(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년09월02일
(11) 등록번호 10-2701851
(24) 등록일자 2024년08월28일

(51) 국제특허분류(Int. Cl.)
G06T 11/00 (2006.01) *G06T 15/04* (2011.01)
G06T 3/00 (2024.01)
(52) CPC특허분류
G06T 11/001 (2024.01)
G06T 15/04 (2013.01)
(21) 출원번호 10-2016-0174769
(22) 출원일자 2016년12월20일
심사청구일자 2021년12월20일
(65) 공개번호 10-2018-0071767
(43) 공개일자 2018년06월28일
(56) 선행기술조사문헌
US20050017983 A1
JP2006244426 A
US06975319 B

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
강석
경기도 용인시 수지구 진산로66번길 10, 523동 1303호(풍덕천동, 진산마을삼성래미안5차아파트)
(74) 대리인
리앤목특허법인

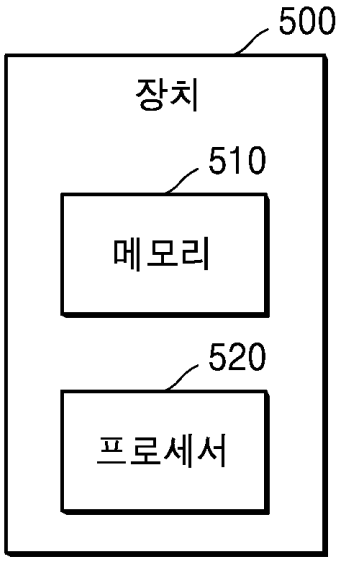
전체 청구항 수 : 총 11 항 심사관 : 이병우

(54) 발명의 명칭 **큐브 맵을 텍스처링하기 위한 LOD(level of detail)를 결정하는 방법 및 장치**

(57) 요 약

픽셀들이 큐브 맵의 서로 다른 면에 대응되는 경우, 픽셀들의 좌표를 보정하고, 보정된 좌표를 이용하여, 픽셀들의 변화율(derivative)을 계산하고, 계산된 변화율에 기초하여, 픽셀들에 큐브 맵을 텍스처링 하기 위한 LOD를 결정하는 방법 및 이를 위한 장치를 제공한다.

대 표 도 - 도5



(52) CPC특허분류
G06T 3/06 (2024.01)
G06T 2219/016 (2013.01)

명 세 서

청구범위

청구항 1

큐브 맵(cube map)을 텍스처링하기 위한 LOD(Level Of Detail)를 결정하는 방법에 있어서,

스크린 스페이스의 제1 및 제2 픽셀의 좌표들을 결정하는 단계 - 상기 제1 및 제2 픽셀은 각각 제1 큐브 맵 좌표들 및 제2 큐브 맵 좌표들에 위치한 상기 큐브 맵의 제1 및 제2 큐브 맵 픽셀들 각각에 매핑됨 -;

상기 제1 및 제2 픽셀들이 각각 상기 큐브 맵의 제1 및 제2 면들에 투영되는 경우 상기 제2 큐브 맵 픽셀의 좌표를 보정하고, 상기 제1 면을 큐브 맵 픽셀 좌표들이 s 축 성분(component) 및 t 축 성분으로 정의된 참조 면(reference face)으로 지정하는 단계 - 상기 t 축은 상기 s 축에 직교함 -;

상기 보정된 좌표를 이용하여 상기 제1 큐브 맵 픽셀과 상기 제2 큐브 맵 픽셀 사이의 상기 s축 및 상기 t 축의 성분 거리들(component distances)을 각각 계산함으로써, 상기 제1 픽셀 및 상기 제2 픽셀의 변화율(derivative) $(\partial s/\partial x)_{p01}$ 및 $(\partial t/\partial x)_{p01}$를 계산하는 단계 - 여기서, x는 상기 스크린 스페이스의 좌표축을 나타내고, p01은 상기 제 1 픽셀과 상기 제 2 픽셀 사이의 관계를 나타냄 -;

상기 제1 큐브 맵 픽셀과 제 3 픽셀이 투영된 제3 큐브 맵 픽셀 사이의 상기 s축 및 상기 t 축의 성분 거리들을 계산함으로써 상기 스크린 스페이스의 상기 제1 픽셀 및 상기 제3 픽셀의 변화율 $(\partial s/\partial x)_{p02}$ 및 $(\partial t/\partial x)_{p02}$를 계산하는 단계 - 여기서 p02는 상기 제1 픽셀과 상기 제3 픽셀 사이의 관계를 나타냄 -; 및

상기 계산된 변화율들에 기초하여 하기 수학식에 따라 상기 큐브 맵을 텍스처링하기 위한 LOD를 결정하는 단계를 포함하고,

$$\lambda = \log_2\left[MAX\left\{\sqrt{(\partial s/\partial x)_{p01}{}^2 + (\partial t/\partial x)_{p01}{}^2}, \sqrt{(\partial s/\partial x)_{p02}{}^2 + (\partial t/\partial x)_{p02}{}^2}\right\}\right]$$

상기 λ는 LOD의 값인, 방법.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 제2 큐브 맵 픽셀의 좌표의 보정은,

상기 큐브 맵 픽셀들의 좌표에 요구되는 보정 연산에 대한 정보를 포함하는 좌표 보정 테이블에 기초하여, 상기 큐브 맵 픽셀들의 좌표를 보정함으로써 수행되는, 방법.

청구항 4

제 1 항에 있어서,

상기 제2 큐브 맵 픽셀의 좌표의 보정은,

상기 제 1 큐브 맵 픽셀의 좌표에 대응하는 상기 제1 면의 원점 및 좌표축을 기준으로, 좌표값을 조정하는 덧셈 연산 및 좌표값의 부호를 반전시키는 반전 연산 중 적어도 하나를 상기 제2 큐브 맵 픽셀의 좌표에 대해 수행하는 것을 포함하는, 방법.

청구항 5

삭제

청구항 6

제 1 항에 있어서,

상기 큐브 맵은,

정규화된 크기를 갖는 6개의 정사각형 면(face)들로 구성된 정육면체의 텍스쳐인, 방법.

청구항 7

제 1 항에 있어서,

상기 제2 큐브 맵 픽셀의 좌표의 보정은,

스크린 스페이스 상에서의 상기 제 1 및 제 2 픽셀들의 방향 벡터를 기준으로, 상기 제 1 및 제 2 픽셀들 각각이 상기 큐브 맵의 서로 다른 면(face)에 대응되는지 여부를 판단하는 것을 포함하는, 방법.

청구항 8

큐브 맵(cube map)을 텍스처링하기 위한 LOD(Level Of Detail)를 결정하는 장치에 있어서,

컴퓨터 실행가능 명령어(computer executable instruction)를 저장하는 메모리; 및

상기 컴퓨터 실행가능 명령어를 실행하는 프로세서를 포함하고,

상기 프로세서는

스크린 스페이스의 제1 및 제2 픽셀의 좌표들을 결정하는 단계 - 상기 제1 및 제2 픽셀은 각각 제1 큐브 맵 좌표들 및 제2 큐브 맵 좌표들에 위치한 상기 큐브 맵의 제1 및 제2 큐브 맵 픽셀들 각각에 매핑됨 -;

상기 제1 및 제2 픽셀들이 각각 상기 큐브 맵의 제1 및 제2 면들에 투영되는 경우 상기 제2 큐브 맵 픽셀의 좌표를 보정하고, 상기 제1 면을 큐브 맵 픽셀 좌표들이 s 축 성분(component) 및 t 축 성분으로 정의된 참조 면(reference face)으로 지정하는 단계 - 상기 t 축은 상기 s 축에 직교함 -;

상기 보정된 좌표를 이용하여 상기 제1 큐브 맵 픽셀과 상기 제2 큐브 맵 픽셀 사이의 상기 s축 및 상기 t 축의 성분 거리들(component distances)을 각각 계산함으로써, 상기 제1 픽셀 및 상기 제2 픽셀의 변화율(derivative) $(\partial s/\partial x)_{p01}$ 및 $(\partial t/\partial x)_{p01}$를 계산하는 단계 - 여기서, x는 상기 스크린 스페이스의 좌표축을 나타내고, p01은 상기 제 1 픽셀과 상기 제 2 픽셀 사이의 관계를 나타냄 -;

상기 제1 큐브 맵 픽셀과 제 3 픽셀이 투영된 제3 큐브 맵 픽셀 사이의 상기 s축 및 상기 t 축의 성분 거리들을 계산함으로써 상기 스크린 스페이스의 상기 제1 픽셀 및 상기 제3 픽셀의 변화율 $(\partial s/\partial x)_{p02}$ 및 $(\partial t/\partial x)_{p02}$를 계산하는 단계 - 여기서 p02는 상기 제1 픽셀과 상기 제3 픽셀 사이의 관계를 나타냄 -; 및

상기 계산된 변화율들에 기초하여 하기 수학식에 따라 상기 큐브 맵을 텍스처링하기 위한 LOD를 결정하는 단계;를 수행하는 상기 컴퓨터 실행가능 명령어를 실행하고,

$$\lambda = \log_2\left[MAX\left\{\sqrt{(\partial s/\partial x)_{p01}{}^2 + (\partial t/\partial x)_{p01}{}^2}, \sqrt{(\partial s/\partial x)_{p02}{}^2 + (\partial t/\partial x)_{p02}{}^2}\right\}\right]$$

상기 λ는 LOD의 값인, 장치.

청구항 9

삭제

청구항 10

제 8 항에 있어서,

상기 프로세서는,

상기 큐브 맵 픽셀들의 좌표에 요구되는 보정 연산에 대한 정보를 포함하는 좌표 보정 테이블에 기초하여, 상기 큐브 맵 픽셀들의 좌표를 보정하는, 장치.

청구항 11

제 8 항에 있어서,

상기 프로세서는,

상기 제 1 큐브 맵 픽셀의 좌표에 대응하는 상기 제1 면의 원점 및 좌표축을 기준으로, 좌표값을 조정하는 덧셈 연산 및 좌표값의 부호를 반전시키는 반전 연산 중 적어도 하나를 상기 제2 큐브 맵 픽셀의 좌표에 대해 수행하는, 장치.

청구항 12

삭제

청구항 13

제 8 항에 있어서,

상기 프로세서는,

스크린 스페이스 상에서의 상기 제 1 및 제 2 픽셀들의 방향 벡터를 기준으로, 상기 제 1 및 제 2 픽셀들 각각이 상기 큐브 맵의 서로 다른 면(face)에 대응되는지 여부를 판단하는, 장치.

청구항 14

제 8 항에 있어서,

상기 프로세서는,

상기 제 1 및 제 2 픽셀들에 대해 상기 큐브 맵 상에서의 상기 좌표들을 결정하는 제 1 연산부;

상기 제 2 큐브 맵 픽셀의 좌표들을 보정하는 제 2 연산부;

상기 큐브 맵 상에서 상기 제 1 픽셀 및 상기 제 2 픽셀의 상기 변화율 및 상기 제 1 픽셀 및 상기 제 3 픽셀의 상기 변화율을 계산하는 제 3 연산부; 및

상기 큐브 맵을 텍스처링 하기 위한 상기 LOD를 결정하는 제 4 연산부를 포함하는, 장치.

청구항 15

제 1 항, 제 3 항, 제 4 항, 제 6 항 및 제 7 항 중에 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

발명의 설명

기 술 분 야

[0001] 본 개시는 큐브 맵을 텍스처링하기 위한 LOD를 결정하는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 3차원 그래픽 시스템에서 보다 현실감 있는 영상을 얻기 위한 방법의 일환으로 텍스처링 또는 텍스쳐 맵핑(texture mapping) 기술이 활용되고 있다. 텍스처링 또는 텍스쳐 맵핑이란, 3차원 물체의 표면에 질감을 주기 위해 2차원 이미지를 3차원 물체 표면에 입히는 것을 의미한다. 여기서, 텍스쳐는 2차원 이미지를 의미하고, 텍

스쳐 내의 각 점들은 텍셀(texel)로서 스크린 스페이스 상의 픽셀에 대응한다. 3차원 그래픽스 파이프라인이 수행되면서 2차원 스크린 스페이스의 각 픽셀에 대응되는 3차원 스페이스 상의 오브젝트 표면이 결정되면, 오브젝트 표면에 해당하는 텍스쳐 좌표를 갖는 텍셀들이 계산되고 이에 따라 픽셀과 텍셀 간의 텍스쳐 매핑이 수행될 수 있다.

발명의 내용

해결하려는 과제

[0003] 큐브 맵을 텍스처링하기 위한 LOD를 결정하는 방법 및 장치를 제공하는데 있다. 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 이하의 실시예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제의 해결 수단

[0004] 일 측면에 따라, 큐브 맵(cube map)을 텍스처링하기 위한 LOD(Level Of Detail)를 결정하는 방법은, 인접하는 픽셀들에 대해 큐브 맵 상에서의 좌표를 결정하는 단계; 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 좌표를 보정하는 단계; 보정된 좌표를 이용하여, 큐브 맵 상에서 픽셀들의 변화율(derivative)을 계산하는 단계; 및 계산된 변화율에 기초하여, 픽셀들에 큐브 맵을 텍스처링 하기 위한 LOD를 결정하는 단계;를 포함할 수 있다.

[0005] 또한, 보정하는 단계는, 픽셀들 중 기준 픽셀에 대응되는 면(face)의 원점 및 좌표축을 기준으로, 픽셀들 중 다른 픽셀의 좌표를 보정할 수 있다.

[0006] 또한, 보정하는 단계는, 픽셀들의 좌표에 요구되는 보정 연산에 대한 정보를 포함하는 좌표 보정 테이블에 기초하여, 픽셀들의 좌표를 보정할 수 있다.

[0007] 또한, 픽셀들의 변화율을 계산하는 단계는, 큐브 맵의 좌표축 상에서 픽셀들 간의 좌표값 차이(difference)를, 픽셀들의 변화율로써, 계산할 수 있다.

[0008] 또한, 큐브 맵은, 정규화된 크기를 갖는 6개의 정사각형 면(face)들로 구성된 정육면체의 텍스쳐일 수 있다.

[0009] 또한, 보정하는 단계는, 스크린 스페이스 상에서의 픽셀들의 방향 벡터를 기준으로, 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는지 여부를 판단하는 단계;를 포함할 수 있다.

[0010] 다른 측면에 따른, 큐브 맵(cube map)을 텍스처링하기 위한 LOD(Level Of Detail)를 결정하는 장치는, 컴퓨터 실행가능 명령어(computer executable instruction)를 저장하는 메모리; 및 컴퓨터 실행가능 명령어를 실행함으로써, 인접하는 픽셀들에 대해 큐브 맵 상에서의 좌표를 결정하고, 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 좌표를 보정하고, 보정된 좌표를 이용하여, 큐브 맵 상에서 픽셀들의 변화율(derivative)을 계산하고, 계산된 변화율에 기초하여, 픽셀들에 큐브 맵을 텍스처링 하기 위한 LOD를 결정하는, 프로세서를 포함할 수 있다.,

[0011] 또 다른 측면에 따라, 큐브 맵(cube map)을 텍스처링하기 위한 LOD(Level Of Detail)를 결정하는 방법을 구현하기 위한 프로그램이 기록된 컴퓨터로 판독 가능한 기록 매체가 제공된다.

발명의 효과

[0012] 본 실시예들에 따르면, 픽셀들의 큐브 맵 상의 보정된 좌표로부터 픽셀들의 변화율(derivative)을 직접 계산할 수 있는바, 변화율을 계산하는 데 필요한 연산기의 개수를 줄일 수 있고, 이에 따라 장치의 면적과 소비 전력을 줄일 수 있다.

도면의 간단한 설명

[0013] 도 1은 일 실시예에 따른 그래픽 처리 장치를 나타낸 도면이다.

도 2는 그래픽 처리 장치가 3차원 그래픽스를 처리하는 과정을 설명하는 도면이다.

도 3은 텍스쳐 맵핑을 설명하기 위한 도면이다.

도 4는 밉맵을 설명하기 위한 도면이다.

도 5는 일 실시예에 따라, 큐브 맵을 텍스쳐링하기 위한 LOD를 결정하는 장치를 나타낸다.

도 6은 스크린 스페이스 상의 좌표계와 큐브 맵 상의 좌표계의 일 실시예를 나타낸다.

도 7은 스크린 스페이스 상의 픽셀에 대해 큐브 맵 상의 좌표를 결정하는 실시예를 나타낸다.

도 8은 스크린 스페이스 상의 픽셀에 대해 큐브 맵 상의 좌표를 결정하는 구체적인 실시예를 나타낸다.

도 9는 좌표 보정 테이블을 이용하여 픽셀들의 변화율을 계산하는 실시예를 나타낸다.

도 10은 픽셀들의 좌표를 보정하여 픽셀들의 변화율을 계산하는 실시예를 나타낸다.

도 11은 픽셀의 좌표를 보정하여 픽셀들의 변화율을 계산하는 다른 실시예를 나타낸다.

도 12는 일 예에 따라, 큐브 맵을 텍스쳐링하기 위한 LOD를 결정하는 방법을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0014] 이하 첨부된 도면을 참조하면서 오로지 예시를 위한 실시예를 상세히 설명하기로 한다. 하기 실시예는 기술적 내용을 구체화하기 위한 것일 뿐 권리 범위를 제한하거나 한정하는 것이 아님은 물론이다. 상세한 설명 및 실시예로부터 해당 기술분야의 전문가가 용이하게 유추할 수 있는 것은 권리범위에 속하는 것으로 해석된다.

[0015] 본 명세서에서 사용되는 '구성된다' 또는 '포함한다' 등의 용어는 명세서 상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.

[0016] 또한, 본 명세서에서 사용되는 '제 1' 또는 '제 2' 등과 같이 서수를 포함하는 용어는 다양한 구성 요소들을 설명하는데 사용할 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만 사용된다.

[0017] 본 실시예들은 큐브 맵이라는 텍스쳐를 처리하는 방법 및 장치에 관한 것으로서 이하의 실시예들이 속하는 기술분야에서 통상의 지식을 가진 자에게 널리 알려져 있는 사항들에 관해서는 자세한 설명을 생략한다.

[0018] 도 1은 일 실시예에 따른 그래픽 처리 장치를 나타낸 도면이다.

[0019] 도 1에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 관련 기술 분야에서 통상의 지식을 가진 자라면 이해할 수 있다.

[0020] 도 1을 참고하면, 그래픽 처리 장치(100)는 래스터라이저(rasterizor)(110), 쉐이더 코어(shader core)(120), 텍스쳐 처리 장치(texture processing unit)(130), 픽셀 처리 장치(pixel processing unit)(140), 타일 버퍼(tile buffer)(150) 등을 포함할 수 있다. 그래픽 처리 장치(100)는 버스(BUS)(300)를 통하여 외부의 메모리(200)와 데이터를 송수신할 수 있다.

[0021] 도 1에 도시된 그래픽 처리 장치(100)는 3차원 그래픽스를 처리하는 장치로서, 타일에 기초한 렌더링(tile based rendering, TBR) 방식을 사용할 수 있다. 다시 말해서, 그래픽 처리 장치(100)는 하나의 프레임에 해당하는 3차원 그래픽스를 생성하기 위해서, 일정한 크기로 분할된 복수 개의 타일들을 래스터라이저(110), 쉐이더 코어(120), 픽셀 처리 장치(140)를 거치도록 하여, 처리 결과를 타일 버퍼(150)에 저장할 수 있다. 그래픽 처리 장치(100)는 프레임을 구성하는 모든 타일들에 대해서, 래스터라이저(110), 쉐이더 코어(120), 및 픽셀 처리 장치(140)로 구성되는 채널을 복수 개 이용하여, 병렬 처리할 수 있다. 그래픽 처리 장치(100)는 하나의 프레임에 해당하는 복수 개의 타일들이 처리되면, 타일 버퍼(150)에 저장된 처리 결과를 메모리(200)의 프레임 버퍼(미도시)로 전송할 수 있다.

[0022] 래스터라이저(110)는 기하 변환 과정을 거쳐 버텍스 쉐이더로부터 생성된 프리미티브에 대해 래스터화(rasterization)를 수행할 수 있다.

[0023] 쉐이더 코어(120)는 래스터라이저(110)로부터 래스터화된 프리미티브를 전달받아, 픽셀 쉐이딩을 수행할 수 있다. 쉐이더 코어(120)는 래스터화를 거쳐 생성된 프리미티브의 프레그먼트들을 포함하는 타일들에 대하여, 타일을 구성하는 모든 픽셀들의 색상을 결정하는 픽셀 쉐이딩을 수행할 수 있다. 쉐이더 코어(120)는 픽셀 쉐이딩 과정에서 입체적이고 현실감있는 3차원 그래픽스를 생성하기 위해 텍스쳐를 이용하여 생성된 픽셀 값을 사용할

수 있다.

[0024] 쉐이더 코어(120)는 픽셀 쉐이더(pixel shader)를 포함할 수 있다. 또한 쉐이더 코어(120)는 버텍스 쉐이더(vertex shader)를 더 포함한 형태이거나, 버텍스 쉐이더와 픽셀 쉐이더가 통합된 형태의 통합 쉐이더일 수도 있다. 쉐이더 코어(120)가 버텍스 쉐이더의 기능을 수행할 수 있는 경우, 오브젝트를 나타내는 프리미티브(primitive)를 생성하여 래스터라이저(110)에 전달할 수 있다.

[0025] 쉐이더 코어(120)가 원하는 픽셀에 대응되는 픽셀 값을 전달해 줄 것을 텍스쳐 처리 장치(130)에 요청하면, 텍스쳐 처리 장치(130)는 미리 준비된 텍스쳐를 처리하여 생성된 픽셀 값을 전달해 줄 수 있다. 텍스쳐는 텍스쳐 처리 장치(130) 내부 또는 외부의 소정의 공간 또는 그래픽 처리 장치(100) 외부의 메모리(200)에 저장되어 있을 수 있다. 텍스쳐 처리 장치(130)는 쉐이더 코어(120)에서 요청한 픽셀 값을 생성하는데 이용되는 텍스쳐가 텍스쳐 처리 장치(130) 내부의 소정의 공간에 없는 경우, 텍스쳐 처리 장치(130) 외부의 공간 또는 메모리(200)로부터 텍스쳐를 가져와 사용할 수 있다.

[0026] 픽셀 처리 장치(140)는 하나의 타일 내의 같은 위치에 대응되는 픽셀들에 대하여, 깊이 테스트 등과 같은 과정을 거쳐, 최종적으로 표시될 픽셀 값을 결정하여 하나의 타일에 해당하는 모든 픽셀 값들을 결정할 수 있다.

[0027] 타일 버퍼(150)는 픽셀 처리 장치(140)로부터 전달된 하나의 타일에 해당하는 모든 픽셀 값들을 저장할 수 있다. 하나의 프레임을 구성하는 모든 타일들에 대한 그래픽 처리 과정이 완료되면, 타일 버퍼(150)에 저장된 처리 결과가 메모리(200)의 프레임 버퍼로 전달될 수 있다.

[0028] 도 2는 그래픽 처리 장치가 3차원 그래픽스를 처리하는 과정을 설명하는 도면이다.

[0029] 3차원 그래픽스를 처리하는 과정은 크게 기하변환, 래스터화, 픽셀 쉐이딩의 3단계로 나눌 수 있으며, 이하 도 2를 참조하여, 보다 세부적인 과정에 대해 설명한다. 구체적으로, 도 2는 단계 11 내지 단계 18을 통해 3차원 그래픽스를 처리하는 과정을 나타낸다.

[0030] 단계 11은 버텍스들(vertices)을 생성하는 단계이다. 버텍스들은 3차원 그래픽스에 포함된 객체(object)들을 나타내기 위해 생성된다.

[0031] 단계 12는 생성된 버텍스들을 쉐이딩(shading)하는 단계이다. 버텍스 쉐이더(vertex shader)는 단계 11에서 생성된 버텍스들의 위치를 지정함으로써, 버텍스들에 대한 쉐이딩을 수행할 수 있다.

[0032] 단계 13은 프리미티브(primitive)들을 생성하는 단계이다. 프리미티브는 하나 이상의 버텍스들을 이용하여 형성되는 점, 선, 다각형(polygon)등을 의미한다. 예를 들어, 프리미티브는 3개의 버텍스들이 연결된 삼각형일 수 있다.

[0033] 단계 14는 프리미티브를 래스터화(rasterization)하는 단계이다. 프리미티브를 래스터화하는 것은 프리미티브를 프레그먼트들(fragments)로 분할하는 것을 의미한다. 프레그먼트는 프리미티브에 대한 그래픽스 처리를 수행하기 위한 기본 단위일 수 있다. 프리미티브는 버텍스에 대한 정보만을 포함하므로, 래스터라이징을 통해 버텍스와 버텍스 사이의 프레그먼트들이 생성됨으로써, 3차원 그래픽스 처리가 수행될 수 있다.

[0034] 단계 15는 픽셀들을 쉐이딩하는 단계이다. 래스터화에 의해 생성된, 프리미티브를 구성하는 프레그먼트들은 픽셀들이 될 수 있다. 당해 분야에서, 프레그먼트와 픽셀이란 용어는 경우에 따라 혼용되어 사용되기도 한다. 예를 들어, 픽셀 쉐이더는 프레그먼트 쉐이더라고 호칭될 수 있다. 일반적으로, 프리미티브를 구성하는 그래픽 처리의 기본 단위를 프레그먼트라고 부르고, 이후, 픽셀 쉐이딩부터 그래픽 처리의 기본 단위를 픽셀이라 지칭할 수 있다. 픽셀 쉐이딩에 의하여 픽셀들의 값들, 속성들 등(예를 들어 픽셀의 색)이 결정될 수 있다.

[0035] 단계 16은 픽셀의 색을 결정하기 위한 텍스처링(texturing) 단계이다. 텍스처링은 미리 준비된 이미지인 텍스쳐를 이용하여 픽셀의 색을 결정하는 과정이다. 이때 다양한 색상과 패턴의 모습을 표현하기 위해서 각각의 픽셀의 색상을 계산하여 결정하는 것은 그래픽 처리에 필요한 데이터 연산량과 그래픽 처리 시간을 증가시키므로, 그래픽 처리 장치는 미리 준비된 텍스쳐를 이용하여 픽셀의 색상을 결정할 수 있다.

[0036] 단계 17은 테스트 및 믹싱(testing and mixing) 단계이다. 깊이 테스트(depth test), 컬링(curling), 클리핑(clipping) 등을 통해 최종적으로 표시될 픽셀 값들이 결정된다.

[0037] 단계 18은 11 내지 단계 17을 통해 생성된 프레임을 프레임 버퍼(frame buffer)에 저장하고, 프레임 버퍼에 저장된 프레임을 디스플레이 장치를 통해 표시하는 단계이다.

[0038] 도 2에서 설명된 3차원 그래픽스를 처리하는 과정은 개괄적인 것으로서, 보다 세부적인 과정들에 대해서는 당해 기술분야의 통상의 기술자에게 자명하다.

[0039] 도 3은 텍스쳐 맵핑을 설명하기 위한 도면이다.

[0040] 도 3을 참고하면, 래스터라이징에 의하여 스크린 스페이스 상의 픽셀들(301)이 생성된 경우, 텍스쳐 처리 장치(130)는 픽셀들(301)에 맵핑될 텍스쳐(302)를 결정할 수 있다. 이때 텍스쳐(302)는 3차원 객체 표면의 색상, 질감 및 패턴 등에 대한 정보를 갖는 이미지로 정의되며, 텍스쳐 공간 상의 텍셀(texel) 단위로 구성된다.

[0041] 한편, 객체의 크기는 스크린 스페이스 상에서 연속적으로 변하기 때문에, 모든 픽셀들(301)에 대응되는 텍스쳐(302)들을 미리 준비하는 것은 어렵다. 따라서, 텍스쳐 처리 장치(130)는 한 장의 텍스쳐 또는 복수의 텍스쳐들을 이용한 보간(interpolation) 작업을 통해, 픽셀들(301)의 값을 추정하는, 텍스쳐 필터링을 수행할 수 있다.

[0042] 한편, 텍스쳐 처리 장치(130)는 스크린 스페이스 상의 인접하는 픽셀들에 매핑될 큐브 맵(cube map)을 결정할 수 있다. 큐브 맵은 6개의 면(face)들로 구성된 정육면체의 텍스쳐를 의미할 수 있다. 예를 들어, 큐브 맵은, 특정 시점(viewpont)에서 바라본 주변 환경을 표현하는 6개의 면들로 구성될 수 있다. 또한, 큐브 맵은 그래픽 처리에 있어 다양한 효과(effect)를 위해 이용될 수 있다. 예를 들어, 큐브 맵은 렌더링 하고자 하는 장면 내에서의 반사(reflection) 및 광 효과(lighting effect)를 위한 소스 데이터로 이용될 수 있다. 따라서, 텍스쳐 처리 장치(130)는 미리 준비된 큐브 맵을 이용하여 텍스쳐링을 수행할 수 있다.

[0043] 도 4는 밉맵을 설명하기 위한 도면이다.

[0044] 텍스쳐 처리 장치(130)는 스크린 스페이스 상에서 변화되는 객체의 크기에 적응적으로 대응할 수 있도록, 미리 준비된 텍스쳐를 이용할 수 있다. 이때, 기본 텍스쳐와 이를 축소시킨 텍스쳐들로 이루어진 비트맵 이미지의 집합을 밉맵(mipmap)이라고 한다.

[0045] 이때, 다른 레벨의 밉맵은 해상도가 상이할 수 있다. 도 4를 참고하면, 레벨 0 밉맵(410)은 텍스쳐들 중에서 해상도가 제일 높은 텍스쳐로, 기본 텍스쳐를 의미할 수 있다. 레벨 0 밉맵(410)을 표현하기 위해, 8×8 텍셀들이 필요하다. 또한, 기본 텍스쳐의 크기보다 1/4 축소된 밉맵은 레벨 1 밉맵(411)이다. 레벨 1 밉맵(411)을 표현하기 위해, 4×4 텍셀들이 필요하다. 또한, 기본 텍스쳐의 크기보다 1/16 축소된 밉맵은 레벨 2 밉맵(412)일 수 있다. 레벨 2 밉맵(412)을 표현하기 위해, 2×2 텍셀들이 필요하다.

[0046] 한편, 시점(view point)과 픽셀간 거리가 변함에 따라, 픽셀에 맵핑되는 텍스쳐가 달라질 수 있다. 예를 들어, 도로의 타일을 스크린 공간에 표시할 때 시점에서 가까운 타일에 대응하는 픽셀들에 맵핑되는 텍스쳐는 해상도가 높은 레벨 0 밉맵(410)일 수 있다. 또한, 시점에서 멀리 떨어진 도로의 타일에 대응하는 픽셀들에 맵핑될 텍스쳐는 해상도가 낮은 레벨 2 밉맵(412)일 수 있다. 즉, 복수의 밉맵을 이용할 경우, 먼 거리에 있는 객체는 낮은 해상도로 표현되기 때문에, 그래픽 처리 장치는 3차원 그래픽을 자연스럽게 표현할 수 있다. 더불어, 상이한 텍스쳐들이 맵핑된 픽셀들의 경계면에서 텍스쳐 필터링이 수행되면, 3차원 그래픽의 품질이 향상될 수 있다.

[0047] 한편, 도 4에서는 2차원의 텍스쳐를 기준으로 복수의 밉맵들을 설명하였지만, 6개의 텍스쳐 면들로 구성된 큐브 맵에도 복수의 밉맵들이 준비되고 이용될 수 있다.

[0048] 또한, 스크린 스페이스 상의 픽셀들에 맵핑시킬 텍스쳐들은, 그래픽의 LOD(level of detail)에 기초하여 결정될 수 있다. 여기서 LOD는 3차원 그래픽 이미지를 표현하는 정밀도를 단계화한 것을 의미한다. 텍스쳐 처리 장치(130)는 픽셀 정보를 수신한 후 그래픽의 LOD 값을 결정하고, 결정된 LOD 값에 따라 텍스쳐 필터링을 수행할 때 필요한 밉맵들을 결정할 수 있다. 예를 들어, 텍스쳐 처리 장치는 2×2 픽셀들로 구성된 쿼드(quad) 단위로 그래픽의 LOD 값을 계산할 수 있으나, LOD 값을 계산할 때 필요한 픽셀의 수는 이에 제한되지 않는다.

[0049] 마찬가지로, 스크린 스페이스 상의 픽셀들에 매핑시킬 큐브맵은, 그래픽의 LOD에 기초하여 결정될 수 있다. 따라서, 텍스쳐 처리 장치(130)는 픽셀 정보를 수신한 후 그래픽의 LOD 값을 결정하고, 결정된 LOD 값에 따라 텍스쳐 필터링을 수행할 때 필요한 밉맵들을 결정할 수 있다.

[0050] 도 5는 일 실시예에 따라, 큐브 맵을 텍스쳐링하기 위한 LOD를 결정하는 장치를 나타낸다.

[0051] 장치(500)는 메모리(510) 및 프로세서(520)를 포함할 수 있다. 한편, 도 5에 도시된 장치(500)는 큐브 맵을 텍스쳐링하기 위한 LOD를 결정하는 실시예들과 관련된 구성요소들만이 도시되어 있다. 따라서, 도 5에 도시된 구성요소들 외에 다른 범용적인 구성요소들이 더 포함될 수 있음을 당해 기술분야의 통상의 기술자라면 이해할 수 있다. 또한, 일 실시예에 따라, 장치(500)는 도 1 의 텍스쳐 처리 장치(130)에 포함될 수 있으며, 텍스쳐 처리

장치(130) 내의 세부 구성들로 구현될 수 있으나, 이에 제한되지 않는다.

[0052] 메모리(510)는 컴퓨터 실행가능 명령어(computer executable instruction)를 저장할 수 있다.

[0053] 프로세서(520)는, 메모리(510)에 저장된 컴퓨터 실행가능 명령어를 실행함으로써, 인접하는 픽셀들에 대해 큐브 맵 상에서의 좌표를 결정할 수 있다. 다시 말해, 프로세서(520)는 인접하는 픽셀들 각각의 스크린 스페이스 상의 좌표를, 텍스쳐 스페이스인 큐브 맵 상의 좌표로 변환할 수 있다. 일 실시예에 따라, 프로세서(520)는 스크린 스페이스 상에서 픽셀의 방향 벡터를 기준으로, 픽셀을 큐브 맵 상으로 투영(project)시킨 지점의 좌표를, 픽셀의 큐브 맵 상의 좌표로 결정할 수 있다. 이하 도 6 내지 8에서, 픽셀의 큐브 맵 상의 좌표를 결정하는 구체적인 실시예를 살펴보기로 한다.

[0054] 도 6은 스크린 스페이스 상의 좌표계와 큐브 맵 상의 좌표계의 일 실시예를 나타낸다.

[0055] 도 6에서, 스크린 스페이스는 서로 직교하는 x, y, 및 z축으로 구성된 3차원 좌표계를 가지며, 큐브 맵은 6개의 면(face)들 각각마다 원점을 기준으로 s_{face} 및 t_{face} 축으로 구성된 2차원 좌표계를 가질 수 있다. 구체적으로, 큐브 맵의 중심에 스크린 스페이스 상이 원점이 위치한다고 가정할 수 있고, 큐브 맵은, +x축과 교차하는 X+ face, -x축과 교차하는 X- face, +y축과 교차하는 Y+ face, -y축과 교차하는 Y- face, +z축과 교차하는 Z+ face 및 -z축과 교차하는 Z- face를 포함할 수 있으며, 6개의 면들 각각은 각 면의 원점을 기준으로 s_{face} 및 t_{face} 축으로 구성된 2차원 좌표계를 가질 수 있다.

[0056] 도 7은 스크린 스페이스 상의 픽셀에 대해 큐브 맵 상의 좌표를 결정하는 실시예를 나타낸다.

[0057] 프로세서(520)는 표(710) 및 수식(720)에 기초하여, 큐브 맵 상에서의 픽셀의 좌표를 결정할 수 있다.

[0058] 스크린 스페이스 상에서의 픽셀의 방향 벡터가 (r_x, r_y, r_z)(단, r_x, r_y, r_z 각각은 실수)인 경우, 프로세서(520)는 픽셀의 r_x, r_y, 및 r_z 각각의 크기 및 부호를 기준으로 Major Axis Direction을 결정할 수 있고, 이어서 Cube map face, s_c, t_c, 및 r_c를 결정할 수 있다.

[0059] 예를 들어, 픽셀의 r_x, r_y, 및 r_z 중 r_x가 크기가 가장 크고 부호가 양수인 경우, 프로세서(520)는 픽셀의 Major Axis Direction이 +x축이라고 결정할 수 있고, 이에 따라 픽셀이 큐브 맵 중 X+ face에 해당한다고 결정할 수 있고, 픽셀의 (s_c, t_c, r_c)를 $(-r_z, -r_y, r_x)$로 결정할 수 있다. 따라서, 프로세서(520)는 결정된 (s_c, t_c, r_c)를 이용하여, X+ face 상에서의 좌표 (s_{face}, t_{face})를 결정할 수 있다.

[0060] 마찬가지로, 픽셀의 r_x, r_y, 및 r_z 중 r_z가 크기가 가장 크고 부호가 음수인 경우, 프로세서(520)는 픽셀의 Major Axis Direction이 -z축이라고 결정할 수 있고, 이에 따라 픽셀이 큐브 맵 중 Z- face에 해당한다고 결정할 수 있고, 픽셀의 (s_c, t_c, r_c)를 $(-r_x, -r_y, r_z)$로 결정할 수 있다. 따라서, 프로세서(520)는 결정된 (s_c, t_c, r_c)를 이용하여, Z- face 상에서의 좌표 (s_{face}, t_{face})를 결정할 수 있다.

[0061] 도 8은 스크린 스페이스 상의 픽셀에 대해 큐브 맵 상의 좌표를 결정하는 구체적인 실시예를 나타낸다.

[0062] 프로세서(520)는 스크린 스페이스 상의 픽셀 P0와 P1에 대해 큐브 맵 상의 좌표를 결정할 수 있다. 다시 말해, 프로세서(520)는, 픽셀 P0와 P1을 큐브 맵 상으로 투영(project)시킨 지점의 좌표를 결정할 수 있다.

[0063] 도 8은 설명 및 이해의 편의를 위해 픽셀 P0 및 P1 각각의 y축 방향의 방향 벡터가 0인 경우를 기준으로 설명한다. 도 9에 도시되어 있듯이, 픽셀 P0의 스크린 스페이스 상의 좌표는 (P0.rx, 0, P0.rz)이고, 픽셀 P1의 스크린 스페이스 상의 좌표는 (P1.rx, 0, P1.rz)이다.

[0064] 프로세서(520)는 픽셀 P0의 방향 벡터 (P0.rx, 0, P0.rz)에 기초하여, 큐브 맵 상의 픽셀 P0의 좌표를, 큐브 맵의 X+ face 상의 $(P0.s_{face}, P0.t_{face})$로 결정할 수 있다. 예를 들어, 프로세서(520)는 도 7의 표(710) 및 수식(720)을 이용하여, 픽셀 P0의 좌표를 큐브 맵의 X+ face 상의 $(P0.s_{face}, P0.t_{face})$로 결정할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P1의 방향 벡터 (P1.rx, 0, P1.rz)에 기초하여, 큐브 맵 상의 픽셀 P1의 좌표를, 큐브 맵의 Z+ face 상의 $(P1.s_{face}, P1.t_{face})$로 결정할 수 있다. 예를 들어, 프로세서(520)는 도 7의 표(710) 및 수식(720)을 이용하여, 픽셀 P1의 좌표를 큐브 맵의 Z+ face 상의 $(P1.s_{face}, P1.t_{face})$로 결정할 수 있다.

[0065] 다시 도 5를 참조하면, 프로세서(520)는 컴퓨터 실행가능 명령어를 실행함으로써, 픽셀들 각각이 큐브 맵의 서

로 다른 면에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 좌표를 보정할 수 있다.

[0066] 먼저, 프로세서(520)는 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는지 여부를 판단할 수 있다. 다시 말해, 프로세서(520)는 스크린 스페이스 상의 픽셀들이 큐브 맵의 서로 다른 면들로 투영되는지 여부를 판단할 수 있다. 일 실시예에 따라, 프로세서(520)는 기 결정된 픽셀들 각각의 큐브 맵 상의 좌표가 큐브 맵의 서로 다른 면에 포함되는지 여부에 따라, 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는 지 여부를 판단할 수 있다.

[0067] 또한, 다른 실시예에 따라, 프로세서(520)는 픽셀들의 스크린 스페이스 상에서의 방향 벡터를 기준으로, 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는지 여부를 판단할 수 있다. 구체적으로, 프로세서(520)는 픽셀들의 스크린 스페이스 상에서의 방향 벡터에 기초하여 픽셀들의 Major Axis Direction을 결정할 수 있고, 픽셀들의 Major Axis Direction이 서로 다를 경우, 픽셀들이 큐브 맵의 서로 다른 면들에 투영된다고 판단할 수 있다. 도 8을 예를 들면, 프로세서(520)는 픽셀 P0의 방향 벡터 (P0.rx, 0, P0.rz)의 좌표축 각각의 크기 및 부호를 기준으로, P0의 Major Axis Direction이 +x축이라고 결정할 수 있고, 픽셀 P1의 방향 벡터 (P1.rx, 0, P1.rz)의 좌표축 각각의 크기 및 부호를 기준으로, 픽셀 P1의 Major Axis Direction이 -z축이라고 결정할 수 있다. 따라서, 프로세서(520)는 픽셀 P0 및 P1 각각이 큐브 맵의 서로 다른 면에 투영된다고 판단할 수 있다.

[0068] 다음으로, 프로세서(520)는 인접하는 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 큐브 맵 상의 좌표를 보정할 수 있다. 일 실시예에 따라, 프로세서(520)는 픽셀들 중 기준 픽셀에 대응되는 면(face)의 원점 및 좌표축을 기준으로, 픽셀들 중 다른 픽셀의 큐브 맵 상의 좌표를 보정할 수 있다. 예를 들어, 제 1 픽셀 및 제 2 픽셀이 서로 다른 면인 X+ face 및 Y- face에 투영되는 경우, 프로세서(520)는 기준 픽셀인 제 1 픽셀이 투영되는 X+ face에서의 s_{face}축 및 t_{face}축을 기준으로, 제 2 픽셀의 큐브 맵 상의 좌표를 보정할 수 있다.

[0069] 또한, 프로세서(520)는 인접하는 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들의 좌표에 요구되는 보정 연산에 대한 정보를 포함하는 좌표 보정 테이블에 기초하여, 픽셀들의 큐브 맵 상의 좌표를 보정할 수 있다. 좌표 보정 테이블은, 픽셀들 각각이 대응되는 면을 기준으로, 픽셀들 각각의 좌표가 어떤 연산을 통해 보정되어야 하는 지에 대한 정보를 포함할 수 있다.

[0070] 예를 들어, 프로세서(520)는 픽셀의 좌표에 대해, 덧셈 연산 및 반전 연산 중 적어도 하나를 수행하여, 픽셀의 좌표를 보정할 수 있다. 구체적인 예로, 프로세서(520)는 픽셀에 대해 +1, -1, 또는 +2에 대한 덧셈 연산을 수행 할 수 있으며, 픽셀의 부호를 반전시키는 반전 연산을 수행할 수 있다. 또한, 프로세서(520)는 픽셀의 좌표 (s_{face}, t_{face})의 각 좌표축 값을 (t_{face}, s_{face})와 같이 서로 교환(swap)시키는 연산을 수행할 수 있다.

[0071] 프로세서(520)는 컴퓨터 실행가능 명령어를 실행함으로써, 픽셀들의 큐브 맵 상의 보정된 좌표를 이용하여, 큐브 맵 상에서 픽셀들의 변화율(derivative)을 계산할 수 있다.

[0072] 픽셀들의 변화율(derivative)이란, 스크린 스페이스 상에서 픽셀들의 변화량 대비 큐브 맵 상에서 픽셀들의 변화량의 비율을 의미할 수 있다. 구체적으로, 픽셀들의 변화율(derivative)은, 스크린 스페이스의 좌표 축 상에서 픽셀들 간의 거리 대비, 큐브 맵의 좌표 축 상에서 픽셀들 간의 거리의 비율을 의미할 수 있다. 또한, 스크린 스페이스의 좌표 축 상에서 픽셀들의 거리가 일정한 경우, 프로세서(520)는 픽셀들의 큐브 맵의 좌표 축 상에서의 거리를 계산하여, 픽셀들의 변화율(derivative)을 계산할 수 있다. 또한, 프로세서(520)는 픽셀들의 변화율을 근사치로써 계산할 수 있는 바, 픽셀들의 큐브 맵의 좌표 축 상에서의 거리를 픽셀들의 변화율로 계산할 수 있다.

[0073] 구체적으로, 프로세서(520)는 큐브 맵의 s_{face}좌표축에 대한 픽셀들의 변화율로써, 큐브 맵의 s_{face}좌표축 상에서 픽셀들 간의 거리를 계산할 수 있다. 다시 말해, 프로세서(520)는 큐브 맵의 s_{face}좌표축 상에서 픽셀들 각각의 좌표값의 차이를 계산할 수 있다. 또한, 프로세서(520)는 큐브 맵의 t_{face}좌표축에 대한 픽셀들의 변화율로써, 큐브 맵의 t_{face}좌표축 상에서 픽셀들의 간의 거리를 계산할 수 있다. 다시 말해, 프로세서(520)는 큐브 맵의 t_{face}좌표축 상에서 픽셀들 각각의 좌표값의 차이를 계산할 수 있다.

[0074] 도 9는 좌표 보정 테이블을 이용하여 픽셀들의 변화율을 계산하는 실시예를 나타낸다.

[0075] 프로세서(520)는 좌표 보정 테이블(910,920)에 기초하여, 픽셀들의 큐브 맵 상의 좌표들을 보정할 수 있다.

[0076] 구체적으로, 좌표 보정 테이블(910)은 픽셀의 s_{face}좌표값을 보정하기 위한 테이블이고, 좌표 보정 테이블(920)은 t_{face}좌표값을 보정하기 위한 테이블이다. 또한 각 좌표 보정 테이블(910,920)에서, "Face of Reference"는 픽셀들 중 기준 픽셀이 투영되는 면을 나타내고, "Face of adjusted Pixel"은 좌표 보정하고자 하는 픽셀이 투영되는 면을 나타낸다.

[0077] 픽셀 P0와 픽셀 P1의 경우를 예로 들어 살펴보면, 픽셀 P0의 큐브 맵 상의 좌표가 X- face 상의 좌표 ($P0.s_{face}$, $P0.t_{face}$)이고, 픽셀 P1의 큐브 맵 상의 좌표가 Y+ face 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)이다.

[0078] 기준 픽셀이 픽셀 P0인 경우, 픽셀 P0의 "Face of Reference"와 "Face of adjusted Pixel"은 동일하게 X- face가 되므로, 좌표 보정 테이블(910,920)에 의해, 픽셀 P0의 조정된 좌표 ($P0.s_{adj}$, $P0.t_{adj}$)는 원래 좌표 ($P0.s_{face}$, $P0.t_{face}$)로 그대로 유지된다. 다만, 픽셀 P1인 경우, 픽셀 P1의"Face of Reference"는 X- face가 되고, 픽셀 P1의 "Face of adjusted Pixel"은 Y+ face가 되므로, 좌표 보정 테이블(910,920)에 의해, 픽셀 P1의 조정된 좌표 ($P1.s_{adj}$, $P1.t_{adj}$)는 ($P1.t_{face}$, $-P1.s_{face}$)가 된다.

[0079] 이어서, 프로세서(520)는 픽셀들의 보정된 좌표를 이용하여, 큐브 맵 상에서 픽셀들의 변화율을 계산할 수 있다. 즉, 프로세서(520)는 픽셀들의 보정된 좌표를 이용하여, 큐브 맵의 s_{face}좌표축 상에서 픽셀들의 변화율인 $\frac{\partial s_{face}}{\partial x}$ 를 계산할 수 있고, 픽셀들의 보정된 좌표를 이용하여, 큐브 맵의 t_{face}좌표축 상에서 픽셀들의 변화율인 $\frac{\partial t_{face}}{\partial x}$ 를 계산할 수 있다.

[0080] 구체적으로, 프로세서(520)는 스크린 스페이스의 좌표 축 상에서 픽셀들의 거리가 일정한 경우, 스크린 스페이스의 x좌표축 상에서 픽셀들의 변화율인 ∂x을 1로 결정할 수 있다. 따라서, 프로세서(520)는 큐브 맵의 s_{face}좌표축 상에서 픽셀들의 변화율인 $\frac{\partial s_{face}}{\partial x}$ 를, s_{face}좌표축 상에서 픽셀 P0와 픽셀 P1의 좌표값의 차이인 $\mathbf{P1.s_{adj} - P0.s_{adj}}$ 로 계산할 수 있다. 마찬가지로, 프로세서(520)는 큐브 맵의 t_{face}좌표축 상에서 픽셀들의 변화율인 $\frac{\partial t_{face}}{\partial x}$ 를, t_{face}좌표축 상에서 픽셀 P0와 픽셀 P1의 좌표값의 차이인 $\mathbf{P1.t_{adj} - P0.t_{adj}}$ 로 계산할 수 있다.

[0081] 도 10은 픽셀들의 좌표를 보정하여 픽셀들의 변화율을 계산하는 실시예를 나타낸다.

[0082] 도 10은 큐브 맵 상의 투영된 픽셀들 P0, P1, P2, 및 P3을 도시한다. 프로세서(520)는 픽셀들 P0, P1, P2, 및 P3이 큐브 맵 상의 서로 다른 면들에 포함되는 경우, 픽셀들 P0, P1, P2, 및 P3의 큐브 맵 상의 좌표를 보정할 수 있다.

[0083] 일 실시예에 따라, 프로세서(520)는 픽셀 P0 및 P1이 서로 다른 면인 Z+ face 및 Y+ face에 포함되어 있으므로, 기준 픽셀인 픽셀 P0이 투영되는 Z+ face의 원점 및 좌표축을 기준으로, 픽셀 P1의 큐브 맵 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)를 보정할 수 있다. 먼저, 프로세서(520)는 Z+ face 및 Y+ face의 s_{face}축의 경우 평행하고 방향이 일치하므로, 픽셀 P1의 s_{face}좌표값은 그대로 $P1.s_{face}$로 결정할 수 있다. 다음으로, 프로세서(520)는, 큐브 맵이 크기 1을 갖는 정규화된 큐브 맵이므로, Y+ face의 원점이 Z+ face의 원점보다 $-t_{face}$ 축 방향으로 1만큼 떨어져 있다고 판단할 수 있다. 따라서, 프로세서(520)는 Y+ face 상에서의 픽셀 P1의 t_{face}좌표값인 $P.t_{face}$를 Z+ face 상에서의 t_{face} 좌표값인 $-(1-P.t_{face})$로 보정할 수 있다. 따라서, 프로세서(520)는 픽셀 P1의 큐브 맵 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)를 좌표 ($P1.s_{face}$, $P1.t_{face}-1$)로 보정할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0와 다른 면에 투영되는 픽셀 P3의 큐브 맵 상의 좌표를 보정할 수 있다.

[0084] 이어서, 프로세서(520)는 픽셀 P0의 좌표 및 보정된 픽셀 P1의 좌표에 기초하여, 픽셀 P0 및 픽셀 P1의 변화율 $\frac{\partial s_{face}}{\partial x}$ 및 $\frac{\partial t_{face}}{\partial x}$ 을 계산할 수 있다. 구체적으로, 프로세서(520)는 픽셀 P0의 좌표 ($P0.s_{face}$, $P0.t_{face}$) 및 보정된 픽셀 P1의 좌표 ($P1.s_{face}$, $P1.t_{face}-1$)에 기초하여, 픽셀 P0 및 픽셀 P1의 변화율 $\frac{\partial s_{face}}{\partial x} = \mathbf{P1.s_{face} - P0.s_{face}}$

및 $\frac{\partial t_{face}}{\partial x} = (P1.t_{face} - 1) - P0.t_{face}$를 계산할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0 및 픽셀 P2의 변화율, 픽셀 P0 및 픽셀 P3의 변화율 등을 계산할 수 있다.

[0085] 다른 실시예에 따라, 프로세서(520)는 픽셀 P0 및 P1이 서로 다른 면인 Z+ face 및 Y+ face에 포함되어 있으므로, 좌표 보정 테이블(910,920)에 기초하여, 픽셀 P0의 큐브 맵 상의 좌표 ($P0.s_{face}$, $P0.t_{face}$) 및 픽셀 P1의 큐브 맵 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)를 보정할 수 있다. 구체적으로, 프로세서(520)는 기준 픽셀을 픽셀 P0로 판단하여, 픽셀 P0의"Face of Reference"와 "Face of adjusted Pixel"은 동일하게 Z+ face가 되므로, 좌표 보정 테이블(910,920)에 의해, 픽셀 P0의 보정된 좌표 ($P0.s_{adj}$, $P0.t_{adj}$)는 원래 좌표 ($P0.s_{face}$, $P0.t_{face}$)로 그대로 유지시킬 수 있다. 다음으로, 프로세서(520)는, 픽셀 P1의 경우,"Face of Reference"는 Z+ face가 되고, "Face of adjusted Pixel"은 Y+ face가 되므로, 좌표 보정 테이블(910,920)에 의해, 픽셀 P1의 조정된 좌표 ($P1.s_{adj}$, $P1.t_{adj}$)를 ($P1.s_{face}$, $P1.t_{face}$-1)로 보정할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0와 다른 면에 투영되는 픽셀 P3의 큐브 맵 상의 좌표를 보정할 수 있다.

[0086] 이어서, 프로세서(520)는 보정된 픽셀 P0의 좌표 ($P0.s_{face}$, $P0.t_{face}$) 및 보정된 픽셀 P1의 좌표 ($P1.s_{face}$, $P1.t_{face}$-1)에 기초하여, 픽셀 P0 및 픽셀 P1의 변화율 $\frac{\partial s_{face}}{\partial x} = P1.s_{adj} - P0.s_{adj} = P1.s_{face} - P0.s_{face}$ 및 $\frac{\partial t_{face}}{\partial x} = P1.t_{adj} - P0.t_{adj} = (P1.t_{face} - 1) - P0.t_{face}$를 계산할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0 및 픽셀 P2의 변화율, 픽셀 P0 및 픽셀 P3의 변화율 등을 계산할 수 있다.

[0087] 도 11은 픽셀의 좌표를 보정하여 픽셀들의 변화율을 계산하는 다른 실시예를 나타낸다.

[0088] 도 11은 큐브 맵 상의 투영된 픽셀들 P0, P1, P2, 및 P3을 도시한다. 프로세서(520)는 픽셀들 P0, P1, P2, 및 P3이 큐브 맵 상의 서로 다른 면들에 포함되는 경우, 픽셀들 P0, P1, P2, 및 P3의 큐브 맵 상의 좌표를 보정할 수 있다.

[0089] 일 실시예에 따라, 프로세서(520)는 픽셀 P0 및 P1이 서로 다른 면인 X+ face 및 Y- face에 포함되어 있으므로, 기준 픽셀인 픽셀 P0이 투영되는 X+ face의 원점 및 좌표축을 기준으로, 픽셀 P1의 큐브 맵 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)를 보정할 수 있다. 먼저, 프로세서(520)는 X+ face의 s_{face}축과 및 Y- face의 t_{face}축이 서로 평행하고 방향이 일치하므로, 픽셀 P1의 s_{face}좌표값을 $P1.t_{face}$로 결정할 수 있다. 다음으로, 프로세서(520)는, 큐브 맵이 크기 1을 갖는 정규화된 큐브 맵이므로, Y- face의 원점이 X+ face의 원점보다 t_{face}축 방향으로 2만큼 떨어져 있다고 판단할 수 있다. 따라서, 프로세서(520)는 Y- face 상에서의 픽셀 P1의 t_{face}좌표값인 $P1.t_{face}$를 X+ face 상에서의 t_{face} 좌표값인 (2-$P1.s_{face}$)로 보정할 수 있다. 따라서, 프로세서(520)는 픽셀 P1의 큐브 맵 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)를 좌표 ($P1.t_{face}$, 2-$P1.s_{face}$)로 보정할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0와 다른 면에 투영되는 픽셀 P2 및 P3의 큐브 맵 상의 좌표를 보정할 수 있다.

[0090] 이어서, 프로세서(520)는 픽셀 P0의 좌표 및 보정된 픽셀 P1의 좌표에 기초하여, 픽셀 P0 및 픽셀 P1의 변화율 $\frac{\partial s_{face}}{\partial x}$ 및 $\frac{\partial t_{face}}{\partial x}$을 계산할 수 있다. 구체적으로, 프로세서(520)는 픽셀 P0의 좌표 ($P0.s_{face}$, $P0.t_{face}$) 및 보정된 픽셀 P1의 좌표 ($P1.t_{face}$, 2-$P1.s_{face}$)에 기초하여, 픽셀 P0 및 픽셀 P1의 변화율 $\frac{\partial s_{face}}{\partial x} = P1.t_{face} - P0.s_{face}$ 및 $\frac{\partial t_{face}}{\partial x} = (2 - P1.s_{face}) - P0.t_{face}$를 계산할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0 및 픽셀 P1의 변화율, 픽셀 P0 및 픽셀 P3의 변화율 등을 계산할 수 있다.

[0091] 다른 실시예에 따라, 프로세서(520)는 픽셀 P0 및 P1이 서로 다른 면인 X+ face 및 Y- face에 포함되어 있으므로, 좌표 보정 테이블(910,920)에 기초하여, 픽셀 P0의 큐브 맵 상의 좌표 ($P0.s_{face}$, $P0.t_{face}$) 및 픽셀 P1의 큐브 맵 상의 좌표 ($P1.s_{face}$, $P1.t_{face}$)를 보정할 수 있다. 구체적으로, 프로세서(520)는 기준 픽셀을 픽셀 P0로 판단하여, 픽셀 P0의"Face of Reference"와 "Face of adjusted Pixel"은 동일하게 X+ face가 되므로, 좌표 보정 테이블(910,920)에 의해, 픽셀 P0의 조정된 좌표 ($P0.s_{adj}$, $P0.t_{adj}$)는 원래 좌표 ($P0.s_{face}$, $P0.t_{face}$)로 그대로 유지시킬 수 있다. 다음으로, 프로세서(520)는, 픽셀 P1의 경우,"Face of Reference"는 X+ face가 되고, "Face

of adjusted Pixel"은 Y- face가 되므로, 좌표 보정 테이블(910,920)에 의해, 픽셀 P1의 조정된 좌표 (P1.s_{adj}, P1.t_{adj})를 (P1.t_{face}, 2-P1.s_{face})로 보정할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0와 다른 면에 투영되는 픽셀 P2 및 픽셀 P3의 큐브 맵 상의 좌표를 보정할 수 있다.

[0092] 이어서, 프로세서(520)는 보정된 픽셀 P0의 좌표 (P0.s_{face}, P0.t_{face}) 및 보정된 픽셀 P1의 좌표 (P1.t_{face}, 2-P1.s_{face})에 기초하여, 픽셀 P0 및 픽셀 P1의 변화율 $\frac{\partial s_{face}}{\partial x} = P1.s_{adj} - P0.s_{adj} = P1.t_{face} - P0.s_{face}$ 및 $\frac{\partial t_{face}}{\partial x} = P1.t_{adj} - P0.t_{adj} = (2 - P1.s_{face}) - P0.t_{face}$를 계산할 수 있다. 마찬가지로, 프로세서(520)는 픽셀 P0 및 픽셀 P2의 변화율, 픽셀 P0 및 픽셀 P3의 변화율 등을 계산할 수 있다.

[0093] 다시 도 5를 참조하면, 프로세서(520)는 계산된 픽셀들의 변화율(derivative)에 기초하여, 픽셀들에 큐브 맵을 텍스처링 하기 위한 LOD를 결정할 수 있다. LOD는 예를 들어, 다음과 같은 수학식 1을 이용하여 계산될 수 있으나, 이에 한정되지 않는다.

수학식 1

[0094]
$$\lambda = \log_2 \left[MAX \left\{ \sqrt{(\partial s/\partial x)_{P01}^2 + (\partial t/\partial x)_{P01}^2}, \sqrt{(\partial s/\partial x)_{P02}^2 + (\partial t/\partial x)_{P02}^2} \right\} \right]$$

[0095] λ는 LOD의 값이고, $(\partial s/\partial x)_{P01}$은 픽셀 P0 및 P1의 s_{face}좌표축 상에서의 변화율이고, $(\partial t/\partial x)_{P01}$은 픽셀 P0 및 P1의 t_{face}좌표축 상에서의 변화율이고, $(\partial s/\partial x)_{P02}$은 픽셀 P0 및 P2의 s_{face}좌표축 상에서의 변화율이고, $(\partial t/\partial x)_{P02}$은 픽셀 P0 및 P2의 t_{face}좌표축 상에서의 변화율이다.

[0096] 또한, 일 실시예에 따라, 프로세서(520)는 인접하는 픽셀들에 대해 상기 큐브 맵 상에서의 좌표를 결정하는 제 1 연산부, 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 좌표를 보정하는 제 2 연산부, 보정된 좌표를 이용하여, 큐브 맵 상에서 픽셀들의 변화율(derivative)을 계산하는 제 3 연산부; 및 계산된 변화율에 기초하여, 픽셀들에 큐브 맵을 텍스처링 하기 위한 LOD를 결정하는 제 4 연산부를 포함할 수 있다.

[0097] 따라서, 장치(500)는 픽셀들의 큐브 맵 상의 보정된 좌표로부터 픽셀들의 변화율(derivative)을 직접 계산할 수 있는바, 변화율을 계산하는 데 필요한 연산기의 개수를 줄일 수 있다. 구체적으로, 종래에는 LOD 결정에 전제가 되는 픽셀들의 변화율을 계산하기 위해, 픽셀들의 큐브 맵 상의 좌표를 결정하기 전에 스크린 스페이스 상에서 변화율을 계산하는 과정 및 기 계산된 변화율을 이용하여 큐브 맵 상에서 픽셀들의 변화율을 계산하는 과정이 요구되는 바, 이러한 과정에서 곱셈기, 덧셈기, 및 나눗셈기 등 다수의 연산기가 필요하였다. 다만, 본 개시에 따르면, 장치(500)는 큐브 맵 상에서의 픽셀들의 좌표에 대해 덧셈 연산 또는 반전 연산 정도만 수행하면 되는 바, 변화율을 계산하는 데 필요한 연산기의 개수를 줄일 수 있고, 이에 따라 장치(500)의 면적과 소비 전력을 줄일 수 있다. 예를 들어, 도 9에서 장치(500)는 s_{face}좌표축 및 t_{face}좌표축 상에서 픽셀들의 변화율 $\frac{\partial s_{face}}{\partial x}$ 및 $\frac{\partial t_{face}}{\partial x}$을 계산하기 위해, 두 픽셀의 좌표 보정에 필요한 연산기 및 보정된 두 픽셀의 좌표 간의 차이를 구하기 위한 연산기만을 필요로 한다.

[0098]

[0099] 도 12는 일 예에 따라, 큐브 맵을 텍스처링하기 위한 LOD를 결정하는 방법을 설명하기 위한 도면이다.

[0100] 도 12에 도시된 방법은, 도 5의 장치(500) 또는 도 1의 텍스쳐 처리 장치(130)의 각 구성요소에 의해 수행될 수

있고, 중복되는 설명에 대해서는 생략한다.

[0101] 단계 s1210에서, 장치(500)는 인접하는 픽셀들에 대해 큐브 맵 상에서의 좌표를 결정할 수 있다. 일 실시예에 따라, 장치(500)는 스크린 스페이스 상에서 픽셀의 방향 벡터를 기준으로, 픽셀을 큐브 맵 상으로 투영(project)시킨 지점의 좌표를, 픽셀의 큐브 맵 상의 좌표로 결정할 수 있다.

[0102] 단계 s1220에서, 장치(500)는 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 좌표를 보정할 수 있다.

[0103] 먼저, 장치(500)는 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는지 여부를 판단할 수 있다. 다시 말해, 장치(500)는 스크린 스페이스 상의 픽셀들이 큐브 맵의 서로 다른 면들로 투영되는지 여부를 판단할 수 있다. 일 실시예에 따라, 장치(500)는 기 결정된 픽셀들 각각의 큐브 맵 상의 좌표가 큐브 맵의 서로 다른 면에 포함되는지 여부에 따라, 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는 지 여부를 판단할 수 있다. 또한, 다른 실시예에 따라, 장치(500)는 픽셀들의 스크린 스페이스 상에서의 방향 벡터를 기준으로, 픽셀들 각각이 큐브 맵의 서로 다른 면에 대응되는지 여부를 판단할 수 있다.

[0104] 다음으로, 장치(500)는 인접하는 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들 중 적어도 하나의 픽셀의 큐브 맵 상의 좌표를 보정할 수 있다. 일 실시예에 따라, 장치(500)는 픽셀들 중 기준 픽셀에 대응되는 면(face)의 원점 및 좌표축을 기준으로, 픽셀들 중 다른 픽셀의 큐브 맵 상의 좌표를 보정할 수 있다. 또한, 장치(500)는 인접하는 픽셀들 각각이 큐브 맵의 서로 다른 면(face)에 대응되는 경우, 픽셀들의 좌표에 요구되는 보정 연산에 대한 정보를 포함하는 좌표 보정 테이블에 기초하여, 픽셀들의 큐브 맵 상의 좌표를 보정할 수 있다.

[0105] 예를 들어, 장치(500)는 픽셀의 좌표에 대해, 덧셈 연산 및 반전 연산 중 적어도 하나를 수행하여, 픽셀의 좌표를 보정할 수 있다. 구체적인 예로, 장치(500)는 픽셀에 대해 +1, -1, 또는 +2에 대한 덧셈 연산을 수행 할 수 있으며, 픽셀의 부호를 반전시키는 반전 연산을 수행할 수 있다. 또한, 장치(500)는 픽셀의 좌표 (s_{face}, t_{face})의 각 좌표축 값을 (t_{face}, s_{face})와 같이 서로 교환(swap)시키는 연산을 수행할 수 있다.

[0106] 단계 s1230에서, 장치(500)는 픽셀들의 큐브 맵 상의 보정된 좌표를 이용하여, 큐브 맵 상에서 픽셀들의 변화율(derivative)을 계산할 수 있다. 구체적으로, 장치(500)는 큐브 맵의 좌표축 상에서 픽셀들 간의 거리를, 픽셀들의 변화율로써, 계산할 수 있다. 예를 들어, 장치(500)는 큐브 맵의 s_{face}좌표축 및 t_{face}좌표축에 대한 픽셀들의 변화율로써, 큐브 맵의 s_{face}좌표축 및 t_{face}좌표축 상 각각에서 픽셀들 간의 거리를 계산할 수 있다. 다시 말해, 장치(500)는 큐브 맵의 s_{face}좌표축 및 t_{face}좌표축 상에서 픽셀들 각각의 좌표값의 차이를 계산할 수 있다.

[0107] 단계 s1240에서, 장치(500)는 계산된 픽셀들의 변화율(derivative)에 기초하여, 픽셀들에 큐브 맵을 텍스쳐링하기 위한 LOD를 결정할 수 있다.

[0108] 상기 살펴 본 실시 예들에 따른 장치는 프로세서, 프로그램 데이터를 저장하고 실행하는 메모리, 디스크 드라이브와 같은 영구 저장부(permanent storage), 외부 장치와 통신하는 통신 포트, 터치 패널, 키(key), 버튼 등과 같은 사용자 인터페이스 장치 등을 포함할 수 있다. 소프트웨어 모듈 또는 알고리즘으로 구현되는 방법들은 상기 프로세서상에서 실행 가능한 컴퓨터가 읽을 수 있는 코드들 또는 프로그램 명령들로서 컴퓨터가 읽을 수 있는 기록 매체 상에 저장될 수 있다. 여기서 컴퓨터가 읽을 수 있는 기록 매체로 마그네틱 저장 매체(예컨대, ROM(read-only memory), RAM(random-access memory), 플로피 디스크, 하드 디스크 등) 및 광학적 판독 매체(예컨대, 시디롬(CD-ROM), 디브이디(DVD: Digital Versatile Disc)) 등이 있다. 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템들에 분산되어, 분산 방식으로 컴퓨터가 판독 가능한 코드가 저장되고 실행될 수 있다. 매체는 컴퓨터에 의해 판독가능하며, 메모리에 저장되고, 프로세서에서 실행될 수 있다.

[0109] 본 실시 예는 기능적인 블록 구성들 및 다양한 처리 단계들로 나타내어질 수 있다. 이러한 기능 블록들은 특정 기능들을 실행하는 다양한 개수의 하드웨어 또는/및 소프트웨어 구성들로 구현될 수 있다. 예를 들어, 실시 예는 하나 이상의 마이크로프로세서들의 제어 또는 다른 제어 장치들에 의해서 다양한 기능들을 실행할 수 있는, 메모리, 프로세싱, 로직(logic), 룩 업 테이블(look-up table) 등과 같은 직접 회로 구성들을 채용할 수 있다. 구성 요소들이 소프트웨어 프로그래밍 또는 소프트웨어 요소들로 실행될 수 있는 것과 유사하게, 본 실시 예는 데이터 구조, 프로세스들, 루틴들 또는 다른 프로그래밍 구성들의 조합으로 구현되는 다양한 알고리즘을 포함하여, C, C++, 자바(Java), 어셈블러(assembler) 등과 같은 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 기능적인 측면들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다. 또한, 본 실시 예는 전자

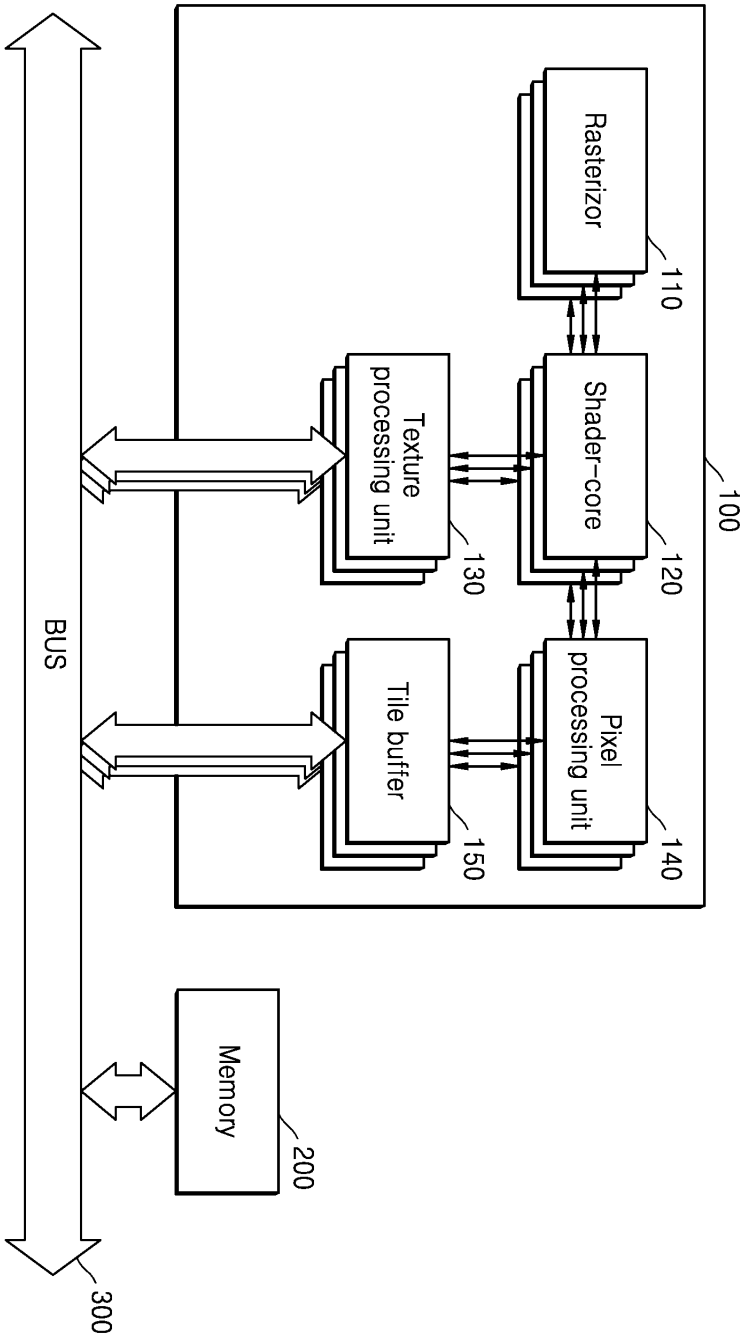
적인 환경 설정, 신호 처리, 및/또는 데이터 처리 등을 위하여 종래 기술을 채용할 수 있다. "매커니즘", "요소", "수단", "구성"과 같은 용어는 넓게 사용될 수 있으며, 기계적이고 물리적인 구성들로서 한정되는 것은 아니다. 상기 용어는 프로세서 등과 연계하여 소프트웨어의 일련의 처리들(routines)의 의미를 포함할 수 있다.

[0110] 본 실시 예에서 설명하는 특정 실행들은 예시들로서, 어떠한 방법으로도 기술적 범위를 한정하는 것은 아니다. 명세서의 간결함을 위하여, 종래 전자적인 구성들, 제어 시스템들, 소프트웨어, 상기 시스템들의 다른 기능적인 측면들의 기재는 생략될 수 있다. 또한, 도면에 도시된 구성 요소들 간의 선들의 연결 또는 연결 부재들은 기능적인 연결 및/또는 물리적 또는 회로적 연결들을 예시적으로 나타낸 것으로서, 실제 장치에서는 대체 가능하거나 추가의 다양한 기능적인 연결, 물리적인 연결, 또는 회로 연결들로서 나타내어질 수 있다.

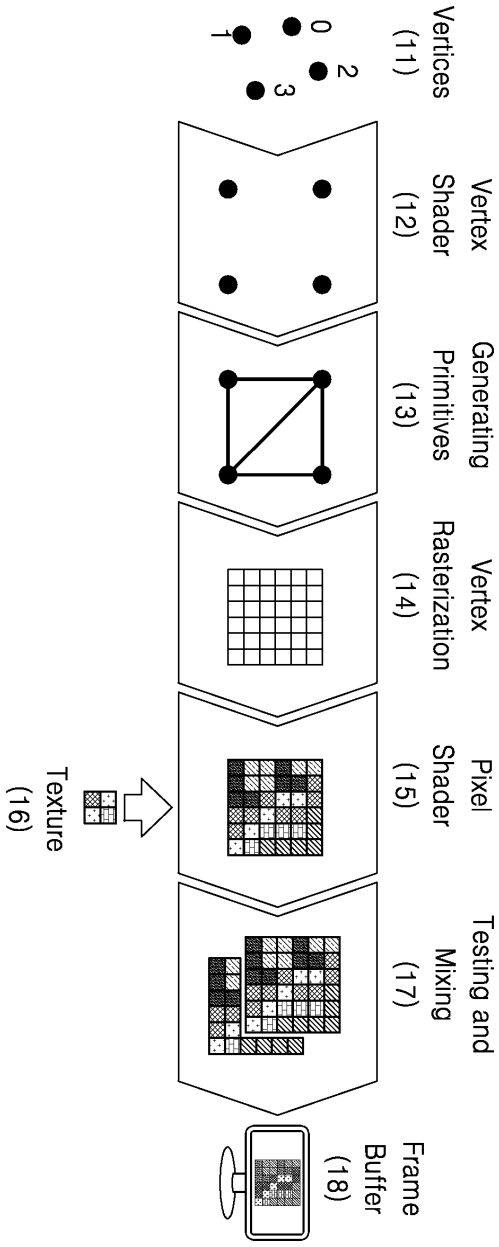
[0111] 본 명세서(특히 특허청구범위에서)에서 "상기"의 용어 및 이와 유사한 지시 용어의 사용은 단수 및 복수 모두에 해당하는 것일 수 있다. 또한, 범위(range)를 기재한 경우 상기 범위에 속하는 개별적인 값을 포함하는 것으로서(이에 반하는 기재가 없다면), 상세한 설명에 상기 범위를 구성하는 각 개별적인 값을 기재한 것과 같다. 마지막으로, 방법을 구성하는 단계들에 대하여 명백하게 순서를 기재하거나 반하는 기재가 없다면, 상기 단계들은 적당한 순서로 행해질 수 있다. 반드시 상기 단계들의 기재 순서에 한정되는 것은 아니다. 모든 예들 또는 예시적인 용어(예들 들어, 등등)의 사용은 단순히 기술적 사상을 상세히 설명하기 위한 것으로서 특허청구범위에 의해 한정되지 않는 이상 상기 예들 또는 예시적인 용어로 인해 범위가 한정되는 것은 아니다. 또한, 당업자는 다양한 수정, 조합 및 변경이 부가된 특허청구범위 또는 그 균등물의 범주 내에서 설계 조건 및 팩터에 따라 구성될 수 있음을 알 수 있다.

도면

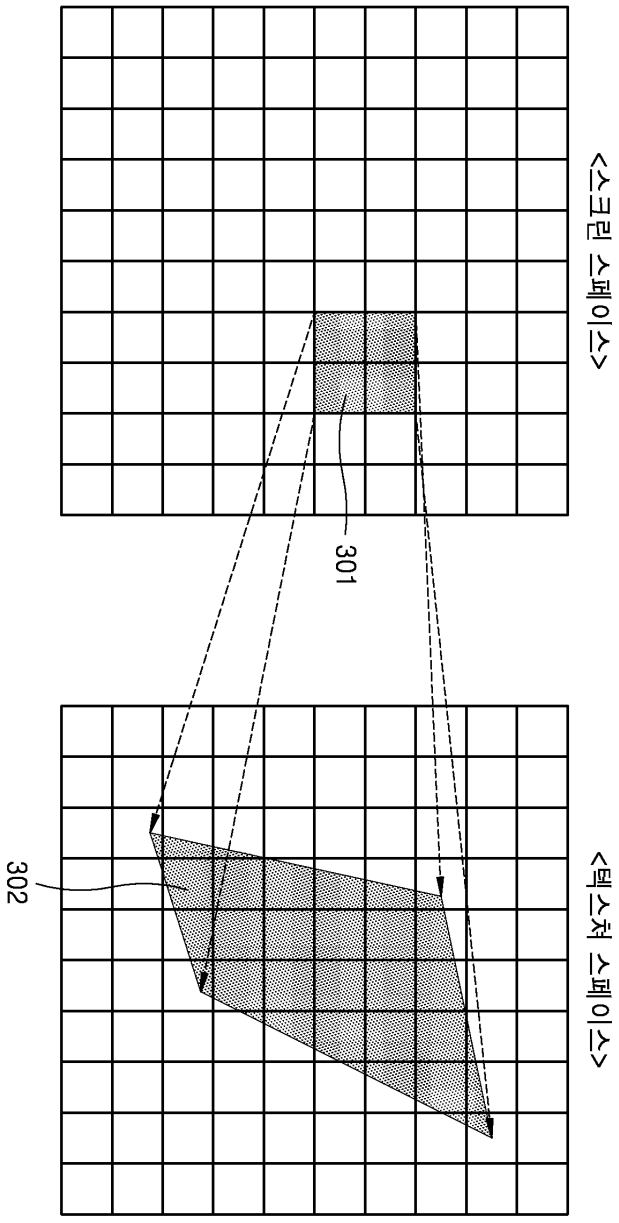
도면1


Rasterizor
110
Shader-core
120
Texture processing unit
130
Pixel processing unit
140
Tile buffer
150
100
BUS
300
Memory
200

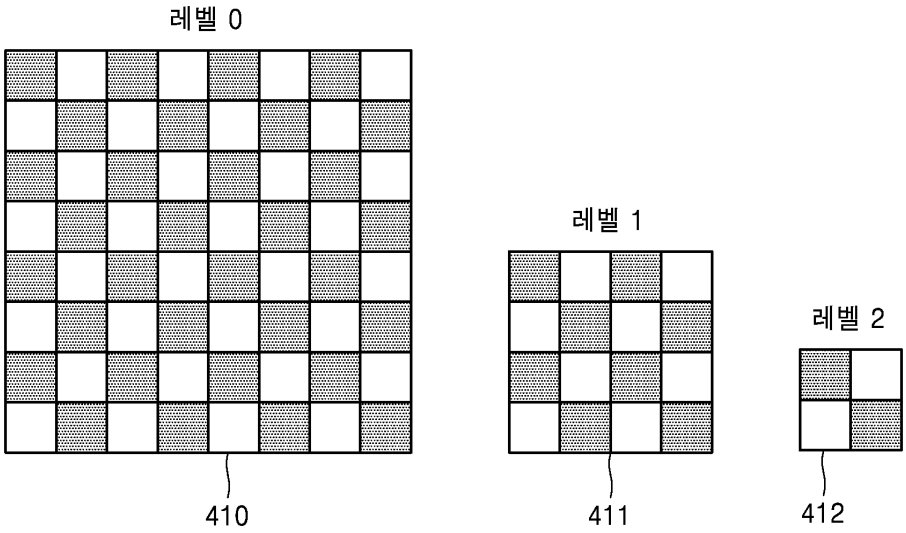
도면2


Vertices
(11)
0
1
2
3
Vertex
Shader
(12)
Generating
Primitives
(13)
Vertex
Rasterization
(14)
Pixel
Shader
(15)
Texture
(16)
Testing and
Mixing
(17)
Frame
Buffer
(18)

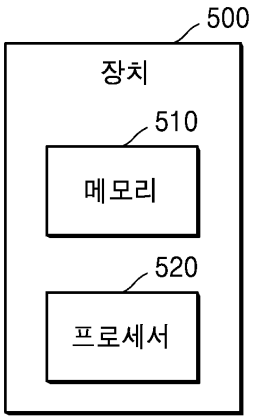
도면3



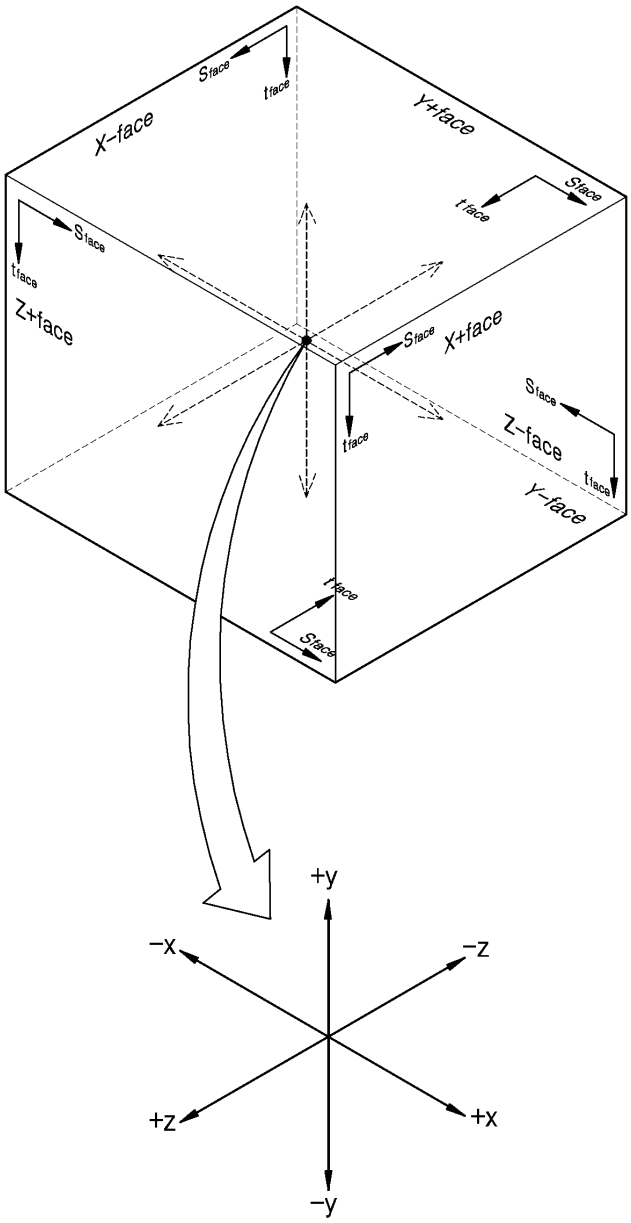
도면4



도면5



도면6



도면7

710

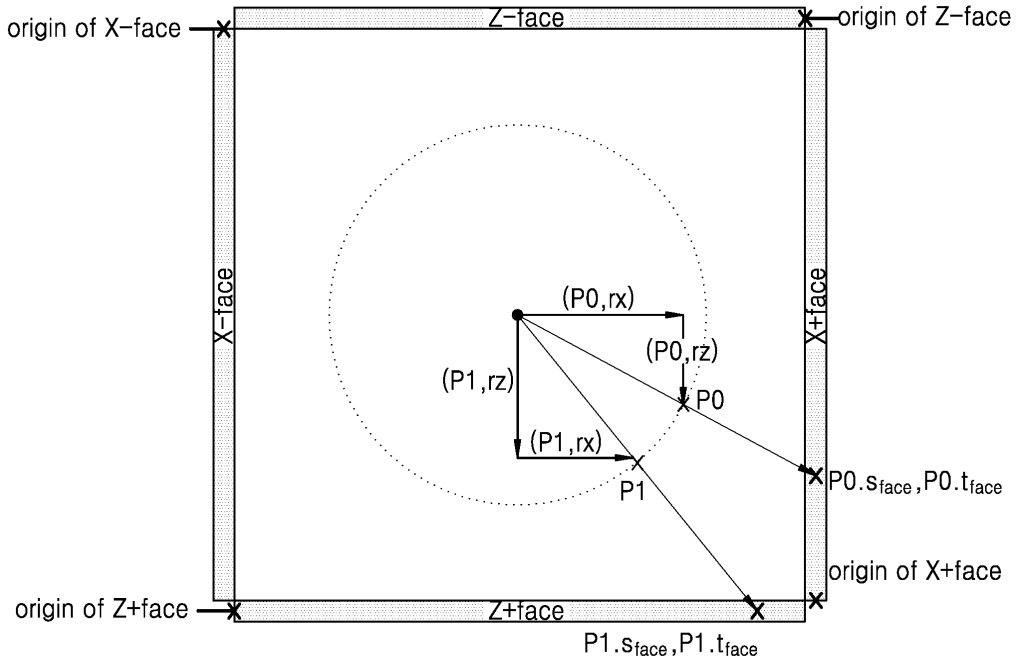
Major Axis Direction	Cube Map Face	s_c	t_c	r_c
$+r_x$	X+ face	$-r_z$	$-r_y$	r_x
$-r_x$	X− face	$+r_z$	$-r_y$	r_x
$+r_y$	Y+ face	$+r_x$	$+r_z$	r_y
$-r_y$	Y− face	$+r_x$	$-r_z$	r_y
$+r_z$	Z+ face	$+r_x$	$-r_y$	r_z
$-r_z$	Z− face	$-r_x$	$-r_y$	r_z

720

$$s_{face} = \frac{1}{2} \times \frac{s_c}{|r_c|} + \frac{1}{2}$$

$$t_{face} = \frac{1}{2} \times \frac{t_c}{|r_c|} + \frac{1}{2}$$

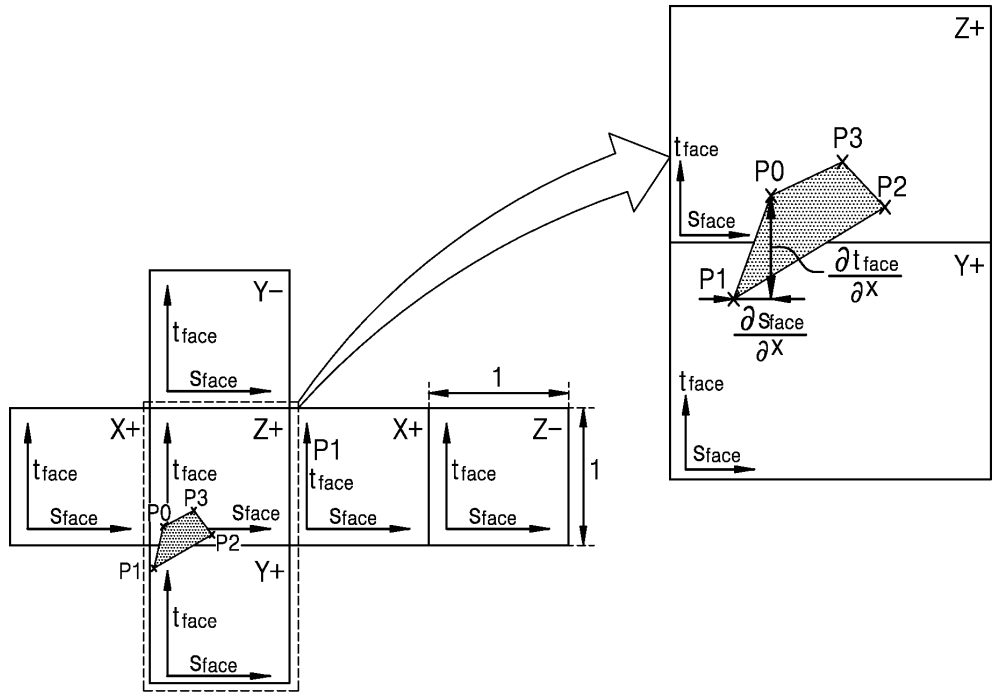
도면8


origin of X-face
Z-face
origin of Z-face
X-face
X+face
(P0,rx)
(P0,rz)
(P1,rz)
P0
(P1,rx)
P1
P0.s_face,P0.t_face
origin of X+face
origin of Z+face
Z+face
P1.s_face,P1.t_face

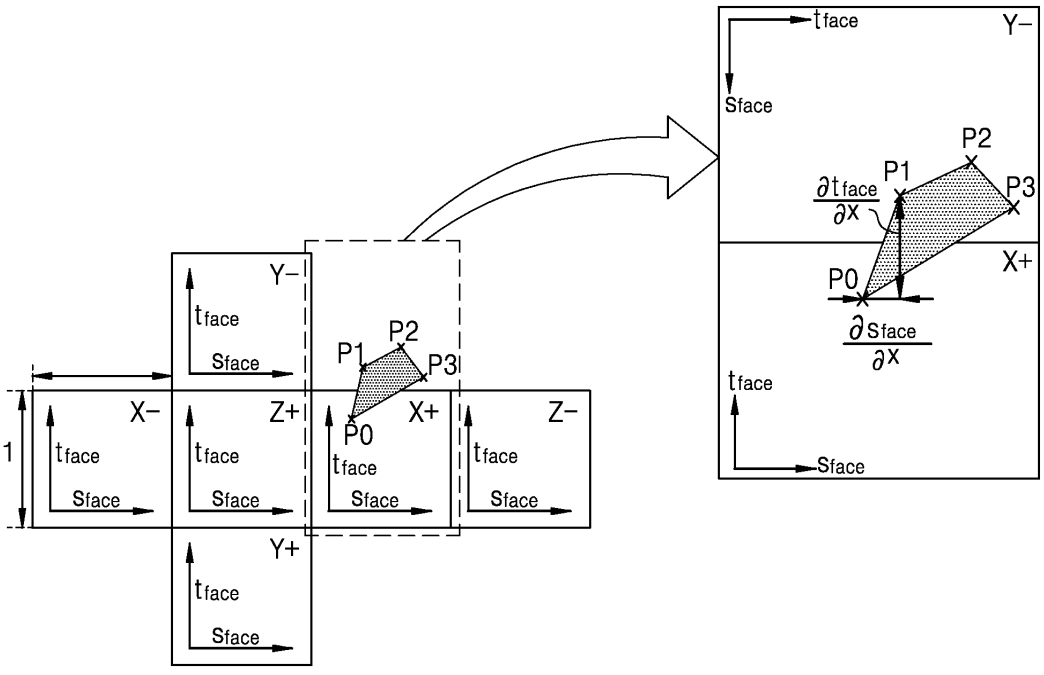
도면9

(P0.sface, P0.tface)
(P1.sface, P1.tface)
sadj
Face of adjusted Pixel
Face of Reference
X+
X−
Y+
Y−
Z+
Z−
910
X+ sface sface −tface+1 tface sface−1 sface+1
X− sface sface tface −tface+1 sface+1 sface−1
Y+ tface+1 −tface sface sface −sface+1 sface
Y− −tface+2 tface−1 sface sface sface −sface+1
Z+ sface+1 sface−1 sface sface sface sface
Z− sface−1 sface+1 −sface+1 sface sface
$\frac{\partial S_{face}}{\partial X} = P1.S_{adj} - P0.S_{adj}$
$\frac{\partial t_{face}}{\partial X} = P1.t_{adj} - P0.t_{adj}$
(P0.sadj, P0.tadj)
(P1.sadj, P1.tadj)
tadj
Face of adjusted Pixel
920
Face of Reference
X+
X−
Y+
Y−
Z+
Z−
X+ tface tface sface−1 −sface+2 tface tface
X− tface tface −sface tface+1 tface tface
Y+ −sface+1 sface tface tface+1 −tface
Y− sface −sface+1 tface tface−1 tface+2
Z+ tface tface tface−1 tface+1 tface tface
Z− tface tface −tface −tface+2 tface tface

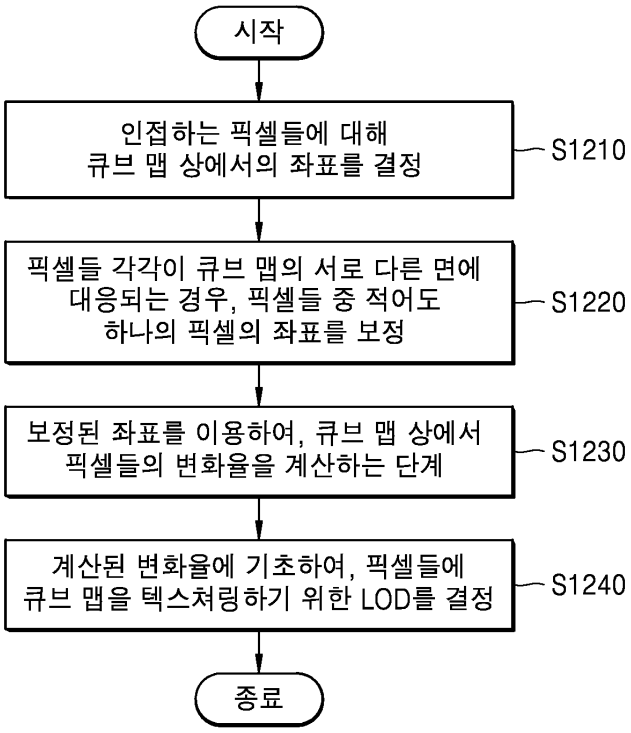
도면10


Z+
tface
Sface
P3
P0
P2
P1
∂tface/∂x
∂Sface/∂x
Y+
Y−
X+
Z+
X+
Z−
1
1
P0
P3
P2
P1

도면11


tface
Y−
Sface
P2
P1
P3
∂tface/∂x
X+
P0
∂Sface/∂x
tface
Sface
Y−
P2
P1
P3
X−
Z+
X+
Z−
P0
1
Y+

도면12


시작
인접하는 픽셀들에 대해
큐브 맵 상에서의 좌표를 결정
S1210
픽셀들 각각이 큐브 맵의 서로 다른 면에
대응되는 경우, 픽셀들 중 적어도
하나의 픽셀의 좌표를 보정
S1220
보정된 좌표를 이용하여, 큐브 맵 상에서
픽셀들의 변화율을 계산하는 단계
S1230
계산된 변화율에 기초하여, 픽셀들에
큐브 맵을 텍스쳐링하기 위한 LOD를 결정
S1240
종료