

# 구역화를 이용한 디지털 격자지형데이터의 단순화 알고리즘

강 윤 식<sup>†</sup> · 박 우 찬<sup>††</sup> · 양 성 봉<sup>†††</sup>

## 요 약

본 논문에서는 지형 높이 정보(terrain height field data)로부터 불규칙 삼각망을 생성하는 갈망 삽입 알고리즘(greedy insertion algorithm)[1, 4, 5]에 기반 한 구역화 알고리즘을 제안한다. 이는 지형 높이 정보를 같은 크기의 사각형 영역으로 구역화 시켜, 각 구역들에 대하여 갈망 삽입 알고리즘을 이용해 단순화를 행한다. 기존의 갈망 삽입 알고리즘은 하나의 정점을 삼각망(triangulation)에 추가하기 위하여, 전체 지형 높이 정보에 대해 검색을 한다. 그러나 구역화 알고리즘은 하나의 정점을 삼각망에 추가하기 위하여, 검색되어야 하는 데이터의 범위를 현재 단순화가 진행되는 구역에 국한하여 수행성능을 향상시킨다. 또한 본 논문에서는 다양한 실험결과를 통하여, 구역화 알고리즘이 기존의 갈망 삽입 알고리즘에 비하여, 적게는 4배에서 많게는 20배 이상의 빠른 수행속도를 가지며, 구역화 알고리즘으로 생성되어진 단순화 지형의 에러는 기존의 방법과 비교해 거의 차이를 보이지 않고 있음을 보여준다. 구역화 알고리즘의 부수적인 이점으로, 구역화를 행함에 있어 각각의 구역 별로 삽입되는 4개의 모서리 정점들이 지형의 경계에 이들 정점들의 간격보다 큰 삼각형이 생성되지 못하게 한다. 이로 인해 기존의 갈망 삽입 알고리즘들에서 지형의 경계 주변에 생성되어지는 길쭉한 삼각형들의 생성을 억제하여, 보다 양질의 삼각망을 생성한다.

## A Digital Terrain Simplification Algorithm with a Partitioning Method

Yoon-Sig Kang<sup>†</sup> · Woo-Chan Park<sup>††</sup> · Sung-Bong Yang<sup>†††</sup>

## ABSTRACT

In this paper we introduce a fast simplification algorithm for terrain height fields to produce a triangulated irregular network, based on the greedy insertion algorithm in [1, 4, 5]. Our algorithm partitions a terrain height data into rectangular blocks with the same size and simplifies blocks one by one with the greedy insertion algorithm. Our algorithm references only to the points and the triangles within each current block for adding a point into the triangulation. Therefore, the algorithm runs faster than the greedy insertion algorithm, which references all input points and triangles in the terrain. Our experiment shows that partitioning method runs from 4 to more than 20 times faster, and it approximates test height fields as accurately as the greedy insertion algorithms. Most greedy insertion algorithms suffer from elongated triangles that usually appear near the boundaries. However, we insert the four corner points into each block to produce the base triangulation of the block before the point addition step begins so that elongated triangles could not appear in the simplified terrain.

† 정 회 원 : 연세대학교 대학원 컴퓨터학과  
 †† 준 회 원 : 연세대학교 대학원 컴퓨터학과  
 ††† 종신회원 : 연세대학교 컴퓨터학과 교수  
 논문접수 : 1999년 9월 30일, 심사완료 : 2000년 1월 21일

## 1. 서 론

3차원 그래픽스는 최근 들어 그 중요도가 더욱 높아지는 컴퓨터 과학의 한 분야로서, 비행/자동차 모의 프로그램, 의학/교육 시각화, 3차원 지형정보 시스템, 그리고 오락 프로그램 등의 다양한 응용 분야에서 매우 중요한 역할을 하고 있다. 이는 결과적으로 많은 3차원 하드웨어 및 3차원 알고리즘들의 개발을 가져왔다[3, 6, 7, 8, 9]. 그럼에도 불구하고, 이와 같은 3차원 하드웨어들은, 높은 해상도의 3차원 데이터를 실시간으로 렌더링하기에 아직 부족한 성능을 가지고 있다. 따라서 실시간 렌더링을 위해서는, 3차원 물체들을 시각적으로 중요도가 낮은 정도에 따라 폴리곤의 수를 줄이고, 이로 인한 렌더링 시간을 줄이는 것이 필요하다. 이와 같은 방법을 단순화라 한다. 단순화는 3차원 물체뿐만 아니라, 3차원 지형에도 매우 중요한 요소의 하나로, 본 논문에서는 3차원 지형 단순화 알고리즘을 제안한다.

지형 단순화의 가장 중요한 목적 중 하나는, 정점의 수를 최소화하면서도, 원 지형의 특징을 최대한 나타내어 주는 양질의 삼각망의 생성이다. 기존의 세밀화 갈망 삽입 알고리즘(refinement greedy insertion)은 높은 정밀도를 제공하는 알고리즘의 하나로써, 가장 단순한 초기 삼각망을 생성 한 후, 각각의 입력 점들에 대한 지역 에러를 구하여, 이들 중 가장 큰 값을 가지고 있는 점을 반복적으로 삽입시킴으로써, 원하는 정밀도의 삼각망을 생성한다. 이 알고리즘은 하나의 점을 삼각망에 삽입하기 위하여 아직 삽입되지 않은 모든 정점들에 대한 각각의 지역 에러를 구하고, 이를 비교한다. 따라서 정점을 삽입하기 위한 반복적인 과정에서, 아직 삼각망에 삽입되지 않는 많은 정점들에 대하여, 반복적인 지역 에러 값 계산과 비교 연산이 필요하다.

본 논문에서는 수행 속도를 개선하기 위하여, 전체 지형을 구역화하고 각 구역별로 세밀화 갈망 삽입 알고리즘을 적용시켜, 단순화를 수행하는 알고리즘을 제안한다. 이는 최대 지역 에러를 갖는 정점을 찾기 위한 범위를 축소하여, 수행 성능을 향상함과 동시에 갈망 삽입 알고리즘의 장점인 양질의 삼각망을 생성한다.

지금까지 지형을 컴퓨터 상에서 표현하고자 하는 많은 연구들이 행해 졌으며[2, 10, 11, 12, 13], 그 중 불규칙 삼각망(triangulated irregular network)은 가장 효율적이고 편리한 표현 방법중의 하나로, 하나의 지형을 3개의 표본 정점들로 연결된 삼각형 조각들의 집합으로

표현한다. 구역화 알고리즘은 지형 높이 정보로부터 불규칙 삼각망을 결과로 출력한다. 지형 높이 정보는 임의의 지형에서  $m \times n$ 의 각 격자위치의 높이를 나타낸다.  $p$ 는  $m \times n$ 의 각 추출 지점이고,  $H(p)$ 는  $p$ 의 위치에 해당하는 지형의 높이일 때,  $(p, H(p))$ 로 표현된다. 또한 불규칙 삼각망은 이들  $p$ 와 이들에 대한 높이 정보  $H(p)$ 의 집합에 대하여, 임의의 조건에 의해 선택된 부분 집합이다. 본 논문의 가장 기본적인 목적은, 지형 높이 정보로부터 가장 중요한 의미 점들을 추출하여, 낮은 에러를 유지하면서 최대한 빠르게 불규칙 삼각망을 생성한다.

본 논문에서는, 이후 2장에서 기본적인 갈망 삽입 알고리즘에 대하여 설명하고, 3장에서 구역화 방법에 대하여 소개를 하고, 4장에서는 실험의 결과를 보인다. 마지막으로 5장에서는 본 논문의 결과를 서술한다.

## 2. 관련 연구

갈망 삽입 알고리즘에서 사용되는 델러니 삼각망은 보로노이 다이어그램의 사이트들을 연결하여, 그 사이트 집합에 대하여 유일한 삼각망을 생성한다. 2장에서는 갈망 삽입 알고리즘에 사용되는 기본적인 알고리즘들에 대하여 설명 한다.

### 2.1 보로노이 다이어그램(Voronoi Diagrams)

보로노이 다이어그램은 유클리디안(Euclidean) 평면을 분할하는 일련의 사이트(sites)의 집합이다.  $P = \{p_1, p_2, \dots, p_n\}$ 는 2차원의 유클리디안 평면 위에 존재하는 정점들의 집합으로 이를 사이트라 하고, 이들 사이트들은 평면 위에 존재하는 모든 정점들을 이들과 가장 가까운 사이트에 할당함으로써 유클리디안 평면을 분할한다. 또한 그 평면에 존재하는 정점들은 보로노이 영역(region)  $V(p_i)$ 에 할당된다.

$$V(p_i) = \{x | p_i - x \leq p_j - x, \forall j \neq i\}$$

가장 가까운 사이트가 2개 이상인 정점들은 보로노이 다이어그램의 에지(edge)에 존재하게 된다. 그리고 하나의 정점이 4개의 최단거리의 사이트를 가지고 있는 경우에는, 보로노이 다이어그램이 그 정점의 4개의 최단거리 사이트를 갖지 않도록 변형된다.

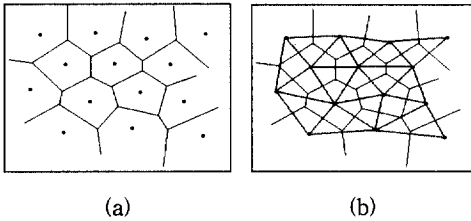
### 2.2 델러니 삼각망(Delaunay Triangulation)

1934년 델러니(R. Delaunay)는 보로노이 다이어그램

으로부터, 그 다이어그램에 유일한 2차원 삼각망을 생성해 내는 조건을 증명하였다[13]. 만약 보로노이 다이어그램내의 3개 사이트의 외접원내에 다른 사이트가 존재하지 않는다면, 그 다이어그램내의 사이트들을 인접한 보로노이 영역의 사이트들과 직선으로 연결함으로써 유일한 평면 삼각망을 생성한다. 이러한 삼각망을 딜러니 삼각망이라 하며, 이 딜러니 삼각망은 2가지의 중요한 특징을 가지고 있다.

- 임의의 보로노이 다이어그램에 대하여, 유일한 딜러니 삼각망이 존재한다.
- 딜러니 삼각망 내의 삼각형들은 모두 예각 삼각형이다.

(그림 1)은 보로노이 다이어그램과 그에 대한 딜러니 삼각망을 나타낸다.



(그림 1) 보로노이 다이어그램(a)과 그에 대한 딜러니 다이어그램(b)

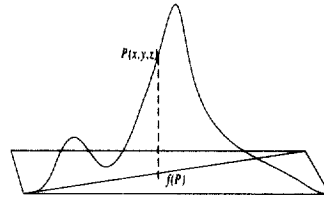
2.3 갈망 삽입 알고리즘(Greedy Insertion Algorithm)

갈망 삽입 알고리즘은 가장 낮은 정밀도를 갖는 삼각망을 생성 후, 원하는 정밀도까지 그 삼각망에 지형의 특징을 가장 잘 표현하는 정점들을 반복적으로 삽입한다. 따라서, 정밀화 알고리즘이라고도 한다.

플로리아니의 갈망 삽입 알고리즘[1]은 지형의 모서리에 해당하는 네 개의 정점으로 초기 삼각망을 생성한다. 이 삼각망은 두 개의 삼각형을 포함하고, 전체 지형을 대표한다. 이 초기 삼각망에 지형의 특징을 가장 잘 표현할 수 있는 특징 점들을 반복적으로 선택/삽입시켜 원하는 정밀도를 갖는 삼각망을 생성한다. 특징 점을 선택하기 위한 방법으로 지역 에러(local error) 측정 법을 사용하며, 지역에러  $e(p)$ 는 점  $p=(x, y, z)$ 의  $z$  값과 이 점  $p$ 를  $z$  방향으로 투영시켰을 때,  $p$ 를 포함하는 평면상의 좌표  $p'=(x, y, z')$ 의  $z'$ 값과의 차이이다(그림 2). 이것은 아래의 식 (1)로 쓰여 질 수 있으며,

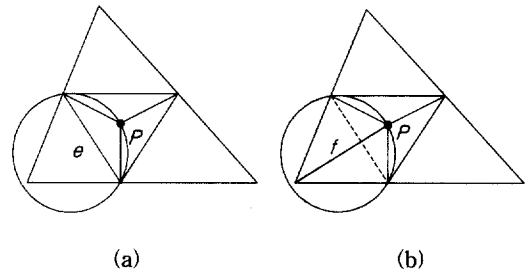
$$e(p) = |z - f(p)| \tag{1}$$

여기서  $f(p)$ 는  $z'$ 의 값을 돌려주는 함수이다.



(그림 2) 지역 에러 측정방법

점  $p$ 가 이를 포함하고 있는 삼각형  $t$ 에 삽입되는 경우, 삼각형  $t$ 는 3개의 작은 삼각형으로 세분화된다. 이들 3개의 삼각형들에 대하여 딜러니 삼각망을 만족하는지의 평가가 이루어진다.



(그림 3) 원 평가 (a)와 면 변형 (b)

플로리아니는 그의 논문[2, 4, 5]에서 원 평가(circle criterion)방법을 이용하여, 이를 평가하였다. 이 평가 방법은 정점들의 집합  $P$ 로 이루어진 삼각망  $T$ 에 대하여,  $P$ 의 어떠한 정점도 삼각망  $T$ 의 임의의 삼각형의 외접원 내에 있지 않을 경우, 삼각망  $T$ 를 딜러니 삼각망이라 정의한다. 만약 추가되어진 점  $p$ 에 의해 쪼개어진 3개의 삼각형들 중 그들의 외접원내에 하나 이상의 점을 포함하고 있는 삼각형은 바깥쪽의 면(edge)  $e$ 를 대각선의 면  $f$ 로 위치 변환하여 딜러니 삼각망을 만족시킨다(그림 3).

3. 구역화를 이용한 지형 단순화

대부분의 갈망 삽입 알고리즘들은 고성능 워크스테이션 환경에서조차 일반적인 크기의 디지털 지형 정보를

단순화하는데, 적게는 수십 초에서 많게는 수십 분의 시간을 필요로 하고 있다. 이것은 이와 같은 일반적인 갈망 삽입 알고리즘들이 단순화과정에서 하나의 정점을 삼각망에 추가하기 위하여, 모든 점들과 삼각형들에 대하여 조건 검사를 행하기 때문이다. 따라서 본 논문에서는 조건 검사를 위한 점들과 삼각형들의 수를 줄이기 위하여, 아래의 구역화 방법을 사용하였다.

**A Fast Digital Terrain Simplification with a Partitioning**

**Method :**

```

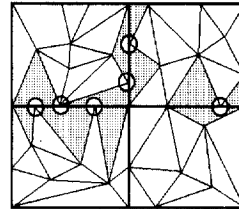
/* 전체 지형의 k x l 개의 구역화 */
Partition a terrain into kx l blocks;
/* 초기 삼각망 생성 */
for each blocks b
    Make_BaseTriangles(b);
end for
/* 구역별 정점삽입 */
for each blocks b
    Init_Triangles(b);
    Init_Error(b);
    Find the local maximum error e and the point p
    by calling Get_Max_Error&Point(b);
    while error <= r do /* r = 일고자하는
    삼각망의 정밀도 */
        Insertion(p);
        Recalculation(b);
        Find the local maximum error e and
        the point p
        by calling Get_Max_Error&Point(b);
    end while
end for
    
```

위에 제시된 알고리즘은 구역화를 이용한 갈망 삽입 알고리즘을 처리하기 위한 슈도(pseudo) 코드이며, 크게 구역화, 초기 삼각망 생성, 그리고 정점 삽입으로 구성된다. 이번 장에서는 이를 자세히 살펴보겠다.

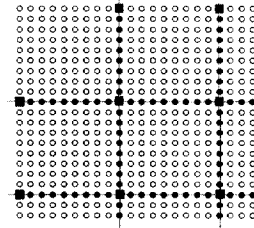
**3.1 지형 정보의 구역화**

지형의 구역화는 전체 지형을 몇 개의 작은 영역으로 나눔으로서, 정점 삽입 단계에서 고속 수행을 보장하는 방법을 제공한다. 가장 일반적인 구역화 방법은, 각 정점들이 유일한 하나의 구역에만 포함되도록 구역화를 수행하는 방법이다. 이 경우, 각 구역들은 인접한 구역과 어떠한 지형적인 정보도 공유하지 않고, 구역별로 독립적인 초기 삼각망이 생성된다. 따라서 정점 삽입 또한 각 구역의 삼각망에 대하여 개별적으로 이루어진 후, 이들 삼각망은 전체 지형으로 통합되는 과정이 필요하다. 또한 각 구역별로 개별적인 삼각망들이 생성이 되고, 이들은 정점이 삽입되는 과정에서 인접한 지형을 전혀 고려할 수 없으므로, 각 구역의 경계가 (그

림 4)와 같이 일치하지 않는 경우가 발생할 수 있다. 따라서 삼각망의 통합 시에 이를 해결하기 위한 처리가 필요하다. 이는 불필요한 삼각형의 생성을 야기할 수 있고, 또한 전체적인 수행 성능의 저하를 가져온다.



(그림 4) 주변의 삼각형들과 일치하지 않는 삼각형들(회색)



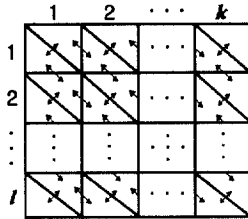
(그림 5) 인접한 블록들에 의해 공유되어지는 점들(검정색)

본 논문에서는 앞에서 언급한 문제점들을 해결하기 위하여, (그림 5)에서 보이고 있는 것과 같이, 각 구역들이 경계 점들을 공유하도록 구역화를 행한다. 여기서 검정색 사각형으로 표현되고 있는 정점들은 인접한 구역과 공유는 동시에, 각 구역의 초기 삼각망을 생성하기 위해 사용된다. 검정색으로 채워진 원으로 표현된 정점들은, 두 개의 구역간에 공유되어지는 면(edge) 상에 존재한다. 따라서 각 구역별 삼각망들은 이들 정점에 의해 서로 연결되어, 지형 전체를 위한 하나의 삼각망만을 생성한다. 또한 하나의 삼각형의 변화는 인접한 삼각형이 어느 구역에 속해 있는지에 관계없이, 그들에게 변형된 상태가 전파된다. 이를 통하여 우리는 매우 간단하고 효율적으로 구역화를 통해 생기는 문제점을 해결하였다.

**3.2 초기 삼각망**

구역화 알고리즘은 각 구역별로 정점을 삽입하기 위

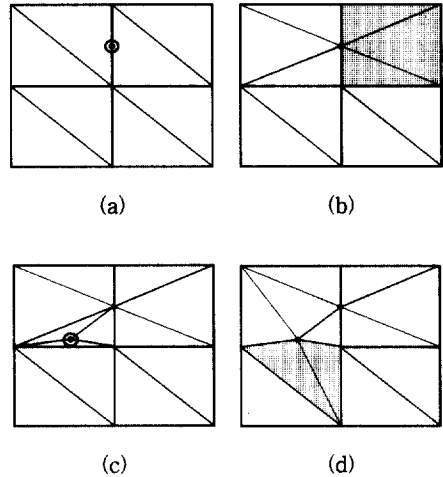
해 (그림 6)과 같이 각 구역별로 두 개씩의 초기 삼각형을 생성한다. 초기 삼각망을 생성하기 위해 구역화 알고리즘에서는  $(k+1) \times (l+1)$ 개의 초기정점을 삽입한다. 이들 초기 정점은 그 지형을 표현한 중요도가 검증이 되지 않은 상태에서 삽입되므로, 생성되는 삼각망의 정밀도를 낮출 수 있는 가능성을 내포하고 있다. 따라서 이들의 수를 최소화하는 것이 생성되어지는 삼각망의 질적인 향상의 중요한 요소이다. 반면, 구역의 수가 증가할수록 수행 속도는 증가하므로, 이 둘을 모두 만족시킬 수 있는 적절한 구역의 수를 결정하는 것이 중요하다. 따라서 본 논문에서는 구역의 수를 달리한 삼각망을 생성하여, 이들의 정밀도를 비교하는 실험을 하였고, 생성되는 삼각망의 정밀도가 높아질수록 초기 정점에 의해 발생하는 에러는 감소하게 되며, 특정 정밀도 이상에서는 기존의 갈망 삽입 알고리즘과 차이를 보이지 않음을 확인하였다.



(그림 6) 구역별 두 개의 삼각형으로 구성된 초기 삼각망

### 3.3 정점 추가

구역화 알고리즘에는 하나의 점을 추가함으로써 현재의 구역에 포함되지 않는 삼각형들의 모양을 변화시키는 두 가지 경우가 있다. 첫 번째는, 하나의 점이 현재 구역의 경계선상에 추가되는 경우로, (그림 7)의 (a)와 같이 두 구역의 경계에 정점이 추가되어 이들 인접한 구역들의 삼각망이 변형된다. 두 번째는 (그림 7)의 (c)와 같이 현재 정점삽입이 발생하고 있는 구역의 가장 바깥쪽 경계에 위치한 삼각형에 점이 추가되어, 이 삼각형이 3개의 작은 삼각형들로 세분화되고, 이들 중 가장 바깥쪽에 위치한 삼각형이 덜러니 삼각망 조건을 만족시키기 위해, 인접한 구역내의 삼각형과 번 교환 (edge swapping)이 발생한 경우이다. 이 두 가지 경우에는 구역화 알고리즘은 정점 삽입이 발생하는 구역의 변화가 인접 구역에 전파된다.



(그림 7) 현재 단순화가 진행되지 않고 있는 블록의 삼각형들의 모양이 변하는 경우. (a) 두 개의 블록의 경계에 점이 추가되어짐. (b) 두 개의 구역에 대하여 삼각형들의 세분화가 발생한다. (c) 구역의 경계에 인접하여 포인트가 추가되어진다. (d) Delaunay triangulation 조건에 의하여 인접한 블록의 삼각형들의 모양이 변한다.

## 4. 실험

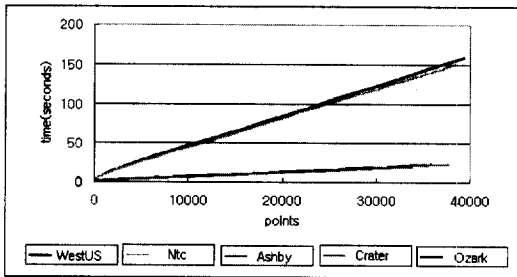
실험을 위한 시스템은 인텔 펜티엄2 400Mhz CPU (512Kbyte L2 캐쉬)와 128Mbyte RAM으로 구성된 PC로, 구역화 알고리즘의 수행 성능과 생성된 삼각망의 질적인 면을 평가하는데 주안점을 두었다. 이를 위해 본 논문의 알고리즘을  $O(m+n) \log m$ 의 ( $m$ :생성된 삼각망을 구성하고 있는 점의 수,  $n$ :입력된 지형데이터의 전체 정점 수) 계산 시간을 갖고 있는 Muchael Garland 와 Paul Heckbert's algorithm III(FFA)[4]에 적용시켜, 이와 구역화 알고리즘을 비교 실험하였으며, 이에 대한 결과를 아래에 그래프 및 3차원 지형 이미지로 나타냈다. 본 실험은 다섯 개의 다양한 지형 <표 1>에 대하여 실험을 하였으며, 이들은 모두 같은 형태의 결과를 보여 주었다. 따라서 Crater 지형에 대한 결과로 나머지들을 대표 하고자 한다.

본 논문의 경우, 지형을  $k \times l$ 의 격자 모양 구역으로 나눌 경우 하나의 max error를 갖는 점을 선택하기 위하여, 현재 갈망 삽입 알고리즘이 적용되고 있는 구역내의 점들만이 비교 검색되고, 각 구역으로부터 같은

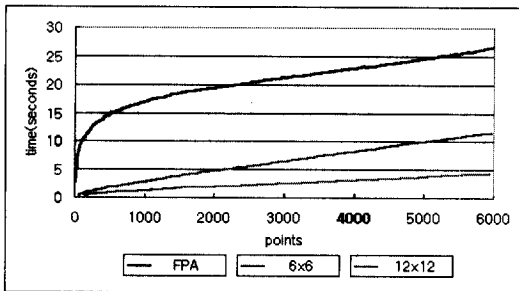
수의 점들이 삼각망에 추가된다고 할 때, 각 구역에 대한 평균 계산시간은  $O(\frac{m+n}{kl} \log \frac{m}{kl})$ 이 되고, 전체 지형에 대해 총  $k \times l$ 개의 구역이 있으므로 구역화 알고리즘의 평균 계산 시간은  $O((m+n) \log \frac{m}{kl})$ 이 된다.

〈표 1〉 테스트를 위한 5개의 지형정보

이름	크기	지형 위치
Ashby	346×452 (156,392 점)	Ashby Gap, Virginia
Crater	336×459 (154,224 점)	West half of Crater Lake, Oregon
Ozark	369×462 (170,478 점)	Desert around Mt. Tiefert, California
NTC	1024×1024 (1,048,576 점)	Ozark, Missouri
West US	1024×1024 (1,048,576 점)	Section of Idaho/Whoming border



(그림 8) 5개의 지형 데이터에 대한 12×12 구역화 방법을 이용한 수행 시간



(그림 9) Crater 지형에 대한 다양한 블록 수를 이용한 구역화 방법의 수행 시간

(그림 8)과 (그림 9)는 다양한 해상도의 삼각망을 생성해 내기 위해 소요된 수행 시간을 보여주고 있다. (그림 8)의 경우, 다섯 가지의 다양한 지형에 대하여 12×

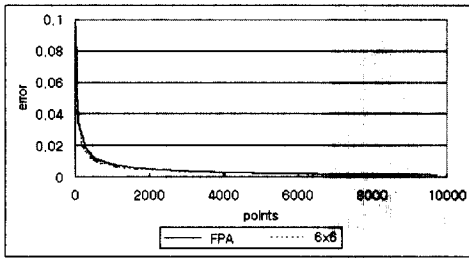
12의 구역으로 구역화 시켜 단순화를 행한 결과로, 입력된 데이터의 크기에 의해 수행시간이 좌우됨을 볼 수가 있다. (그림 9)의 경우는 하나의 지형에 대한 구역의 수를 달리한 경우를 보여주며, 지형을 많은 수의 구역으로 나누어줄수록 수행 속도가 감소함을 보여 준다. 여기서 불과 6×6 구역화를 수행했을 경우에도 네 배에 달하는 높은 수행 성능의 향상을 보여주는데, 이는 구역화 알고리즘이 추가될 정점을 선택하기 위해 탐색해야하는 데이터를 지형의 일부로 국한 시켜 데이터의 로컬리티가 증가되었고, 따라서 캐쉬의 효율이 향상된 점에서도 원인을 찾을 수 있다.

다음으로 생성되어진 삼각망의 질적인 측면을 평가하기 위하여, 생성되어진 삼각망에 대해 지형데이터의 모든 정점의 높이 차를 평균한 값을 (그림 10)과 (그림 11)에서 비교하였다.  $PN$ 은 입력된 점의 수이고,  $h_{max}$ 와  $h_{min}$ 은 입력된 정점들 중 가장 큰/작은 높이 값을 의미한다, 그리고  $RP_i$ 는 입력된 각 점의 높이 값을 의미하고,  $SP_i$ 는 입력된 점들을 생성되어진 삼각망상에  $z$ 축에 평행하게 투영시킨 위치의 높이 값들을 나타낸다고 했을 때, 평가 식을 아래 식 (2)와 같이 나타낼 수 있다.

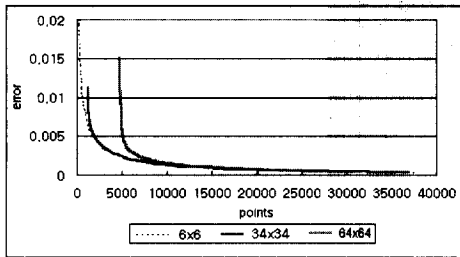
$$\frac{1}{PN} \sum_{i=0}^{PN} \left( \frac{|RP_i - SP_i|}{h_{max} - h_{min}} \right) \quad (2)$$

(그림 10)은 지형을 구역화 한 경우와 그렇지 않은 경우의 에러를 비교하고 있으며, 6×6으로 구역화를 행하였을 경우 구역화를 행하지 않은 경우와 비교하여 생성되어진 단순화 지형의 에러가 거의 같은 수치를 보이고 있다. (그림 11)의 경우 구역화를 좀더 세밀하게 할수록, 생성되어지는 단순화 지형의 에러 변화를 비교하여 보여준다. 구역의 수가 많아질수록 낮은 정밀도의 삼각망에 대해서는 높은 에러를 갖고 있음을 보여 주고 있다. 그러나 특정 정밀도 이상의 삼각망에서는 에러 차가 거의 없음을 보여 준다. 즉, 낮은 정밀도의 삼각망을 얻고자 할 경우에는 구역의 수를 적게 해주어 에러를 최소화 할 수 있으며, 높은 정밀도의 삼각망의 경우는 에러가 크게 증가하지 않는 범위 내에서 구역의 수를 증가시켜 높은 수행 성능을 얻을 수 있다.

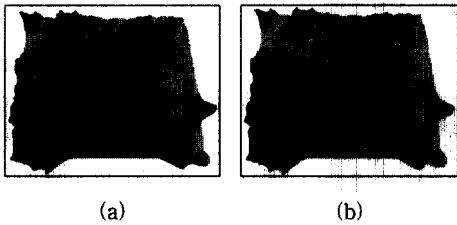
(그림 12)와 (그림 13)은 Crater DEM에 대하여 재 생성되어진 다양한 해상도의 image들이다. (그림 13)의 (b)는 단지 471개의 점만으로 이루어진 지형 image를 보이며, 그럼에도 계곡, 산등성이, 그리고 봉우리 등과 같은 지형의 특징을 충분히 가지고 있다.



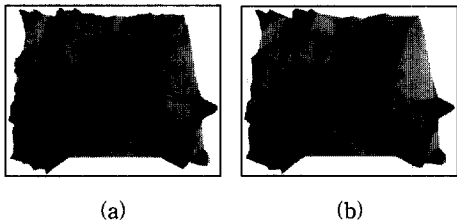
(그림 10) Crater DEM에 대한 FPA와 구역화 방법으로 생성된 삼각망의 평균 에러



(그림 11) Crater DEM에 대한 다양한 블록 수에 대한 구역화 방법으로 생성된 삼각망의 평균 에러



(그림 12) 154,224개의 포인트로 구성된 Crater lake DEM의 원본 이미지(a)와 52,800개의 포인트로 이루어진 48×48 구역화 단순 지형(b)



(그림 13) 4,711개의 포인트로 이루어진 24×24 구역화 단순지형(a)와 471개의 포인트로 이루어진 6×6 구역화 단순지형(b)

## 5. 결 론

단순화의 가장 중요한 요소 중의 하나는 가장 적은 수의 점들만을 이용해 원하는 정밀도의 삼각망을 생성하는데 있다. 이를 위해서는 입력된 지형 데이터로부터 그 지형을 가장 잘 표현할 수 있는 특징 점들만으로 이루어진 양질의 삼각망을 생성하는 것이 필요하다. 갈망 삼입 알고리즘은 이와 같은 조건을 만족시킬 수 있는 단순화 알고리즘 중의 하나이다. 그러나, 갈망 삼입 알고리즘의 경우 가장 중요한 특징 점을 찾기 위하여 하나의 포인트를 추가할 때마다 전체 입력된 정점에 대하여 연산을 행하므로 수행 속도에서의 문제점을 갖고 있다. 본 논문은 이러한 문제점을 해결하기 위하여 입력 데이터에 대하여 구역화를 행하여, 실험에서 보여주고 있는 것과 같은 수배에서 수십 배에 달하는 수행 성능의 향상과, 갈망 삼입 알고리즘에 못지 않은 질적인 성능을 보이고 있다.

이와 같은 뛰어난 수행성능의 향상은, 구역화 알고리즘이 구역화를 통하여 갈망 삼입 알고리즘의 수행 성능을 개선시키고 동시에 데이터의 로컬리티를 증가시켜 캐시의 효율성을 증대시킨 데서 찾을 수 있다.

마지막으로 제안하고 있는 알고리즘은, 그 특성상 각 구역별로 정점의 삼입이 이루어지고 있으므로 이들을 병렬 시스템에서 각 프로세서에 할당해 병렬 수행을 할 경우 높은 속도의 향상을 이루어 낼 수 있을 것이다. 따라서 구역화 알고리즘을 병렬시스템에 적용시키기 위한 연구가 진행되어야 할 것이다.

## 참 고 문 헌

- [1] L. D. Floriani, B. Falcidieno, and C. Pienovi, "A Delaunay-based method for surface approximation," *Eurographics '83*, pp.33-350, 1983.
- [2] L. D. Floriani, B. Falcidieno, G. Nagy, and C. Pienovi, "A hierarchical structure for surface approximation," *Computers and Graphics*, Vol.8, No.2, pp.183-193, 1984.
- [3] L. D. Floriani. "Surface representations based on triangular grids," *The Visual Computer*, Vol.3, pp. 27-50, 1987.
- [4] L. D. Floriani, "A pyramidal data structure for

triangle-based surface description," *IEEE Computer Graphics and Application*, Vol.9, No.2, pp.67-78, March 1989.

- [5] M. Garland and P. S. Heckbert, "Fast polygonal approximation of terrains and height fields," Technical report, CS Dept., Carnegie Mellon U., Sept. 1995. CMUCS-95-181, <http://www.cs.cmu.edu/~garland/scape>.
- [6] K. Anjyo, Y. Usami, T. Kurihara, "A Simple Method for Extracting the Natural Beauty of Hair," *SIGGRAPH '92*, Vol.26, No.2, pp.111-120, July 1992.
- [7] P. S. Heckbert, "Survey of texture mapping," *IEEE Transaction on Computer Graphics and Applications*, pp.56-57, Nov. 1986
- [8] M. Chow. "Optimized geometry compression for realtime rendering," *IEEE Visualization '97*, pp. 347-354, Oct. 1997.
- [9] N. Greene, "Environment mapping and other applications of world projections," *IEEE Computer Graphics and Applications*, Vol.6, No.11, pp.21-29, 1986.
- [10] S. L. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," *Computer Graphics and Image Preocessing*, Vol.4, No.2, pp.104-119, June 1975.
- [11] L. Williams, "Pyramidal parametrics," *Computer Graphics(SIGGRAPH'83 Proc.)*, Vol.17, No.3, pp. 1-11, July 1983.
- [12] D. Gomez and A. Guzman, "Digital model for three-dimensional surface representation," *GeoProcessing*, Vol.1, pp.53-70, 1979.
- [13] H. Samet, *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.



### 강 윤 식

e-mail : noori@mythos.yonsei.ac.kr  
 1993년 경희대학교(학사)  
 1999년 연세대학교 컴퓨터과학과  
 (석사)  
 관심분야 : 3D graphics algorithm.



### 박 우 찬

e-mail : chan@kurene.yonsei.ac.kr  
 1993년 연세대학교 컴퓨터과학과  
 (학사)  
 1995년 연세대학교 컴퓨터 과학과  
 (석사)  
 2000년 연세대학교 컴퓨터 과학과  
 (박사)

관심분야 : 3-D 그래픽 가속기, ASIC 디자인, Computer Arithmetic, High performance Computer Architecture



### 양 성 봉

e-mail : yang@mythos.yonsei.ac.kr  
 1981년 연세대학교 공과대학(학사)  
 1984년 University of Oklahoma  
 컴퓨터 과학(학부과정)  
 1986년 University of Oklahoma  
 컴퓨터과학(석사)

1992년 University of Oklahoma 컴퓨터과학(박사)  
 1992년~1992년 University of Oklahoma, Adjunct Assistant Professor  
 1993년~1994년 전주대학교 전자계산학과 전임강사  
 1994년~현재 연세대학교 공과대학 컴퓨터과학과 부교수  
 관심분야 : 병렬처리, Computational geometry, Spatial data processing, 3D graphics algorithm