



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0098690  
(43) 공개일자 2018년09월04일

(51) 국제특허분류(Int. Cl.)  
G10L 15/06 (2006.01) G10L 15/08 (2006.01)  
(52) CPC특허분류  
G10L 15/06 (2013.01)  
G10L 15/08 (2013.01)  
(21) 출원번호 10-2018-7024396  
(22) 출원일자(국제) 2017년01월04일  
심사청구일자 2018년08월23일  
(85) 번역문제출일자 2018년08월23일  
(86) 국제출원번호 PCT/US2017/012130  
(87) 국제공개번호 WO 2017/146820  
국제공개일자 2017년08월31일  
(30) 우선권주장  
15/053,930 2016년02월25일 미국(US)

(71) 출원인  
켈컴 인코퍼레이티드  
미국 92121-1714 캘리포니아주 샌 디에고 모어하  
우스 드라이브 5775  
(72) 발명자  
오베르트 유라이  
미국 32817 플로리다주 올랜도 패브리 서클 2408  
(74) 대리인  
특허법인코리아나

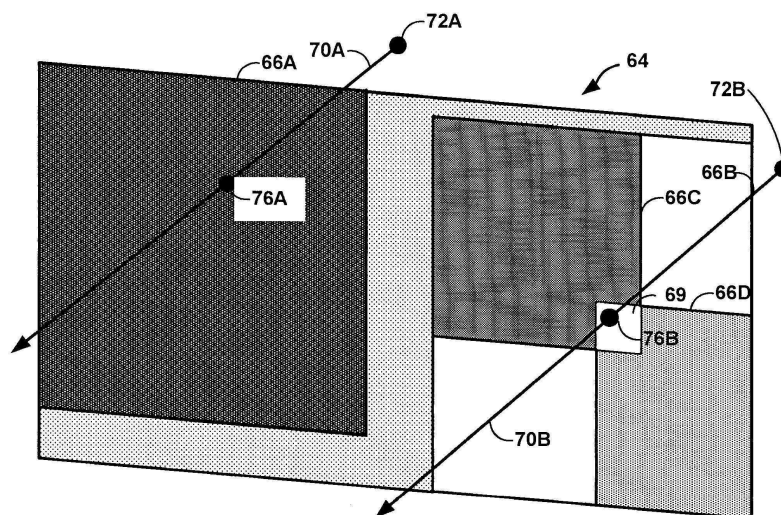
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 그래픽 프로세싱에서 셰도우 레이들에 대한 트리 횡단을 위한 시작 노드 결정

(57) 요약

적어도 하나의 프로세서는 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화할 수도 있으며, 여기서 복수의 바운딩 볼륨들(bounding volumes)이 계층적 데이터 구조의 복수의 노드들과 연관된다. 적어도 하나의 프로세서는 메모리에서의 오프-스크린(off-screen) 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화할 수도 있다. 적어도 하나의 프로세서는, 장면에서 레이에 맵핑하는 오프-스크린 렌더 타겟에서의 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 난-루트(non-root) 노드를 결정할 수도 있다. 적어도 하나의 프로세서는 장면 내의 레이가 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단할 수도 있다.

대표도 - 도5



(52) CPC특허분류  
G06T 2207/20016 (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

방법으로서,

적어도 하나의 프로세서에 의해, 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하는 단계로서, 복수의 바운딩 볼륨들이 상기 계층적 데이터 구조의 노드들과 연관되는, 상기 복수의 프리미티브들을 조직화하는 단계;

상기 적어도 하나의 프로세서에 의해, 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현들을 래스터화하는 단계;

상기 적어도 하나의 프로세서에 의해 그리고 레이에 맵핑하는 상기 오프-스크린 렌더 타겟에서의 화소에 적어도 부분적으로 기초하여, 상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 년-루트 노드를 결정하는 단계; 및

상기 적어도 하나의 프로세서에 의해, 상기 장면 내의 상기 레이가 상기 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 상기 시작 노드로부터 시작하여 상기 계층적 데이터 구조의 노드들의 세트를 횡단하는 단계를 포함하는, 방법.

#### 청구항 2

제 1 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 단계는,

상기 적어도 하나의 프로세서에 의해, 복수의 컬러 값들 중 상이한 컬러 값을 상기 계층적 데이터 구조의 복수의 상기 노드들 각각과 연관시키는 단계; 및

상기 복수의 바운딩 볼륨들의 각각의 바운딩 볼륨에 대해, 상기 적어도 하나의 프로세서에 의해, 상기 각각의 바운딩 볼륨과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나와 연관된 상기 상이한 컬러 값들 중 하나를 갖는 화소들의 블록을 래스터화하는 단계를 더 포함하는, 방법.

#### 청구항 3

제 2 항에 있어서,

상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 년-루트 노드를 결정하는 단계는,

상기 적어도 하나의 프로세서에 의해, 상기 화소의 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나의 노드를 결정하는 단계; 및

상기 적어도 하나의 프로세서에 의해, 상기 계층적 데이터 구조의 횡단을 시작하기 위해 상기 시작 노드로서 상기 화소의 상기 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드를 설정하는 단계를 더 포함하는, 방법.

#### 청구항 4

제 2 항에 있어서,

상기 적어도 하나의 프로세서에 의해, 상기 바운딩 볼륨들의 제 1 바운딩 볼륨의 제 1 표현이 상기 바운딩 볼륨들의 제 2 바운딩 볼륨의 제 2 표현과 중첩한다고 결정하는 단계로서, 상기 제 1 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 1 노드와 연관되고, 상기 제 2 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 2 노드와 연관되는, 상기 결정하는 단계; 및

상기 적어도 하나의 프로세서에 의해, 상기 제 1 노드 및 상기 제 2 노드의 최저 공통 조상 노드와 연관된 노드

컬러 값에 상기 제 1 표현 및 상기 제 2 표현의 중첩의 영역 내의 하나 이상의 화소들의 컬러 값들을 설정하는 단계를 더 포함하는, 방법.

#### 청구항 5

제 1 항에 있어서,

상기 레이가 상기 복수의 프리미티브들 중 하나와 교차한다고 결정하는 것에 응답하여, 상기 적어도 하나의 프로세서에 의해, 상기 레이가 광원을 향해 발하는 상기 장면 내의 로케이션이 상기 광원으로부터 폐색된다고 결정하는 단계를 더 포함하고,

상기 레이는 새도우 레이를 포함하는, 방법.

#### 청구항 6

제 1 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 단계는,

상기 적어도 하나의 프로세서에 의해, 광원의 관점으로부터 상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 상기 표현을 래스터화하는 단계를 더 포함하는, 방법.

#### 청구항 7

제 1 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 단계는,

상기 적어도 하나의 프로세서에 의해, 상기 복수의 바운딩 볼륨들 각각의 표현들로서 상기 오프-스크린 렌더 타겟으로 복수의 플랫폼-셰이딩된 정육면체들을 래스터화하는 단계; 및

상기 적어도 하나의 프로세서에 의해, 각각의 바운딩 볼륨의 형상과 매치시키기 위해 상기 복수의 플랫폼-셰이딩된 정육면체들 각각을 스케일링 및 변환하는 단계를 더 포함하는, 방법.

#### 청구항 8

제 1 항에 있어서,

상기 적어도 하나의 프로세서에 의해, 디스플레이 디바이스에 의한 디스플레이를 위해 상기 장면을 렌더링하는 단계를 더 포함하는, 방법.

#### 청구항 9

그래픽 데이터를 프로세싱하도록 구성된 장치로서,

메모리; 및

적어도 하나의 프로세서를 포함하고,

상기 적어도 하나의 프로세서는,

계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하는 것으로서, 복수의 바운딩 볼륨들이 상기 계층적 데이터 구조의 노드들과 연관되는, 상기 복수의 프리미티브들을 조직화하고;

상기 메모리 내의 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현들을 래스터화하며;

상기 장면에서 레이에 맵핑하는 상기 오프-스크린 렌더 타겟에서의 화소에 적어도 부분적으로 기초하여, 상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 난-루트 노드를 결정하고; 및

상기 장면 내의 상기 레이가 상기 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 상기 시작 노드로부터 시작하여 상기 계층적 데이터 구조의 노드들의 세트를 횡단하도록

구성된, 그래픽 데이터를 프로세싱하도록 구성된 장치.

## 청구항 10

제 9 항에 있어서,

상기 적어도 하나의 프로세서는 또한,

복수의 컬러 값들 중 상이한 컬러 값을 상기 계층적 데이터 구조의 복수의 상기 노드들 각각과 연관시키고; 및

상기 복수의 바운딩 볼륨들의 각각의 바운딩 볼륨에 대해, 상기 각각의 바운딩 볼륨과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나와 연관된 상기 상이한 컬러 값들 중 하나를 갖는 화소들의 블록을 래스터화하도록

구성되는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

## 청구항 11

제 10 항에 있어서,

상기 적어도 하나의 프로세서는 또한,

상기 화소의 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나의 노드를 결정하고; 및

상기 계층적 데이터 구조의 횡단을 시작하기 위해 상기 시작 노드로서 상기 화소의 상기 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드를 설정하도록

구성되는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

## 청구항 12

제 10 항에 있어서,

상기 적어도 하나의 프로세서는, 또한,

상기 바운딩 볼륨들의 제 1 바운딩 볼륨의 제 1 표현이 상기 바운딩 볼륨들의 제 2 바운딩 볼륨의 제 2 표현과 중첩한다고 결정하는 것으로서, 상기 제 1 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 1 노드와 연관되고, 상기 제 2 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 2 노드와 연관되는, 상기 제 1 표현이 제 2 표현과 중첩한다고 결정하고; 및

상기 제 1 노드 및 상기 제 2 노드의 최저 공통 조상 노드와 연관된 노드 컬러 값에 상기 제 1 표현 및 상기 제 2 표현의 중첩의 영역 내의 하나 이상의 화소들의 컬러 값들을 설정하도록

구성되는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

## 청구항 13

제 9 항에 있어서,

상기 적어도 하나의 프로세서는 또한,

상기 레이가 상기 복수의 프리미티브들 중 하나와 교차한다고 결정하는 것에 응답하여, 상기 레이가 광원을 향해 발하는 상기 장면 내의 로케이션이 상기 광원으로부터 폐색된다고 결정하도록 구성되고,

상기 레이는 새도우 레이를 포함하는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

## 청구항 14

제 9 항에 있어서,

상기 적어도 하나의 프로세서는 또한,

광원의 관점으로부터 상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 상기 표현을 래스터화하도록 구성되는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

## 청구항 15

제 9 항에 있어서,

상기 적어도 하나의 프로세서는 또한,

상기 복수의 바운딩 볼륨들 각각의 표현들로서 상기 오프-스크린 렌더 타겟으로 복수의 플랫-셰이딩된 정육면체들을 래스터화하고; 및

각각의 바운딩 볼륨의 형상과 매치시키기 위해 상기 복수의 플랫-셰이딩된 정육면체들 각각을 스케일링 및 변환하도록

구성되는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

#### 청구항 16

제 9 항에 있어서,

상기 장치는 디스플레이 디바이스를 더 포함하고,

상기 적어도 하나의 프로세서는 또한 상기 디스플레이 디바이스에 의한 디스플레이를 위해 상기 장면을 렌더링하도록 구성되는, 그래픽 데이터를 프로세싱하도록 구성된 장치.

#### 청구항 17

장치로서,

계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하는 수단으로서, 복수의 바운딩 볼륨들이 상기 계층적 데이터 구조의 노드들과 연관되는, 상기 복수의 프리미티브들을 조직화하는 수단;

오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현들을 래스터화하는 수단;

상기 장면에서 레이에 맵핑하는 상기 오프-스크린 렌더 타겟에서의 화소에 적어도 부분적으로 기초하여, 상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 난-루트 노드를 결정하는 수단; 및

상기 장면 내의 상기 레이가 상기 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 상기 시작 노드로부터 시작하여 상기 계층적 데이터 구조의 노드들의 세트를 횡단하는 수단을 포함하는, 장치.

#### 청구항 18

제 17 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 수단은,

복수의 컬러 값들 중 상이한 컬러 값을 상기 계층적 데이터 구조의 복수의 상기 노드들 각각과 연관시키는 수단; 및

상기 복수의 바운딩 볼륨들의 각각의 바운딩 볼륨에 대해, 상기 각각의 바운딩 볼륨과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나와 연관된 상기 상이한 컬러 값들 중 하나를 갖는 화소들의 블록을 래스터화하는 수단을 더 포함하는, 장치.

#### 청구항 19

제 18 항에 있어서,

상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 난-루트 노드를 결정하는 수단은,

상기 화소의 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나의 노드를 결정하는 수단; 및

상기 계층적 데이터 구조의 횡단을 시작하기 위해 상기 시작 노드로서 상기 화소의 상기 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드를 설정하는 수단을 더 포함하는, 장치.

## 청구항 20

제 18 항에 있어서,

상기 바운딩 볼륨들의 제 1 바운딩 볼륨의 제 1 표현이 상기 바운딩 볼륨들의 제 2 바운딩 볼륨의 제 2 표현과 중첩한다고 결정하는 수단으로서, 상기 제 1 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 1 노드와 연관되고, 상기 제 2 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 2 노드와 연관되는, 상기 결정하는 수단; 및

상기 제 1 노드 및 상기 제 2 노드의 최저 공통 조상 노드와 연관된 노드 컬러 값에 상기 제 1 표현 및 상기 제 2 표현의 중첩의 영역 내의 하나 이상의 화소들의 컬러 값들을 설정하는 수단을 더 포함하는, 장치.

## 청구항 21

제 17 항에 있어서,

상기 레이가 상기 복수의 프리미티브들 중 하나와 교차한다고 결정하는 것에 응답하여, 상기 레이가 광원을 향해 발하는 상기 장면 내의 로케이션이 상기 광원으로부터 폐색된다고 결정하는 수단을 더 포함하고,

상기 레이는 새도우 레이를 포함하는, 장치.

## 청구항 22

제 17 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 수단은,

광원의 관점으로부터 상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 상기 표현을 래스터화하는 수단을 더 포함하는, 장치.

## 청구항 23

제 17 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 수단은,

상기 복수의 바운딩 볼륨들 각각의 표현들로서 상기 오프-스크린 렌더 타겟으로 복수의 플랫폼-셰이딩된 정육면체들을 래스터화하는 수단; 및

각각의 바운딩 볼륨의 형상과 매치시키기 위해 상기 복수의 플랫폼-셰이딩된 정육면체들 각각을 스케일링 및 변환하는 수단을 더 포함하는, 장치.

## 청구항 24

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 하나 이상의 프로그램가능 프로세서들로 하여금,

계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하게 하는 것으로서, 복수의 바운딩 볼륨들이 상기 계층적 데이터 구조의 노드들과 연관되는, 상기 복수의 프리미티브들을 조직화하게 하고;

메모리 내의 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현들을 래스터화하게 하며;

상기 장면에서 레이에 맵핑하는 상기 오프-스크린 렌더 타겟에서의 화소에 적어도 부분적으로 기초하여, 상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 난-루트 노드를 결정하게 하고; 및

상기 장면 내의 상기 레이가 상기 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 상기 시작 노드로부터 시작하여 상기 계층적 데이터 구조의 노드들의 세트를 횡단하게 하는, 컴퓨터 판독가능 저장 매체.

## 청구항 25

제 24 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 것은,

복수의 컬러 값들 중 상이한 컬러 값을 상기 계층적 데이터 구조의 복수의 상기 노드들 각각과 연관시키는 것; 및

상기 복수의 바운딩 볼륨들의 각각의 바운딩 볼륨에 대해, 상기 각각의 바운딩 볼륨과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나와 연관된 상기 상이한 컬러 값들 중 하나를 갖는 화소들의 블록을 래스터화하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 26

제 25 항에 있어서,

상기 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 상기 화소와 연관된 상기 계층적 데이터 구조의 난-루트 노드를 결정하는 것은,

상기 화소의 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드들 중 하나의 노드를 결정하는 것; 및

상기 계층적 데이터 구조의 횡단을 시작하기 위해 상기 시작 노드로서 상기 화소의 상기 화소 컬러 값과 연관되는 상기 계층적 데이터 구조의 상기 노드를 설정하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 27

제 25 항에 있어서,

실행될 때, 하나 이상의 프로그램가능 프로세서들로 하여금,

상기 바운딩 볼륨들의 제 1 바운딩 볼륨의 제 1 표현이 상기 바운딩 볼륨들의 제 2 바운딩 볼륨의 제 2 표현과 중첩한다고 결정하게 하는 것으로서, 상기 제 1 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 1 노드와 연관되고, 상기 제 2 바운딩 볼륨은 상기 계층적 데이터 구조의 상기 노드들의 제 2 노드와 연관되는, 상기 제 1 표현이 제 2 표현과 중첩한다고 결정하게 하고; 및

상기 제 1 노드 및 상기 제 2 노드의 최저 공통 조상 노드와 연관된 노드 컬러 값에 상기 제 1 표현 및 상기 제 2 표현의 중첩의 영역 내의 화소들의 컬러 값들을 설정하게 하는

명령들을 더 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 28

제 24 항에 있어서,

실행될 때, 하나 이상의 프로그램가능 프로세서들로 하여금,

상기 레이가 상기 복수의 프리미티브들 중 하나와 교차한다고 결정하는 것에 응답하여, 상기 레이가 광원을 향해 발하는 상기 장면 내의 로케이션이 상기 광원으로부터 폐색된다고 결정하게 하는 명령들을 더 포함하고,

상기 레이는 새도우 레이를 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 29

제 24 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 것은,

광원의 관점으로부터 상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 상기 표현을 래스터화하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 30

제 24 항에 있어서,

상기 오프-스크린 렌더 타겟으로 상기 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 것은,



상기 복수의 바운딩 볼륨들 각각의 표현들로서 상기 오프-스크린 렌더 타겟으로 복수의 플랫-셰이딩된 정육면체들을 래스터화하는 것; 및

각각의 바운딩 볼륨의 형상과 매치시키기 위해 상기 복수의 플랫-셰이딩된 정육면체들 각각을 스케일링 및 변환하는 것을 더 포함하는, 컴퓨터 판독가능 저장 매체.

## 발명의 설명

### 기술 분야

[0001] 본 개시는 새도우 레이 트레이싱 (shadow ray tracing) 을 위한 레이-프리미티브 교차 (ray-primitive intersection) 를 결정하기 위해 계층적 데이터 구조를 횡단하는 것을 포함하는 그래픽 처리에 관한 것이다.

### 배경 기술

[0002] 컴퓨터 그래픽에서, 새도우 렌더링은 장면의 특정의 로케이션들이 광원에 의해 조명되는지 여부에 기초하여 3차원 (3D) 장면에서 새도우들이 추가되는 기법이다. 그래픽 프로세싱 유닛 (GPU) 은 그 로케이션으로부터 광원을 향해 새도우 레이라고 지칭되는 벡터를 발함으로써 3D 장면의 특정의 로케이션에 대한 그러한 새도우 렌더링을 수행할 수도 있다. GPU 가 새도우 레이와 장면 지오메트리에서 프리미티브와 교차한다고 결정하면, GPU 는 소스 로케이션이 그늘져 있고 광원에 의해 조명되지 않는다고 결정할 수도 있다.

[0003] 새도우 레이-프리미티브 교차들을 발견하는 프로세스를 가속화하기 위해, GPU 는 3D 장면의 장면 지오메트리를 장면 프리미티브들 (예를 들어, 삼각형들) 을 계층적으로 그룹핑하는 가속화 데이터 구조 (acceleration data structure: ADS) 로 배열할 수도 있다. GPU 는 새도우 레이와 장면의 프리미티브와 교차하는지 여부를 결정하기 위해 장면 프리미티브들의 계층에 대한 새도우 레이 교차 테스트들을 수행함으로써 ADS 를 회귀적으로 횡단할 수도 있다. GPU 가 특정의 로케이션으로부터 발하는 새도우 레이와 ADS 의 횡단에 기초하여 프리미티브와 교차한다고 결정하면, GPU 는 그 특정의 로케이션이 적어도 그 프리미티브에 의해 광원으로부터 폐색된다고 결정할 수도 있다.

## 발명의 내용

### 해결하려는 과제

### 과제의 해결 수단

[0004] 하나의 양태에서, 본 개시는 방법으로 지향된다. 방법은 적어도 하나의 프로세서에 의해 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하는 단계를 포함하며, 여기서 복수의 바운딩 볼륨들 (bounding volumes) 은 계층적 데이터 구조의 복수의 노드들과 연관된다. 방법은 적어도 하나의 프로세서에 의해 오프-스크린 (off-screen) 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 단계를 더 포함한다.

방법은 적어도 하나의 프로세서에 의해 그리고 오프-스크린 렌더 타겟에서 제 1 레이와 교차하는 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 난-루트 (non-root) 노드를 결정하는 단계를 더 포함한다. 방법은 적어도 하나의 프로세서에 의해, 장면 내의 제 2 레이와 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단하는 단계를 더 포함한다.

[0005] 다른 양태에서, 본 개시는 그래픽 데이터를 프로세싱하도록 구성된 장치에 지향된다. 장치는 메모리를 포함한다. 장치는 적어도 하나의 프로세서를 더 포함하며, 그 적어도 하나의 프로세서는, 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하는 것으로서, 복수의 바운딩 볼륨들은 계층적 데이터 구조의 복수의 노드들과 연관되는, 상기 복수의 프리미티브들을 조직화하고; 메모리에서 오프-스크린 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화하며; 오프-스크린 렌더 타겟에서 제 1 레이와 교차하는 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 난-루트 노드를 결정하고; 장면 내의 제 2 레이와 복수의 프리미티브들 중 하나와 교차하는지 여부를 결

정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단하도록 구성된다.

[0006] 다른 양태에서, 본 개시는 장치에 지향된다. 장치는 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하는 수단을 포함하며, 여기서, 복수의 바운딩 볼륨들은 계층적 데이터 구조의 복수의 노드들과 연관된다. 장치는 오프-스크린 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화하는 수단을 더 포함한다. 장치는 오프-스크린 렌더 타겟에서 제 1 레이와 교차하는 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 난-루트 노드를 결정하는 수단을 더 포함한다. 장치는 장면 내의 제 2 레이가 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단하는 수단을 더 포함한다.

[0007] 다른 양태에서, 본 개시는 명령들을 저장하는 컴퓨터 판독가능 저장 매체로 지향된다. 명령들은, 실행될 때, 하나 이상의 프로그램가능 프로세서로 하여금, 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화하게 하는 것으로서, 복수의 바운딩 볼륨들은 계층적 데이터 구조의 복수의 노드들과 연관되는, 상기 복수의 프리미티브들을 조직화하게 하고; 메모리에서 오프-스크린 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화하게 하며; 오프-스크린 렌더 타겟에서 제 1 레이와 교차하는 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 난-루트 노드를 결정하게 하고; 장면 내의 제 2 레이가 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단하게 한다.

[0008] 본 개시의 하나 이상의 양태들의 상세들은 첨부하는 도면들 및 이하의 설명에서 진술된다. 본 개시의 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구범위로부터 명백할 것이다.

### 도면의 간단한 설명

[0009] 도 1 은 본 개시의 하나 이상의 양태들을 구현하도록 구성될 수도 있는 예시의 컴퓨팅 디바이스를 도시하는 블록도이다.

도 2 는 도 1 의 CPU, GPU, 및 시스템 메모리의 예시의 구현들을 더욱 상세히 도시하는 블록도이다.

도 3 은 GPU 가 새도우 레이 트레이싱을 수행할 수도 있는 예시의 그래픽 장면 및 바운딩 볼륨들로의 그래픽 장면의 예시의 파티셔닝을 도시하는 개념도이다.

도 4 는 도 3 에 도시된 예시의 바운딩 볼륨들 및 프리미티브들과 연관되는 노드들을 갖는 예시의 계층적 데이터 구조를 도시한다.

도 5 는 예시의 오프-스크린 렌더 타겟으로 래스터화된 바운딩 볼륨들의 예시의 표현을 도시하는 블록도이다.

도 6 은 새도우 레이-프리미티브 교차를 발견하기 위해 예시의 계층적 데이터 구조를 횡단하기 위한 시작 노드를 결정하는 예시의 프로세스를 도시하는 플로우차트이다.

### 발명을 실시하기 위한 구체적인 내용

[0010] 일반적으로, 본 개시는 GPU 가 장면의 광원을 향해 지향된 새도우 레이들이 ADS 와 같은 계층적 데이터 구조에서 배열되는 프리미티브들과 교차하는지 여부를 결정함으로써 그래픽 장면에 대한 새도우 렌더링을 더 효율적으로 수행하는 기법들을 기술한다. 루트 노드로부터 계층적 데이터 구조를 횡단하는 대신에, GPU 는 대신에 루트 노드 이외의 노드로부터 계층적 데이터 구조의 횡단을 시작하며, 이것에 의해 GPU 가 수행할 수도 있는 레이 교차 테스트들의 수를 감소시킬 수도 있다. GPU 는 오프-스크린 렌더 타겟으로 계층적 데이터 구조의 내부 노드들과 그리고 리프 (leaf) 노드들과 연관된 바운딩 볼륨들의 서브셋을 래스터화하기 위해 그것의 그래픽 프로세싱 파이프라인으로부터의 새이더 유닛들을 이용함으로써 횡단을 시작할 계층적 데이터 구조의 난-루트 노드를 결정할 수도 있다. GPU 는, 오프-스크린 렌더 타겟으로부터, 계층적 데이터 구조의 횡단을 시작할 내부 난-루트 노드를 결정하고, 이것에 의해 계층적 데이터 구조를 횡단하기 위해 GPU 가 수행할 수도 있는 새도우 레이 교차 테스트들의 수를 감소시킬 수도 있다. 오프-스크린 렌더 타겟으로 래스터화하는 것은 레이 교차 테스트들을 수행하는 것보다 상대적으로 덜 계산적으로 고가이기 때문에, GPU 는 루트 노드로부터 계층적 데이터 구조를 횡단하는 것에 비해 오프-스크린 렌더 타겟으로 래스터화하는 것 및 계층적 데이터 구조의 횡단을 시작할 내부 난-루트 노드를 결정하는 것에 의해 새도우 렌더링 성능에서의 실질적인 증가를 실감할 수도 있다.

[0011] 본 개시의 양태들에 따르면, GPU 는 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화할 수도 있으

며, 여기서 복수의 바운딩 볼륨들은 계층적 데이터 구조의 복수의 노드들과 연관된다. GPU 는 또한 메모리에서 오프-스크린 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화할 수도 있다. GPU 는 또한 오프-스크린 렌더 타겟에서 제 1 레이와 교차하는 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 난-루트 노드를 결정할 수도 있다.

GPU 는 또한 장면 내의 제 2 레이가 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단할 수도 있다.

[0012] 도 1 은 본 개시의 하나 이상의 양태들을 구현하도록 구성될 수도 있는 예시의 컴퓨팅 디바이스를 도시하는 블록도이다. 도 1 에 도시된 바와 같이, 디바이스 (2) 는 비디오 디바이스들, 미디어 재생기들, 셋톱 박스들, 이동 전화들 및 소위 스마트폰들과 같은 무선 핸드셋들, 개인용 휴대정보단말들 (PDAs), 데스크톱 컴퓨터들, 랩톱 컴퓨터들, 게이밍 콘솔들, 비디오 회의 유닛들, 태블릿 컴퓨팅 디바이스들, 홈 어프라이언스들, 산업용 어플라이언스들, 키오스크들 등을 포함하지만 이들에 제한되지 않는 컴퓨팅 디바이스일 수도 있다. 도 1 의 예에서, 디바이스 (2) 는 중앙 프로세싱 유닛 (CPU) (6), 시스템 메모리 (10), 및 GPU (12) 를 포함할 수도 있다.

디바이스 (2) 는 또한 디스플레이 프로세서 (14), 송수신기 모듈 (3), 사용자 인터페이스 (4), 및 디스플레이 (8) 를 포함할 수도 있다. 송수신기 모듈 (3) 및 디스플레이 프로세서 (14) 는 양자 모두가 CPU (6) 및/또는 GPU (12) 와 동일한 집적회로 (IC) 의 부분일 수도 있거나, 양자 모두가 CPU (6) 및/또는 GPU (12) 를 포함하는 IC 또는 IC 들의 외부에 있을 수도 있거나, CPU (6) 및/또는 GPU (12) 를 포함하는 IC 의 외부에 있는 IC 에 형성될 수도 있다.

[0013] 디바이스 (2) 는 명확성의 목적상 도 1 에 도시되지 않은 추가의 모듈들 또는 유닛들을 포함할 수도 있다. 예를 들어, 디바이스 (2) 는 디바이스 (2) 가 이동 무선 전화인 경우의 예들에서의 전화 통신들을, 또는 디바이스 (2) 가 미디어 재생기인 경우의 스피커를 유효하게 하기 위해, 도 1 에 도시되지 않은 스피커 및 마이크를 포함할 수도 있다. 디바이스 (2) 는 또한 비디오 카메라를 포함할 수도 있다. 더욱이, 디바이스 (2) 에 도시된 여러 모듈들 및 유닛들은 디바이스 (2) 의 모든 예에서 필요하지는 않을 수도 있다. 예를 들어, 사용자 인터페이스 (4) 및 디스플레이 (8) 는 디바이스 (2) 가 외부 사용자 인터페이스 또는 디스플레이와 인터페이싱하도록 구비되는 데스크톱 컴퓨터 또는 다른 디바이스인 경우의 예들에서 디바이스 (2) 의 외부에 있을 수도 있다.

[0014] 사용자 인터페이스 (4) 의 예들은 트랙볼, 마우스, 키보드, 및 다른 타입들의 입력 디바이스들을 포함하지만 이들에 제한되지 않는다. 사용자 인터페이스 (4) 는 또한 터치 스크린일 수도 있고 디스플레이 (8) 의 부분으로서 통합될 수도 있다. 송수신기 모듈 (3) 은 디바이스 (2) 와 다른 디바이스 또는 네트워크 사이의 무선 또는 유선 통신을 허용하는 회로를 포함할 수도 있다. 송수신기 모듈 (3) 은 변조기들, 복조기들, 증폭기들 및 유선 또는 무선 통신을 위한 다른 그러한 회로를 포함할 수도 있다.

[0015] CPU (6) 는 실행을 위한 컴퓨터 프로그램의 명령들을 프로세싱하도록 구성된 중앙 프로세싱 유닛 (CPU) 과 같은 마이크로프로세서일 수도 있다. CPU (6) 는 디바이스 (2) 의 동작을 제어하는 범용 또는 특수 목적 프로세서를 포함할 수도 있다. 사용자는 CPU (6) 가 하나 이상의 소프트웨어 애플리케이션들을 실행하게 하기 위해 디바이스 (2) 로 입력을 제공할 수도 있다. CPU (6) 상에서 실행되는 소프트웨어 애플리케이션들은 예를 들어 운영 시스템, 워드 프로세서 애플리케이션, 이메일 애플리케이션, 스프레드시트 애플리케이션, 미디어 재생기 애플리케이션, 비디오 게임 애플리케이션, 그래픽 사용자 인터페이스 애플리케이션 또는 다른 프로그램을 포함할 수도 있다. 추가적으로, CPU (6) 는 GPU (12) 의 동작을 제어하기 위한 GPU 구동기 (22) 를 실행할 수도 있다. 사용자는 키보드, 마우스, 마이크, 터치 패드 또는 사용자 인터페이스 (4) 를 통해 디바이스 (2) 에 커플링되는 다른 입력 디바이스와 같은 하나 이상의 입력 디바이스들 (미도시) 을 통해 디바이스 (2) 로 입력을 제공할 수도 있다.

[0016] CPU (6) 상에서 실행되는 소프트웨어 애플리케이션들은 CPU (6) 에게 디스플레이 (8) 로의 그래픽 데이터의 렌더링을 야기하도록 명령하는 하나 이상의 그래픽 렌더링 명령들을 포함할 수도 있다. 일부 예들에서, 소프트웨어 애플리케이션 명령들은 예를 들어 개방 그래픽 라이브러리 (OpenGL<sup>®</sup>) API, 개방 그래픽 라이브러리 임베디드 시스템들 (OpenGL ES) API, Direct3D API, X3D API, RenderMan API, WebGL API, 또는 임의의 다른 공용 또는 사유의 표준 그래픽 API 와 같은 그래픽 애플리케이션 프로그램 인터페이스 (API) 에 따를 수도 있다. 그래픽 렌더링 명령들을 프로세싱하기 위해, CPU (6) 는 GPU (12) 로 하여금 그래픽 데이터의 렌더링의 일부 또는 전부를 수행하게 하기 위해 (예를 들어, GPU 구동기 (22) 를 통해) GPU (12) 로 하나 이상의 그래픽 렌더링 커맨드들을 발행할 수도 있다. 일부 예들에서, 렌더링될 그래픽 데이터는 그래픽 프리미티브들, 예를 들

어, 점들, 선들, 삼각형들, 사각형들, 삼각형 스트립들 등의 리스트를 포함할 수도 있다.

[0017] GPU (12) 는 디스플레이 (8) 로 하나 이상의 그래픽 프리미티브들을 렌더링하기 위해 그래픽 동작들을 수행하도록 구성될 수도 있다. 따라서, CPU (6) 상에서 실행되는 소프트웨어 애플리케이션들 중 하나가 그래픽 프로세싱을 요구할 때, CPU (6) 는 디스플레이 (8) 로 렌더링하기 위해 GPU (12) 에게 그래픽 커맨드들 및 그래픽 데이터를 제공할 수도 있다. 그래픽 데이터는 예를 들어 드로잉 커맨드들, 상태 정보, 프리미티브 정보, 텍스처 정보 등을 포함할 수도 있다. GPU (12) 는 일부 예들에서 CPU (6) 보다 복잡한 그래픽 관련 동작들의 더 효율적인 프로세싱을 제공하는 고도로 병렬 구조로 구축될 수도 있다. 예를 들어, GPU (12) 는 다수의 정점들 또는 화소들에 대해 병렬 방식으로 동작하도록 구성되는, 셰이더 유닛들과 같은 복수의 프로세싱 엘리먼트들을 포함할 수도 있다. GPU (12) 의 고도로 병렬 특성은, 일부 예들에서, GPU (12) 가 CPU (6) 를 사용하여 디스플레이 (8) 에 직접 장면들을 드로잉하는 것보다 더 빠르게 디스플레이 (8) 상으로 그래픽 이미지들 (예를 들어, GUI 들 및 2차원 (2D) 및/또는 3차원 (3D) 그래픽 장면들) 을 드로잉하는 것을 허용할 수도 있다.

[0018] GPU (12) 는, 일부 예들에서, 디바이스 (2) 의 마더보드로 통합될 수도 있다. 다른 예들에서, GPU (12) 는 디바이스 (2) 의 마더보드 내의 포트에 설치되는 그래픽 카드상에 존재할 수도 있거나 다르게는 디바이스 (2) 와 상호작용하도록 구성된 주변 디바이스 내에 통합될 수도 있다. GPU (12) 는 하나 이상의 마이크로프로세서들, 주문형 반도체들 (ASICs), 필드 프로그램가능 게이트 어레이들 (FPGAs), 디지털 신호 프로세서들 (DSPs), 또는 다른 등가의 집적된 또는 이산 논리 회로와 같은 하나 이상의 프로세서들을 포함할 수도 있다. GPU (12) 는 또한 GPU (12) 가 멀티-코어 프로세서로서 지칭될 수 있도록 하나 이상의 프로세서 코어들을 포함할 수도 있다.

[0019] GPU (12) 는 그래픽 메모리 (40) 에 직접 커플링될 수도 있다. 따라서, GPU (12) 는 버스를 사용하지 않고 그래픽 메모리 (40) 로부터 데이터를 판독하고 그래픽 메모리 (40) 로 데이터를 기입할 수도 있다. 즉, GPU (12) 는 오프-칩 메모리 대신에 로컬 스토리지, 즉, 그래픽 메모리 (40) 를 사용하여 로컬로 데이터를 프로세싱할 수도 있다. 그러한 그래픽 메모리 (40) 는 온-칩 메모리로서 지칭될 수도 있다. 이것은 과중한 버스 트래픽을 경험할 수도 있는, 버스를 통해 데이터를 판독하고 기입할 GPU (12) 의 필요를 제거함으로써 GPU (12) 가 더 효율적인 방식으로 동작하는 것을 허용한다. 일부 예들에서, 그러나, GPU (12) 는 별개의 메모리들을 포함하지 않을 수도 있으며, 대신에 버스를 통해 시스템 메모리 (10) 를 이용할 수도 있다. 그래픽 메모리 (40) 는 예를 들어 랜덤 액세스 메모리 (RAM), 정적 RAM (SRAM), 동적 RAM (DRAM), 소거가능 프로그램가능 ROM (EPROM), 전기적으로 소거가능 프로그램가능 ROM (EEPROM), 플래시 메모리, 자기 데이터 매체들 또는 광학 스토리지 매체들과 같은 하나 이상의 휘발성 또는 비휘발성 메모리들 또는 스토리지 디바이스들을 포함할 수도 있다.

[0020] 일부 예들에서, GPU (12) 는 시스템 메모리 (10) 에 완전히 형성된 이미지를 저장할 수도 있다. 디스플레이 프로세서 (14) 는 시스템 메모리 (10) 로부터 이미지를 추출하고 디스플레이 (8) 의 화소들로 하여금 이미지를 디스플레이하기 위해 조명하게 하는 값들을 출력할 수도 있다. 디스플레이 (8) 는 GPU (12) 에 의해 생성된 이미지 콘텐츠를 디스플레이하는 디바이스 (2) 의 디스플레이일 수도 있다. 디스플레이 (8) 는 액정 디스플레이 (LCD), 유기 발광 다이오드 디스플레이 (OLED), 음극선관 (CRT) 디스플레이, 플라즈마 디스플레이, 또는 다른 타입의 디스플레이 디바이스일 수도 있다.

[0021] 본 개시의 양태들에 따르면, GPU (12) 는 계층적 데이터 구조에서 장면의 복수의 프리미티브들을 조직화할 수도 있으며, 여기서, 복수의 바운딩 볼륨들은 계층적 데이터 구조의 복수의 노드들과 연관된다. GPU (12) 는 또한 메모리에서의 오프-스크린 렌더 타겟으로 복수의 바운딩 볼륨들 각각의 표현을 래스터화할 수도 있다. GPU (12) 는 또한 오프-스크린 렌더 타겟에서 제 1 레이와 교차하는 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 년-루트 노드를 결정할 수도 있다. GPU (12) 는 또한 장면 내의 제 2 레이가 복수의 프리미티브들 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터 시작하여 계층적 데이터 구조를 횡단할 수도 있다.

[0022] 도 2 는 도 1 의 CPU (6), GPU (12), 및 시스템 메모리 (10) 의 예시의 구현들을 더욱 상세히 도시하는 블록도이다. 도 2 에 도시된 바와 같이, CPU (6) 는 각각이 CPU (6) 상에서 실행하는 하나 이상의 소프트웨어 애플리케이션들 또는 서비스들일 수도 있는 적어도 하나의 소프트웨어 애플리케이션 (18), 그래픽 API (20), 및 GPU 구동기 (22) 를 포함할 수도 있다.

[0023] CPU (6) 및 GPU (12) 에게 이용가능한 메모리는 시스템 메모리 (10) 및 프레임 버퍼 (16) 를 포함할 수도 있다.



프레임 버퍼 (16) 는 시스템 메모리 (10) 의 부분일 수도 있거나 시스템 메모리 (10) 와는 별개일 수도 있고, 렌더링된 이미지 데이터를 저장할 수도 있다.

[0024] 소프트웨어 애플리케이션 (18) 은 GPU (12) 의 기능성을 이용하는 임의의 애플리케이션일 수도 있다. 예를 들어, 소프트웨어 애플리케이션 (18) 은 GUI 애플리케이션, 운영 시스템, 휴대용 맵핑 애플리케이션, 엔지니어링 또는 미술 애플리케이션들을 위한 컴퓨터 원조 설계 프로그램, 비디오 게임 애플리케이션, 또는 2D 또는 3D 그래픽을 사용하는 다른 타입의 소프트웨어 애플리케이션일 수도 있다.

[0025] 소프트웨어 애플리케이션 (18) 은 그래픽 사용자 인터페이스 (GUI) 및/또는 그래픽 장면을 렌더링하도록 GPU (12) 에게 명령하는 하나 이상의 드로잉 명령들을 포함할 수도 있다. 예를 들어, 드로잉 명령들은 GPU (12) 에 의해 렌더링될 하나 이상의 그래픽 프리미티브들의 세트를 정의하는 명령들을 포함할 수도 있다. 일부 예들에서, 드로잉 명령들은, 집합적으로, GUI 에서 사용되는 복수의 윈도우잉 표면들의 전부 또는 부분을 정의할 수도 있다. 추가적인 예들에서, 드로잉 명령들은, 집합적으로, 애플리케이션에 의해 정의된 모델 공간 또는 세계 공간 내의 하나 이상의 그래픽 오브젝트들을 포함하는 그래픽 장면의 전부 또는 부분을 정의할 수도 있다.

[0026] 소프트웨어 애플리케이션 (18) 은 디스플레이가능한 그래픽 이미지들로 하나 이상의 그래픽 프리미티브들을 렌더링하기 위해 GPU (12) 로 하나 이상의 커맨드들을 발생하기 위해, 그래픽 API (20) 를 통해 GPU 구동기 (22) 를 호출할 수도 있다. 예를 들어, 소프트웨어 애플리케이션 (18) 은 GPU (12) 로 프리미티브 정의들을 제공하기 위해 그래픽 API (20) 를 통해 GPU 구동기 (22) 를 호출할 수도 있다. 일부 예들에서, 프리미티브 정의들은 드로잉 프리미티브들, 예를 들어, 삼각형들, 직사각형들, 삼각형 팬들, 삼각형 스트립들 등의 리스트의 형태로 GPU (12) 로 제공될 수도 있다.

[0027] GPU (12) 로 제공된 프리미티브 정의들은 렌더링될 프리미티브들과 연관된 하나 이상의 정점들을 특징하는 정점 사양들을 포함할 수도 있다. 정점 사양들은 각각의 정점에 대한 위치 좌표들, 및 일부 예들에서, 예를 들어 컬러 좌표들, 법선 벡터들, 및 텍스처 좌표들과 같은, 정점과 연관된 다른 속성들을 포함할 수도 있다. 프리미티브 정의들은 또한 프리미티브 타입 정보 (예를 들어, 삼각형, 직사각형, 삼각형 팬, 삼각형 스트립 등), 스케일링 정보, 회전 정보 등을 포함할 수도 있다.

[0028] GPU 구동기 (22) 로 소프트웨어 애플리케이션 (18) 에 의해 발행된 명령들에 기초하여, GPU 구동기 (22) 는 프리미티브를 렌더링하기 위해 GPU (12) 가 수행할 하나 이상의 동작들을 특징하는 하나 이상의 커맨드들을 포괄 레이팅할 수도 있다. GPU (12) 가 CPU (6) 로부터 커맨드를 수신할 때, 프로세서 클러스터 (46) 는 커맨드를 디코딩하기 위해 그래픽 프로세싱 파이프라인을 실행할 수도 있고 커맨드에서 특정된 동작을 수행하도록 그래픽 프로세싱 파이프라인을 구성할 수도 있다. 예를 들어, 그래픽 프로세싱 파이프라인의 커맨드 엔진은 프리미티브 데이터를 판독하고 그래픽 프로세싱 파이프라인에서의 다른 그래픽 파이프라인 스테이지들에 의한 사용을 위해 프리미티브들 내로 그 데이터를 어셈블링할 수도 있다. 특정된 동작들을 수행한 후, GPU (12) 는 디스플레이 디바이스와 연관된 프레임 버퍼 (16) 로 렌더링된 데이터를 출력한다.

[0029] 프레임 버퍼 (16) 는 GPU (12) 에 대한 목적지 화소들을 저장한다. 각각의 목적지 화소는 고유한 스크린 화소 로케이션과 연관될 수도 있다. 일부 예들에서, 프레임 버퍼 (16) 는 각각의 목적지 화소에 대한 컬러 컴포넌트들 및 목적지 알파값을 저장할 수도 있다. 예를 들어, 프레임 버퍼 (16) 는 각각의 화소에 대한 적색, 녹색, 청색, 알파 (RGBA) 컴포넌트들을 저장할 수도 있으며, 여기서 "RGB" 컴포넌트들은 컬러 값들에 대응하고 "A" 컴포넌트는 목적지 알파값에 대응한다. 프레임 버퍼 (16) 는 또한 각각의 목적지 화소에 대한 깊이 값들을 저장할 수도 있다. 이러한 방식으로 프레임 버퍼 (16) 는 그래픽 데이터 (예를 들어, 표면) 을 저장한다고 말해질 수도 있다. 프레임 버퍼 (16) 및 시스템 메모리 (10) 가 별개의 메모리 유닛들인 것으로서 도시되지만, 다른 예들에서, 프레임 버퍼 (16) 는 시스템 메모리 (10) 의 부분일 수도 있다. 일단 GPU (12) 가 프레임 버퍼 (16) 로 프레임의 모든 화소들을 렌더링했다면, 프레임 버퍼는 디스플레이를 위해 디스플레이 (8) 로 완성된 프레임을 출력할 수도 있다.

[0030] 프로세서 클러스터 (46) 는 하나 이상의 프로그램가능 프로세싱 유닛들 (42) 및/또는 하나 이상의 고정 기능 프로세싱 유닛들 (44) 을 포함할 수도 있다. 프로그램가능 프로세싱 유닛 (42) 은 예를 들어 CPU (6) 로부터 GPU (12) 로 다운로드되는 하나 이상의 셰이더 프로그램들을 실행하도록 구성되는 프로그램가능 셰이더 유닛들을 포함할 수도 있다. 일부 예들에서, 프로그램가능 프로세싱 유닛들 (42) 은 "셰이더 프로세서들" 또는 "통합된 셰이더들" 로서 지칭될 수도 있고, 그래픽을 렌더링하기 위해 지오메트리, 정점, 화소, 또는 다른 셰이딩 동작들을 수행할 수도 있다. 셰이더 유닛들은 각각 페칭 (fetching) 및 디코딩 동작들을 위한 하나 이상

의 컴포넌트들, 산술 계산들을 수행하기 위한 하나 이상의 ALU 들, 하나 이상의 메모리들, 캐시들, 및 레지스터들을 포함할 수도 있다.

[0031] GPU (12) 는 그래픽 프로세싱 파이프라인에서의 정점 셰이더 스테이지, 테셀레이션 스테이지들, 지오메트리 셰이더 스테이지, 래스터화 스테이지 및 프래그먼트 셰이더 스테이지 중 하나 이상을 실행하기 위해 프로그램가능 프로세싱 유닛들 (42) 로 커맨드들을 전송함으로써 정점 셰이딩, 허 (hull) 셰이딩, 도메인 셰이딩, 지오메트리 셰이딩, 프래그먼트 셰이딩 등과 같은 다양한 셰이딩 동작들을 수행하기 위해 프로그램가능 프로세싱 유닛들 (42) 을 지정할 수도 있다. 일부 예들에서, GPU 구동기 (22) 는 CPU (6) 상에서 실행하는 컴파일러로 하여금 하나 이상의 셰이더 프로그램들을 컴파일하고, GPU (12) 내에 포함된 프로그램가능 프로세싱 유닛들 (42) 상으로 그 컴파일된 셰이더 프로그램들을 다운로드하게 할 수도 있다.

[0032] 셰이더 프로그램들은 예를 들어 OpenGL 셰이딩 언어 (GL SL), 고레벨 셰이딩 언어 (HLSL), 그래픽을 위한 C (Cg) 셰이딩 언어, OpenCL C 커널 등과 같은 고레벨 셰이딩 언어로 작성될 수도 있다. 컴파일된 셰이더 프로그램들은 GPU (12) 내의 프로그램가능 프로세싱 유닛들 (42) 의 동작을 제어하는 하나 이상의 명령들을 포함할 수도 있다. 예를 들어, 셰이더 프로그램들은 정점 셰이더 스테이지의 기능들을 수행하기 위해 프로그램가능 프로세싱 유닛들 (42) 에 의해 실행될 수도 있는 정점 셰이더 프로그램들, 테셀레이션 스테이지들의 기능들을 수행하기 위해 프로그램가능 프로세싱 유닛들 (42) 에 의해 실행될 수도 있는 테셀레이션 셰이더 프로그램들, 지오메트리 셰이더 스테이지의 기능들을 수행하기 위해 프로그램가능 프로세싱 유닛들 (42) 에 의해 실행될 수도 있는 지오메트리 셰이더 프로그램들, 및/또는 프래그먼트 셰이더 스테이지의 기능들을 수행하기 위해 프로그램가능 프로세싱 유닛들 (42) 에 의해 실행될 수도 있는 프래그먼트 셰이더 프로그램들을 포함할 수도 있다. 정점 셰이더 프로그램은 프로그램가능 정점 셰이더 유닛 또는 통합된 셰이더 유닛의 실행을 제어하고, 하나 이상의 정점 마다의 동작들을 특징하는 명령들을 포함할 수도 있다.

[0033] 프로세서 클러스터 (46) 는 또한 고정 기능 프로세싱 유닛들 (44) 을 포함할 수도 있다. 고정 기능 프로세싱 유닛들 (44) 은 소정의 기능들을 수행하도록 하드-와이어링되는 하드웨어 논리 회로를 포함할 수도 있다. 고정 기능 프로세싱 유닛들 (44) 은 상이한 기능들을 수행하기 위해, 예를 들어, 하나 이상의 제어 신호들을 통해 구성가능할 수도 있지만, 고정 기능 하드웨어는 통상 사용자-컴파일된 프로그램들을 수신할 수 있는 프로그램 메모리를 포함하지 않는다. 일부 예들에서, 프로세서 클러스터 (46) 내의 고정 기능 프로세싱 유닛들 (44) 은, 예를 들어, 그래픽 프로세싱 파이프라인의 래스터화 스테이지의 기능들을 수행하기 위해, 예를 들어, 깊이 테스트, 시저스 (scissors) 테스트, 알파 블렌딩, 저해상도 깊이 테스트 등과 같은 래스터 동작들을 수행하는 프로세싱 유닛들을 포함할 수도 있다.

[0034] 그래픽 메모리 (40) 는 GPU (12) 의 집적회로로 물리적으로 통합되는 온-칩 스토리지 또는 메모리일 수도 있다. 그래픽 메모리 (40) 가 온-칩이기 때문에, GPU (12) 는 시스템 버스를 통해 시스템 메모리(10) 로부터 값들을 판독하거나 시스템 메모리(10) 로 값들을 기입하는 것보다 더 빠르게 그래픽 메모리 (40) 로부터 값들을 판독하거나 그래픽 메모리 (40) 로 값들을 기입할 수 있다.

[0035] 본 개시의 양태들에 따르면, 프로세서 클러스터 (46) 는 3D 장면 내의 하나 이상의 그래픽 오브젝트들을 구성하는 복수의 프리미티브들을 렌더링하는 것을 포함하는, 모델 공간 또는 세계 공간 내에 하나 이상의 그래픽 오브젝트들을 포함하는 3차원 (3D) 그래픽 장면을 렌더링하기 위해 그래픽 프로세싱 파이프라인을 실행하기 위해 상술된 동작들을 수행할 수도 있다. 프로세서 클러스터 (46) 는 또한 3D 그래픽 장면의 어느 화소들이 광원에 의해 조명되는지를 결정하기 위해, 3D 그래픽 장면의 화소들을 통한 광원으로부터의 광의 경로를 트레이싱함으로써 3D 그래픽 장면의 레이 트레이싱을 수행할 수도 있다.

[0036] 3D 그래픽 장면의 레이 트레이싱을 수행하는 것의 부분으로서, 프로세서 클러스터 (46) 는 광원에 의해 조명되지 않는 (및 따라서 그늘져 있는) 3D 그래픽 장면의 표면들을 결정하기 위해 3D 그래픽 장면의 새도우 렌더링을 수행할 수도 있다. 그러한 표면들은 하나 이상의 다른 솔리드 (solid) 표면들이 광원에 의해 방출된 광선들이 이들 표면들에 도달하는 것을 차단하기 때문에 그늘져 있을 수도 있다. 3D 그래픽 장면 내의 특징의 로케이션이 표면에 의해 광원으로부터 셰이딩되는지 여부를 결정하기 위해, 프로세서 클러스터 (46) 는 광원의 방향으로 그 특징의 로케이션으로부터 새도우 레이라고 지칭되는 벡터를 캐스트할 수도 있다. 프로세서 클러스터 (46) 가 그 로케이션으로부터 캐스트된 새도우 레이가 그 로케이션과 광원 사이에 위치해 있는 프리미티브와 교차한다고 결정하는 경우, 프로세서 클러스터 (46) 는 새도우 레이가 발원하는 그 로케이션이 광원으로부터 폐색되는 것을 간주할 수도 있다.

[0037] 3D 그래픽 장면의 특징의 로케이션으로부터 발원하고 3D 그래픽 장면에 대한 광원을 향해 지향되는 특징의 새도우

우 레이가 3D 그래픽 장면 내의 프리미티브와 교차하는지 여부를 결정하기 위해, GPU (12) 는 3D 그래픽 장면 내의 프리미티브들을 장면 프리미티브들 (예를 들어, 삼각형들) 을 계층적으로 그룹핑하는, 가속화 데이터 구조 (ADS) (41) 와 같은 계층적 구조로 조직화할 수도 있다. GPU (12) 는 그래픽 메모리 (40) 에, 시스템 메모리 (10) 에, 프로세서 클러스터 (46) 의 셰이더 메모리 (미도시) 에, 또는 공유된 시스템/그래픽 메모리 (미도시) 에 ADS (41) 를 저장할 수도 있다. GPU (12) 가 새도우 레이-프리미티브 교차들을 결정하기 위해 ADS (41) 를 사용하는 방법에 대한 상세들은 도 3 및 도 4 를 참조하여 더욱 상세히 논의된다.

[0038] 도 3 은 GPU (12) 가 새도우 레이 트레이싱을 수행할 수도 있는 예시의 그래픽 장면 및 바운딩 볼륨들로의 그래픽 장면의 예시의 파티셔닝을 도시하는 개념도이다. 도 3 에 도시된 바와 같이, 그래픽 장면 (50) 은 프리미티브들 (52A-52E) (이하 "프리미티브들 (52)") 을 포함하는 2D 또는 3D 그래픽 장면일 수도 있다. 새도우 맵핑 프로세스의 부분으로서, GPU (12) 는 그래픽 장면 (50) 내의 특징의 로케이션에 대해, 광원을 향해 그 특징의 로케이션으로부터 발원하는 새도우 레이가 프리미티브들 (52) 중 하나와 교차하는지 여부를 결정할 수도 있다. GPU (12) 가 새도우 레이 광원과 새도우 레이가 발원하는 로케이션 사이에 위치해 있는 프리미티브와 교차한다고 결정하는 경우, 새도우 레이가 발원하는 그 로케이션은 교차된 프리미티브에 의해 광원으로부터 새도우잉되고, 따라서 광원에 의해 조명되지 않는다.

[0039] GPU (12) 는 그래픽 장면 (50) 을 분할하고, 그래픽 장면 (50) 의 분할된 부분들을 계층적으로 배열하며, 그래픽 장면 (50) 의 분할된 부분들의 계층을 회귀적으로 횡단하는 것에 의해 프리미티브들 (52) 중의 프리미티브가 특징의 새도우 레이와 교차하는지 여부를 조직적으로 결정할 수도 있다. GPU (12) 는 프리미티브들 (52) 을 바운딩 볼륨들 (56A-56E) ("바운딩 볼륨들 (56)") 로 개념적으로 파티셔닝할 수도 있다. 바운딩 볼륨들 (56) 은 인클로징된 프리미티브들의 모든 점들이 놓일 수도 있는 최소화된 영역을 갖는 바운딩 박스들일 수도 있는 축-정렬 바운딩 박스들 (axis-aligned bounding boxes: AABBs) 일 수도 있다. 바운딩 박스들은 그 박스들의 에지들이 좌표축 (예를 들어, x, y, 및 z 축) 에 평행할 수 있도록 축-정렬될 수도 있다.

[0040] 바운딩 볼륨 (56A) 은 그래픽 장면 (50) 의 모든 프리미티브들 (52) 을 바운딩하는 바운딩 박스일 수도 있다. 바운딩 볼륨들 (56B 및 56C) 은 바운딩 볼륨들 (56B 및 56C) 이 바운딩 볼륨 (56A) 에 의해 바인딩된 장면 (50) 의 부분의 서브셋을 바운딩한다는 점에서 바운딩 볼륨 (56A) 의 서브셋들일 수도 있다. 바운딩 볼륨 (56B) 은 프리미티브들 (52A 및 52B) 을 바인딩할 수도 있고, 바운딩 볼륨 (56C) 은 프리미티브들 (52C, 52D, 및 52E) 을 바인딩할 수도 있다. 바운딩 볼륨들 (56D 및 56E) 은 바운딩 볼륨 (56C) 의 서브셋들일 수도 있고, 바운딩 볼륨 (56B) 에 의해 바인딩된 장면 (50) 의 부분의 서브셋을 바인딩할 수도 있다. 바운딩 볼륨 (56D) 은 프리미티브들 (52C 및 52D) 을 바운딩할 수도 있고, 바운딩 볼륨 (56E) 은 프리미티브 (52E) 를 바운딩할 수도 있다.

[0041] 도 3 에 도시된 예에서, GPU (12) 는 프리미티브들 (52) 을 5 개의 바운딩 볼륨들 (56A-56E) 로 파티셔닝할 수도 있다. GPU (12) 는 5 개의 바운딩 볼륨들 (56A-56E) 에 제한되는 것이 아니라, 장면 및 장면 내의 프리미티브들의 수에 따라, 5 개의 바운딩 볼륨들 (56A-56E) 보다 더 많거나 더 적은 바운딩 볼륨들을 사용할 수도 있다. 일부 예들에서, GPU (12) 는 각각 프리미티브들 (52A 및 52B) 을 개별적으로 바운딩하기 위해 바운딩 볼륨 (56B) 의 서브셋들로서 추가적인 바운딩 볼륨들을 생성할 수도 있다. 일부 예들에서, CPU (6) 는 또한 프리미티브들 (52) 을 바운딩 볼륨들 (56) 로 파티셔닝하도록 구성될 수도 있다.

[0042] 바운딩 볼륨들 (56) 은 GPU (12) 가 가능한 새도우 레이-프리미티브 교차들을 결정하기 위해 계층적 구조를 횡단할 수 있도록 계층적 구조 내로 배열될 수도 있다. 도 4 는 도 3 에 도시된 바운딩 볼륨들 (56) 및 프리미티브들 (52) 과 연관되는 노드들을 갖는 예시의 계층적 데이터 구조를 도시한다. 상술된 바와 같이, 장면의 장면 프리미티브들은 ADS (41) 와 같은 계층적 구조 내로 조직화될 수도 있고, GPU (12) 는 가능한 새도우 레이-프리미티브 교차들을 결정하기 위해 ADS (41) 를 횡단할 수도 있다. 도 4 에 도시된 바와 같이, ADS (41) 의 하나의 예는 그래픽 장면 (50) 의 바운딩 볼륨들 (56) 및 프리미티브들 (52) 과 연관된 바운딩 볼륨 계층 (bounding volume hierarchy: BVH) 트리 (60) 의 노드들 (62A-62E) 이 트리-형상 구조로 계층적으로 배열되는 BVH 트리 (60) 일 수도 있다.

[0043] 구체적으로, GPU (12) 는 다른 바운딩 볼륨을 인클로징하는 바운딩 볼륨과 연관된 노드가 인클로징된 바운딩 볼륨과 연관된 노드의 부모 노드일 수 있도록 BVH 트리 (60) 를 배열할 수도 있다. 도 3 의 예에서, 바운딩 볼륨 (56C) 은 바운딩 볼륨 (56C) 의 서브셋들인 바운딩 볼륨들 (56D 및 56E) 을 인클로징하기 때문에, 바운딩 볼륨 (56C) 과 연관된 노드 (62C) 는 각각 바운딩 볼륨들 (56D 및 56E) 과 연관된 노드들 (62D 및 62E) 의 부모 노드일 수도 있다. 따라서, 루트 노드 (62A) 는 바운딩 볼륨 (56) 과 연관될 수도 있고, 내부 노드

(62C)는 바운딩 볼륨 (56C)와 연관될 수도 있으며, 리프 노드들 (62B, 62D, 및 62E)은 각각 바운딩 볼륨들 (56B, 56D, 및 56E)와 연관될 수도 있다.

[0044] 루트 노드 (62A) 이외의 BVH 트리 (60)의 노드들은 BVH 트리 (60)의 난-루트 노드들로서 지칭될 수도 있다.

예를 들어, 내부 노드 (62C) 및 리프 노드들 (62B, 62D, 및 62E)은 BVH 트리 (60)의 난-루트 노드들로서 지칭될 수도 있다. 리프 노드들 (62B, 62D, 및 62E)은 각각 프리미티브들 (52)중 적어도 하나의 프리미티브와 링크될 수도 있다. 예를 들어, 리프 노드 (62B)는 리프 노드 (62B)와 연관된 바운딩 볼륨 (56B)이 프리미티브들 (52A 및 52B)을 인클로징하기 때문에 프리미티브들 (52A 및 52B)과 링크될 수도 있고, 리프 노드 (62D)는 리프 노드 (62D)와 연관된 바운딩 볼륨 (56D)이 프리미티브들 (52C 및 52D)을 인클로징하기 때문에 프리미티브들 (52C 및 52D)과 링크될 수도 있고, 리프 노드 (62E)는 리프 노드 (62E)와 연관된 바운딩 볼륨 (56E)이 프리미티브 (52E)을 인클로징하기 때문에 프리미티브 (52E)과 링크될 수도 있다. BVH 트리 (60)는 계층적 데이터 구조 (60)의 각각의 난-리프 노드가 많아야 2개의 차일드 노드들을 갖기 때문에, 그리고 리프 노드들 (62B, 62D, 및 62E)이 동일하지 않은 깊이들을 가질 수도 있기 때문에 불균형 이진 트리로 고려될 수도 있다.

[0045] GPU (12)는 BVH 트리 (60)의 노드들 (62)과 연관된 바운딩 볼륨들 (56)에 대해 레이-박스 교차 테스트들을 수행함으로써 새도우 레이가 프리미티브들 (52)중의 프리미티브와 교차하는지 여부를 결정하기 위해 BVH 트리 (60)를 횡단할 수도 있다. GPU (12)는 루트 노드 (62A)와 연관된 바운딩 볼륨 (56A)에 대해 레이-박스 교차 테스트를 수행함으로써 BVH 트리 (60)의 횡단을 시작할 수도 있다. GPU (12)가 새도우 레이가 바운딩 볼륨 (56A)과 교차한다고 결정하는 경우, GPU (12)는 노드 (62B)로 BVH 트리 (60)를 계속 횡단할 수도 있고, GPU (12)는 노드 (62B)와 연관된 바운딩 볼륨 (56B)에 대해 레이-박스 교차 테스트를 수행할 수도 있다.

[0046] GPU (12)가 새도우 레이가 바운딩 볼륨 (56B)과 교차하지 않는다고 결정하는 경우, GPU (12)는 노드 (62A)까지 위로 및 노드 (62C)까지 아래로 BVH 트리를 회귀적으로 횡단할 수도 있고, GPU (12)는 노드 (62C)와 연관된 바운딩 볼륨 (56C)에 대해 레이-박스 교차 테스트를 수행할 수도 있다. GPU (12)가 새도우 레이가 바운딩 볼륨 (56C)과 교차한다고 결정하는 경우, GPU (12)는 노드 (62D)로 BVH 트리 (60)를 계속 횡단할 수도 있고, GPU (12)는 노드 (62D)와 연관된 바운딩 볼륨 (56D)에 대해 레이-박스 교차 테스트를 수행할 수도 있다. GPU가 새도우 레이가 바운딩 볼륨 (56D)과 교차한다고 결정하는 경우, GPU (12)는 노드 (62D)에 링크된 프리미티브들에 대해 레이-프리미티브 교차 테스트들을 수행할 수도 있다.

[0047] 따라서, GPU (12)는 새도우 레이가 프리미티브 (52C) 또는 프리미티브 (52D)와 교차하는지 여부를 결정하기 위해 프리미티브 (52C)에 대해 레이-프리미티브 교차 테스트를 수행할 수도 있고, 또한 프리미티브 (52D)에 대해 레이-프리미티브 교차 테스트를 수행할 수도 있다. GPU (12)는 새도우 레이가 프리미티브 (52D)와 교차한다는 것을 프리미티브 (52D)에 대한 레이-프리미티브 교차 테스트로부터 결정할 수도 있다. 새도우 레이가 프리미티브 (예를 들어, 프리미티브 (52D))와 교차한다고 결정한 때에, GPU (12)는 새도우 레이가 발원하는 그래픽 장면 (50)내의 로케이션이 광원으로부터 폐색된다고 결정할 수도 있다.

[0048] GPU (12)가 새도우 레이가 프리미티브 (52D)와 교차하지 않는다고 결정하는 경우, GPU (12)는 노드 (62C)로 위로 그리고 노드 (62E)로 아래로 BVH 트리 (60)를 회귀적으로 횡단하기를 계속할 수도 있고, GPU (12)는 노드 (62E)와 연관된 바운딩 볼륨 (56E)에 대해 레이-박스 교차 테스트를 수행할 수도 있다. GPU (12)는, 그 레이-박스 교차 테스트에 기초하여, 새도우 레이가 바운딩 볼륨 (56E)과 교차하는지 여부를 결정할 수도 있고, 그 결정을 행한 때에 새도우 레이에 대한 BVH 트리 (60)의 횡단을 종료할 수도 있다.

[0049] BVH 트리 (60)가 내부 노드 (62C) 또는 리프 노드들 (62B, 62D, 및 62E)중 하나와 같은 난-루트 노드로부터 시작하여 횡단될 수 있는 경우, GPU (12)는 루트 노드 (62A)로부터 BVH 트리 (60)의 횡단을 시작하는 것에 비해 그것이 수행하는 레이 교차 테스트들의 수를 감소시킬 수도 있으며, 이것에 의해 레이에 의해 교차되는 프리미티브를 결정하는 효율을 증가시킨다. GPU (12)는 BVH 트리 (60)의 난-루트 (내부)노드와 연관된 바운딩 볼륨이 특정의 새도우 레이에 의해 교차된다고 결정함으로써 BVH 트리 (60)의 난-루트 노드로부터 BVH 트리 (60)의 횡단을 시작할 수도 있다. GPU (12)는 그래픽 메모리 (40)내의 오프-스크린 렌더 타겟으로 바운딩 볼륨들 (56)의 적어도 서브세트를 래스터화할 수도 있다. GPU (12)는, 오프-스크린 렌더 타겟으로 복수의 바운딩 볼륨들 (56)을 래스터화하는 것에 기초하여, BVH 트리 (60)의 횡단을 시작하기 위해 BVH 트리 (60)내의 시작 노드로서 BVH 트리 (60)의 난-루트 노드를 결정할 수도 있다. GPU (12)는 레이 (54)에 의해 교차되는 프리미티브를 결정하기 위해 시작 노드로부터 시작하여 BVH 트리 (60)를 횡단할 수도 있다.



- [0050] 새도우 레이들을 사용하여 그래픽 장면 (50) 과 같은 그래픽 장면의 새도우 렌더링을 수행하기 위해, GPU (12) 는 그래픽 장면 (50) 에 대한 광원의 관점으로부터 BVH 트리 (60) 의 바운딩 볼륨들 (56) 의 적어도 일부의 표현을 렌더링할 수도 있다. 바운딩 볼륨들 (56) 의 적어도 일부의 표현들이 광원의 관점으로부터 렌더링될 때, GPU (12) 는 그래픽 장면 (50) 내의 로케이션들로부터 발원하는 새도우 레이들이 프리미티브들 (52) 중의 프리미티브와 교차하는 경우 이들 로케이션들이 광원에 의해 조명되지 않는다고 결정할 수 있다.
- [0051] 상술된 바와 같이, GPU (12) 는 그래픽 장면 (50) 내의 특정의 로케이션으로부터 발원하는 새도우 레이가 프리미티브들 (52) 중의 프리미티브와 교차하는지 여부를 결정하기 위해 BVH 트리 (60) 와 같은 계층적 구조를 횡단할 수도 있다. GPU (12) 가, BVH 트리 (60) 의 횡단을 통해, 새도우 레이가 프리미티브들 (52) 중의 프리미티브와 교차한다고 결정하면, GPU (12) 는 새도우 레이가 발원하는 그래픽 장면 (50) 내의 특정의 로케이션이 그래픽 장면 (50) 에 대한 광원에 의해 조명되지 않는다고 결정할 수도 있다. GPU (12) 는 통상 특정의 로케이션이 장면 (50) 에 대한 광원에 의해 조명되는지 여부를 결정하기 위해 새도우 레이와 노드들 (62) 과 연관된 바운딩 볼륨들 (56) 사이의 레이-박스 교차 테스트들 및/또는 새도우 레이와 프리미티브들 (52) 사이의 레이-프리미티브 교차 테스트들을 수행함으로써 루트 노드 (62A) 로부터 BVH 트리 (60) 를 횡단할 수도 있다.
- [0052] 그러나, BVH 트리들은 다수의 다중 레벨들 깊이일 수 있다. 예를 들어, BVH 트리가 16 개의 레벨들을 포함한다면, GPU (12) 는 BVH 트리의 루트로부터 시작하는 것 대신에 BVH 트리의 난-루트 노드로부터 BVH 트리의 횡단을 시작함으로써 새도우 레이가 프리미티브를 교차하는지 여부를 더 효율적으로 결정할 수 있다. GPU (12) 가 오프-스크린 렌더 타겟으로 화소들을 빠르게 래스터화하기 위해 그리고 오프-스크린 렌더 타겟 내의 화소들을 빠르게 샘플링하기 위해 프로세서 클러스터 (46) 상에서 실행하는 화소 셰이더 프로그램들 및/또는 정점 셰이더 프로그램들을 사용할 수도 있기 때문에, GPU (12) 는 BVH 트리의 횡단을 시작할 난-루트 노드들을 결정하기 위해 이들 셰이더 프로그램들의 성능 특성들을 이용할 수도 있다.
- [0053] 도 5 는 예시의 오프-스크린 렌더 타겟으로 래스터화된 바운딩 볼륨들의 예시의 표현을 도시하는 블록도이다. GPU (12) 는, BVH 트리의 특정된 수의 상부 레벨들에 대해, 광원의 관점으로부터 그래픽 메모리 또는 시스템 메모리 (10) 내의 오프-스크린 렌더 타겟으로 BVH 트리의 노드들과 연관된 바운딩 볼륨들의 표현들을 래스터화하기 위해 프로세서 클러스터 (46) 상에서 실행하는 그것의 그래픽 프로세싱 파이프라인의 화소 셰이더 및/또는 정점 셰이더를 사용할 수도 있다. GPU (12) 는 GPU (12) 가 그래픽 장면 (50) 의 관점의 특정의 광원으로부터 바운딩 볼륨들의 표현들을 래스터화하도록 투영 행렬로 바운딩 볼륨들의 표현들을 변환할 수도 있다.
- [0054] 그래픽 장면 (50) 이 다수의 광원들을 포함하는 경우, GPU (12) 는 오프-스크린 렌더 타겟으로 바운딩 볼륨들의 표현들의 그러한 래스터화를, 각 광원에 대해 한번씩, 다수회 수행할 수도 있다. GPU (12) 는 바운딩 볼륨들의 표현들이 각 광원의 관점으로부터 래스터화되도록 각 광원에 대해 상이한 투영 행렬로 그러한 래스터화를 수행할 수도 있다.
- [0055] 일부 예들에서, 16-레벨 BVH 트리가 주어지는 경우, GPU (12) 가 BVH 트리의 상위, 예를 들어, 5-6 레벨들과만 연관된 바운딩 볼륨들의 표현을 래스터화할 수 있도록, GPU (12) 는 오프-스크린 렌더 타겟으로 BVH 트리의 특정된 상위 레벨들만을 래스터화할 수도 있다. 따라서, GPU (12) 는 일부 예들에서 오프-스크린 렌더 타겟 (64) 으로 BVH 트리의 모든 난-루트 노드들보다 적은 난-루트 노드들과 연관된 바운딩 볼륨들의 표현들을 래스터화할 수도 있다.
- [0056] 도 5 에 도시된 바와 같이, GPU (12) 는 그래픽 장면 (50) 에 대한 광원의 관점으로부터 오프-스크린 렌더 타겟 (64) 으로 BVH 트리 (60) 의 노드들 (62B-62E) 과 연관된 그래픽 장면 (50) 의 바운딩 볼륨들 (56B-56E) 의 표현들 (66A-66D) ("표현들 (66)") 을 래스터화할 수도 있다. GPU (12) 는 투영 행렬로 표현들 (66) 을 래스터화할 수도 있다. GPU (12) 는 그래픽 메모리 (40), 시스템 메모리 (10), 또는 임의의 다른 적합한 메모리에 오프-스크린 렌더 타겟 (64) 을 저장할 수도 있다.
- [0057] GPU (12) 는 바운딩 볼륨들 (56B-56E) 을 투영할 수도 있고, GPU (12) 의 프로세서 클러스터 (46) 의 셰이더 유닛들은 다각형들, 정육면체들 등과 같은 2차원 또는 3차원 표현들로서 BVH 트리 (60) 의 난-루트 노드들 (62B-62E) 과 연관된 바운딩 볼륨들 (56B-56E) 의 표현들을 래스터화할 수도 있다. 예를 들어, GPU (12) 의 하드웨어 래스터화기는 바운딩 볼륨들 (56) 각각을 렌더 타겟 (64) 내의 화소들로 주사-변환할 수도 있다. 하나의 예에서, GPU (12) 는 바운딩 볼륨들 (56B-56E) 의 표현들 (66) 로서 오프-스크린 렌더 타겟 (64) 으로 복수의 플랫 셰이딩된 정육면체들을 래스터화할 수도 있다. GPU (12) 는 또한 표현들 (66) 이 광원의 관

점으로부터 래스터화되도록 관점 행렬을 통해 표현들 (66) 각각을 스케일링 및 변환할 수도 있다. 일부 예들에서, GPU (12) 는 그래픽 장면 (50) 내의 바운딩 볼륨들 (56B-56E) 에 비해 상대적으로 더 낮은 해상도로 바운딩 볼륨들 (56B-56E) 의 표현들 (66) 을 래스터화할 수도 있다. 이러한 방식으로, GPU (12) 는 레이-프리미티브 교차들을 결정함에 있어서의 그것의 성능을 더 증가시킬 수도 있다.

[0058] GPU (12) 는 상이한 컬러 값을 BVH 트리 (60) 의 노드들 (62) 각각과 연관시킬 수도 있고, 바운딩 볼륨들 (56) 의 각각의 바운딩 볼륨에 대해, 표현들 (66) 의 바운딩 볼륨의 연관된 표현으로서, 각각의 바운딩 볼륨과 연관되는 BVH 트리 (60) 의 노드와 연관된 컬러 값을 갖는 화소들의 블록을 래스터화할 수도 있다. 이러한 방식으로, GPU (12) 는 표현들 (66) 각각의 컬러가 BVH 트리 (60) 내의 연관된 노드를 표시하는 노드 인덱스를 표현할 수 있도록 상이한 컬러로 바운딩 볼륨들 (56B-56E) 의 표현들 (66) 각각을 래스터화할 수도 있다.

[0059] GPU (12) 는 투영된 바운딩 볼륨들 (56B-56E) 의 표준 깊이 테스트를 수행함으로써 및 투영된 바운딩 볼륨들 (56B-56E) 의 상이한 깊이들을 표현하기 위해 표현들 (66) 의 화소들에 상이한 컬러 값들을 할당함으로써 표현들 (66) 의 컬러를 결정할 수도 있다. GPU (12) 는 표현들 (66) 의 화소들에 할당된 컬러 값들을 BVH 트리 (60) 의 노드들 (62) 과 연관시킬 수도 있다. 이러한 방식으로, GPU (12) 는 화소의 컬러 값을 샘플링함으로써 렌더 타겟 (64) 내의 화소와 연관된 BVH 트리 (60) 내의 노드를 결정할 수도 있다. 렌더 타겟 (64) 으로 표현들 (66) 을 래스터화하는 것의 부분으로서, 렌더 타겟 (64) 내의 화소에 대해, GPU (12) 가 그것을 가능하게는 로케이션 (72A) 으로부터 발원하는 새도우 레이 (70A) 또는 위치 (72B) 로부터 발원하는 새도우 레이 (70B) 와 같은 하나 이상의 새도우 레이들과 교차하는 것으로서 맵핑할 수 있도록, GPU (12) 는 렌더 타겟 (64) 내의 화소들로의 새도우 레이들의 맵핑을 결정할 수도 있다. 일부 예들에서, 렌더 타겟 (64) 은 화소와 새도우 레이 사이의 일대일 맵핑을 가질 수도 있다. 다른 예들에서, 표현들 (66) 이 렌더 타겟 (64) 으로 (대응하는 바운딩 볼륨들 (56) 의 해상도에 비해) 상대적으로 더 낮은 해상도로 래스터화된다면, 화소는 다수의 새도우 레이들로 맵핑될 수도 있다.

[0060] 그래픽 장면 내의 특징의 로케이션으로부터 발원하는 특징의 새도우 레이에 대한 잠재적인 새도우 레이-프리미티브 교차들을 결정하기 위해, GPU (12) 는 각각의 레이에 대해 그 새도우 레이로 맵핑하는 렌더 타겟 (64) 내의 화소 로케이션을 결정할 수도 있다. 시점 및 방향을 갖는 새도우 레이가 주어지는 경우, GPU (12) 는 광원의 투영 행렬에 기초하여, 렌더 타겟 (64) 내의 화소 로케이션에 새도우 레이를 맵핑할 수도 있다. 그러한 화소 로케이션의 경우, GPU (12) 는 화소의 컬러 값을 샘플링하고 BVH 트리 (60) 의 횡단을 시작하기 위해 시작 노드로서 그 샘플링된 컬러 값과 연관된 노드를 결정할 수도 있다.

[0061] 도 5 의 예에서, GPU (12) 는 화소 (76A) 가 새도우 레이 (70A) 로 맵핑되는 렌더 타겟 (64) 내의 화소라고 결정할 수도 있다. GPU (12) 는 화소 (76A) 의 컬러 값을 결정하기 위해 렌더 타겟 (64) 내의 화소 (76A) 를 샘플링하고, 화소 (76A) 의 그 샘플링된 컬러 값이 노드 (62B) 에 대한 노드 인덱스와 연관된 컬러 값과 동일하다고 결정할 수도 있고, 이것에 의해 그 화소 (76A) 를 노드 (62B) 와 연관시킬 수도 있다. 따라서, GPU (12) 는 BVH 트리 (60) 를 횡단하기 위한 시작 노드로서 노드 (62B) 를 설정할 수도 있고, 노드 (62B) 로부터 BVH 트리 (60) 의 횡단을 시작할 수도 있다.

[0062] GPU (12) 가 노드 (62B) 로부터 시작하여 BVH 트리 (60) 를 횡단함에 따라, GPU (12) 는 먼저 노드 (62B) 와 연관된 바운딩 볼륨 (56B) 에 대한 새도우 레이-박스 교차 테스트를 수행할 수도 있다. GPU (12) 가 새도우 레이 (70A) 가 바운딩 볼륨 (56B) 과 교차한다고 결정하는 경우, GPU (12) 는 노드 (62D) 에 링크되는 프리미티브 (52A) 에 대해 레이-프리미티브 교차 테스트를 수행할 수도 있다. GPU (12) 가 새도우 레이 (70A) 가 프리미티브 (52A) 와 교차하지 않는다고 결정하는 경우, GPU (12) 는 노드 (62B) 까지 위로 BVH 트리 (60) 를 회귀적으로 횡단할 수도 있고, 노드 (62B) 에 링크된 프리미티브 (52B) 에 대해 레이-프리미티브 교차 테스트를 수행할 수도 있다. GPU (12) 는 새도우 레이 (70A) 가 프리미티브 (52B) 와 교차하지 않는다는 것을 프리미티브 (52B) 에 대한 레이-프리미티브 교차 테스트로부터 결정할 수도 있다. GPU (12) 가 내부 노드 (62B) 로부터 BVH 트리 (60) 의 횡단을 시작하고 새도우 레이 (70A) 가 프리미티브들 (52A 또는 52B) 와 교차하지 않는다고 결정하기 때문에, GPU (12) 는 새도우 레이 (70A) 에 대한 BVH 트리 (60) 의 횡단을 종료할 수도 있다.

따라서, GPU (12) 는 새도우 레이 (70A) 가 장면 (50) 내의 임의의 프리미티브들과 교차하지 않기 때문에, 로케이션 (72A) 이 그로부터 새도우 레이 (70A) 가 그래픽 장면 (50) 에 대한 광원에 의해 조명되는 (및 광원으로부터 폐색되지 않는) 로케이션이라고 결정할 수도 있다. 알 수 있듯이, 프리미티브가 새도우 레이 (70A) 와 교차하는지 여부를 결정하기 위한 BVH 트리 (60) 의 횡단은 GPU (12) 가 루트 노드로부터 시작하여 BVH 트리 (60) 를 횡단하도록 요구되는 경우와 대조적으로 상대적으로 더 적은 레이-박스 교차 테스트들을 수행하는 것을

포함할 수도 있다.

[0063] 일부 예들에서, 뷰잉 각도에 따라, 임의의 2 개의 노드들의 바운딩 볼륨들의 표현들은 스크린상에 투영될 때 중첩할 수도 있다. 이러한 경우에, BVH 트리 (60) 의 횡단은 루트 노드가 아닐 수도 있는, 2 개의 중첩 바운딩 볼륨들의 최저 공통 조상으로부터 시작할 수도 있다. 바운딩 볼륨들의 2 개의 표현들이 중첩하는 경우, 그리고 레이가 그 2 개의 바운딩 볼륨들의 중첩된 영역에서 교차하는 경우, GPU (12) 는 바운딩 볼륨들과 연관된 노드들 중 최저 공통 조상 노드를 결정할 수도 있고, 그 2 개의 중첩 표현들에 의해 표현된 바운딩 볼륨들과 연관된 노드들 중 최저 공통 조상 노드로부터 BVH 트리 (60) 의 횡단을 시작할 수도 있다. 예를 들어, 렌더 타겟 (64) 으로 표현들 (66) 을 래스터화하는 동안, GPU (12) 는 표현 (66C) 및 표현 (66D) 이 영역 (69) 에서 중첩한다고 결정할 수도 있으며, 여기서 표현 (66C) 은 노드 (62D) 와 연관되고 표현 (66D) 은 노드 (62E) 와 연관되고, 여기서 노드 (62D) 및 노드 (62E) 는 BVH 트리 (60) 내의 동일한 레벨에 있다. GPU (12) 는 노드들 (62D 및 62E) 의 최저 공통 조상 노드와 연관된 컬러 값을 결정할 수도 있고, 영역 (69) (즉, 중첩의 영역) 내의 화소들의 컬러 값을 노드들 (62D 및 62E) 의 최저 공통 조상 노드와 연관된 동일한 컬러 값으로 설정할 수도 있다. 이러한 예에서, GPU (12) 는 노드 (62C) 가 노드들 (62D 및 62E) 의 최저 공통 조상 노드라고 결정할 수도 있고, 노드 (62C) 와 연관되는 표현 (66B) 의 컬러 값으로 영역 (69) 내의 화소들의 컬러 값을 설정할 수도 있다. 특정의 레이가 영역 (69) 내의 화소 로케이션에 맵핑하는 경우, GPU 는 임의의 가능한 레이-프리티미티브 교차들을 결정하기 위해 루트 노드 (62A) 가 아니고 노드 (62C) 로부터 BVH 트리 (60) 의 횡단을 시작할 수도 있다.

[0064] 예를 들어, GPU (12) 가 로케이션 (72B) 으로부터 발하는 새도우 레이 (70A) 가 영역 (69) 내에 놓여 있는 화소 (76B) 에 맵핑한다고 결정하는 경우, GPU (12) 는 맵핑된 화소 (76B) 의 컬러를 샘플링하고 샘플 화소의 컬러 값이 노드 (62C) 와 연관된 컬러 값과 동일하다고 결정함으로써, 루트 노드 (62A) 가 아닌 노드 (62C) 로부터 BVH 트리 (60) 의 횡단을 시작하기로 결정할 수도 있다. 노드 (62C) 가 렌더 타겟 (64) 내의 표현 (66B) 에 의해 표현된 바운딩 볼륨 (56C) 과 연관되기 때문에, GPU (12) 는 노드 (62C) 와 연관된 바운딩 볼륨 (56C) 에 대한 레이-박스 교차 테스트를 수행할 수도 있다.

[0065] GPU (12) 가 새도우 레이 (70B) 가 바운딩 볼륨 (56C) 과 교차한다고 결정하는 경우, GPU (12) 는 노드 (62D) 까지 BVH 트리 (60) 를 횡단할 수도 있다. GPU (12) 는 노드 (62D) 와 연관된 바운딩 볼륨 (56D) 에 대해 레이-박스 교차 테스트를 수행할 수도 있다. GPU 가 새도우 레이 (70B) 가 바운딩 볼륨 (56D) 과 교차한다고 결정하는 경우, GPU (12) 는 노드 (62D) 에 링크된 프리티미티브 (52C) 에 대해 레이-프리티미티브 교차 테스트를 수행할 수도 있다. GPU 가 새도우 레이 (70B) 가 프리티미티브 (52C) 와 교차하지 않는다고 결정하는 경우, GPU (12) 노드 (62D) 까지 위로 BVH 트리 (60) 를 회귀적으로 횡단할 수도 있고, 노드 (62D) 에 링크된 프리티미티브 (52D) 에 대해 레이-프리티미티브 교차 테스트를 수행할 수도 있다. GPU (12) 는 새도우 레이 (70B) 가 프리티미티브 (52D) 와 교차한다고 프리티미티브 (52D) 에 대한 레이-프리티미티브 교차 테스트로부터 결정할 수도 있다.

[0066] GPU (12) 가 새도우 레이 (70B) 가 프리티미티브들 (52) 중 하나와 교차한다고 결정할 때, GPU (12) 는 새도우 레이 (70B) 가 발원하는 로케이션 (72B) 이 그래픽 장면 (50) 의 광원으로부터 폐색된다고 (광원에 의해 조명되지 않는다고) 결정할 수도 있고, 새도우 레이 (70B) 에 대한 BVH 트리 (60) 의 횡단을 종료할 수도 있다. 알 수 있듯이, GPU (12) 는 렌더 타겟 (64) 으로 그래픽 장면 (50) 의 표현을 렌더링하고, BVH 트리 (60) 의 횡단을 시작할 뉀-루트 노드를 결정하기 위해 렌더 타겟 (64) 으로부터 화소들을 샘플링함으로써 BVH 트리 (60) 의 횡단을 가속화할 수도 있다.

[0067] 도 6 은 새도우 레이-프리티미티브 교차를 발견하기 위해 예시의 계층적 데이터 구조를 횡단하기 위한 시작 노드를 결정하는 예시의 프로세스를 도시하는 플로우차트이다. 도 6 에 도시된 바와 같이, 프로세스는 CPU (6) 또는 GPU (12) 와 같은 적어도 하나의 프로세서에 의해, 계층적 데이터 구조 (예를 들어, BVH 트리 (60)) 에서 그래픽 장면 (예를 들어, 그래픽 장면 (50)) 의 복수의 프리티미티브들 (52) 을 조직화하는 것을 포함할 수도 있고, 여기서 복수의 바운딩 볼륨들 (56) 이 계층적 데이터 구조의 노드들 (62) 과 연관된다 (702). 프로세스는 CPU (6) 또는 GPU (12) 에 의해, 오프-스크린 렌더 타겟 (64) 으로 복수의 바운딩 볼륨들 (56) 각각의 표현들을 래스터화하는 것을 더 포함할 수도 있다 (704). 프로세스는, CPU (6) 또는 GPU (12) 에 의해 그리고 그래픽 장면 (50) 에서 레이에 맵핑하는 오프-스크린 렌더 타겟 (64) 에서의 화소에 적어도 부분적으로 기초하여, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 뉀-루트 노드를 결정하는 것을 더 포함할 수도 있다 (706). 프로세스는, CPU (6) 또는 GPU (12) 에 의해, 그래픽 장면 (50) 내의 레이가 복수의 프리티미티브들 (52) 중 하나와 교차하는지 여부를 결정하기 위해 시작 노드로부터

시작하여 계층적 데이터 구조의 노드들의 세트를 횡단하는 것을 더 포함할 수도 있다 (708).

- [0068] 적어도 하나의 프로세서는 복수의 로케이션들 각각으로부터 새도우 레이를 발하고, 상술된 프로세스에 따라 새도우 레일들 각각에 대해 가능한 새도우 레이-프리미티브 교차들을 결정함으로써 그래픽 장면 (50) 내의 복수의 로케이션들에 대한 새도우 렌더링을 수행할 수도 있다. 적어도 하나의 프로세서가 특정의 로케이션으로부터 발하는 새도우 레이가 그래픽 장면 (50) 내의 프리미티브와 교차하기 때문에 그래픽 장면 (50) 내의 특정의 로케이션이 광원으로부터 폐색된다고 결정하는 경우, 적어도 하나의 프로세서는 그 특정의 로케이션의 화소 값들을 변경할 수도 있다. 예를 들어, 적어도 하나의 프로세서는 그 특정의 로케이션이 광원으로부터 폐색되는 것을 나타내는 흑색 컬러 값 또는 다른 적합한 컬러 값으로 그 특정의 로케이션의 컬러 값들을 변경할 수도 있다.
- [0069] 또한, 오프-스크린 렌더 타겟 (64) 으로 복수의 바운딩 볼륨들 (56) 각각의 표현들을 래스터화하고, 새도우 레이에 맵핑하는 오프-스크린 렌더 타겟 (64) 내의 화소에 기초하여 계층적 데이터 구조의 횡단을 시작할 계층적 데이터 구조의 년-루트 노드를 결정함으로써, 프로세스는 년-루트 시작 노드를 결정함으로써 계층적 데이터 구조를 더 효율적으로 횡단할 방법에 대한 그래픽 프로세싱에서의 근본적인 기술적 문제에 대한 기술적 솔루션을 제공하며, 이것에 의해 적어도 하나의 프로세서가 장면에 대한 새도우 렌더링을 더 효율적으로 수행하는 것을 가능하게 한다.
- [0070] 일부 예들에서, 오프-스크린 렌더 타겟 (64) 으로 복수의 바운딩 볼륨들 (56) 각각의 표현을 래스터화하는 것은, CPU (6) 또는 GPU (12) 에 의해, 복수의 컬러 값들 중 상이한 컬러 값을 계층적 데이터 구조의 복수의 노드들 (62) 각각과 연관시키는 것; 및 복수의 바운딩 볼륨들 (56) 의 각각의 바운딩 볼륨에 대해, CPU (6) 또는 GPU (12) 에 의해, 각각의 바운딩 볼륨과 연관되는 계층적 데이터 구조의 노드들 중 하나와 연관된 상이한 컬러 값들 중 하나를 갖는 화소들의 블록을 래스터화하는 것을 더 포함할 수도 있다.
- [0071] 일부 예들에서, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소와 연관된 계층적 데이터 구조의 년-루트 노드를 결정하는 것은, CPU (6) 또는 GPU (12) 에 의해, 화소의 화소 컬러 값과 연관되는 계층적 데이터 구조의 노드를 결정하는 것; 및 CPU (6) 또는 GPU (12) 에 의해, 계층적 데이터 구조의 횡단을 시작하기 위해 시작 노드로서 화소 컬러 값과 연관되는 계층적 데이터 구조의 노드를 설정하는 것을 더 포함할 수도 있다.
- [0072] 일부 예들에서, 프로세스는, CPU (6) 또는 GPU (12) 로, 바운딩 볼륨들의 제 1 바운딩 볼륨의 제 1 표현이 바운딩 볼륨들의 제 2 바운딩 볼륨의 제 2 표현과 중첩한다고 결정하는 것으로서, 제 1 바운딩 볼륨은 계층적 데이터 구조의 제 1 노드와 연관되고, 제 2 바운딩 볼륨은 계층적 데이터 구조의 제 2 노드와 연관되는, 상기 결정하는 것; 및 CPU (6) 또는 GPU (12) 에 의해, 제 1 노드 및 제 2 노드의 최저 공통 조상 노드와 연관된 노드 컬러 값에 제 1 표현 및 제 2 표현의 중첩의 영역 내의 하나 이상의 화소들의 컬러 값들을 설정하는 것을 더 포함할 수도 있다.
- [0073] 일부 예들에서, 프로세스는, 레이가 복수의 프리미티브들 (52) 중 하나와 교차한다고 결정하는 것에 응답하여, CPU (6) 또는 GPU (12) 에 의해, 레이가 광원을 향해 발하는 그래픽 장면 (50) 내의 로케이션이 광원으로부터 폐색된다고 결정하는 것을 더 포함할 수도 있고, 여기서 레이는 새도우 레이를 포함한다. 일부 예들에서, 오프-스크린 렌더 타겟 (64) 으로 복수의 바운딩 볼륨들 (56) 각각의 표현을 래스터화하는 것은, CPU (6) 또는 GPU (12) 에 의해, 오프-스크린 렌더 타겟 (64) 으로 계층적 데이터 구조의 모든 년-루트 노드들보다 적은 수의 년-루트 노드와 연관된 복수의 바운딩 볼륨들 (56) 의 표현들을 래스터화하는 것을 더 포함할 수도 있다.
- [0074] 일부 예들에서, 오프-스크린 렌더 타겟 (64) 으로 복수의 바운딩 볼륨들 (56) 각각의 표현을 래스터화하는 것은, CPU (6) 또는 GPU (12) 에 의해, 복수의 바운딩 볼륨들 (56) 각각의 표현들로서 오프-스크린 렌더 타겟 (64) 으로 복수의 플랫폼-셰이딩된 정육면체들을 래스터화하는 것; 및 CPU (6) 또는 GPU (12) 에 의해, 각각의 바운딩 볼륨의 형상과 매치시키기 위해 복수의 플랫폼-셰이딩된 정육면체들 각각을 스케일링 및 변환하는 것을 더 포함할 수도 있다. 일부 예들에서, 프로세스는 디스플레이 디바이스에 의한 디스플레이를 위해 장면을 렌더링하는 것을 더 포함할 수도 있다.
- [0075] 하나 이상의 예들에서, 기술된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체에 하나 이상의 명령들 또는 코드로서 저장되거나 송신될 수도 있다. 컴퓨터 판독가능 매체들은 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들 또는 컴퓨터 데이터 저장 매체들을 포함할 수도



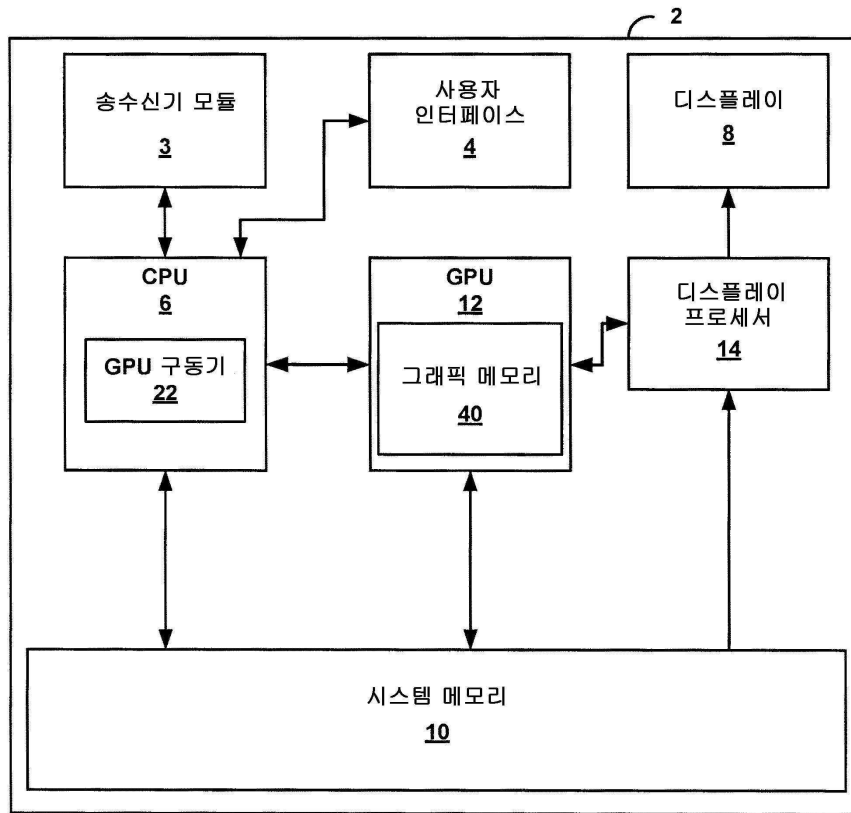
있다. 데이터 저장 매체들은 본 개시에 기술된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 제한이 아닌 예로서, 그러한 컴퓨터 판독가능 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 반송하거나 또는 저장하는데 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 연결이 컴퓨터 판독가능 매체로 적절히 지칭된다. 예를 들어, 소프트웨어가 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버 또는 다른 원격 소스로부터 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 디스크 (disk) 및 디스크 (disc) 는, 본 명세서에서 사용되는 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 데이터를 자기적으로 보통 재생하지만, 디스크 (disc) 들은 레이저들로 데이터를 광학적으로 재생한다. 또한, 상기의 조합들도 컴퓨터 판독가능 매체들의 범위 내에 포함된다.

[0076] 코드는 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 반도체들 (ASICs), 필드 프로그램가능 로직 어레이들 (FPGAs), 또는 다른 등가의 집적된 또는 이산의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 이에 따라, 여기에서 사용되는 바와 같은 용어 "프로세서" 및 "프로세싱 유닛" 은 상술된 구조 또는 여기에 기술된 기법들의 구현을 위해 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또, 일부 양태들에서, 여기에 기술된 기능성은 인코딩 및 디코딩을 위해 구성된, 또는 결합된 코덱에 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에서 제공될 수도 있다. 또한, 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있을 것이다.

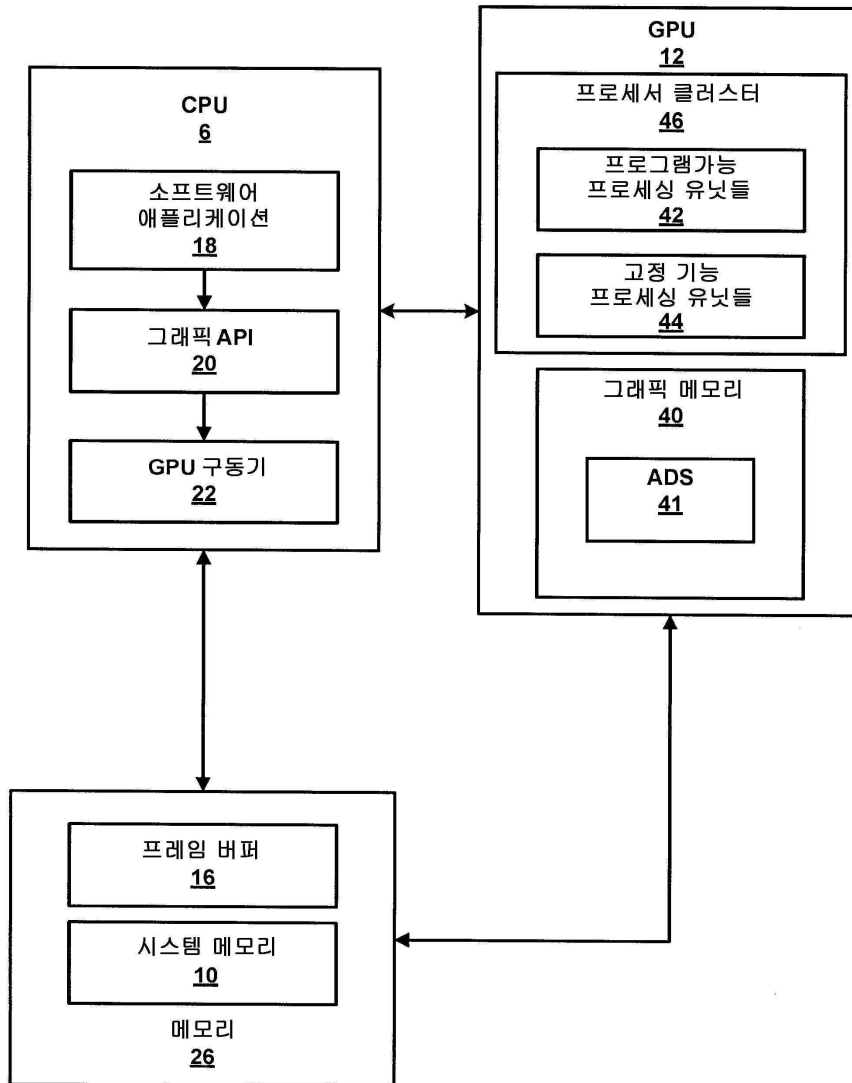
[0077] 본 개시의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC 들이 세트 (즉, 칩셋) 를 포함하는 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 여러 컴포넌트들, 모듈들 또는 유닛들이 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에 기술되지만, 반드시 상이한 하드웨어 유닛들에 의한 실현을 요구하지는 않는다. 오히려, 상술된 바와 같이, 여러 유닛들이 적합한 소프트웨어 및/또는 펌웨어와 결합하여, 상술된 하나 이상의 프로세서들을 포함하는, 상호동작적 하드웨어 유닛들의 컬렉션에 의해 제공되거나 코덱 하드웨어 유닛에 결합될 수도 있다. 이들 및 다른 예들은 다음의 청구범위의 범위 내에 있다.

도면

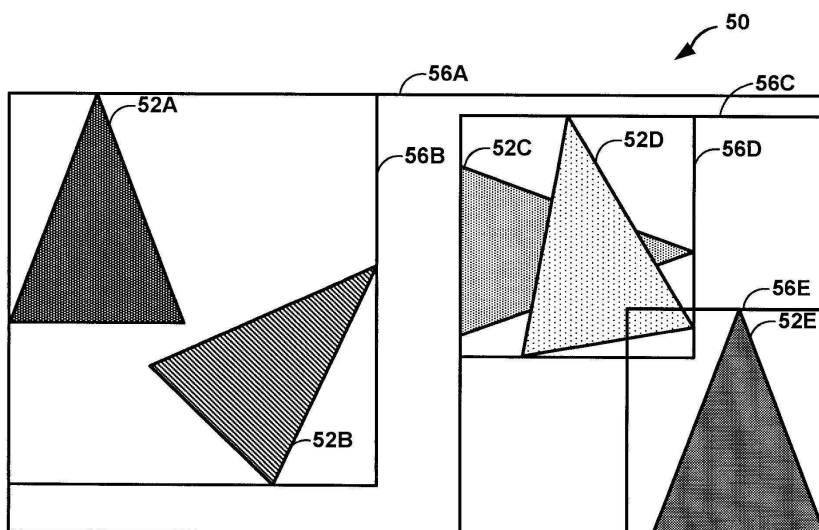
도면1



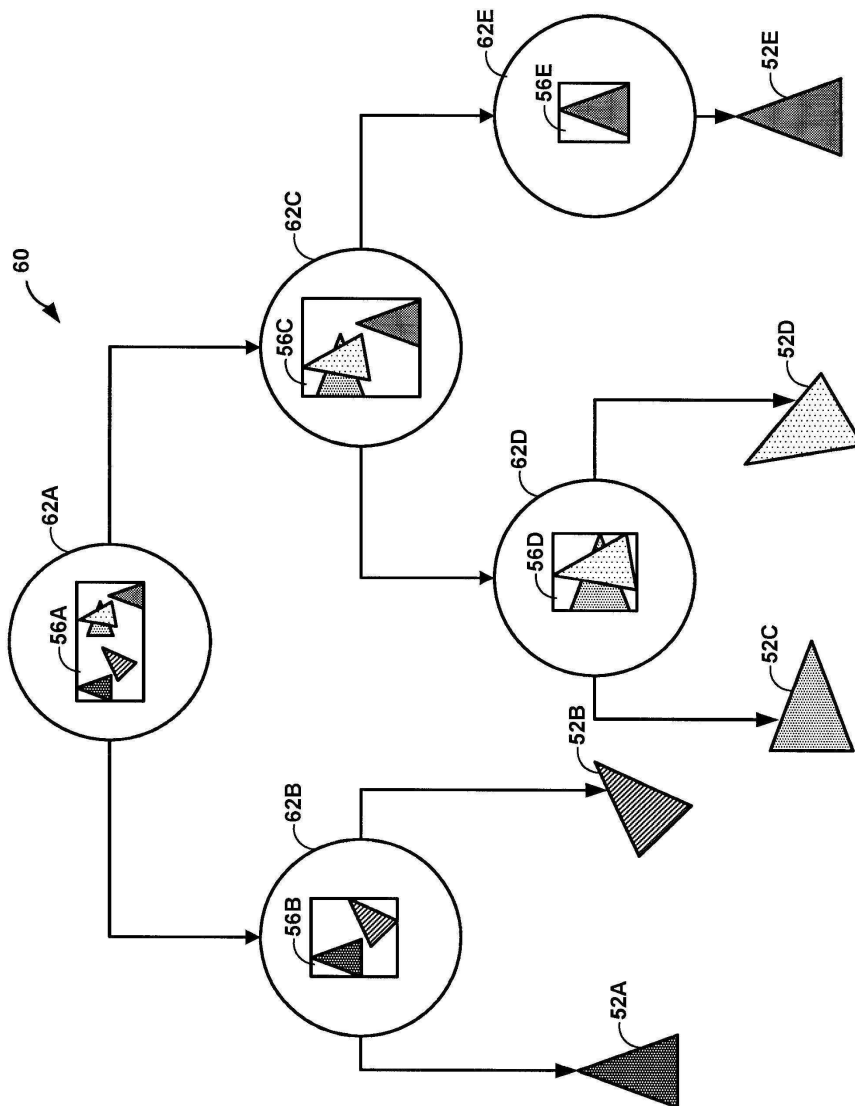
도면2



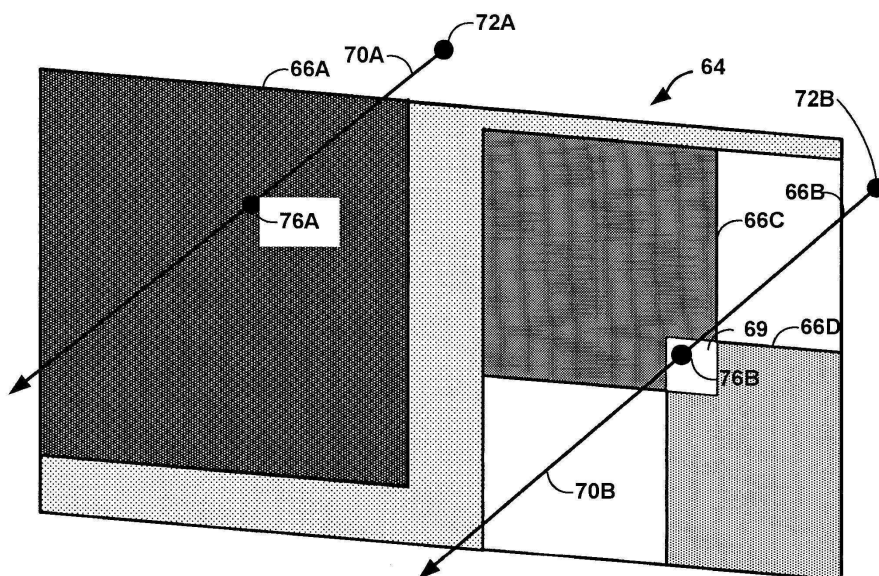
도면3



도면4



도면5





도면6

