

지형 데이터의 효율적 단순화

박상철^{1†} · 김정훈² · 정용호³

¹이주대학교 산업정보시스템공학부, ²국방과학연구소, ³이주대학교 산업공학과

Efficient Simplification of a Height Map

Sang Chul Park^{1†}, Junghoon Kim², and Yong Ho Chung³

¹Dept. of Industrial & Information Systems Engineering, Ajou Univ.

²Agency for Defense Development

³Dept. of Industrial Engineering, Ajou Univ.

Received 14 June 2011; received in revised from 15 February 2012; accepted 14 March 2012

ABSTRACT

Presented in the paper is a procedure to extract simplified triangular mesh from a height map (terrain data). The proposed algorithm works directly on a height map that extracts a simplified triangular mesh. For the simplification, the paper employs an iterative method of edge contractions. To determine an edge to be contracted, the contraction cost of an edge is evaluated through the QEM method. Normally, an edge contraction will remove two triangles sharing the edge. Although the edge contraction can be implemented easily on a triangular mesh, it is not viable to implement the operation on a height map due to the irregular topology. To handle the irregular topology during the simplification procedure, a new algorithm is introduced.

Key Words : Height map, Simplification, Terrain data

1. 서 론

지형 데이터는 일반적으로 Height Map으로 표현하게 된다^[1-3]. Height Map이란 Fig. 1과 같이 균일한 이차원 Grid 상에서 각 점에 해당하는 높이 값을 저장하고 있는 데이터 구조를 의미하며, CAD/CAM 및 컴퓨터 그래픽스 분야에서 다양한 용도로 사용되어 왔다. 특히 최근 컴퓨터 시뮬레이션 분야인 공중, 육상, 수중 국방 무기체계 시뮬레이션뿐만 아니라 컴퓨터 게임 분야에도 지형의 효율적인 가시화가 중요해 짐에 따라서 Height Map

의 활용도가 커지고 있는 상황이다. 특히 국방 시뮬레이션과 컴퓨터 게임의 경우에는 사용자의 입력에 즉각적으로 반응해야 하는 경우가 대부분이

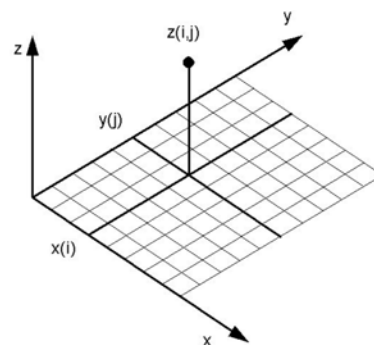


Fig. 1 Height Map

[†]Corresponding Author, scpark@ajou.ac.kr
©2012 Society of CAD/CAM Engineers

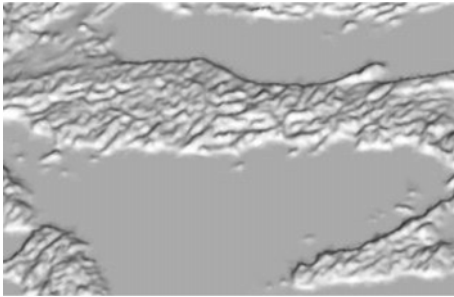


Fig. 2 Rendered terrain represented with a height map

므로 Height Map의 빠르면서도 정교한 가시화 기술에 대한 관심이 높아지고 있다.

Height Map의 가시화를 위해서는 하나의 Cell을 두 개의 삼각형으로 나누어 가시화 하는 것이 일반적이다. 이때 흔히 Cell을 구성하는 네 개의 점 중에 대각선 방향으로 가까운 점 두 개를 이어서 삼각형을 얻어내게 된다. 이런 식으로 개념적으로 볼 때 Height Map의 가시화는 매우 간단하지만, 실제 Height Map의 데이터 양이 많은 경우에는 좋은 성능의 컴퓨터와 그래픽 카드를 사용하더라도 가시화에 많은 시간이 걸리는 문제가 존재하고 있다.

예를 들어 Fig. 2와 같이 가로 세로의 길이가 10 km인 지형이 있다고 가정하자. 이러한 지형을 1 m 해상도로 높이 값을 저장하려면 10,000 * 10,000의 Height Map이 필요하게 된다. 10,000 * 10,000의 Height Map을 가시화 하기 위해서는 2억 개의 삼각형을 그려야 한다. 비록 고성능의 컴퓨터와 그래픽가속기의 도움을 받는다 해도 2억 개의 삼각형을 변하는 시점(Viewpoint)에 따라 즉각적으로 실시간에 가시화 한다는 것은 거의 불가능하다.

이러한 문제점을 해결하기 위해서는 Height Map을 단순화 시켜서 효율적인 가시화 기술의 개발이 필수적이다. Height Map의 효율적인 가시화를 위해서 기존에 많은 연구들이 있어왔다. 기존 연구들을 크게 두 가지 접근 방법으로 나누어 볼 수 있다. 첫 번째는 일반적인 3차원 삼각망을 단순화 시키는 방법^[4-14]이고, 두 번째는 Height Map이 2.5차원이라는 점을 활용하는 방법^[1-3]이다. Fig. 3은 일반적인 3차원 삼각망의 단순화를 보여주며, Fig. 4는 Height Map이 2.5차원이라는 점을 이용하여 삼각망을 단순화 하는 예를 보여준다.

일반적인 3차원 삼각망의 단순화를 위해 개발된 알고리즘은 매우 다양하다. 왜냐하면 삼각망은 전

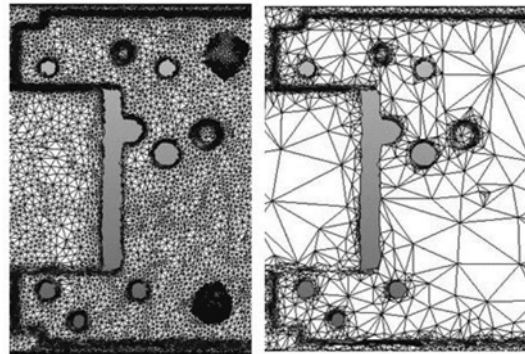


Fig. 3 Simplification of a general 3D triangular mesh

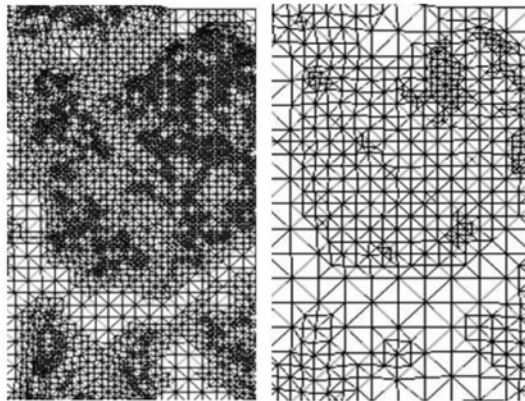


Fig. 4 Quad-tree based simplification of a Height Map

통적으로 컴퓨터 그래픽스, 캐드캠 분야 등에서 널리 사용되어 왔으며, 최근에는 게임과 영화에 이르기까지 다양한 분야에서 사용되고 있기 때문이다. 삼각망의 활용도가 높은 만큼 삼각망을 효율적으로 다루는 알고리즘을 개발하는 것은 매우 중요하고 또한 그 파급효과도 매우 크다. 비록 일반적인 3차원 삼각망을 단순화하는 다양한 알고리즘이 존재하고 이들을 Height Map에 적용하는 것도 가능하지만 비효율적인 요소가 발생한다. 우선 Height Map을 일반적인 3차원 삼각망으로 전환해야 하는데, 이는 개념상으로 매우 간단하지만, 실제적으로는 20배 가량의 더 많은 메모리 공간을 필요로 하는 작업이다. 왜냐하면 Height Map이 단순히 각 Grid 값에서 높이 값만 가지는 반면, 일반적인 3차원 삼각망은 Geometry 뿐 아니라 이웃하는 삼각형들과의 관계가 정의된 Topology 정보도 함께 유지해야 하기 때문이다. 단순화가 필요할 정도로 데이터 량이 큰 Height Map을 다룬다고 가정하면, 20배 더 많은 메모리는 현실적으로 큰 장

애물이 될 뿐 아니라 많은 시간 소모를 유발한다. 하지만 이러한 3차원 삼각망을 대상으로 하는 알고리즘들은 지형의 특징 형상을 효율적으로 보존하면서 단순화 할 수 있다는 장점이 있다.

Height Map 단순화에 대한 두 번째 접근 방법은 Height Map이 2.5차원 곡면임을 활용하는 것이 특징이다. 2.5차원 곡면을 쉽게 설명하면 위에서 내려다 보았을 때 모든 부분이 다 보이는 곡면이라 할 수 있다. 이러한 두 번째 접근방법에 속하는 방법들은 대부분 Quad-tree 기술에 기반하고 있다. Quad-tree란 하나의 node에 네 개의 child node가 존재할 수 있는 데이터 구조라고 할 수 있으며, 이런 속성은 Height Map을 효율적으로 표현할 수 있게 해준다. Height Map의 단순화를 위한 허용 가능한 에러 값을 주면, 그 에러 값을 넘지 않는 범위까지만 세분화해서 그려주는 방식이다. 단순하지만 효율적인 방식이라 할 수 있다. 하지만 지형의 특징형상 보다는 Quad-tree라는 데이터 구조에 의지하기 때문에 서로 다른 단계의 메쉬와 메쉬간의 연결이 매끄럽지 못해 발생하는 크랙(crack)현상^[14]을 발생시킬 수 있다. 또한 예를 들어 날카로운 사선 방향의 굴곡이 지형에 있을 경우 Quad-tree에서는 사선 방향의 계단처럼 왜곡되어 표현될 수 있다.

기존 연구를 종합해 보면, 3차원 곡면을 대상으로 하는 알고리즘은 지형의 특징형상을 효율적으로 표현하는 반면 너무 많은 메모리를 요구하여 대형 Height Map인 경우에는 적용이 곤란한 단점이 있다. 반면 Height Map이 2.5차원 곡면임을 감안하여 Quad-tree기법을 적용하는 알고리즘들은 간단하고 메모리 소모 측면에서 효율적이지만, 지형의 특징형상보다는 Quad-tree라는 데이터 구조에 의존하므로 지형의 특징형상을 잘 보존하면서 단순화 하는 용도로는 부족한 부분이 있다.

본 연구에서는 이러한 기존 연구들의 한계점을 극복하기 위하여 기존의 두 가지 방식의 장점들을 취하여 새로운 Height Map 단순화 방법론을 제안하고자 한다. 제안된 방법론은 기존 3차원 곡면용 알고리즘들처럼 Height Map의 특징형상을 최대한 고려하여 단순화 하면서도 기존 방법들에 비해서는 지극히 적은 양의 메모리만을 사용하게 된다. 다음 2장에서는 Height Map 단순화에 대한 본 논문의 접근방법을 설명하고, 3장에서는 구체적인 알고리즘을 설명한다. 4장에서는 제안된 알고리즘

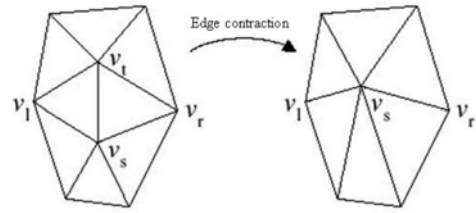


Fig. 5 Edge Contraction

에 대한 평가를 한다. 마지막으로 4장에서 결론을 맺도록 한다.

2. 접근 방법

기존에 삼각 망을 단순화하는 방법에는 여러 가지가 있지만 가장 일반적인 방법은 변(Edge)을 반복해서 제거하는 것이다. 변이 하나가 없어진다는 것은 해당 변을 정의하는 두 개의 꼭지점이 하나로 합쳐진다는 의미이며 Fig. 5가 보여주고 있다.

본 논문에서는 일반적인 3차원 삼각망을 대상으로 개발된 QEM (Quadratic Error Metric) 방법^[11]을 Height Map에 적용한다. 이때 원래의 QEM과는 달리 Height Map이 2.5차원 곡면이라는 특징을 최대한 이용함으로써 소모되는 메모리를 최소화 하는 방법을 개발한다. QEM 방법은 삼각망을 단순화 할 때 주어진 허용 에러를 넘지 않는 범위 내에서 변을 반복적으로 소거함(Iterative Edge Contraction)으로써 삼각망을 단순화 한다.

삼각망에서 하나의 꼭지점 v 는 여러 개의 삼각형에 의하여 공유된다. 이때 여러 개의 삼각형은 각각 하나의 무한 평면 상에 존재하게 되는데, QEM은 공유되는 꼭지점 v 가 이러한 무한 평면이 공통 교차하는 지점이라 본다. 만약 우리가 어떠한 이유로든 간에 이 꼭지점 v 의 위치를 바꾸어야 한다면 그로 인하여 발생하는 에러는 원래 그 꼭지점을 공유했던 평면들부터 얼마나 위치가 떨어지는 가로 평가할 수 있을 것이다. 이를 정량적으로 수식으로 나타내면 아래와 같다.

$$\Delta(v) = \Delta([v_x, v_y, v_z, 1]^T) = \sum_{p \in \text{planes}(v)} (p^T v)^2$$

위의 수식에서 평면 $p = [a, b, c, d]^T$ 는 $ax + by + cz + d = 0$, $(a^2 + b^2 + c^2 = 1)$ 라는 평면의 방정식으로 표현된다. 이를 고려하면 위의 수식에 표현된

에러는 아래와 같이 quadratic Form으로 표현될 수 있다.

$$\Delta(v) = \sum_{p \in \text{planes}(v)} (v^T p)(p^T v)^2 = v^T(Q)v$$

이런 방식으로 모든 꼭지점 v 에 대하여 위의 수식에 나타난 Q matrix를 간단히 구할 수 있다. 우선 모든 꼭지점에 대한 Q 를 계산하고, 이를 edge contraction의 여부를 계산할 때 에러의 정도를 평가하기 위해 사용한다. 임의의 edge $[V_i, V_j]$ 를 소거할지 말지를 계산하는 과정을 QEM 방식으로 해보면 다음과 같다. 우선 edge를 이루는 두 꼭지점을 어떠한 점으로 합쳐서 표현할 것인지를 결정하고, 그 후에 새로운 점으로 합쳐졌을 때의 에러를 계산하여 주어진 에러보다 작으면 해당 edge를 소거하는 방식을 취한다. 그 결과 해당 edge가 없어지고 기존의 edge를 이루는 두 꼭지점의 Q matrix를 합쳐서 새로운 통합 꼭지점의 Q matrix로 할당하게 된다.

하지만 이러한 방식은 본 연구에서 다루는 Height Map에 적용하기에는 적합하지 않다. 왜냐하면 Height Map이란 일정한 격자 위에 높이 값이 정의 되는 것이므로 자유롭게 점의 위치를 옮기는 것이 가능하지 않기 때문이다.

본 연구에서는 지형을 표현하는 Height Map의 이러한 특성을 반영하여 edge $[V_i, V_j]$ 가 만약 소거되어 하나의 새로운 꼭지점으로 표현된다면 그 새로운 꼭지점은 V_i 혹은 V_j 가 되도록 제약하였다. 즉 다시 말하면 단순화를 수행하는 과정에서 기존에 없는 새로운 꼭지점이 탄생하는 일은 없다는 것이다.

그러나, Height Map의 특성을 고려하여 edge contraction을 수행 시 새로운 점이 만들어지지 않는다는 제약을 둔다 하더라도 모든 문제가 해결되는 것은 아니다. 가장 큰 문제는 edge contraction이 진행되면서 삼각형들의 topology가 자유롭게 변하는데 이러한 과정을 표현할 방법이 용이하지 않다는 것이다. 물론 일반적인 3D 삼각망의 데이터 구조를 이용하면 충분히 변화하는 topology를 표현할 수 있지만 그럴 경우는 이미 서론에서 언급한 바와 같이 과도한 메모리를 요구하게 되는 어려움이 있다. 이러한 문제점에 대한 해결책이 다음 장에 설명되어 있다.

3. Height Map 위의 삼각망 Topology 유지

본 장에서는 이미 언급된 바와 같이 Height Map을 단순화하는 과정에서 topology의 변화가 일어나게 되는데 이를 일반적인 3D 삼각망 데이터 구조를 사용하지 않고 표현하는 방법론을 기술한다. 이를 위해서 가장 주의 깊게 숙지해야 하는 사실은 다음과 같다.

하나의 edge $[V_i, V_j]$ 를 소거하게 되면 새로운 점은 생기지 않으며, V_i 혹은 V_j 로 통합된다. 이때 통합하는 점을 'merger vertex'라고 부르고 통합 당하여 소거 당하는 점을 'mergee vertex'라 부르기로 하자. 만약 두 개의 vertex 사이에 이러한 관계가 성립되면 대응 관계는 다음과 같다. 임의의 vertex가 통합 당하여 소거 당했을 때는 반드시 1개의 merger vertex를 가지고 있으며 이를 포인팅할 수 있다. 즉 포인팅하고 있는 merger vertex가 없다면 해당 vertex는 소거 되지 않고 있으며 아직 유효한 점이라 볼 수 있다. 반면 소거 되었거나 되지 않았거나 모든 Height Map의 vertex들은 0개 혹은 복수개의 mergee vertex를 가질 수 있다.

이런 식으로 mergee vertex가 어떤 merger vertex에 합병되었는지를 기억함으로써 edge의 contraction

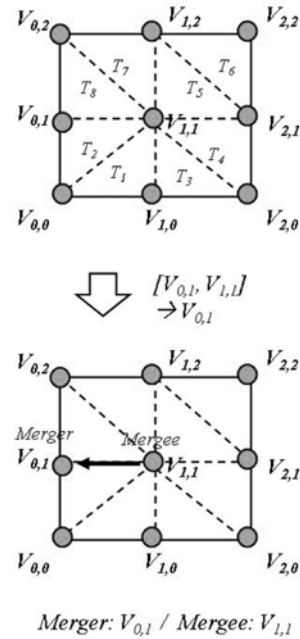


Fig. 6 Merger & Merges

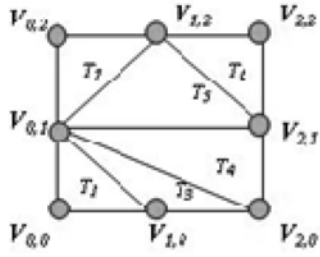


Fig. 7 Extraction of valid triangles

이 일어날 때 마다 발생하게 되는 topology의 변화를 유지하는 것이 가능하다.

Fig. 6에서 보여지는 예를 활용하여 본 연구에서 어떻게 topology의 변화를 유지하는지를 설명하도록 한다. Fig. 6에서 처음 주어지는 height map은 9개의 점으로 구성이 되며 8개의 삼각형으로 이루어진다. 여기에서 $V_{0,i}$ 과 $V_{1,i}$ 가 정의하는 변을 QEM 방법론에 따라 소거할 수 있으며 이 두 개의 꼭지점은 $V_{0,i}$ 로 통합되어 합쳐진다고 하자. 그러면 merger는 $V_{0,i}$ 가 되고, mergee는 $V_{1,i}$ 가 된다. 이러한 병합의 결과로 topology가 변하게 되는데, 이 정보는 간단하게 pointer 하나를 도입함으로써 해결할 수 있다. Fig. 6에서와 같이 항상 mergee가 자신을 병합한 merger를 기억하는 pointer 하나만 유지하면 된다. 하나의 vertex가 가리킬 수 있는 merger는 최대 1개이므로 이를 recursive하게 추적하면 병합이 여러 단계 되었어도 결국 최종 유효한 꼭지점을 찾아 갈 수 있다.

이런 식으로 QEM 방식으로 단순화를 수행하면서도 변화하는 topology는 mergee가 merger를 pointing 함으로써 정보를 유지하게 한다. 그리고 최종적으로 원하는 정밀도의 단순화가 완료되면 이러한 merger 및 mergee 관계를 이용하여 최종적으로 유효한 삼각형들을 추출한다. Fig. 6에서 보여지는 예제에 대해서 최종적으로 유효한 삼각형들을 추출하면 Fig. 7과 같다.

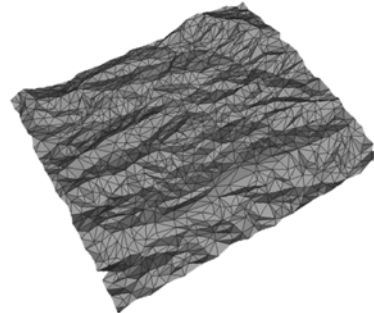
구체적인 알고리즘을 기술하기 전에 하나의 함수를 정의하고자 한다. 임의의 vertex V_i 에 대해서 $\text{find_merger}(V_i)$ 라는 함수는 V_i 가 병합되었을 경우 최종적으로 병합된 vertex를 추적해 주는 함수라고 가정한다. 예를 들어서 V_i 가 V_j 에 병합되었고 또 V_j 가 다시 V_k 에 병합이 되었다면 $\text{find_merger}(V_i)$ 는 V_k 를 return하게 된다. 만약 V_i 가 병합되지 않은 경우라면 $\text{find_merger}(V_i)$ 는 그냥 자신을 return



(a) Original height map (269,581 triangles)



(b) Simplified triangular mesh (20,958 triangles)



(c) Simplified triangular mesh (2,094 triangles)

Fig. 8 Simplification of a height map

하게 되는 함수이다. 이 함수를 이용하여 단순화 후에 유효한 삼각형을 추출하는 알고리즘은 아래와 같이 간단하게 기술될 수 있다. Fig. 8에서는 제안된 알고리즘을 이용해 지형데이터가 단순화되는 예를 보여준다. Fig. 9-10은 제안된 알고리즘을 이용하여 단순화된 더 큰 모델들의 예를 보여준다. Table 1은 제안된 알고리즘을 적용하기 전과 후를 비교한다.

단순화 후 삼각형 추출 알고리즘

// 입력: Mergee, Merger 관계가 포함된 Height Map

// 출력: Tri-S(유효 삼각형 집합)

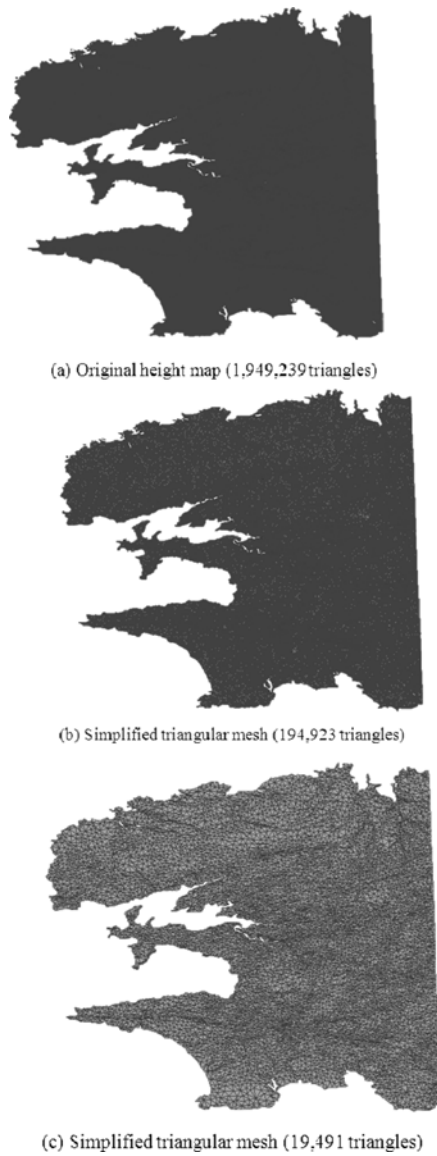


Fig. 9 Simplification of a height map

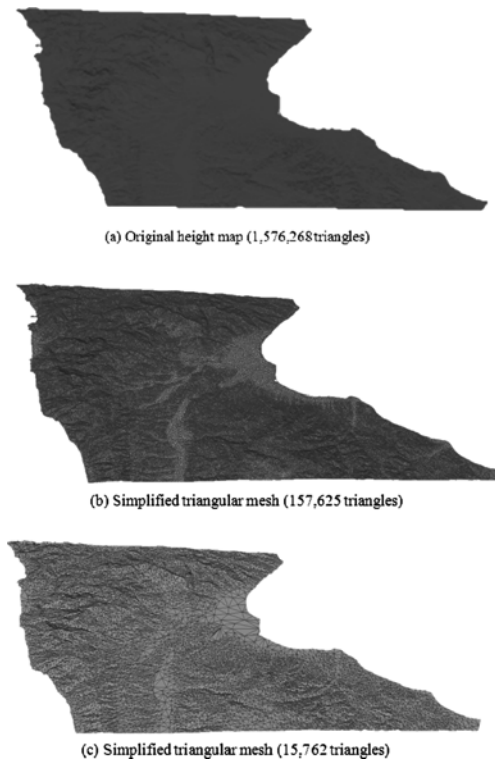


Fig. 10 Simplification of a height map

For each triangle T_i {

- 1) 삼각형을 구성하는 세 꼭지점(V_1, V_2, V_3)를 Height Map에서 가져온다;
- 2) $MV_1 = \text{find_merger}(V_1)$;
- 3) $MV_2 = \text{find_merger}(V_2)$;
- 4) $MV_3 = \text{find_merger}(V_3)$;
- 5) MV_1, MV_2, MV_3 중에 중복되는 위치가 존재하지 않으면 T_i 를 Tri-S에 추가한다;

}

Table 1. Comparison of required memory sizes

Caption	Contents	100%	10%	1%
Fig. 8	Number of triangles	209,581	20,958	2,094
	Memory size of a triangular mesh (KB)	10,223	1,024	101
	Memory size of a range image and a flag map (KB)	707	71	7
Fig. 9	Number of triangles	1,949,239	194,923	19,491
	Memory size of a triangular mesh (KB)	95,003	9,522	951
	Memory size of a range image and a flag map (KB)	6579	659	66
Fig. 10	Number of triangles	1,576,268	157,625	15,762
	Memory size of a triangular mesh (KB)	77,189	7,719	755
	Memory size of a range image and a flag map (KB)	5345	535	52

4. 평 가

QEM은 Quadric error metric이 표면 곡률과 직접적으로 연관되어 있기 때문에 수행 속도가 빠르며 높은 품질의 결과를 나타낸다. 이러한 이점에도 불구하고 QEM은 메모리 할당의 많은 양에 대한 한계점이 있다. QEM은 모든 점에 대해서 Q matrix(16 doubles 64 bytes)를 유지해야 한다. 그리고 몇몇 어플리케이션에서 점의 수는 매우 방대해진다. QEM의 메모리 할당은 Q matrix를 유지하기 위한 메모리와 geometry를 나타내기 위한 메모리로 나누어진다. Q matrices를 위한 메모리는 줄일 수 없기 때문에, geometry를 위한 메모리를 줄이는 것이 중요하다. 메모리 오버헤드를 감소시키기 위해서, 본 논문은 일반적인 triangular mesh 대신에 Flag map을 사용하였다. Flag map의 각 셀은 0.5 bytes를 필요로 한다. 1bit는 triangulation의 타입을 저장하기 위한 것이고, 3 bits는 merge vertex의 이동 방향(merger vertex로 합쳐지는 방향)을 저장하기 위한 것이다. Range image와 flag map을 모두 고려하였을 때, 제안되는 방법의 요구되는 총 메모리는 $4.5*n$ bytes ($=n*4 + n*0.5$)이며, 여기서 n 은 점의 수이다. Triangular mesh가 $65*n$ bytes ($=2n*24 + n*17$)이며, 그렇기 때문에 기존의 QEM에 비해 제안된 알고리즘이 메모리 사용량이 14.4배가 적다.

5. 결 론

본 논문에서는 지형데이터의 단순화를 위한 알고리즘을 제안하였다. 일반적으로 지형데이터는 height map으로 나타내어지므로 본 연구에서 제안된 알고리즘 역시 height map에 대해 적용하도록 설계되었다. 기존에도 지형데이터를 단순화 시키는 연구가 있지만, 대부분은 height map을 삼각망으로 바꾼 후에 삼각망 단순화 알고리즘을 적용하는 방식이다. 이러한 방식의 문제점은 대규모 지형데이터의 경우 삼각망으로 바꾸면 초기 데이터의 메모리량이 지나치게 커져서 효율성에 심각한 문제를 초래한다는 것이다.

이러한 문제점을 해결하기 위해서 본 연구에서는 height map을 삼각망으로 전환하지 않은 상태에서 QEM을 적용할 수 있는 방법론을 제안하였다. 이를 위해서는 QEM 적용 도중에 변이 소거되

면서 발생하는 topology 변화 정보를 유지하는 것이 필수적이다. 이러한 topology 변화 정보를 표현하기 위해서 merger와 mergee 개념을 도입하였고 항상 mergee가 해당 merger를 pointing하게 하였다. 그 결과 삼각망을 유지하지 않고서도 height map 기반으로 단순화를 수행할 수 있게 되었고, 단순화 이후 유효한 삼각형도 간단하게 추출할 수 있었다.

감사의 글

본 연구는 방위사업청(UD10009DD), 국방과학연구소(UD110006MD) 및 한국연구재단(2011-0026545)의 지원으로 수행되었습니다.

참고문헌

1. Hebert, D. and Kim, H., 1995, Image encoding with triangulation wavelets, *In Proc SPIE*, pp. 381-392.
2. Lindstrom, P. and Koller, D., 1996, Real-time Continuous Level of Detail Rendering of Height Fields, *In Proc SIGGRAPH*, pp. 109-118.
3. Sivan, R. and Samet, H., 1992, Algorithms for Constructing Quadtree Surface Maps, *In Proc 5th International Symposium on Spatial Data Handling*, pp.361-370.
4. Kim, H. S., Choi H. K. and Lee, K. H., 2006, Novel Evaluation of the Edge Cost for Mesh Simplification with Vertex Color Attribute, *Proceedings of the Society of CAD/CAM Engineers Conference*, pp. 764-770.
5. Kim, H. S., Choi, H. K. and Lee, K. H., 2007, Mesh Simplification for Preserving Geometric and Color Feature", *Proceedings of the Society of CAD/CAM Engineers Conference*, pp. 253-259.
6. Li, W. D., Cai, Y. L. and Lu, W. F., 2007, A 3D Simplification Algorithm for Distributed Visualization, *Computers in Industry*, 58(3), pp. 211-226.
7. Lee, K. H., Woo, H. and Suk, T., 2001, Data Reduction Methods for Reverse Engineering, *The International journal of advanced manufacturing technology*, 17(10), pp. 735-743.
8. Hoppe, H., 1996, Progressive Meshes, *In Proc SIGGRAPH*, pp. 99-108.
9. Garland, M. and Heckbert, P. S., 1997, Surface Simplification Using Quadric Error Metrics, *In Proc SIGGRAPH*, pp. 209-216.

10. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., 1993, Mesh Optimization, *In Proc SIGGRAPH*, pp. 19-26.
11. Schroeder, W. J., Zarge, J. A. and Lorensen, W. E., 1992, Decimation of Triangle Meshes, *Computer Graphics*, 26(3), pp. 65-70.
12. Soucy, M. and Laurendeau D., 1996, "Multiresolution Surface Modeling Based on Hierarchical Triangulation, *Computer Vision and Image Understanding*, 63(1), pp. 1-14.
13. Reinhard, K., 1998, Multiresolution Representations for Surfaces Meshes Based on the Vertex Decimation Method, *Computer & Graphics*, 22(1), pp. 13-26.
14. Paul S. Heckbert and Michael Garland, 1999, Optimal Triangulation and Quadric-Based Surface Simplification, *Journal of Computational Geometry: Theory and Applications*, 14, pp. 49-65.
15. Jian, W., 2010, A New Quadtree-based Terrain LOD Algorithm, *Journal of Software*, 5(7), pp. 769-776.



박 상 철

1994년 한국과학기술원 산업공학과 학사
 1996년 한국과학기술원 산업공학과 석사
 2000년 한국과학기술원 산업공학과 박사
 2004년~현재 아주대학교 산업정보시스템공학부 부교수
 관심분야: Digital Manufacturing System, CAD/CAM, CAPP, Manufacturing System Modeling & Simulation



김 정 훈

1998년 서울대학교 해양학과 학사
 2001년 서울대학교 지구환경과학부 석사
 2001년~현재 국방과학연구소 제6기술연구본부
 관심분야: Defense Modeling & Simulation, Weapon Systems Effectiveness Analysis, Battle Experimentation



정 용 호

2011년 아주대학교 산업정보시스템공학부 학사
 2011년~현재 아주대학교 산업공학과 석사과정
 관심분야: Modeling & Simulation, Defense Simulation, Synthetic Environment, Dynamic Terrain