상을 모눈선(grid) 613개 위에 놓은 후 모든 정점들을 직접 지정해서 해부구조물의 3차원 모델을 만드는 방식을 이용하였다. 본 논문에서 제안하는 방법의 유효성을 보이기 위해 Barequet가 제안한 방법을 동일한 샘플 데이터를 이용하여 기하정보를 재구성하고 소요 시간을 측정하였다.

그림 13(a)는 인체 기관 중 불기뻐에 대해서 외곽선 정점들을 수작업으로 3차원 영상을 모델링한 결과이고 (b)는 제안한 알고리즘을 적용한 3차원 영상을 모델링한 결과를 렌더링한 것이다. 기존의 수작업한 모델과 거

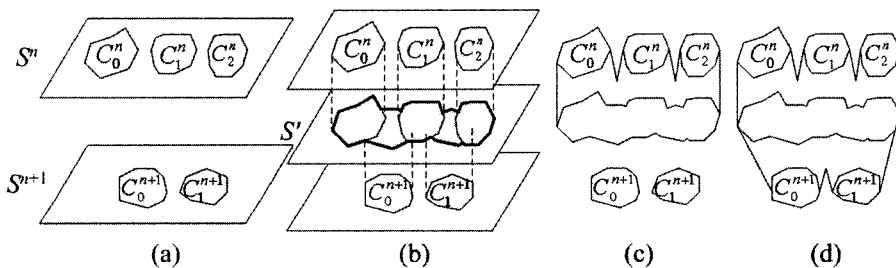


그림 11 3:2 분기 영역의 타일화 과정 (a) 3:2 분기가 발생한 경우 (b) 중간 평면에 모든 외곽선을 투영하여 하나의 외곽선 추출 (c) 중간 외곽선과  $S^n$ 의 외곽선들을 타일화 (d) 중간 외곽선과  $S^{n+1}$ 의 외곽선들을 타일화

의 차이가 없는 것을 알 수 있다. 그림 13(c), (d)는 인체 기관 중 기관지를 모델링 한 후 렌더링한 모습이다. 그림 14는 인체 기관 중 소화계통과 피부를 모델링 한 후 렌더링한 모습이다.

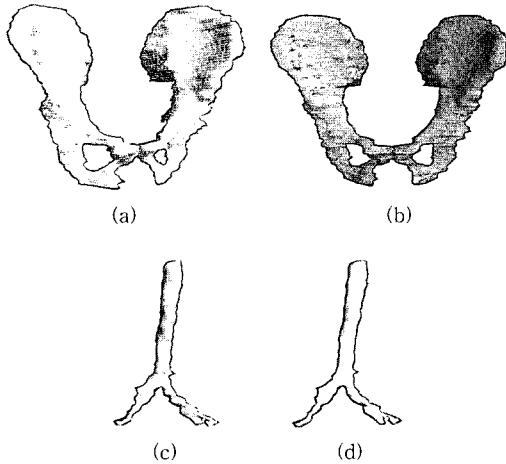


그림 13 불기뻐(위)와 기관지(아래) 모델링. 수작업으로 모델링 한 결과 (a, c), 본 논문의 방법으로 모델링 한 결과 (b, d)

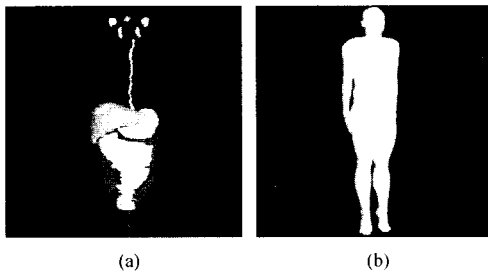


그림 14 인체기관 모델링 결과 (a) 소화기관 모델링 (b) 피부 모델링

그림 15는 Barequet이 고안한 방법과 이 논문에서 제안하는 방법의 처리시간을 비교한 것이다. 그림 16은 인접한 외곽선을 이용하여 분기 부분을 재구성한 것이다. 그림에서 볼 수 있듯이 본 논문의 방법을 사용하면 분기 영역을 좀 더 매끄럽게 표현할 수 있다. 실험에 의하면 본 논문에서 제안하는 방법이 Barequet 방법에 비해 10배이상 빠른 속도를 보이는 것을 알 수 있다.

그림 17은 본 논문의 방법으로 2D 등고선을 3D 지형 모델로 만든 영상이다. 복잡한 영역을 갖는 등고선에 대해 정확한 지형모델을 생성하였다. 표 1은 여러 2D 등고선을 이용하여 3D 모델을 생성한 수행시간을 비교하였다. 그림 18은 본 논문의 방법으로 1028\*1430 크기의 2D 등고선 영상을 3D 지형 모델로 만든 영상이다.

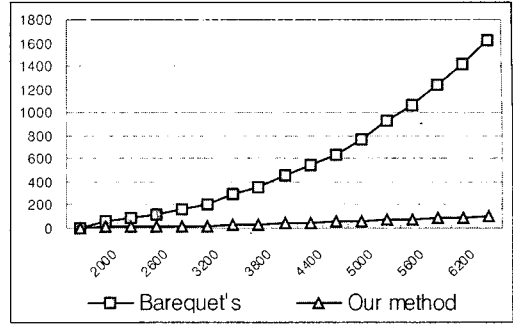


그림 15 두 가지 방법을 적용한 경우 모델 생성 시간 비교. X축은 삼각형의 개수이고 Y축은 시간 (msec)을 표시한다.

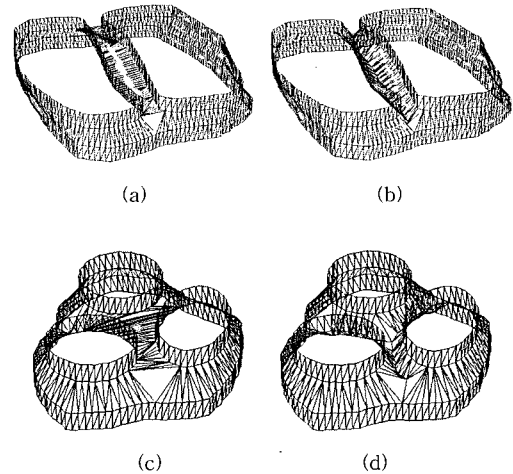


그림 16 동일한 슬라이스 데이터를 이용하여 Barequet의 방법으로 생성한 모델 (a, c)와 본 논문에서 제안하는 방법으로 생성한 모델 (b, d)를 비교한 그림

표 1 2D 등고선 영상을 이용한 3D 모델 생성 시간 비교

	단면영상 개수	삼각형 개수	수행시간(sec)
data A	7	1393	0.672
data B	10	2136	1.094
data C	7	1241	0.547

### 5. 결론

본 논문에서는 기존의 표면 재구성 알고리즘에 비해 처리속도가 빠른 알고리즘을 제안하고 실험하였다. 단순 정합되는 부분과 클레프트 부분으로 나누어 동적계획법으로 타일화 하는 기존 방법과는 달리, 분기가 없는 단순 영역은 형태에 관계없이 한 번에 외곽선의 최단거리를 따라 타일화 한다. 또한 분기가 있는 경우에는 정합이 잘되는 기본영역과 정합이 되지 않는 클레프트로 나

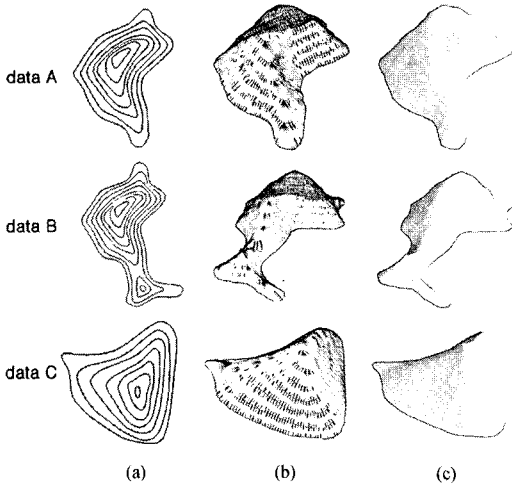


그림 17 2D 등고선을 이용한 지형 모델링의 예 (a) 2D 등고선 (b) 와이어프레임(wireframe) (c) 셰이딩(shading)

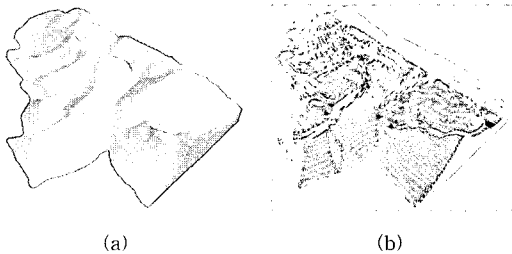


그림 18 1028\*1430 크기의 2D 등고선을 이용하여 지형 모델을 생성한 예 (a) 셰이딩한 결과 (b) 와이어프레임으로 렌더링한 결과

누어 정점들 간의 최단거리 정점을 사용하여 중점을 구함으로써 간단하고 빠르게 타일화 하는 방법을 제안하였다. 실험에서는 제안한 표면 재구성 방법으로 3차원 모델을 생성하였다. 표면 재구성 시스템을 통해 생성된 모델의 영상은 수작업으로 만든 모델의 영상과 큰 차이가 없는 것을 확인하였다. 또한 기존 방법보다도 빠른 시간에 기하모델을 생성할 수 있음을 확인하였다.

참 고 문 헌

[1] Levoy, M., "Display of Surfaces from Volume Data," IEEE Computer Graphics and Applications, Vol. 8, No. 3, pp. 29-37, 1988.  
 [2] Lacroute, P. and Levoy, M., "Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation," Computer Graphics (SIGGRAPH 94 Proceedings), pp. 451-458, Orlando, Florida, 1994.

[3] Lorensen, W. and Cline, H., "Marching cubes: A High Resolution 3D Surface Construction Algorithm," Computer Graphics, Vol. 21, pp. 163-169, 1987.  
 [4] Barequet, G. and Sharir, M., "Piecewise-linear Interpolation between Polygonal Slices," Computer Vision and Image Understanding, Vol. 63, No. 2, pp. 251-272, 1996.  
 [5] Barequet, G., Shapiro, D. and Tal, A., "Multilevel Sensitive Reconstruction of Polyhedral Surfaces from Parallel Slices," The Visual Computer, Vol. 16, No. 2, pp. 116-133, 2000.  
 [6] Klincsek, G., "Minimal Triangulations of Polygonal Domains," Annals of Discrete Mathematics, Vol. 9, pp. 121-123, 1980.  
 [7] Lederbuch, P., "Reconstruction of Surfaces from Planar Contours," Computer Engineering and Informatics, pp. 218-223, 1999.  
 [8] Meyers, D., Skinner, S. and Sloan, K., "Surfaces from Contours," ACM Trans. on Graphics, Vol. 11, No. 3, pp. 228-258, 1992.  
 [9] Wang, Y. F., and Aggarwal, J. K., "Surface Reconstruction and Representation of 3-D Scenes," Pattern Recognition, Vol. 19, pp. 197-207, 1986.  
 [10] Bajaj, C., Coyle, E. and Lin, K., "Arbitrary Topology Shape Reconstruction from Planar Cross Sections," Graphical Models and Image Processing, Vol. 58, No. 6, pp. 524-543, Nov. 1996.



정 회 상  
 2003년 원광대학교 공과대학 컴퓨터정보공학과. 2005년 인하대학교 대학원 컴퓨터공학과 석사. 2005년~현재 (주)인프라웨어 주임연구원. 관심분야는 Surface Rendering, 실시간 렌더링



권 구 주  
 1999년 우석대학교 공과대학 컴퓨터공학과. 2001년 인하대학교 대학원 전자계산공학 석사. 2001년~현재 인하대학교 대학원 컴퓨터정보공학과 박사과정. 관심분야는 볼륨 그래픽스, 의료영상, 실시간 렌더링



신 병 석  
 1990년 서울대학교 공과대학 컴퓨터공학과. 1992년 서울대학교 대학원 컴퓨터공학과 석사. 1997년 서울대학교 대학원 컴퓨터공학과 박사. 2000년~현재 인하대학교 컴퓨터공학부 조교수. 관심분야는 실시간 렌더링, 볼륨 그래픽스, 차세대 PC 인터페이스