(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년05월20일
(11) 등록번호 10-0959349
(24) 등록일자 2010년05월14일

(51) Int. Cl.
G06T 17/50 (2006.01) *G06T 17/00* (2006.01)
(21) 출원번호 10-2009-0054996
(22) 출원일자 2009년06월19일
심사청구일자 2009년06월19일
(56) 선행기술조사문헌
KR1020060128803 A
KR1020060100157 A

(73) 특허권자
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
(72) 발명자
신병석
경기 안양시 동안구 갈산동 샘마을한양아파트 114동 402
이은석
서울특별시 서초구 반포1동 서초한양아파트 3동 1207호
강동수
인천 남구 용현동 169-1 세운팔레스 오피스텔 201호
(74) 대리인
특허법인태동

전체 청구항 수 : 총 5 항 심사관 : 문남두

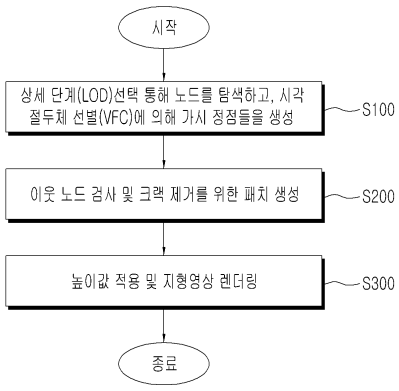
(54) 그래픽스 처리 유닛을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법

(57) 요 약

본 발명은 그래픽스 처리 유닛(Graphics Processing Unit; GPU)을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에 관한 것으로서, 보다 구체적으로는 그래픽스 처리 유닛(Graphics Processing Unit, 이하 GPU)의 렌더링 파이프라인을 이용한 사진트리 기반의 지형 렌더링 방법에 있어서, (1) 상세단계(LOD) 선택을 통해 각 블록에 포함된 지형을 구성할 노드들을 탐색하고, 상기 탐색된 노드들의 각 정점(Vertex)들 중 시각 절두체 선별(Hierarchical View Frustum Culling, 이하 VFC)에 의해 시야의 안쪽에 있는 정점들을 사진트리로 증식시키며 복수의 가시 정점(visible vertices)들을 생성하는 단계; (2) 상기 생성된 복수의 가시 정점들로부터 삼각 메시를 생성하고, 상기 생성된 메시들 간의 이웃 노드 검사를 통해 크랙 제거를 위한 패치를 생성하여 상기 메시를 패치된 블록 메시로 변환하는 단계; 및 (3) 상기 패치된 블록들로 구성된 삼각 메시에 고도 필드로부터 높이 값을 적용하고, 노말 맵으로 쉐이딩 하여 지형 영상으로 렌더링하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

본 발명에서 제안하고 있는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에 따르면, 노드 탐색 과정에서 시각 절두체 선별을 함께 진행함으로써, 너무 많은 정점들의 생성으로 인한 속도 저하를 방지하여 성능을 향상시킬 수 있다. 또한 정점의 좌표값들 중 사용하지 않는 좌표값들을 버퍼로 이용함으로써, GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화할 수 있다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	RTI05-03-01
부처명	산업자원부
연구사업명	지자체 주도 연구개발지원사업
연구과제명	실감형 3D 입체 그래픽스 처리 기술
주관기관	(재)인천광역시정보산업진흥원
연구기간	2008년 05월 01일 ~ 2009년 04월 30일

특허청구의 범위

청구항 1

그래픽스 처리 유닛(Graphics Processing Unit, 이하 GPU)을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법으로서,

(1) 상세단계(LOD) 선택을 통해 각 블록에 포함된 지형을 구성할 노드들을 탐색하고, 상기 탐색된 노드들의 각 정점(Vertex)들 중 시각 절두체 선별(Hierarchical View Frustum Culling, 이하 VFC)에 의해 시야의 안쪽에 있는 정점들을 사진트리로 증식시키며 복수의 가시 정점(visible vertices)들을 생성하는 단계;

(2) 상기 생성된 복수의 가시 정점들로부터 삼각 메시를 생성하고, 상기 생성된 메시들 간의 이웃 노드 검사를 통해 크랙 제거를 위한 패치를 생성하여 상기 메시를 패치된 블록 메시로 변환하는 단계; 및

(3) 상기 패치된 블록들로 구성된 삼각 메시에 고도 필드로부터 높이 값을 적용하고, 노말 맵으로 쉐이딩 하여 지형 영상으로 렌더링 하는 단계

를 포함하는 것을 특징으로 하는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법.

청구항 2

제1항에 있어서, 상기 GPU의 렌더링 파이프라인은, 기하 음영 단계(Geometry Shader stage, 이하 GS) 및 스트림 출력 단계(Stream Output stage, 이하 SO)를 포함하고,

복수의 가시 정점들을 생성하는 상기 단계 (1)은,

(a) 상기 렌더링 파이프라인의 초기 값으로 상기 사진트리의 루트 노드에 해당하는 단일 정점을 입력받는 단계;

(b) 상기 기하 음영 단계(GS)에서 상기 입력받은 단일 정점이 각각의 노드에 대해 상위 레벨로의 탐색 여부를 판별하는 플래그 값(f)에 따른 탐색 조건을 만족하는지 상세단계(LOD) 선택을 하는 단계;

(c) 상기 상세단계(LOD) 선택 결과, 상기 단일 정점이 탐색 조건을 만족하지 않으면 상기 단일 정점을 기준으로 분할되는 4개의 서브블록의 중점에 자식 노드들의 정점을 생성하되, 상기 각각의 자식 노드의 경계구(p)가 시각 절두체의 바깥에 위치하면 해당 자식 노드에서는 정점을 생성하지 않으며, 4개의 자식 노드들이 모두 시각 절두체의 바깥에 위치할 경우 상기 자식 노드들에 대한 정점들뿐만 아니라 상기 단일 정점 자신도 삭제하는 시각 절두체 선별(VFC) 연산을 수행하는 단계;

(d) 상기 단계 (c)의 결과를 스트림 출력 단계(SO)를 통해 1차원 버퍼에 저장하는 단계; 및

(e) 각 블록들이 위치할 장소에 각 블록의 레벨을 지정한 정점들을 생성하기 위하여 상기 저장된 결과를 상기 렌더링 파이프라인으로 제공하여 각 자식 노드들의 정점마다 생성해야 하는 사진트리의 깊이만큼 상기 단계 (a) 내지 단계 (d)의 과정을 반복 수행하는 단계를 포함하는 것을 특징으로 하는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법.

청구항 3

제2항에 있어서, 상기 플래그 값(f)은 아래 수학식을 만족하는 것을 특징으로 하는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법.

$$f = \begin{cases} 0 & \text{if } |v,p| < \frac{\lambda}{\ell \times e} \\ 1 & \text{otherwise} \end{cases}$$

여기서, v는 view point, p는 경계구(boundary sphere of a region), λ는 미리 지정된 임계값, ℓ은 블록의 레벨을 나타내며, e는 표면 거칠기 값을 각각 나타낸다.

청구항 4

제1항에 있어서, 상기 GPU의 렌더링 파이프라인은, 기하 음영 단계(Geometry Shader stage, 이하 GS) 및 스트림 출력 단계(Stream Output stage, 이하 SO)를 포함하고,

상기 단계 (3)은,

(f) 상기 생성된 복수의 가시 정점들이 나타내는 블록의 상하좌우에 위치한 동일한 레벨의 블록을 갖는 4개의 이웃 정점들의 위치를 계산하는 단계;

(g) 상기 위치가 계산된 이웃 정점들에 대하여 아래 수학식을 만족하는 플래그 값(f)을 이용하여 크랙 발생 여부를 판단하고, 상기 판단 결과에 따라 패치를 생성하여 상기 가시 정점들을 패치된 블록 메시로 변환하는 단계; 및

(h) 상기 변환된 블록 메시를 상기 SO를 통해 다시 상기 렌더링 파이프라인으로 피드백 시키는 단계를 포함하는 것을 특징으로 하는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법.

$$f = \begin{cases} 0 & \text{if } |v,p| < \frac{\lambda}{\ell \times e} \\ 1 & \text{otherwise} \end{cases}$$

여기서, v는 view point, p는 경계구(boundary sphere of a region), λ는 미리 지정된 임계값, ℓ은 블록의 레벨을 나타내며, e는 표면거칠기값을 각각 나타낸다.

청구항 5

제1항에 있어서, 상기 정점들은 (x, y, z, w)의 4개의 좌표값을 가지고, 상기 4개의 좌표값 중, 사용하지 않는 좌표값을 버퍼로 이용하며,

상기 삼각 메시를 구성하는 세 정점에 대하여,

상기 삼각 메시에서 삼각형의 양 빗변을 공유하는 정점의 (x, y, z, w) 좌표값 중, z 좌표값에 T-정점(T)과 이웃 정점(N)과의 거리를, w 좌표값에 패치 생성 여부를 판별하는 플래그 값(f)을 저장하고,

빗변에 이웃하는 두 정점 중 하나의 정점의 (x, y, z, w) 좌표값 중, z 좌표값에 이웃 정점(N)의 방향을 저장하는 것을 특징으로 하는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법.

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 지형 렌더링 방법을 가속화하는 방법에 관한 것으로서, 보다 구체적으로는 그래픽스 처리 유닛(Graphics Processing Unit; GPU)을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에 관한 것이다.

배 경 기 술

[0002] 지형 렌더링은 컴퓨터 게임이나 비행 시뮬레이션 등에서 사실적인 야외 장면을 표현하는데 많이 사용된다. 지형 렌더링에서는 거대한 입력 데이터를 다루기 때문에, 이와 동일한 해상도의 메시를 생성하면 대용량의 비디오 메모리(video memory)를 탑재한 최신 그래픽스 하드웨어에서조차도 처리하기 힘들다. 따라서 결과 영상의 화질을 적절히 유지하는 메시 간략화 기법들을 적용해야 한다.

[0003] 사진트리(quadtree)는 2차원 공간을 효율적으로 표현하는 계층 자료구조이다. 사진트리는 지형 렌더링에서 많이 사용되는 메시 간략화 기법인 연속 상세단계(Continuous level of detail) 기법과 시각 절두체 선별 기법에 효과적이다. 사진트리는 고화질의 지형 영상을 실시간으로 생성할 수 있게 하지만, 그래픽스 하드웨어의 렌더링 파이프라인에서는 vertex array나 텍스처 등 순차데이터(sequential data)만 처리가 가능하므로, 사진트리와 같은 계층 자료구조는 사용할 수 없다. 따라서 사진트리를 이용하는 방법들은 CPU에서 수행되고 이로 인해 화질이 떨어지거나 속도가 저하되는 문제가 생긴다. 따라서 지형 렌더링에 필요한 사진트리를 그래픽스 하드웨어의 렌더링 파이프라인에서 사용 가능하도록 함으로써 GPU만을 이용하여 사진트리 기반의 지형 모델을 렌더링하는 기법을 고안할 필요성이 있다.

[0004] 그러나 GPU만을 이용하여 사진트리 기반의 지형 모델을 렌더링하는 방법은 불필요한 연산과 지나치게 많은 정점들의 생성으로 인하여 속도 저하가 발생할 수 있으므로, GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법을 고안할 필요성 또한 있다.

발명의 내용

해결 하고자하는 과제

[0005] 본 발명은 기존에 제안된 방법들의 상기와 같은 문제점들을 해결하기 위해 제안된 것으로서, 노드 탐색 과정에서 시각 절두체 선별을 함께 진행함으로써, 또한 정점의 좌표 값들 중 사용하지 않는 좌표 값들을 버퍼로 이용함으로써, GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법을 제공하는 것을 그 목적으로 한다.

과제 해결수단

[0006] 상기한 목적을 달성하기 위한 본 발명의 특징에 따른, 그래픽스 처리 유닛(Graphics Processing Unit, 이하 GPU)을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법은,

[0007] (1) 상세단계(LOD) 선택을 통해 각 블록에 포함된 지형을 구성할 노드들을 탐색하고, 상기 탐색된 노드들의 각 정점(Vertex)들 중 시각 절두체 선별(Hierarchical View Frustum Culling, 이하 VFC)에 의해 시야의 안쪽에 있는 정점들을 사진트리로 증식시키며 복수의 가시 정점(visible vertices)들을 생성하는 단계;

[0008] (2) 상기 생성된 복수의 가시 정점들로부터 삼각 메시를 생성하고, 상기 생성된 메시들 간의 이웃 노드 검사를 통해 크랙 제거를 위한 패치를 생성하여 상기 메시를 패치된 블록 메시로 변환하는 단계; 및

[0009] (3) 상기 패치된 블록들로 구성된 삼각 메시에 고도 필드로부터 높이 값을 적용하고, 노말 맵으로 쉐이딩 하여 지형 영상으로 렌더링하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

[0010] 바람직하게는, 상기 렌더링 파이프라인은, 기하 음영 단계(Geometry Shader stage, 이하 GS) 및 스트림 출력 단계(Stream Output stage, 이하 SO)를 포함하고,

[0011] 복수의 가시 정점들을 생성하는 상기 단계 (1)은,

[0012] (a) 상기 렌더링 파이프라인의 초기 값으로 상기 사진트리의 루트 노드에 해당하는 단일 정점을 입력받는 단계;

[0013] (b) 상기 기하 음영 단계(GS)에서 상기 입력받은 단일 정점이 각각의 노드에 대해 상위 레벨로의 탐색 여부를 판별하는 플래그 값(f)에 따른 탐색 조건을 만족하는지 상세단계(LOD) 선택을 하는 단계;

[0014] (c) 상기 상세단계(LOD) 선택 결과, 상기 단일 정점이 탐색 조건을 만족하지 않으면 상기 단일 정점을 기준으로 분할되는 4개의 서브블록의 중점에 자식 노드들의 정점을 생성하되, 상기 각각의 자식 노드의 경계구(p)가 시각 절두체의 바깥에 위치하면 해당 자식 노드에서는 정점을 생성하지 않으며, 4개의 자식 노드들이 모두 시각 절두체의 바깥에 위치할 경우 상기 자식 노드들에 대한 정점들뿐만 아니라 상기 단일 정점 자신도 삭제하는 시각 절두체 선별(VFC) 연산을 수행하는 단계;

[0015] (d) 상기 단계 (c)의 결과를 스트림 출력 단계(SO)를 통해 1차원 버퍼에 저장하는 단계; 및

[0016] (e) 각 블록들이 위치할 장소에 각 블록의 레벨을 지정한 정점들을 생성하기 위하여 상기 저장된 결과를 상기 렌더링 파이프라인으로 제공하여 각 자식 노드들의 정점마다 생성해야 하는 사진트리의 깊이만큼 상기 단계 (a) 내지 단계 (d)의 과정을 반복 수행하는 단계를 포함할 수 있다.

[0017] 더욱 바람직하게는, 상기 플래그 값(f)은 다음의 수학식을 만족할 수 있다.

[0018] $$f = \begin{cases} 0 & \text{if } |v,p| < \frac{\lambda}{\ell \times e} \\ 1 & \text{otherwise} \end{cases}$$

[0019] 여기서, v는 view point, p는 경계구(boundary sphere of a region), λ는 미리 지정된 임계값, ℓ은 블록의 레벨을 나타내며, e는 표면 거칠기 값을 각각 나타낸다.

[0020] 더욱 바람직하게는, 상기 정점들은 (x, y, z, w)의 4개의 좌표 값을 가지고, 상기 4개의 좌표 값들 중, 사용하지 않는 좌표 값을 버퍼로 이용하며,

[0021] 상기 각각의 정점에 대하여 수행하는 단계 (b)의 상세단계 선택과 단계 (c)의 시각 절두체 선별(VFC)의 수행에 있어서,

[0022] 상기 정점들의 좌표값들 중,

[0023] 상기 z 좌표값에는 상기 정점의 위치를 이용하여 계산된 경계구(p)의 반지름값을 저장하고,

[0024] 상기 w 좌표값에는 상기 상세단계 선택 시에 상기 정점의 레벨보다 상위 레벨로의 탐색 여부를 판별하는 플래그 값(f)을 저장할 수 있다.

[0025] 더더욱 바람직하게는, 상기 정점들은 (x, y, z, w)의 4개의 좌표값을 가지고, 상기 4개의 좌표값 중, 사용하지 않는 좌표값을 버퍼로 이용하며,

[0026] 상기 삼각 메시를 구성하는 세 정점에 대하여,

[0027] 상기 삼각 메시에서 삼각형의 양 빗변을 공유하는 정점의 (x, y, z, w) 좌표값 중, z 좌표값에 T-정점(T)과 이웃 정점(N)과의 거리를, w 좌표값에 패치 생성 여부를 판별하는 플래그 값(f)을 저장하고,

[0028] 빗변에 이웃하는 두 정점 중 하나의 정점의 (x, y, z, w) 좌표값 중, z 좌표값에 이웃 정점(N)의 방향을 저장할 수 있다.

효 과

[0029] 본 발명에서 제안하고 있는 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에 따르면, 노드 탐색 과정에서 시각 절두체 선별을 함께 진행함으로써, 너무 많은 정점들의 생성으로 인한 속도 저하를 방지하여 성능을 향상시킬 수 있다. 또한 정점의 좌표값들 중 사용하지 않는 좌표값들을 버퍼로 이용함으로써, GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화할 수 있다.

발명의 실시를 위한 구체적인 내용

[0030] 이하에서는 첨부된 도면들을 참조하여, 본 발명에 따른 실시예에 대하여 상세하게 설명하기로 한다.

[0031] 도 1은 본 발명의 일실시예에 따른 그래픽스 처리 유닛(Graphics Processing Unit, 이하 GPU)을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법의 흐름도이다. 도 1에 도시된 바와 같이, 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법은, 그래픽스 처리 유닛(Graphics Processing Unit, 이하 GPU)의 렌더링 파이프라인을 이용한 사진트리 기반의 지형 렌더링 방법에 있어서, 상세단계(LOD) 선택을 통해 각 블록에 포함된 지형을 구성할 노드들을 탐색하고, 탐색된 노드들의 각 정점(Vertex)들 중 시각 절두체 선별(Hierarchical View Frustum Culling, 이하 VFC)에 의해 시야의 안쪽에 있는 정점들을 사진트리로 증식시키며 복수의 가시 정점(visible vertices)들을 생성하는 단계(S100), 생성된 복수의 가시 정점들로부터 삼각 메시를 생성하고, 생성된 메시들 간의 이웃 노드 검사를 통해 크랙 제거를 위한 패치를 생성하여 메시를 패치된 블록 메시로 변환하는 단계(S200), 및 패치된 블록들로 구성된 삼각 메시에 고도 필드로부터 높이 값을 적용하고, 노말 맵으로 쉐이딩하여 지형 영상으로 렌더링 하는 단계(S300)를 포함한다.

[0032] 본 발명은 단계 S100에서의 가시 정점 생성에 앞서, 지형 영상의 렌더링을 위한 전처리 과정을 필요로 하므로 이에 대해 먼저 설명한다. 전처리 과정에서는 상세단계(LOD) 조절을 위한 표면 거칠기 값을 포함하는 사진트리 텍스처(quadtree texture)와 쉐이딩(shading)을 위한 노말 맵(Normal Map)을 생성한다. 사진트리는 각 노드(node)가 4개의 자식노드(child node)를 갖는 계층 자료 구조로서, 2차원 공간을 4개의 4분면으로 재귀적으로 분할하여 공간을 효율적으로 표현한다. 사진 트리는 각 노드마다 부모노드(parent node), 자식노드(child node), 그리고 이웃 노드(neighbor node)의 포인터(pointer) 정보들, 분할된 블록의 레벨값, 고도값(height value), 및 표면거칠기 값(σ)을 포함하는 표면거칠기 정보들을 갖는다. 사진트리 텍스처는 이러한 정보들을 렌더링 파이프 라인에서 사용하기 위해 사진트리를 2차원 텍스처 형태로 변환한 것을 말하며, 이러한 사진트리 텍스처의 구성은 도 2에 잘 나타나 있다.

[0033] 도 2는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 생성되는 사진트리 텍스처를 나타낸 도면이다. 도 2에 도시된 본 발명의 일실시예에 따른 사진트리 텍스처는 전처리 단계에서 생성되며 입력된 고도필드(height field)와 동일한 해상도를 갖는다. 이 텍스처는 도 2와 같이 중앙에 루트 노드(root node)가 저장되고, 이 노드를 중심으로 하여 4개로 분할된 정사각형 서브 블록(sub block)들의 중앙에 자식 노드들을 저장한다. 이것은 트리의 최대 레벨값(L_{max})만큼 재귀적으로 반복한다. 본 발명에서는 표면거칠기 값(σ)을 가진 사진트리 텍스처를 생성한다. 표면 거칠기 값(σ)의 계산은 도 3 및 도 4를 통해 설명한다.

[0034] 도 3은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 블록에 포함된 정점 및 기하오차 값을 나타낸 도면이고, 도 4는 블록의 크기에 따라 가중치가 부여된 모습을 나타낸 도면이다. 도 3에 도시된 바와 같이, 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서 표면거칠기값(σ)은 상위 단계의 정점(vertex)(vn)과 하위 단계의 에지(edge)(vn')에 대한 고도(height) 차이인 기하 오차 값(geometry error value)(δ)을 사용하여 계산된다. 이 기하 오차 값(δ)이 같은 값일 경우 큰 블록보다 작은 블록에서 상대적으로 더 차이가 커보인다. 따라서 이러한 문제를 해결하기 위하여, 도 4와 같이 블록의 레벨에 따라 가중치(w) 값을 주며, 가중치 값은 상위 레벨일수록 큰 값을 갖는다. 가중치(w) 값이 주어진 기하 오차 값(δ)은 다음의 수학식 1을 통해 구할 수 있다.

수학식 1

[0035] $$\delta_k^l = \upsilon n_k^l - \upsilon n'^l_k * w^l$$

[0036] 여기서, vn은 사진트리에서 각 블록마다의 상위 단계 정점의 고도 값을 나타내고, vn'은 하위 단계의 에지의 중점에 대한 고도 값을 나타내며, w는 블록의 레벨에 따른 가중치를 나타낸다. 또한, l은 블록의 레벨, k는 사진트리의 레벨 단계를 나타낸다.

[0037] 이러한 과정을 거쳐 각 블록마다 4 개의 기하 오차 값들이 계산된다. 각 노드에 저장될 표면거칠기값은 노드 자신과 하위 노드들의 최대 기하 오차 값이다. 하위 레벨의 값들을 포함하는 이유는 상세단계(Level Of Detail, 이하 LOD)를 적용할 경우에 특정 상세단계의 오차 값(error value)이 더 큼에도 불구하고, 상위 레벨에서 탐색을 중단할 수 있기 때문이다. 수학식 1을 통해 구해진 기하 오차 값을 이용하여 표면 거칠기 값(σ)을 표시하면 다음의 수학식 2와 같이 표현할 수 있다.

수학식 2

[0038] $$\sigma^l = \begin{cases} \max(\delta_k^l) & \text{if } l = L_{max} \\ \max(\max(\delta_k^l), \sigma^{l+1}) & \text{if } l < L_{max} \end{cases}$$
$$\text{where } k \in [0,3]$$

[0039] 여기서, δ는 사진트리에서 상위 단계의 정점(vn)와 하위 단계의 에지의 중점(vn')에 대한 고도 차이를 나타내

는 기하 오차 값이고, l은 블록의 레벨, k는 사진트리의 레벨 단계를 나타낸다.

[0040] 이렇게 전처리 단계에서는 지형 렌더링 시에 지형을 더욱 상세하게 표현하기 위해 표면 거칠기 값이 포함된 사진 트리 텍스처와 렌더링의 대상이 되는 지형을 표현한 일반적인 지도(normal map)를 생성한다.

[0041] 단계 S100은, 상세단계(LOD) 선택을 통해 각 블록에 포함된 지형을 구성할 노드들을 탐색하고, 탐색된 노드들의 각 정점(Vertex)들 중 시각 절두체 선별(Hierarchical View Frustum Culling, 이하 VFC)에 의해 시야의 안쪽에 있는 정점들을 사진트리로 증식시키며 복수의 가시 정점(visible vertices)들을 생성한다. 본 발명에서는 노드 탐색 과정에서 시각 절두체 선별을 함께 진행한다. 일반적으로 사진트리는 2차원 공간을 효율적으로 표현하는 계층 자료 구조로서, 지형 렌더링에서 많이 사용되는 메시 간략화 기법인 연속 상세 단계(Continuous Level Of Detail, 이하 CLOD)와 시각 절두체 선별 기법에 효과적이다. 여기서 상세단계(LOD)란 어떤 물체가 작거나 렌더링 이미지에 별로 기여하는 것이 없을 경우에 좀 더 간략하게 표현된 것을 사용하자는 것으로서 먼 거리의 모델은 덜 정밀하게 표현되어도 시각적으로 큰 차이 없이도 그래픽 파이프라인으로 넘어가는 다각형(polygon)의 개수를 줄여 렌더링의 성능을 향상시키는 기법을 말한다. 그 중 연속 상세 단계(CLOD)는 지형 데이터를 시각화하는 기법 중의 하나로서 시점의 위치와 거리에 따라 동적으로 모델의 상세단계를 조절하게 된다. 고도 필드(height field)로 표현되는 지형 데이터의 경우 실행 시간에 삼각형 집합을 동적으로 생성하는 것이 가능하기 때문에 거리가 변함에 따라 지형의 정점들을 나누거나 합치는 방법으로 상세단계를 동적으로 조절할 수 있다. 사진트리 기반의 연속 상세 단계(CLOD)를 이용한 지형의 표현은 지형과 시점 사이의 거리와 관측방향에 따라 능동적으로 지형의 상세단계를 결정하여 실시간 지형 영상을 얻을 수 있다는 장점이 있다. 본 발명에서 제안하는 연속 상세 단계(CLOD) 방법에는 두 가지 기준이 적용되는데, 그 첫 번째는 관측자와의 거리이고, 두 번째는 표면 거칠기이다. 표현 거칠기를 고려한 방법은 도 5와 도 6을 통해 설명한다.

[0042] 도 5는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 시점 거리를 갖는 상세단계(LOD)의 와이어프레임의 이미지를 나타낸 도면이고, 도 6은 표면 거칠기 값이 적용된 연속 상세 단계(CLOD)에서의 이미지를 나타낸 도면이다. 도 5에 도시된 바와 같이, 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 시점 종속적 상세단계(View dependent LOD)는 관측자로부터 가까운 부분은 자세히, 먼 부분은 덜 자세하게 표현하는 기법으로서, 다음과 같은 수학식 3에 의해 더 상위 레벨로의 탐색 여부를 판별하는 플래그 값 (f)을 구한다.

수학식 3

[0043]
$$f = \begin{cases} 0 & \text{if } |v,p| < \frac{\lambda}{\ell} \\ 1 & \text{otherwise} \end{cases}$$

[0044] 여기서, v는 관측지점(view point), p는 경개구역(boundary sphere of a region), λ는 미리 정해진 문턱값(predefined threshold), ℓ은 블록의 레벨(Block's level)을 나타낸다.

[0045] 연속 상세 단계(CLOD) 방법에서 표면거칠기를 고려한 방법은 전처리단계에서 생성한 사진트리 텍스처에 저장된 값을 이용한다. 이 방법은 도 6과 같이 블록의 표면거칠기 값이 작은 경우엔 관측자가 가까워도 덜 자세하게 보여주고, 큰 경우엔 관측자와의 거리가 멀어도 자세하게 보여주는 문제가 생긴다. 따라서, 본 발명에서는 연속 상세 단계(CLOD)에서 표면거칠기를 고려한 방법을 시점 종속적 상세 단계(View dependent LOD)와 같이 관측자로부터 가까운 부분은 자세히, 먼 부분은 덜 자세하게 나타내기 위해 수학식 3을 다음과 같이 변형한다. 수학식 4에서는 수학식 3을 통해 구한 더 상위 레벨로의 탐색 여부를 판별하는 플래그 값 (f)을 전처리 단계인 S100에서 구한 표면거칠기 값(e)으로 나누어 구한 값을 플래그 (f)값으로 한다.

수학식 4

[0046]
$$f = \begin{cases} 0 & \text{if } |v,p| < \frac{\lambda}{\ell \times e} \\ 1 & \text{otherwise} \end{cases}$$

[0047] 여기서, v는 관측지점(view point), p는 경개구역(boundary sphere of a region), λ는 미리 정해진 문턱값(predefined threshold), ℓ은 블록의 레벨(Block's level)을 나타내며, e는 표면거칠기값을 나타낸다.

[0048] 단계 S100에서 시각 절두체 선별(Hierarchical View Frustum Culling, 이하 VFC)은 선별 기법의 하나로서, 선별 기법(Culling Techniques)이란 컴퓨터 그래픽스에서 렌더링하고자 하는 전체 장면 중 최종 이미지에 기여할 것으로 보이지 않는 부분들을 없애는 것을 의미한다. 물론 이때 장면의 최종 이미지에 기여하는 나머지 부분은 렌더링 파이프라인으로 보내진다. 그 중 시각 절두체 선별(VFC)은 상세 레벨 선택 과정을 통해 생성된 삼각형 메시의 수를 줄일 수 있는 가시성 결정을 위한 최적화 방법으로써 시점을 중심으로 시야(field of view) 영역의 바깥쪽에 있는 다각형(polygon)들을 제거하고 시각 절두체에 포함되는 기하 요소들만을 렌더링하는 것이다. 시각 절두체 선별 과정을 통하여 불필요한 연산을 줄이고, GPU로 전송되어야 할 다각형의 수, 즉 데이터를 감소시킬 수 있기 때문에 렌더링 속도가 더 빨라지게 된다.

[0049] 단계 S100에서는 상세단계(LOD) 선택을 통해 각 블록에 포함된 지형을 구성할 노드들을 탐색하고, 탐색된 노드들의 각 정점들 중 시각 절두체 선별(VFC)에 의해 시야의 안쪽에 있는 정점들을 사진트리로 증식시키며 복수의 가시 정점들을 생성한다. 이를 위해 본 발명에서는 그래픽스 하드웨어의 렌더링 파이프라인에서 기하 음영 단계(Geometry Shader stage, 이하 GS) 및 스트림 출력 단계(Stream Output stage, 이하 SO)을 이용하여 병렬적으로 지형을 구성할 노드를 탐색하는 방법을 제안한다. 이 방법은 효과적으로 지형을 렌더링하기 위해 탑-다운(top-down) 방식으로 노드를 탐색한다. 기하 음영 단계(GS)는 정점들로 구성된 point, line segment, 또는 triangle 등과 같이 이미지를 구성하는 프리미티브(primitive)를 입력받아 기하정보를 수정하거나 하나 이상의 프리미티브를 구성하는 정점들을 생성하거나 삭제할 수 있다. 렌더링 파이프라인의 특성상 기하 음영 단계(GS)에서 각 프리미티브는 병렬적으로 처리되다. 기하 음영 단계(GS)를 통해 새로 생성되거나 수정된 정점 정보들을 담은 서브셋(subset)은 스트림 출력 단계(SO)를 통하여 1D 출력 버퍼(Output Buffer)로 순차적으로 출력된다. 이러한 기하 음영 단계(GS)와 스트림 출력 단계(SO)의 기능과 각 정점의 (x,y) 좌표값을 사용해 기존 사진트리에서 문제가 되었던 재귀연산과 포인터 문제를 해결할 수 있다. 시각 절두체 선별(VFC)에 의해 복수의 가시 정점들을 생성하는 과정은 도 7과 같은 흐름에 의해 이루어진다.

[0050] 도 7은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 시각 절두체 선별(VFC)에 의해 복수의 가시 정점을 생성하는 과정을 나타낸 흐름도이다. 도 7에 도시된 바와 같이, 본 발명의 일실시예에 따른 시각 절두체 선별(VFC)에 의한 복수의 가시 정점 생성에서 렌더링 파이프라인은, 기하 음영 단계(GS) 및 스트림 출력 단계(SO)를 포함하고, 렌더링 파이프라인의 초기 값으로 사진트리의 루트 노드에 해당하는 단일 정점을 입력받는 단계(S110), 기하 음영 단계(GS)에서 입력받은 단일 정점이 각각의 노드에 대해 상위 레벨로의 탐색 여부를 판별하는 플래그 값(f)에 따른 탐색 조건을 만족하는지 상세단계(LOD) 선택을 하는 단계(S120), 상세단계(LOD) 선택 결과, 단일 정점이 탐색 조건을 만족하지 않으면 단일 정점을 기준으로 분할되는 4개의 서브블록의 중점에 자식 노드들의 정점을 생성하되, 각각의 자식 노드의 경계구(p)가 시각 절두체의 바깥에 위치하면 해당 자식 노드에서는 정점을 생성하지 않으며, 4개의 자식 노드들이 모두 시각 절두체의 바깥에 위치하면 자식 노드들에 대한 정점들뿐만 아니라 단일 정점 자신도 삭제하는 시각 절두체 선별(VFC) 연산을 수행하는 단계(S130), 단계 S130의 결과를 스트림 출력 단계(SO)를 통해 1차원 버퍼에 저장하는 단계(S140), 및 각 블록들이 위치할 장소에 각 블록의 레벨을 지정한 정점들을 생성하기 위하여 저장된 결과를 렌더링 파이프라인으로 제공하여 각 자식 노드들의 정점마다 생성해야 하는 사진트리의 깊이만큼 단계 S110 내지 단계 S140의 과정을 반복 수행하는 단계(S150)를 포함한다.

[0051] 단계 S110은, 렌더링 파이프라인의 초기 값으로 사진트리의 루트 노드에 해당하는 단일 정점을 입력받는 과정이

다. 렌더링 파이프라인은 초기 값으로 사진트리의 루트 노드에 해당하는 단일 정점(single vertex)을 입력받는다. 이 정점은 사진트리 텍스처에서와 같이 노드가 의미하는 블록의 중심 포인트(center point)에 위치한다.

[0052] 단계 S120은, 입력받은 단일 정점이 기하 음영 단계(GS)의 각 노드에 대해 상위 레벨로의 탐색 여부를 판별하는 플래그 값(f)에 따른 탐색 조건을 만족하는지 여부를 판단하는 단계이다. 루트 노드에 해당하는 정점은 GS에서 각각의 노드에 대해 앞의 수학식 3 및 수학식 4의 플래그 값(f), 즉, 더 상위 레벨로의 탐색 여부를 판별하는 플래그 값 (f)에 따라 탐색 조건을 만족하는지를 판단한다. 만약 단계 S120에서의 탐색 결과에 따라 다음 단계에서 더 상위 레벨이 존재하면 탐색을 계속하고, 그렇지 않은 경우는 트리의 최대 레벨까지 탐색된 경우이므로 처음으로 회귀하게 된다.

[0053] 단계 S130은, 단계 S120에서의 상세단계(LOD) 선택 결과, 단일 정점이 탐색 조건을 만족하지 않으면 단일 정점을 기준으로 분할되는 4개의 서브블록의 중점에 자식 노드들의 정점을 생성하되, 각각의 자식 노드의 경계구(P)가 시각 절두체의 바깥에 위치할 경우 해당 자식 노드에서는 정점을 생성하지 않으며, 4개의 자식 노드들이 모두 시각 절두체의 바깥에 위치할 경우 자식 노드들에 대한 정점들뿐만 아니라 단일 정점 자신도 삭제하는 시각 절두체 선별(VFC) 연산을 수행한다. 본 발명의 노드 탐색 과정에서 시각 절두체 선별을 함께 진행하는 방법은 도 8을 통해 설명한다.

[0054] 도 8은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서의 시각 절두체 선별(VFC) 방법을 나타낸 도면이다. 도 8에 도시된 시각 절두체 선별(VFC)은 앞서 살펴본 바와 같이, 시야의 바깥에 있는 정점들을 선별하여 렌더링하지 않는 방법으로서, 노드 탐색을 할 때 4개의 자식 노드들을 탐색할 때 각 자식 노드의 경계구 (P)가 시각 절두체의 바깥에 있는 경우 그 노드에서는 정점을 생성하지 않는다. 만약 4개의 자식 노드가 전부 시각 절두체의 바깥에 있다면 아무런 정점도 생성하지 않고 자기 자신도 삭제한다. 이렇게 노드를 탐색하는 과정에서 시각 절두체 선별을 적용하면 불필요한 가시 정점들에 대한 연산을 미리 차단할 수 있다.

[0055] 단계 S140은, 단계 S130에서의 생성 및 삭제된 결과를 SO(Stream Output stage)를 통해 1차원 버퍼에 저장한다.

[0056] 단계 S150은, 각 블록들이 위치할 장소에 각 블록의 레벨을 지정한 정점들을 생성하기 위해 저장된 결과를 렌더링 파이프라인으로 제공하여 각 자식 정점마다 생성해야 하는 사진트리의 깊이만큼 단계 S110 내지 단계 S140의 과정을 반복 수행한다. 단계 S140에서 저장된 결과는 다시 렌더링 파이프라인으로 피드백되어 각 정점마다 생성해야 하는 사진트리의 깊이(depth)만큼 같은 일을 반복 수행한다. 이 결과로 지형 데이터를 분할한 각 블록들이 위치할 장소에 블록의 레벨을 지정한 정점들이 생성된다. GS의 특징상 각각의 정점마다 병렬 알고리즘이 처리되기 때문에 CPU에서 트리를 검색(traverse)하는 방법보다 빠르다.

[0057] 단계 S200은, 생성된 복수의 가시 정점들로부터 삼각 메시를 생성하고, 생성된 메시들 간의 이웃 노드 검사를 통해 크랙 제거를 위한 패치를 생성하여 정점들을 패치된 블록 메시로 변환한다. 단계 S200에서 생성된 복수의 가시 정점들은 이웃(neighbor) 노드들과 블록 레벨(block level)을 비교한 후, 패치된 블록(pached block)으로 변환된다. 이 과정에서 각 노드들 간의 레벨 차로 인하여 크랙(crack)이 발생하게 되는데, 이는 도 9를 통해 알 수 있다.

[0058] 도 9는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 메시 생성 시에 각 노드들 간의 레벨 차로 인하여 크랙이 발생한 것을 나타낸 도면이다. 도 9에 도시된 것과 같은 크랙의 발생을 없애기 위해서 이웃(neighbor) 정점들을 검사해야 한다. 기존의 방법들에서는 크랙 제거를 위해 가시성 전파 작업을 수행했는데 이 작업은 CPU에서만 이루어졌다. 따라서 본 발명에서는 GPU에서 크랙을

제거하기 위한 방법을 제안하며, 이는 도 10 내지 도 14를 통해 설명한다.

[0059] 도 10는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 크랙 제거를 위한 패치를 생성하여 블록을 변환하는 과정을 나타낸 흐름도이고, 도 11은 크랙의 발생 여부에 따라 각각 생성되는 패치를 나타낸 도면이다. 또한, 도 12는 이웃 체크(Neighbor Check)로 찾는 T-정점(T-vertex)를 나타낸 도면이고, 도 13은 패치가 생성되는 과정을, 도 14는 크랙이 제거된 모습을 나타낸 도면이다. 도 10에 도시된 바와 같이, 본 발명의 일실시예에 따른 크랙 제거를 위한 패치를 생성하여 블록을 변환하는 과정은, 단계 S100에서 생성된 복수의 가시 정점들이 나타내는 블록의 상하좌우에 위치한 동일한 레벨의 블록을 갖는 4개의 이웃 정점들의 위치를 계산하는 단계(S210), 위치가 계산된 이웃 정점들에 대하여 앞의 수학식 4를 만족하는 플래그 값(f)을 이용하여 크랙 발생 여부를 판단하고, 판단 결과에 따라 패치를 생성하여 가시 정점들을 패치된 블록 메시로 변환하는 단계(S230), 변환된 블록 메시를 SO를 통해 다시 렌더링 파이프라인으로 피드백하는 단계(S250)를 포함한다.

[0060] 단계 S210은, 단계 S100에서 생성된 복수의 가시 정점들이 나타내는 블록의 상하좌우에 위치한 동일한 레벨의 블록을 갖는 4개의 이웃 정점들의 위치를 계산한다. 각 정점은 도 11의 (a)와 같이 자신이 의미하는 블록의 상하좌우에 위치한 동일한 레벨의 블록을 갖는 4개의 이웃 노드들의 위치를 계산할 수 있다.

[0061] 단계 S230은, 위치가 계산된 이웃 정점들에 대하여 앞의 수학식 4를 만족하는 플래그 값(f)을 이용하여 크랙 발생 여부를 판단하고, 판단 결과에 따라 패치를 생성하여 가시 정점들로 구성된 메시를 패치된 블록 메시로 변환한다. 단계 S230에 대하여는 도 13을 통해 설명한다.

[0062] 도 13에 도시된 바와 같이, 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서 패치를 생성하여 패치된 블록 메시로 변환하는 단계 S230은, 위치가 계산된 이웃 정점들에 해당하는 각 삼각형(triangle)들을 입력받아 이웃 노드 검사를 실시한 결과, 크랙이 발생한 경우에는 T-정점이 발견된 위치를 기준으로 삼각형을 2조각으로 쪼개는 단계(S232), 및 단계 S232를 사진 트리의 최대 레벨(L_{max}) - 1 번 반복하여 패치를 생성하는 단계(S234)를 포함한다.

[0063] 단계 S232는, 위치가 계산된 이웃 정점들에 해당하는 각 삼각형(triangle)들을 입력받아 이웃 노드 검사를 실시한 결과, 크랙이 발생한 경우에는 T-정점이 발견된 위치를 기준으로 삼각형을 2조각으로 쪼갠다. 단계 S232는 단계 S210에서 계산된 이웃 노드들의 위치에 상세단계(LOD)에서 구해진 식을 적용했을 때 더 탐색이 가능한지를 판별한다. 전부 탐색이 불가능하다면 이웃한 블록들이 자신보다 하위 레벨이므로 T-정점(T-vertex)이 발생하지 않는다. 이 경우엔 정점을 도 11(b)의 하단과 같이 2개의 삼각형으로 된 블록 메시로 변환하여 준다. 그리고 더 탐색이 가능하다면 도 12과 같이 하위 레벨 블록의 한 변 위에 T-정점이 생성되기 때문에 크랙이 발생할 수 있다. 도 12에서 T는 T-정점이고, N은 이웃(Neighbor)이다. 이 경우엔 도 11(b)의 상단과 같이 4개의 삼각형으로 구성된 블록 메시로 변환한다. 이와 같이 변환된 메시는 스트림 출력 단계(SO)를 통하여 다시 렌더링 파이프라인으로 피드백되는데, 이때 이웃 검사 결과 크랙이 발생하므로 4개의 삼각형으로 생성된 블록에 패치를 생성해줘야 한다.

[0064] 단계 S234는 단계 S232를 사진 트리의 최대 레벨(L_{max}) - 1 번 반복하여 패치를 생성하는 과정으로서, 이때 생성되는 패치는, 단계 S230에서의 판단 결과, 크랙이 발생하지 않으면 이웃 노드들을 이용하여 생성된 2개의 삼각형으로 구성된 블록 메시이고, 크랙이 발생하면 이웃 노드들을 이용하여 생성된 4개의 삼각형으로 구성된 블록 메시이다. 도 13은 패치를 생성하는 모습을 나타낸다. 기하 음영 단계(GS)에선 각 삼각형들을 입력받아 이웃 검사를 한 번씩만 한다. 이웃은 도 12의 AB의 중점에서 AT 혹은 BT의 길이만큼 AB에 수직한 정점으로 계산한다. 이 이웃 노드가 더 탐색이 가능하다면 T-정점이 발견된 위치를 기준으로 삼각형을 2조각으로 쪼갠다. 이 결과를 트리의 L_{max} - 1번 반복해주어 발생할 모든 크랙에 대해서 패치를 생성해 준다. 그러나 만

약 관측자와의 거리만 고려하였다면 이웃과의 상세 레벨 차이는 1 이상 나지 않는다. 그러므로 관측자와의 거리만 고려한 경우에는 1번만 수행하여 연산량을 줄인다. 도 14는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 크랙이 제거된 모습을 나타낸 도면이다.

[0065] 단계 S300은, 패치된 블록들로 구성된 삼각 메시에 고도 필드(height field)로부터 높이 값을 적용하고, 노말 맵으로 쉐이딩하여 지형 영상으로 렌더링한다.

[0066] 그러나, 앞서 설명한 방법은 GPU에서도 CPU와 마찬가지로 사진 트리와 같은 계층 자료 구조를 이용할 수 있도록 함에 비하여 각 정점 혹은 삼각형마다 많은 계산이 필요하다. 정점의 경우엔 상세 단계 선택(LOD)과 시각 절두체 선별(VFC)을 수행하고, 삼각형(Triangle)의 경우엔 이웃 노드(neighbor node) 검사를 하고 패치를 생성한다. 따라서 본 발명에서는 이러한 연산들을 줄이기 위해서 정점의 (x,y,z,w) 값들 중 사용하지 않는 좌표값을 버퍼(buffer)로 이용하여 불필요한 가시 정점들에 대한 연산을 미리 차단하도록 한다.

[0067] 단계 S120에서 각각의 정점에 대하여 수행하는 상세 단계 선택과 단계 S130의 시각 절두체 선별(VFC) 연산 수행에 있어서 각 정점들은 (x, y, z, w)의 4개의 좌표값을 가지고, 이러한 4개의 좌표값 중, 사용하지 않는 좌표값인 (z, w) 값을 버퍼로 이용한다.

[0068] 정점별로 수행하는 상세 단계 선택(LOD)과 시각 절두체 선별(VFC)은 연산과정에서 경계구(boundary sphere)(수학식 3 또는 4에서의 P)의 반지름을 가장 많이 사용하며, 이값은 매번 정점의 위치를 이용하여 계산을 해야 한다. 이러한 비효율적인 계산을 줄이기 위해 한 정점의 4개의 좌표값 중 z 좌표값에 정점의 위치를 이용하여 계산된 경계구(P)의 반지름값을 저장하여 사용한다. 상위 레벨의 반지름을 구할 경우에도 하위 레벨의 1/2을 취해주면 되기 때문에 이 방법은 효율적이다. 또한, w 좌표값은 상세 단계 선택 시에 현재 정점의 레벨보다 더 상위 레벨로의 탐색 여부를 판별하는 플래그 값(f)을 저장한다. 이로써 상세 단계 선택(LOD)을 마친 노드들에 더 이상 불필요한 상세 단계 선택(LOD)과 시각 절두체 선별(VFC) 판별 공식을 적용하지 않아도 된다.

[0069] 또한, 사용하지 않은 좌표값을 버퍼로 이용하는 것은 삼각 메시의 경우에도 적용이 가능한데, 이웃 노드 검사 시에 패치 생성을 위해서 삼각형을 분할하는데 이때 매번 단계 S210과 같이 이웃 노드의 위치를 계산해 주어야 한다. 이 경우, 도 12에서 삼각 메시를 구성하는 세 정점에 대하여 삼각 메시에서 삼각형의 양 빗변을 공유하는 정점(C)의 (x, y, z, w) 좌표값 중, z 좌표값에 T-정점(T)과 이웃 정점(N)과의 거리를 저장하고, 빗변에 이웃하는 두 정점(A, B) 중 하나의 정점(A)의 z 좌표값에는 이웃 정점(N)의 방향을 저장하면 이러한 연산을 줄일 수 있다. 삼각형을 분할한 후에 각 정점 (C)엔 기존 정점(C) 값의 1/2을 저장하면 된다.

[0070] 단계 S120에서 수행하는 패치 생성은 스트림 출력 단계(SO)를 통하여 반복 연산을 하게 되는데 이때 기존의 방법에선 매번 모든 삼각형에 대하여 T정점이 있는지 없는지를 체크해야 했다. 이러한 불필요한 연산을 줄이기 위해서 패치 생성을 마치거나 크랙이 발생하지 않는 삼각형과 크랙이 발생하는 삼각형을 구분해야 한다. 따라서 삼각형의 한 꼭지점인 정점 (C)의 w 좌표값에 패치 생성 여부를 판별하는 플래그 값을 저장함으로써 반복적인 복잡한 과정의 수행함이 없이 동일한 효과를 얻을 수 있게 된다.

[0071] 본 발명은 Intel Core2Duo E8400 CPU에 4GB main memory 를 갖는 시스템에서 수행되었다. 그래픽스 하드웨어는 1GB의 Video memory를 갖는 nVidia 9800GTX(+)를 사용하였고 DirectX 10 라이브러리와 Shader4.0모델을 사용하였다. viewport의 크기는 1024768로 하였다. Puget Sound(20492049)와 Grand Canyon(20492049) Data를 사용하였다.

[0072] 다음 표 1은 Puget Sound(20492049)을 렌더링 시에 Threshold 값에 따른 프레임, triangle 수 그리고 CPU의 이

용 빈도들의 변화를 나타낸다. 이 값들은 카메라의 이동에 따라 바뀌므로 같은 시점에서 바라본 경우를 체크했다.

표 1

[0073]

Threshold	FPS	Triangle	CPU 이용율
20000	236	24k	2%
30000	139	46k	1%
40000	101	65k	1%
50000	74	78k	0%
60000	60	102k	0%
70000	45	130k	0%

[0074] 도 15는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서 74 프레임을 보이는 Threshold=5000의 스크린 샷을 나타낸 도면이다. Threshold 값이 늘어나면 상세단계 간격이 커져 정교한 지형을 렌더링할 수 있게 되고 값을 줄이면 상세단계 간격이 촘촘해져 상대적으로 덜 세밀한 지형이 생성된다. CPU에서는 카메라 행렬과 관측 평면(View plane)을 매 프레임 생성을 하기 때문에 GPU에서 하는 연산을 줄여 Frame rate를 높이면 CPU의 사용빈도가 높아진다.

[0075] VS는 생성된 vertex들의 geometry morphing, PS는 normal-map을 이용한 shading에만 사용된다. 즉, VS와 PS는 총 4개의 pass중에서 마지막 1개의 Pass에서만 사용되므로 활용빈도가 낮다. 이에 반하여, GS는 총 3개의 패스에서 사용이 되며 SO를 통한 반복수행도 많기 때문에 가장 활용 비율이 높다.

[0076] 다음 표 2와 표 3은 LOD 방법에 따른 성능차이를 비교한 것이다. 표 3의 threshold 값은 표 2의 threshold 값을 표면거칠기(e) 값으로 나눈 숫자이다. 사진트리 텍스처에 저장된 표면거칠기 값들의 평균이 0.00231이므로 대략 400배를 해준 값으로 비교하였다. 표면거칠기가 적용된 경우 적용되지 않은 경우에 비하여 Threshold의 변화에 따른 삼각형 개수와 FPS의 변화폭이 안정적인 수치를 보였다. 렌더링 수행중에도 geometry popping이 표면거칠기를 적용한 방법이 훨씬 적게 나타났다. 하지만 표면 거칠기를 고려하기 위해 사진트리 텍스처에 접근하는 과정이 추가가 되었기 때문에 같은 triangle 수를 렌더링했을 때 렌더링 속도가 관측자와의 거리만 고려한 방법보다 떨어진다.

표 2

[0077]

Threshold	Triangle	FPS	CPU 이용율
100	10k	327	3%
200	65k	143	1%
300	85k	76	0%
400	130k	55	0%
500	200k	34	0%
600	260k	24	0%

표 3

[0078]

Threshold	Triangle	FPS	CPU 이용율
40000	45k	102	1%
50000	56k	78	0%
60000	70k	65	0%
70000	85k	52	0%
80000	110k	46	0%
90000	125k	36	0%

[0079] 도 16은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서 LOD 방식에 따른 결과 이미지를 나타낸 도면이다. 여기서 (a)는 시각 종속적 LOD만 적용된 (2049 x 2049) puget sound의 모습이고, (b)는 표면거칠기가 고려된 (2049 x 2049) puget sound의 모습이다. 여기서 상단 (a)image는 threshold가 작을 때의 결과이고, 하단 (b)image는 threshold를 크게 했을 때의 결과이다.

[0080] 도 17은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서의 렌더링 파이프라인에서의 시각 절두체 선별(VFC) 결과를 나타낸 도면이다. 도 17에서 (a)는 다른 시각에서 바라본 결과이고, (b)는 원래의 시각에서 바라본 영상이다.

[0081] 다음 표 4는 시각 절두체 선별(VFC)을 통하여 Culling된 지형을 렌더링한 결과이다. 시각 절두체 선별(VFC)을 적용한 결과 위의 표 1에 비하여 생성되는 Triangle 개수가 줄어들었고 FPS는 상승하였다.

표 4

[0082]

Threshold	Triangle	FPS	CPU 이용율
40000	100k	39	0%
50000	135k	26	0%
60000	155k	20	0%
70000	180k	16	0%
80000	205k	13	0%
90000	230k	11	0%

[0083] 본 발명에서는 사진트리-기반 지형을 그래픽스 하드웨어의 렌더링 pipeline만으로 시각화하는 방법을 소개하였다. Shader Model 4.0의 GS와 SO를 통하여 parallel하게 트리 탐색을 하는 동시에 VFC를 수행하여 전체적인 렌더링 성능이 기존의 사진트리-기반의 렌더링 방법들에 비해 향상되었음을 확인할 수 있었다.

[0084]

[0085] 이상 설명한 본 발명은 본 발명이 속한 기술분야에서 통상의 지식을 가진 자에 의하여 다양한 변형이나 응용이 가능하며, 본 발명에 따른 기술적 사상의 범위는 아래의 특허청구범위에 의하여 정해져야 할 것이다.

도면의 간단한 설명

[0086] 도 1은 본 발명의 일실시예에 따른 그래픽스 처리 유닛(Graphics Processing Unit, 이하 GPU)을 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법의 흐름도.

[0087] 도 2는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 생성되는 사진트리 텍스처를 나타낸 도면.

[0088] 도 3은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 블록에 포함된 정점 및 기하오차값을 나타낸 도면.

[0089] 도 4는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 블록의 크기에 따라 가중치가 부여된 모습을 나타낸 도면.

[0090] 도 5는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 시점 거리를 갖는 상세단계(LOD)의 와이어프레임의 이미지를 나타낸 도면.

[0091] 도 6은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법의 표면거칠기 값이 적용된 연속 상세 단계(CLOD)에서의 이미지를 나타낸 도면.

[0092] 도 7은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 시각 절두체 선별(VFC)에 의해 복수의 가시 정점을 생성하는 과정을 나타낸 흐름도.

[0093] 도 8은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서의 시각 절두체 선별(VFC) 방법을 나타낸 도면.

[0094] 도 9는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 메시 생성 시에 각 노드들 간의 레벨 차로 인하여 크랙이 발생한 것을 나타낸 도면.

[0095] 도 10은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 크랙 제거를 위한 패치를 생성하여 블록을 변환하는 과정을 나타낸 흐름도

[0096] 도 11은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 크랙의 발생여부에 따라 각각 생성되는 패치를 나타낸 도면.

[0097] 도 12은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 이웃 체크(Neighbor Check)로 찾는 T-정점(T-vertex)를 나타낸 도면.

[0098] 도 13는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 패치가 생성되는 과정을 나타낸 도면.

[0099] 도 14은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법을 가속화하는 방법에서 크랙이 제거된 모습을 나타낸 도면.

[0100] 도 15는 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서 74 프레임을 보이는 Threshold=5000 의 스크린 샷을 나타낸 도면.

[0101] 도 16은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서 LOD 방식에 따른 결과 이미지를 나타낸 도면.

[0102] 도 17은 본 발명의 일실시예에 따른 GPU를 이용한 사진트리 기반의 지형 렌더링 방법에서의 렌더링 파이프라인에서의 시각 절두체 선별(VFC) 결과를 나타낸 도면.

[0103] <도면 중 주요 부분에 대한 부호의 설명>

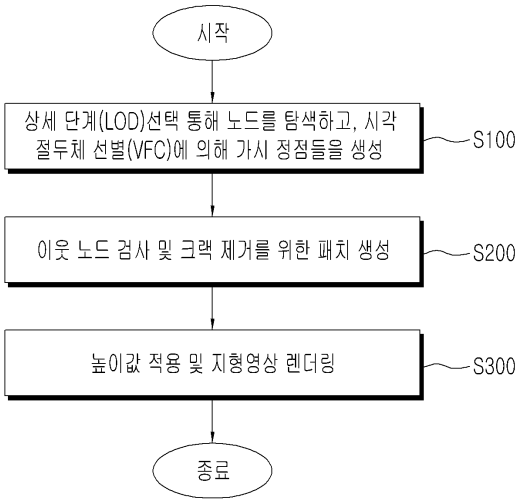
[0104] S100 : 상세단계(LOD) 선택을 통해 노드를 탐색하고, 시각 절두체 선별(VFC)에 의해 가시 정점(visible vertices)들을 생성하는 단계

[0105] S200 : 이웃 노드 검사 및 크랙 제거를 위한 패치를 생성하는 단계

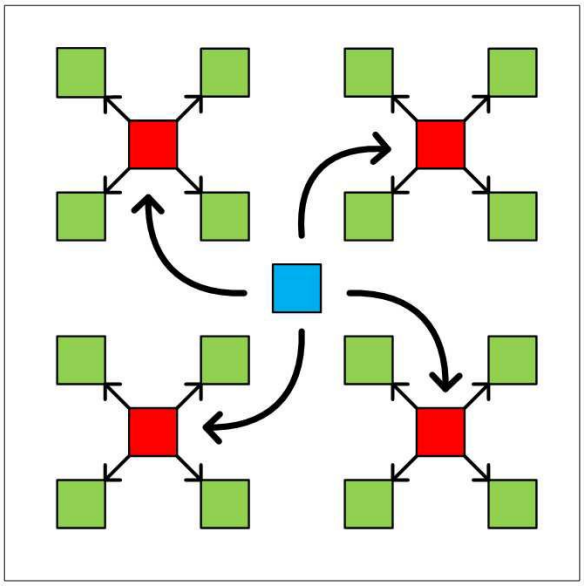
[0106] S300 : 높이 값 적용 및 지형 영상으로 렌더링하는 단계

도면

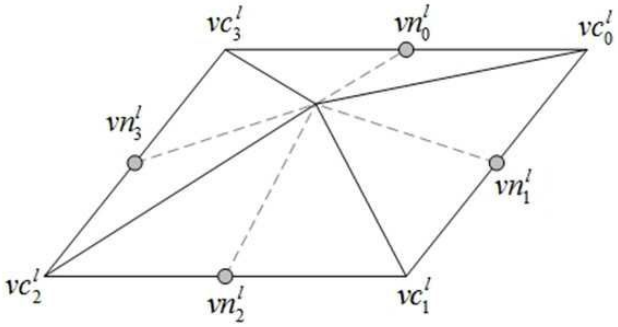
도면1


시작
상세 단계(LOD)선택 통해 노드를 탐색하고, 시각 절두체 선별(VFC)에 의해 가시 정점들을 생성
S100
이웃 노드 검사 및 크랙 제거를 위한 패치 생성
S200
높이값 적용 및 지형영상 렌더링
S300
종료

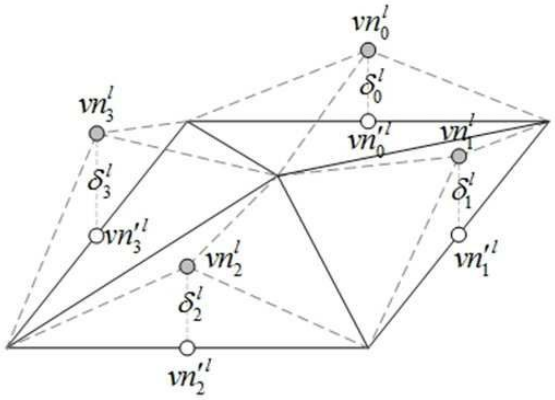
도면2



도면3

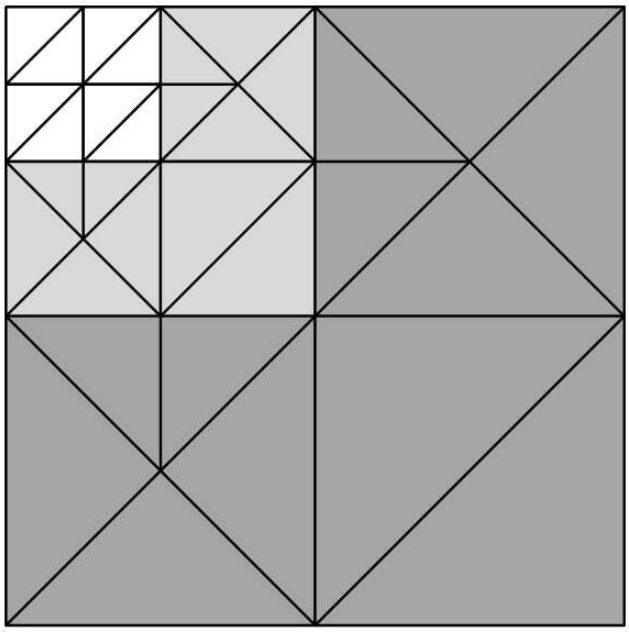

vc_3^l
vn_0^l
vc_0^l
vn_3^l
vn_1^l
vc_2^l
vn_2^l
vc_1^l

(a)

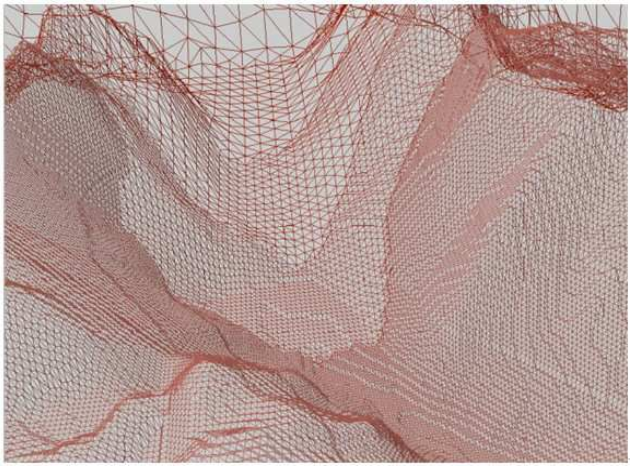

vn_0^l
δ_0^l
vn_3^l
$vn_0'^l$
vn_1^l
δ_3^l
δ_1^l
$vn_3'^l$
vn_2^l
$vn_1'^l$
δ_2^l
$vn_2'^l$

(b)

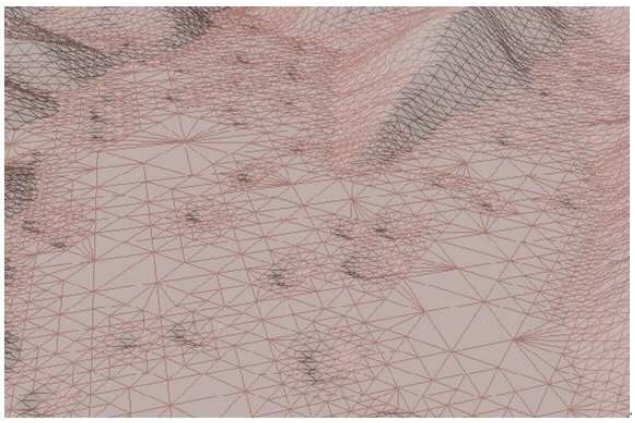
도면4



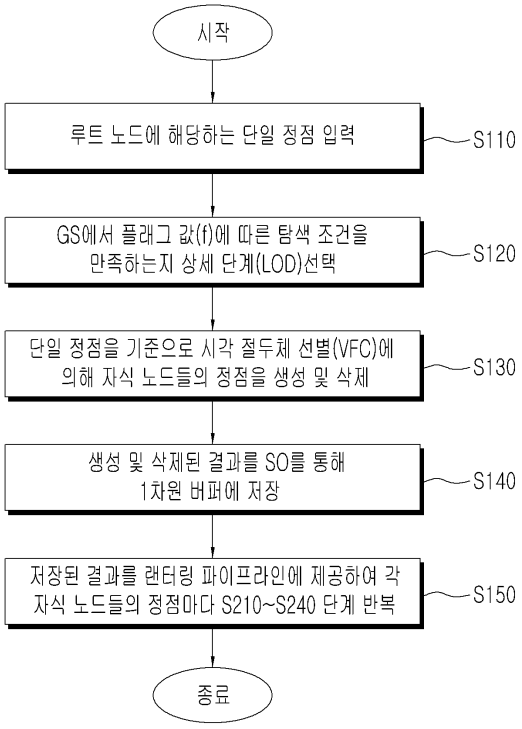
도면5



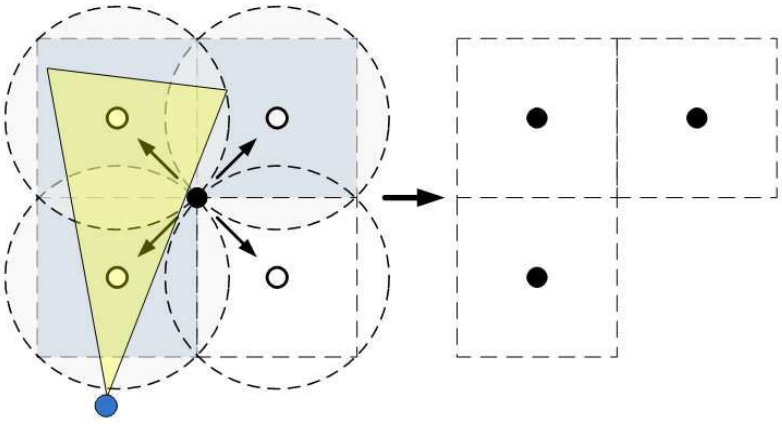
도면6



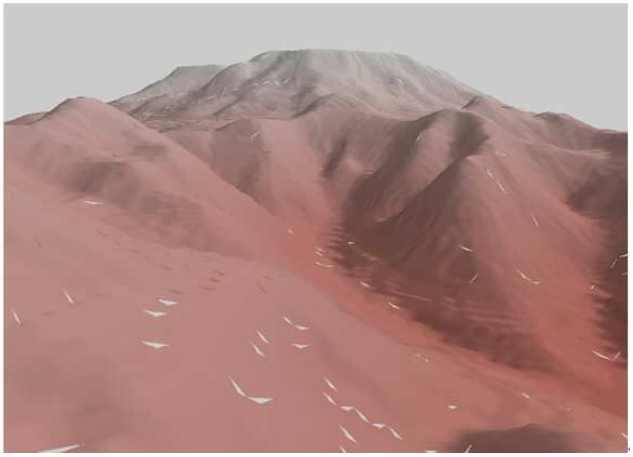
도면7


시작
루트 노드에 해당하는 단일 정점 입력
S110
GS에서 플래그 값(f)에 따른 탐색 조건을 만족하는지 상세 단계(LOD)선택
S120
단일 정점을 기준으로 시각 절두체 선별(VFC)에 의해 자식 노드들의 정점을 생성 및 삭제
S130
생성 및 삭제된 결과를 SO를 통해 1차원 버퍼에 저장
S140
저장된 결과를 랜더링 파이프라인에 제공하여 각 자식 노드들의 정점마다 S210~S240 단계 반복
S150
종료

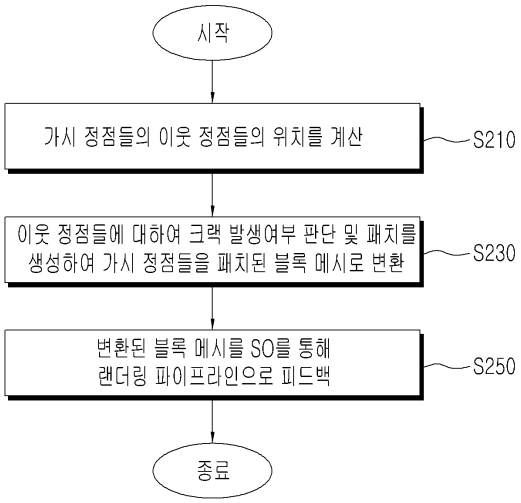
도면8



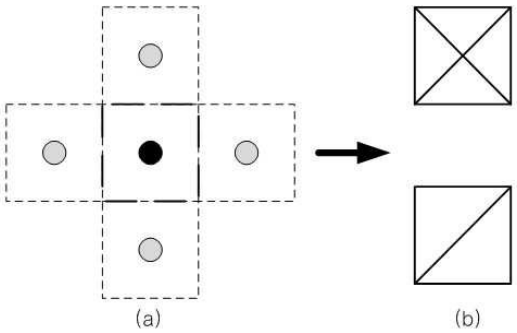
도면9



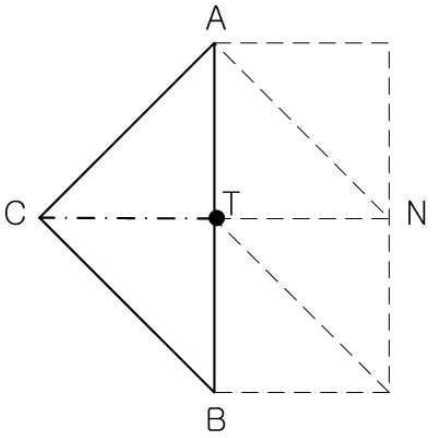
도면10



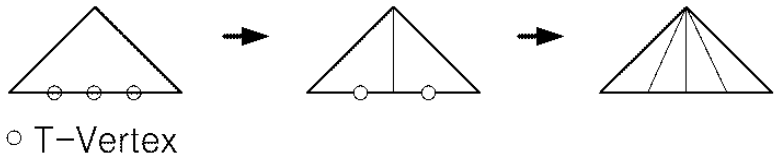
도면11



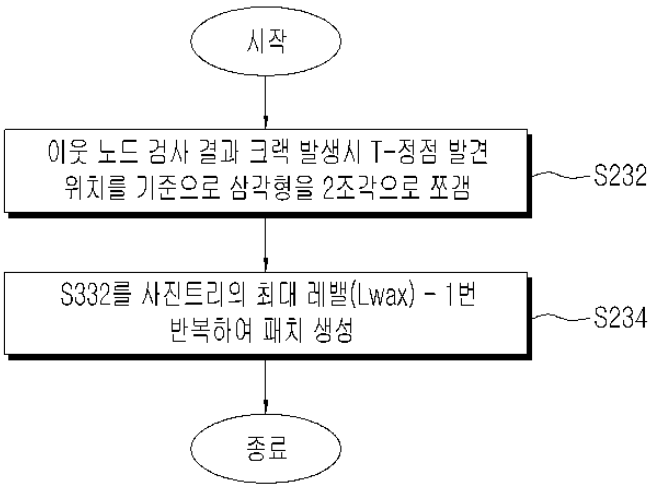
도면12



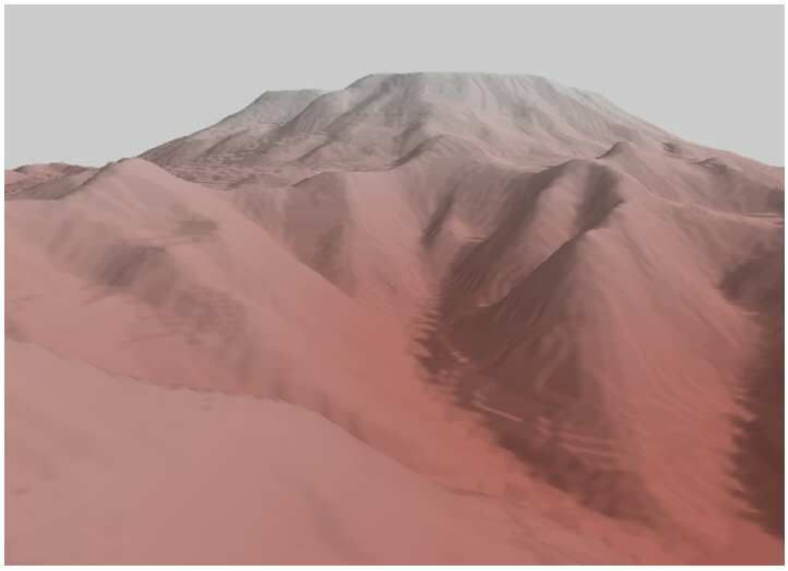
도면13



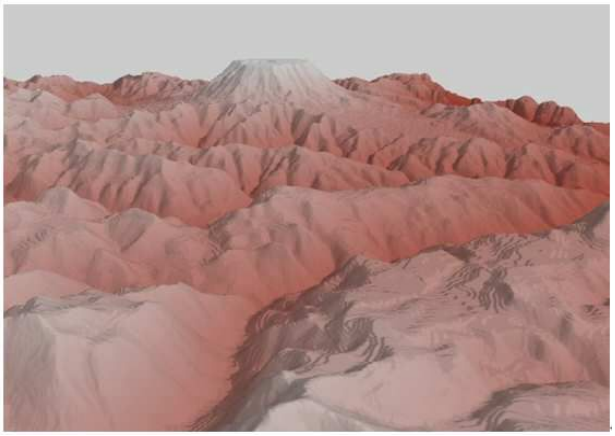
○ T-Vertex



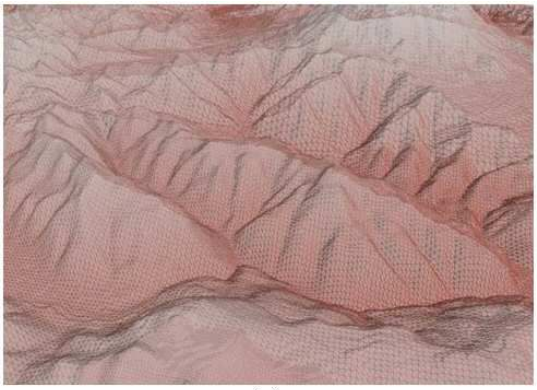
도면14



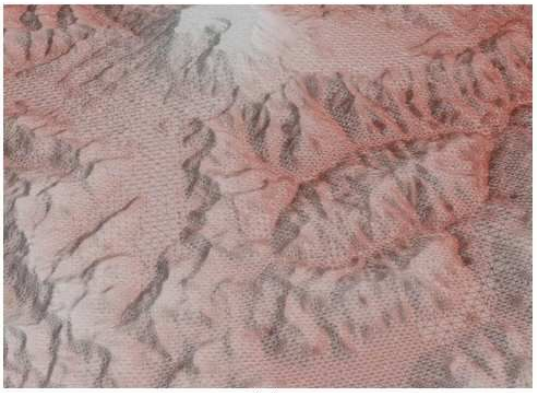
도면15



도면16

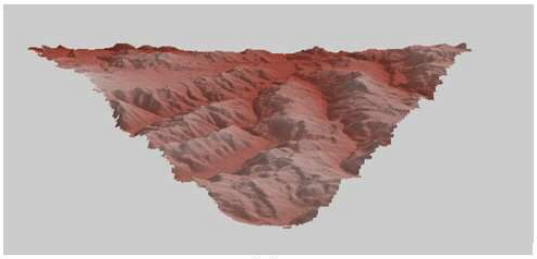


(a)

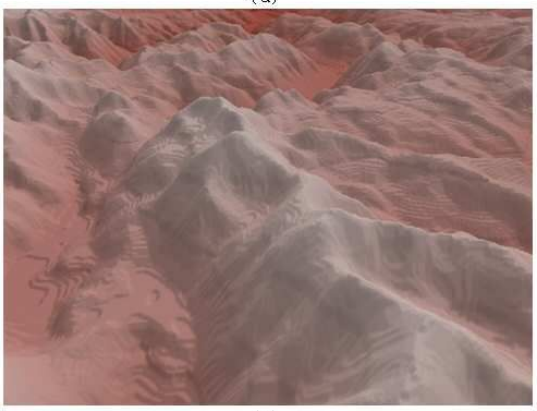


(b)

도면17



(a)



(b)