

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2014-0103407 (43) 공개일자 2014년08월27일
(51) 국제특허분류(Int. Cl.) G06T 17/05 (2011.01) G06T 17/10 (2006.01) (21) 출원번호 10-2013-0016690 (22) 출원일자 2013년02월18일 심사청구일자 2013년02월18일		(71) 출원인 인하대학교 산학협력단 인천광역시 남구 인하로 100, 인하대학교 (용현동) (72) 발명자 신병석 서울 서초구 방배로 14, 3동 903호 (방배동, 임광아파트) 이은석 서울 서초구 서초중앙로 220, 3동 1207호 (반포동, 서초한양아파트) (74) 대리인 양성보

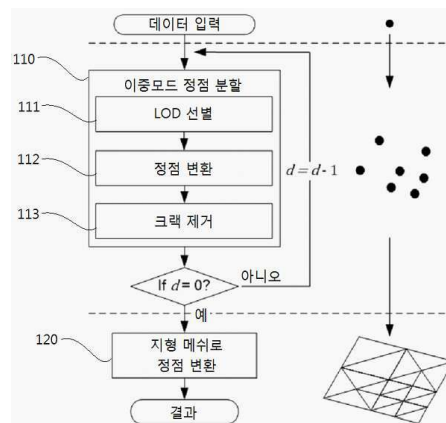
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 **이중모드 정점 분할기법을 이용한 지형 렌더링 가속화 방법**

(57) 요약

이중모드 정점 분할기법을 이용한 지형 렌더링 가속화 방법이 개시된다. 사진트리(Quadtree) 삼각화 기법을 이용한 트리 탐색을 통해 지형 메쉬(mesh)를 생성하는 그래픽 처리 시스템에서의 지형 렌더링 가속화 방법은, 각 노드에 저장된 지형 패치를 대신하는 정점에 대하여, 상세 단계(Level-of-Detail)를 선별하는 정점 분할 모드 및 서로 다른 상세 단계를 가지는 지형 패치 간에 발생하는 크랙(crack)을 제거하는 삼각형 분할 모드를 동시에 수행하는 이중모드 정점 분할 단계; 및 상기 이중모드 정점 분할 단계를 통해 분할된 정점을 지형 메쉬(mesh)로 변환하는 정점 변환 단계를 포함할 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 2012-0005329

부처명 교육과학기술부

연구사업명 핵심연구(개인)

연구과제명 기가바이트급 대용량 비기하 데이터의 HD급 고해상도 가시화 기술

기 여 율 1/1

주관기관 인하대학교 산학협력단

연구기간 2011.05.01 ~ 2014.04.30

특허청구의 범위

청구항 1

사진트리(Quadtree) 삼각화 기법을 이용한 트리 탐색을 통해 지형 메쉬(mesh)를 생성하는 그래픽 처리 시스템에서의 지형 렌더링 가속화 방법에 있어서,

각 노드에 저장된 지형 패치를 대신하는 정점에 대하여, 상세 단계(Level-of-Detail)를 선별하는 정점 분할 모드 및 서로 다른 상세 단계를 가지는 지형 패치 간에 발생하는 크랙(crack)을 제거하는 삼각형 분할 모드를 동시에 수행하는 이중모드 정점 분할 단계; 및

상기 이중모드 정점 분할 단계를 통해 분할된 정점을 지형 메쉬(mesh)로 변환하는 정점 변환 단계를 포함하는 지형 렌더링 가속화 방법.

청구항 2

제1항에 있어서,

상기 이중모드 정점 분할 단계는,

화면 공간 상의 오차가 문턱 값 이상인 경우 상기 정점 분할 모드를 선택하여 상기 정점을 분할하는 단계; 및

상기 화면 공간 상의 오차가 상기 문턱 값 미만인 경우 상기 상세 단계의 선별을 중단하고 크랙 제거 모드인 상기 삼각형 분할 모드를 선택하여 상기 정점을 분할하는 단계

를 포함하는 지형 렌더링 가속화 방법.

청구항 3

제1항에 있어서,

상기 삼각형 분할 모드의 프리미티브 타입(primitive type)을 상기 정점 분할 모드의 프리미티브 타입과 동일하기 위해 하나의 정점으로 압축된 삼각형 정점(triangle vertex)을 이용하는 것

을 특징으로 하는 지형 렌더링 가속화 방법.

청구항 4

제1항에 있어서,

상기 정점 분할 모드에서 분할된 정점은 해당 패치를 구성하는 삼각형 정보가 저장된 4개의 삼각형 정점으로 변환되고,

상기 삼각형 분할 모드는,

상기 삼각형 정보를 이용하여 상기 4개의 삼각형 정점 중 크랙을 유발하는 삼각형 정점을 분할하는 것

을 특징으로 하는 지형 렌더링 가속화 방법.

청구항 5

제4항에 있어서,

상기 삼각형 정보는,

상기 삼각형 정점의 위치 좌표, 상기 삼각형 정점과 해당 삼각형의 빗변 간 수직 거리, 상기 빗변의 방향을 나타내는 방향 플래그, 상기 빗변의 첫 정점의 위치에 비례하는 값인 빗변 인자, 및 상기 삼각형 정점에 대한 정점 분할 회수를 포함하는 것

을 특징으로 하는 지형 렌더링 가속화 방법.

명세서

기술분야

[0001] 본 발명의 실시예들은 지형 렌더링 방법에 관한 것으로서, 보다 구체적으로는 이중모드 정점 분할기법을 이용하여 지형 렌더링을 수행하는 방법에 관한 것이다.

배경기술

[0002] 대용량 지형을 실시간으로 렌더링(rendering) 하는 것은 최근 컴퓨터 게임, 비행 시뮬레이션, 지리정보 시스템(Geographic Information System; GIS) 등에서 가장 중요한 이슈이다. 정교한 지형을 렌더링 하기 위해 DEM(Digital Elevation Model) 데이터가 점차 대용량화 되면서 최근의 고성능 GPU(Graphic Processing Unit)를 사용함에도 불구하고 실시간으로 지형을 렌더링 하기에는 비디오 메모리와 연산 능력의 한계가 있다.

[0003] 이러한 문제점을 해결하기 위하여, 한국공개특허 제10-2012-0112999호(공개일 2012년 10월 12일)에는 DEM 데이터를 대신하여 정점 응집맵을 이용하여 지형 데이터를 가시화 하는 기술이 개시되어 있다.

[0004] 이와 같은 목적에서 최근에는 많은 CLOD(Continuous Level-of-Detail) 기법들이 꾸준히 연구되고 있다. 그 중 사진트리 삼각화 기법은 널리 사용되는 CLOD 기법으로 트리 탐색을 통하여 실시간에 최적의 메쉬를 생성할 수 있도록 한다. 최근의 기하 분할(geometry splitting) 기법은 기존의 CPU 기반 방법들과 달리 최근의 GPU를 이용하여 사진트리 삼각화를 병렬로 처리할 수 있다. 이는 기하 셰이더(geometry shader)에서 지형의 상세단계를 선별하기 위해서 정점을 4분할하며 사진트리를 탐색하고(vertex splitting) 상세단계가 서로 다른 인접한 패치 사이에서 발생하는 크랙을 제거하기 위해서 삼각형을 2분할하는 것(triangle splitting)이다.

[0005] 그러나, 상기한 방법은 기하 셰이더의 스트림 출력(stream output)이 메인 메모리에 출력이 되며 이를 다시 GPU로 재전송 하는 과정에서 병목현상을 보이게 된다. 따라서, 스트림 출력단계가 반복될수록 재전송되는 기하 데이터의 양이 증가하게 된다. 기하 분할기법의 경우 전체 트리를 탐색하는 정점 분할과 삼각형 분할을 따로 수행하기 때문에 트리의 최대 깊이값의 2배만큼 재귀적인 연산을 수행해야 하기 때문에 데이터의 전송량이 많다는 문제가 있다.

발명의 내용

해결하려는 과제

[0006] 정점 분할과 삼각형 분할 단계를 한번에 수행하는 이중모드 정점 분할기법을 이용하여 지형 렌더링을 수행하는 지형 렌더링 가속화 방법을 제공한다.

과제의 해결 수단

[0007] 본 발명의 실시예에 따르면, 사진트리(Quadtree) 삼각화 기법을 이용한 트리 탐색을 통해 지형 메쉬(mesh)를 생성하는 그래픽 처리 시스템에서의 지형 렌더링 가속화 방법은, 각 노드에 저장된 지형 패치를 대신하는 정점에 대하여, 상세 단계(Level-of-Detail)를 선별하는 정점 분할 모드 및 서로 다른 상세 단계를 가지는 지형 패치 간에 발생하는 크랙(crack)을 제거하는 삼각형 분할 모드를 동시에 수행하는 이중모드 정점 분할 단계; 및 상기 이중모드 정점 분할 단계를 통해 분할된 정점을 지형 메쉬(mesh)로 변환하는 정점 변환 단계를 포함할 수 있다.

[0008] 일 측면에 따르면, 상기 이중모드 정점 분할 단계는, 화면 공간 상의 오차가 문턱 값 이상인 경우 상기 정점 분할 모드를 선택하여 상기 정점을 분할하는 단계; 및 상기 화면 공간 상의 오차가 상기 문턱 값 미만인 경우 상기 상세 단계의 선별을 중단하고 크랙 제거 모드인 상기 삼각형 분할 모드를 선택하여 상기 정점을 분할하는 단계를 포함할 수 있다.

[0009] 다른 측면에 따르면, 상기 삼각형 분할 모드의 프리미티브 타입(primitive type)을 상기 정점 분할 모드의 프리미티브 타입과 통일하기 위해 하나의 정점으로 압축된 삼각형 정점(triangle vertex)을 이용할 수 있다.

[0010] 또 다른 측면에 따르면, 상기 정점 분할 모드에서 분할된 정점은 해당 패치를 구성하는 삼각형 정보가 저장된 4개의 삼각형 정점으로 변환될 수 있으며, 이때 상기 삼각형 분할 모드는 상기 삼각형 정보를 이용하여 상기 4개의 삼각형 정점 중 크랙을 유발하는 삼각형 정점을 분할할 수 있다.

[0011] 또 다른 측면에 따르면, 상기 삼각형 정보는, 상기 삼각형 정점의 위치 좌표, 상기 삼각형 정점과 해당 삼각형

의 빗변 간 수직 거리, 상기 빗변의 방향을 나타내는 방향 플래그, 상기 빗변의 첫 정점의 위치에 비례하는 값인 빗변 인자, 및 상기 삼각형 정점에 대한 정점 분할 회수를 포함할 수 있다.

발명의 효과

[0012] 본 발명의 실시예에 따르면, 이중모드의 정점 분할기법을 이용함으로써 기존에 따로 수행되던 정점 분할과 삼각형 분할을 한번의 트리탐색으로 통합하여 재귀 연산량을 절반으로 줄일 수 있으며, 삼각형 정보를 하나의 정점으로 압축하여 삼각형 분할기법에서 전송되던 기하 데이터의 양을 현저히 줄여 총 데이터의 전송량과 연산시간을 줄일 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일 실시예에 있어서, 지형 렌더링 가속화 방법의 처리 과정을 도시한 순서도이다.
 도 2는 본 발명의 일 실시예에 있어서, 상세 단계 선별의 개념을 설명하기 위한 예시 도면이다.
 도 3은 본 발명의 일 실시예에 있어서, 상세 단계 선별의 의사 코드를 도시한 것이다.
 도 4는 본 발명의 일 실시예에 있어서, 사진트리를 이용한 삼각화 기법에서의 크랙을 설명하기 위한 예시 도면이다.
 도 5는 본 발명의 일 실시예에 있어서, 삼각형 분할을 이용한 크랙 제거 과정을 설명하기 위한 예시 도면이다.
 도 6은 본 발명의 일 실시예에 있어서, 삼각형 정점의 형식을 설명하기 위한 예시 도면이다.

발명을 실시하기 위한 구체적인 내용

[0014] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0015] 본 발명의 실시예들은 지형 렌더링 과정의 가속화를 위한 방법으로, 대용량 지형 데이터를 렌더링 하는 그래픽 처리 시스템에 적용될 수 있다.

[0016] 본 명세서에서는 정점 분할 단계와 삼각형 분할 단계를 한번에 수행하는 이중모드의 정점 분할기법을 제안한다. GPU에서는 한번의 패스에서 한 가지 타입의 프리미티브(primitive)만이 사용 가능하기 때문에 프리미티브 타입을 통일해야 한다. 이에, 본 실시예에서는 삼각형 분할 단계에서 사용되던 삼각형 프리미티브들을 삼각형 정점(triangle vertex)이라 하는 하나의 정점으로 압축하여 이를 통일하기로 한다. 각 단계에서는 정점이 상세단계 선별을 수행해야 하는지(정점 분할) 크랙 제거를 수행해야 되는지(삼각형 분할)를 우선 판별한 후, 그에 해당하는 모드를 이용하여 정점(혹은 삼각형 정점)을 분할할 수 있다.

[0017] 이하에서는 이중모드 정점 분할기법을 이용한 지형 렌더링 가속화 방법을 구체적으로 설명하기로 한다.

[0018] 사진트리 삼각화는 일반적으로 LOD(Level-of-Detail) 선별과 크랙 제거라는 2가지 단계를 갖는다. 본 발명에서 제안하는 방법은 렌더링 과정의 가속화를 위하여 GPU 상에서 상기한 2가지 단계를 동시에 수행한다.

[0019] 도 1은 본 발명의 일 실시예에 있어서, 지형 렌더링 가속화 방법의 처리 과정을 도시한 순서도이다.

[0020] 일 실시예에 따른 지형 렌더링 가속화 방법은 이중모드 정점 분할 단계(110)를 수행하고, 이후 지형 메쉬로의 정점 변환 단계(120)를 수행할 수 있다.

[0021] 여기서, 입력 데이터로는 루트 노드에 저장된 사각형 지형 패치를 대신하는 하나의 정점이 사용된다. 본 발명에서 제안하는 방법에서는 두 개의 서로 다른 분할 모드 중 하나를 선택하여 정점을 재귀적으로 나누어 지형 메쉬를 생성할 수 있다.

[0022] LOD 선별 모드는 화면 공간 상의 오차 값에 기반하여 최적의 상세 단계를 찾을 수 있다. 좀 더 상세한 메쉬가 필요할 때에는 정점을 재귀적으로 나눈다. LOD 선별 모드는 정점이 최적의 상세 단계에 도달할 때까지 계속 될 것이다.

[0023] 사진트리 삼각화의 특성 때문에 다른 상세 단계를 가지는 2개의 인접한 패치들에서 크랙이 발생할 수 있다. 이웃 정점이 좀 더 적절한 상세 단계에 도달하기 위해 분할될 때, 크랙 현상이 나타나게 된다. 이러한 크랙들을 제거하기 위해 T-정점이 있는 패치의 삼각형을 분할한다. 그러나, GPU의 셰이더 단계에서는 정점 프리미티브와

삼각형 프리미티브를 함께 사용할 수 없다. 그러므로 정점이 적절한 상세 단계에 도달할 때 그것을 지형 패치를 구성하는 삼각형의 정보를 저장하는 4개의 삼각형 정점으로 바꾼다. 모든 지형 패치는 직각 삼각형이므로 빗변 정보를 사용하여 효율적으로 삼각형을 정점으로 압축할 수 있다.

- [0024] T-정점은 삼각형의 빗변 위에 나타나므로 T-정점의 위치를 기반으로 삼각형을 분할하는 것과 같은 삼각형의 정점을 분할하는 방법으로 T-정점을 제거할 수 있다. 이 모드는 다른 정점들이 LOD 선택 모드를 진행하는 동안 효율적으로 균열을 제거한다.
- [0025] 이중모드의 정점 분할(110)을 d번 수행한 뒤, 정점들을 지형 메쉬로 변환할 수 있다(120). 도 1에서 d의 초기 값은 사인트리의 깊이 값이다. 따라서, 두 가지 모드의 정점 분할 과정을 d번 반복한다는 것은 삼각화 과정에서 트리의 모든 노드들을 병렬로 탐색할 수 있다는 것을 의미할 수 있다. 또한, 모든 분할 단계에서 감소하는 d값으로부터 현재의 깊이 값을 판단할 수 있다.
- [0026] LOD 선별 모드(111)
- [0027] LOD 선별 모드는 사인트리에 저장된 최대 기하 오차 값을 사용한 지형 간략화를 수행한다. 기본 개념은 도 2에 도시한 바와 같이 정규 격자의 분할을 병렬로 하는 것이다.
- [0028] 도 2를 참조하면, 지형 패치를 대신하는 하나의 정점은 해당 패치의 중앙 지역에 위치한다. 패치의 속성들을 저장하기 위해 위치(x,y,z)와 색(R,G,B) 채널을 사용한 정점 레이아웃을 정의한다. 정점으로 대신한 지형 패치의 중앙점이 해당 정점의 x, y 필드에 저장될 것이다. 지형 패치의 대각선 길이는 z 필드에 저장된다. 그리고, 색 채널의 R 필드는 분할 모드를 저장한다. 정점을 분할할 때 자식 정점들은 부모의 위치를 기준으로 분할된 서브패치들의 중앙에 위치할 것이다.
- [0029] 도 3은 LOD 선별 모드의 의사 코드를 보여준다.
- [0030] 본 실시예에서는 지형 패치가 시각 절두체의 안쪽에 위치하는지 아닌지를 확인한다. 만약 안쪽에 위치하지 않는다면 해당 지형 패치는 결과 이미지에 관여하지 않기 때문에 처리할 필요는 없다. 따라서, 이 정점을 삭제하는 것으로 총 연산시간을 절약할 수 있다. 그렇지 않고 정점이 시각 절두체 안에 위치한다면 적정 상세 단계를 선택한다. 화면 공간 상의 오차가 문턱값 τ 보다 크다면 4개의 자식 노드에 해당하는 정점들로 정점을 분할한다. 그렇지 않으면 LOD 선별을 멈추고 크랙 제거 모드로 바꾼다. 하지만 경계구 안에 시점이 있다면 화면 공간 상의 오차를 측정을 할 수 없다. 그러므로 최대 기하 오차가 0이어서 화면 공간 상의 오차가 0이 될 때를 제외하고 정점을 분할한다. 최대 기하 오류가 0일 경우엔 상세 단계 선별 모드를 중단하고 크랙 제거 모드를 수행한다.
- [0031] 삼각형을 압축한 삼각형 정점 변환(112)
- [0032] 도 4는 사인트리를 이용한 삼각화 기법에서의 크랙을 설명하기 위한 예시 도면으로, 여기서 회색 정점은 크랙을 유발하는 T-정점을 의미한다.
- [0033] 사인트리 삼각화 기법에서는 일반적으로 도 4에 도시한 바와 같이 인접한 패치들이 서로 다른 상세 단계를 갖는 경우 크랙이 발생한다.
- [0034] 회색 정점은 T-정점이다. 하향식으로 상세 단계를 선별할 경우 T-정점은 한 정점이 LOD 선별을 마치고 그 점의 이웃점들은 계속 분할해야 할 경우 나타난다. T-정점을 제거하기 위해서 삼각형 분할기법을 이용한다.
- [0035] 도 5는 삼각형 분할을 이용한 크랙 제거 과정을 설명하기 위한 예시 도면이다. 이때, 삼각형 t_{right} 를 재귀적으로 분할하여 T-정점(회색 정점)을 제거할 수 있다.
- [0036] 그러나, 삼각형 분할기법은 GPU 상에서 삼각형 프리미티브를 사용한다. LOD 선별 모드(111)에서 정점 프리미티브를 사용하므로 정점 변환 단계(112)는 LOD 선별 모드(111)와 동시에 수행할 수 없다. 그러므로, 삼각형 데이터는 하나의 정점으로 압축할 수 있다. 이것을 '삼각형 정점'이라 부른다. 따라서, 크랙 제거 모드(113)에서도 정점 프리미티브와 같은 형태의 프리미티브를 사용함으로써 GPU의 동일한 패스 상에서 크랙 제거와 LOD 선별을 수행할 수 있다. 상세 단계 선별 과정에서 적절한 상세 단계를 선별하면 바로 크랙 제거 모드(113)가 수행될 수 있다. 따라서, 한 정점의 이웃 노드가 LOD 선별 모드(111)를 수행하고 있을 때 삼각형 분할을 동시에 수행하여 효과적으로 크랙을 제거할 수 있다.
- [0037] 도 6의 (a)는 정점(검정색 정점)을 삼각형 정점(회색 정점)으로 변환하는 모습을 도시한 것이고, (b)는 삼각형

정점에 저장되는 t_{right} 의 속성을 도기한 것이다.

[0038] 도 6을 참조하여, 삼각형 정점의 형식을 자세히 설명한다. 각 정점은 적정 상세 단계에 도달했을 경우 네 개의 삼각형 정점으로 변환된다(도 6의 (a)). 각 삼각형 정점들은 사진트리의 노드를 대신하는 사각형 패치를 구성하는 삼각형들의 정보를 압축하여 저장한다. 도 6의 (b)는 삼각형 정점의 압축 스키마를 보여준다. 사각형 패치의 중심점인 C의 xy 좌표는 x와 y 채널에 저장될 것이다. z채널에는 점 C와 빗변 AB 사이의 거리 값인 r 이 저장된다. 이 값은 LOD 선별 모드(111)에 의해 결정된 적정 상세 단계를 계산하는데 쓰인다.

[0039] 색 채널은 도 6의 (b)에서 보듯이 압축된 삼각형의 정보를 담고 있다. R 필드에는 방향 플래그가 저장된다. 방향 플래그는 삼각형의 빗변 방향(위, 아래, 왼쪽, 오른쪽)을 의미한다. G 필드에는 빗변 인자 H가 저장된다. H는 빗변의 첫 정점의 위치에 비례하는 값이다. 빗변 분할을 간단하게 연산하기 위해 빗변의 길이 AB를 1로 정규화 한다. 최종적으로, B 필드에는 크랙 제거 과정이 종료된 시점에서의 상세 단계 값을 저장하기 위해 d 값을 저장한다.

[0040] 크랙 제거(113)

[0041] 도 5를 통해 설명한 바와 같이, 빗변의 중심점 T를 기반으로 삼각형을 분할하면 삼각형은 더 이상 직각 삼각형이 아니게 된다. T-정점(도 5의 회색 정점)을 제거하기 위해 삼각형을 분할할 때 분할된 첫 번째 삼각형은 $\triangle CAT$ 이고 두 번째 삼각형은 $\triangle CTB$ 가 된다. $\triangle CAT$ 에서 A는 AT 모서리의 첫 번째 정점이다. 변 AB의 시작점이 A이므로 H 는 0.0이 될 것이다. $\triangle CTB$ 의 경우 모서리 TB의 첫 번째 정점은 T이다. 이 T는 변 AB의 중앙에 위치하므로 H 는 0.5이다. 그러므로 삼각형 정점을 분할할 때 아래 수학적 식 1과 같이 분할된 자식 삼각형 정점들의 H 값인 H' 을 구할 수 있다. 여기서 h 는 빗변의 길이이다.

수학적 식 1

$$\begin{cases} H'_0 = H \\ H'_1 = H + \frac{h_{d-1}}{2r} \end{cases}$$

[0042]

[0043] LOD 선택 모드에서 h 를 전체 지형의 대각선 길이 I_{diag} 와 d 값을 이용하여 구할 수 있다(수학적 식 2를 참조). 따라서, 수학적 식 2와 같이 크랙 제거 모드의 마지막 과정에서 최종 d 값을 저장해야 마지막 삼각형 정점($\triangle CTB$ 를 예로 들 경우 정점 B)을 구할 수 있다.

수학적 식 2

$$h_l = \frac{I_{\text{diag}}}{2^{d_{\text{max}} - d} \sqrt{2}}$$

[0044]

[0045] 삼각형 정점의 압축 해제는 간단하다. 예를 들면, $\triangle CTB$ 를 저장한 삼각형 정점을 사용할 경우, 수학적 식 3으로 삼각형을 재구성할 수 있다.

수학적 식 3

$$V_{\triangle CTB} = \begin{cases} C(x, y) \\ T(x + r, y + r - 2rH) \\ B(x + r, y + r - 2rH - h_d) \end{cases}$$

[0046]

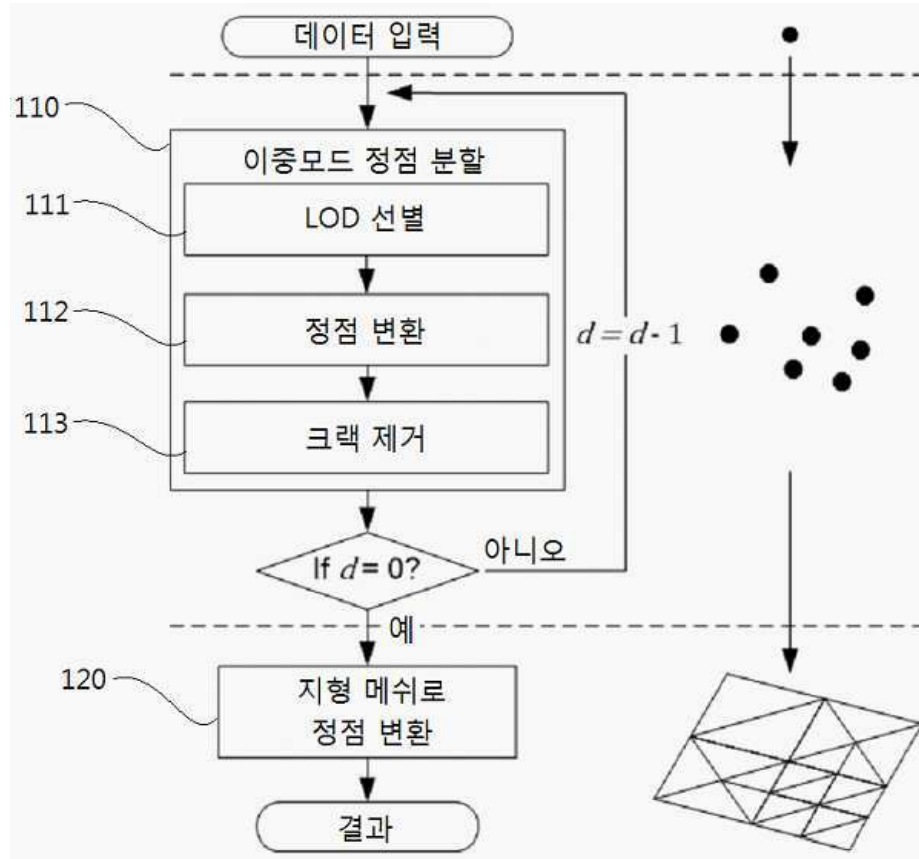
- [0047] 방향 플래그를 이용하면 나머지 삼각형 또한 쉽게 압축 해제를 할 수 있다. 본 발명에서의 압축 기술은 크랙 제거모드와 LOD 선별 모드에서 같은 입력 프리미티브를 사용할 수 있도록 함으로써 크랙 제거와 동시에 LOD 선택을 할 수 있도록 한다.
- [0048] 지형 메쉬로 정점 변환하기(120)
- [0049] 정점 분할을 수행하면 많은 수의 정점이 생성된다. 몇 정점들은 LOD 선택 모드를 마지막으로 분할 과정을 끝냈을 것이며 나머지 정점들은 삼각형 정점들로 변환된 후 분할을 종료할 것이다. 이러한 정점들은 각각의 지형 패치들로 변환해야 한다.
- [0050] LOD 선택 모드에서 사용되었던 정점들은 사각형 패치로 변환한다. 삼각형 정점들은 삼각형으로 변환한다. 이 기하 데이터들은 최종 지형 메쉬를 표현할 수 있다.
- [0051] 이와 같이, 본 발명의 실시예에 따르면, 이중모드의 정점 분할기법에서는 기존에 따로 수행되던 정점 분할과 삼각형 분할을 한번의 트리탐색으로 통합하여 재귀 연산량을 절반으로 줄일 수 있으며, 삼각형 정보를 하나의 정점으로 압축하여 삼각형 분할기법에서 전송되던 기하 데이터의 양을 1/3으로 줄여 총 데이터의 전송량과 연산시간을 줄일 수 있고, 기존의 사진트리 기반 광선 투사법에 비하여 6배에 가까운 속도 증가율을 달성할 수 있다.
- [0052] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0053] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0054] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0055] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태

로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

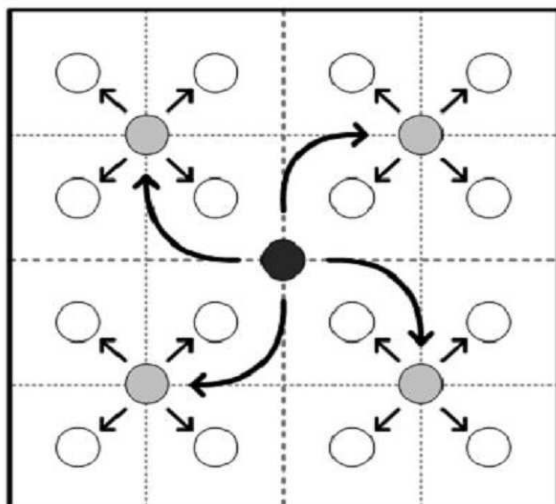
[0056] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

도면1



도면2



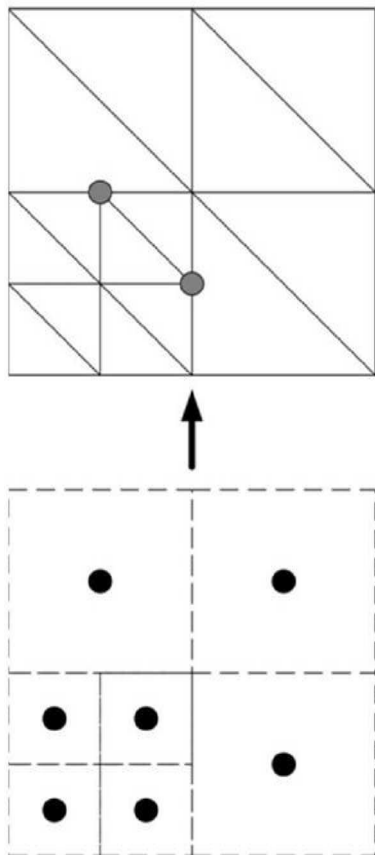
도면3

```

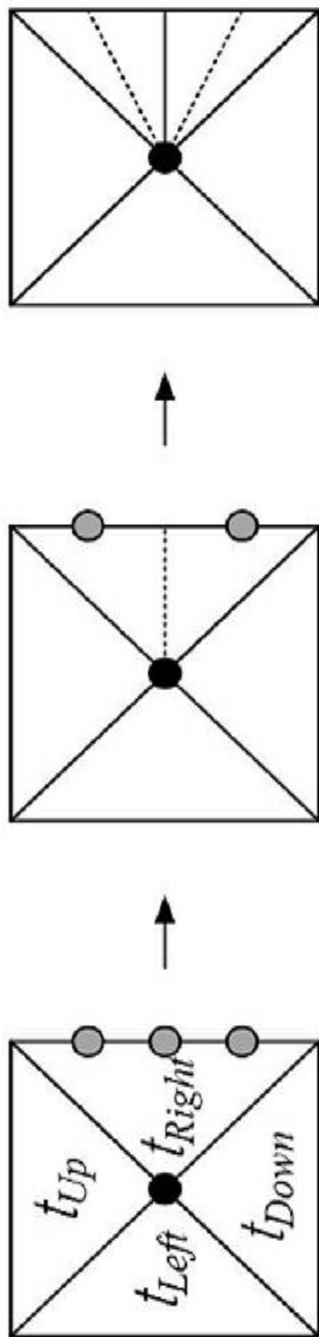
LOD_Selection ( vertex )
{
  if ( is_in_frustum ( vertex ) )
    if ( is_in_bounding_sphere ( eye ) )
      if ( max_geometric_error ( vertex ) > 0 )
        LOD_split ( vertex )
      else
        change_mode_to_crack_removal ( vertex )
    else
      if ( max_screen_space_error ( vertex ) >  $\tau$  )
        propagate_vertex ( vertex )
      else
        change_mode_to_crack_removal ( vertex )
  else
    delete_vertex ( vertex )
}

```

도면4



도면5



도면6

