

 (19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2012-0092926 (43) 공개일자 2012년08월22일
(51) 국제특허분류(Int. Cl.) G06T 15/06 (2011.01) (21) 출원번호 10-2011-0012860 (22) 출원일자 2011년02월14일 심사청구일자 2011년02월14일	(71) 출원인 주식회사 실리콘아츠 서울특별시 강동구 양재대로89길 68, 201호 (성내동) 세종대학교산학협력단 서울특별시 광진구 능동로 209 (군자동, 세종대학교) (72) 발명자 윤형민 서울특별시 송파구 올림픽로 366, 파라다이스빌 901호 (방이동) 박우찬 서울특별시 종로구 무악동 82번지 무악현대아파트 108동 1604호 (74) 대리인 정부연

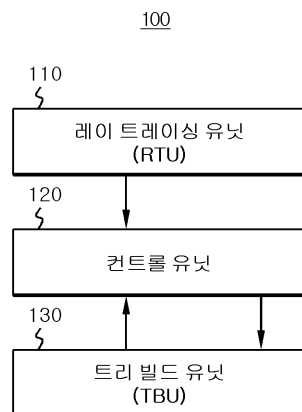
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 **레이 트레이싱 코어 및 레이 트레이싱 처리 방법**

(57) 요약

레이 트레이싱 코어는 레이 트레이싱 유닛(Ray Tracing Unit, RTU), 컨트롤 유닛 및 트리 빌드 유닛(Tree Build Unit, TBU)을 포함한다. 상기 레이 트레이싱 유닛은 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray Tracing)을 수행한다. 상기 컨트롤 유닛은 상기 레이 트레이싱 유닛의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산한다. 상기 트리 빌드 유닛 상기 계산된 복잡도를 가지는 공간 분할 구조체를 구축한다. 일 실시예에서, 상기 부하 상태는 해당 유닛에서 처리되는 프레임 비율(Frame Rate)을 기초로 결정될 수 있다. 다른 일 실시예에서, 상기 공간 분할 구조체는 KD 트리(K-Dimensional Tree)를 적용할 수 있다. 예를 들어, 상기 복잡도는 KD 트리 구조에 대한 리프노드(Leaf Node)의 최대 프리미티브 수(Maximum Primitive Number) 및/또는 트리 깊이(Tree Depth)에 따라 변경될 수 있다.

대표도 - 도1



특허청구의 범위

청구항 1

공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray Tracing)을 수행하는 레이 트레이싱 유닛(Ray Tracing Unit, RTU);

상기 레이 트레이싱 유닛의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산하는 컨트롤 유닛; 및

상기 계산된 복잡도를 가지는 공간 분할 구조체를 구축하는 트리 빌드 유닛(Tree Build Unit, TBU)을 포함하는 레이 트레이싱 코어(Ray Tracing Core).

청구항 2

제1항에 있어서, 상기 부하 상태는

해당 유닛에서 처리되는 프레임 비율(Frame Rate)을 기초로 결정되는 것을 특징으로 하는 레이 트레이싱 코어.

청구항 3

제1항에 있어서, 상기 공간 분할 구조체는 BVH(Bounding Volume Hierarchy) 또는 KD 트리(K-Dimensional Tree)인 것을 특징으로 하는 레이 트레이싱 코어.

청구항 4

제3항에 있어서, 상기 복잡도는

상기 BVH 또는 KD 트리의 구조에 대한 리프노드(Leaf Node)의 최대 프리미티브 수(Maximum Primitive Number) 및/또는 트리 깊이(Tree Depth)에 따라 변경되는 것을 특징으로 하는 레이 트레이싱 코어.

청구항 5

제1항에 있어서, 상기 레이 트레이싱 코어는

프리미티브 씬(Primitive Scene)을 상기 트리 빌드 유닛에 제공하는 프리미티브 캐시(Primitive Cache); 및

상기 트리 빌드 유닛으로부터 상기 프리미티브 씬에 대한 처리 결과를 전송받는 AS 결과 버퍼(Acceleration Structure Result Buffer)를 더 포함하는 것을 특징으로 하는 레이 트레이싱 코어.

청구항 6

제1항에 있어서, 상기 레이 트레이싱 코어는

상기 트리 빌드 유닛에 의해 처리된 프리미티브 씬에 대한 트리구축정보를 상기 레이 트레이싱 유닛에 제공하는 AS 캐시(Acceleration Structure Cache);

텍스처(Texture)를 상기 레이 트레이싱 유닛에 제공하는 텍스처 캐시; 및

상기 레이 트레이싱 유닛으로부터 상기 트리구축정보 및 텍스처에 대한 처리 결과를 전송받는 컬러 결과 버퍼(Color Result Buffer)를 더 포함하는 것을 특징으로 하는 레이 트레이싱 코어.

청구항 7

(a) 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray Tracing)을 수행하는 단계;

(b) 상기 레이 트레이싱이 수행될 때의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산하는 단계; 및

(c) 상기 계산된 복잡도를 가지는 공간 분할 구조체를 구축하는 단계를 포함하는 레이 트레이싱 처리 장치에서 수행되는 레이 트레이싱 방법.

청구항 8

제7항에 있어서, 상기 (b) 단계는

(b1) 상기 레이 트레이싱 유닛 및 트리 빌드 유닛의 부하 상태를 모니터링하는 단계; 및

(b2) 상기 레이 트레이싱 유닛(Ray Tracing Unit, RTU) 및 트리 빌드 유닛(Tree Build Unit, TBU)의 부하를 비교하여 상기 공간 분할 구조체의 복잡도를 조절하는 단계; 및

(b3) 상기 조절된 복잡도를 상기 트리 빌드 유닛에 제공하는 단계를 포함하는 것을 특징으로 하는 레이 트레이싱 처리 방법.

청구항 9

제8항에 있어서, 상기 (b1) 단계는

(b1-1) 레이 트레이싱 유닛의 프레임 처리 속도를 확인하는 단계;

(b1-2) 트리 빌드 유닛의 프레임 처리 속도를 확인하는 단계;

(b1-3) 단위 시간당 상기 레이 트레이싱 유닛에서 처리되는 프레임과 상기 트리 빌드 유닛에서 처리되는 프레임의 비율을 산출하는 단계; 및

(b1-4) 상기 산출된 프레임 비율(Frame Rate)을 기초로 부하 상태를 판단하는 단계를 포함하는 것을 특징으로 하는 레이 트레이싱 처리 방법.

청구항 10

제8항에 있어서, 상기 공간 분할 구조체는 BVH(Bounding Volume Hierarchy) 또는 KD 트리(K-Dimensional Tree)를 포함하고,

상기 복잡도는 상기 BVH 또는 KD 트리의 구조에 대한 리프노드(Leaf Node)의 최대 프리미티브 수(Maximum Primitive Number) 및/또는 트리 깊이(Tree Depth)에 따라 변경되는 것을 특징으로 하는 레이 트레이싱 처리 방법.

청구항 11

제10항에 있어서, 상기 (b2) 단계는

상기 레이 트레이싱 유닛의 부하가 상기 트리 빌드 유닛의 부하보다 큰 경우에는 최대 프리미티브 수의 증가 및/또는 트리 깊이(Tree Depth)의 감소로 상기 공간 분할 구조체의 복잡도를 낮추는 것을 특징으로 하는 레이 트레이싱 처리 방법.

청구항 12

제10항에 있어서, 상기 (b2) 단계는

상기 레이 트레이싱 유닛의 부하가 상기 트리 빌드 유닛의 부하보다 작은 경우에는 최대 프리미티브 수의 감소 및/또는 트리 깊이(Tree Depth)의 증가로 상기 공간 분할 구조체의 복잡도를 높이는 것을 특징으로 하는 레이 트레이싱 처리 방법.

청구항 13

3차원 어플리케이션을 실행하는 CPU(Central Processing Unit);

상기 3차원 어플리케이션에 필요한 그래픽 데이터를 저장하는 시스템 메모리; 및

상기 그래픽 데이터 정보를 기초로 공간 분할 구조체를 구축하고, 상기 생성된 공간 분할 구조체를 기초로 레이 트레이싱을 수행하며, 상기 레이 트레이싱 속도를 모니터링하여 상기 그래픽 데이터 정보에 대한 공간 분할 구조체를 재구축하는 DRTX(Dynamic Ray Tracing Accelerator)를 포함하는 레이 트레이싱 처리 장치.

청구항 14

제13항에 있어서, 상기 레이 트레이싱 처리 장치는

상기 DRTX와 연동하며, 상기 어플리케이션에 필요한 그래픽 데이터에 따라 구축된 공간 분할 구조체의 가속 구조(Acceleration Structure)를 저장하고, 상기 공간 분할 구조체의 가속 구조를 상기 DRTX로 제공하는 외부 메모리를 더 포함하는 것을 특징으로 하는 레이 트레이싱 처리 장치.

청구항 15

제13항에 있어서, 상기 시스템 메모리는

상기 3차원 어플리케이션에 필요한 고정 씬(Scene) 정보를 저장하는 PSS(Primitive Static Scene) 영역;

상기 3차원 어플리케이션에 필요한 동적 씬 정보를 저장하는 PDS(Primitive Dynamic Scene) 영역; 및

텍스처 매핑을 위한 밍맵(MIP-MAP)을 저장하는 텍스처 맵(Texture Map) 영역을 포함하는 것을 특징으로 하는 레이 트레이싱 처리 장치.

청구항 16

제13항에 있어서, 상기 DRTX는

공간 분할 구조체(Spatial Partitioning Tree)를 기초로 고정 씬 및/또는 동적 씬에 대한 레이 트레이싱(Ray Tracing)을 수행하는 레이 트레이싱 유닛(Ray Tracing Unit, RTU);

상기 레이 트레이싱 유닛의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산하는 컨트롤 유닛; 및

동적 씬에 대하여 상기 계산된 복잡도를 가지는 공간 분할 구조체를 구축하는 트리 빌드 유닛(Tree Build Unit, TBU)을 포함하는 것을 특징으로 하는 레이 트레이싱 처리 장치.

청구항 17

제16항에 있어서, 상기 레이 트레이싱 처리 장치는

상기 트리 빌드 유닛에서 구축된 해당 고정 씬 및/또는 동적 씬에 따라 구축된 공간 분할 구조체의 가속 구조(Acceleration Structure)를 저장하고, 상기 공간 분할 구조체의 가속 구조를 상기 레이 트레이싱 유닛으로 제공하는 외부 메모리를 더 포함하는 것을 특징으로 하는 레이 트레이싱 처리 장치.

명세서

기술 분야

[0001] 개시된 기술은 3차원 그래픽 처리에 관한 것으로, 특히 레이 트레이싱 코어 및 레이 트레이싱 처리 방법에 관한 것이다.

배경 기술

[0002] 3차원 그래픽 기술은 컴퓨팅에 저장된 기하학적 데이터(Geometric data)의 3차원 표현을 사용하는 그래픽 기술로, 오늘날 미디어 산업과 게임 산업을 포함하는 다양한 산업에서 널리 사용되고 있다. 일반적으로 3차원 그래픽 기술은 많은 연산량으로 인하여 별개의 고성능 그래픽 프로세서를 요구한다.

[0003] 최근 프로세서의 발전에 따라 매우 현실적인 3차원 그래픽을 생성할 수 있는 레이 트레이싱(Ray Tracing) 기술이 연구되고 있다.

[0004] 레이 트레이싱(Ray Tracing) 기술은 전역 조명(Global Illumination)에 따른 렌더링(Rendering)방식으로, 다른 물체에서 반사되거나 굴절된 빛이 현재 물체의 영상에 미치는 영향을 고려하여 반사, 굴절, 그림자 효과가 자연스럽게 제공되기 때문에 현실감 있는 3D 영상을 생성할 수 있다.

발명의 내용

[0005] 실시예들 중에서, 레이 트레이싱 코어는 레이 트레이싱 유닛(Ray Tracing Unit, RTU), 컨트롤 유닛 및 트리 빌

드 유닛(Tree Build Unit, TBU)을 포함한다. 상기 레이 트레이싱 유닛은 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray Tracing)을 수행한다. 상기 컨트롤 유닛은 상기 레이 트레이싱 유닛의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산한다. 상기 트리 빌드 유닛은 상기 계산된 복잡도를 가지는 공간 분할 구조체를 구축한다. 일 실시예에서, 상기 부하 상태는 해당 유닛에서 처리되는 프레임 처리율(Frame Rate)을 기초로 결정될 수 있다. 다른 일 실시예에서, 상기 공간 분할 구조체는 BVH(Bounding Volume Hierarchy) 또는 KD 트리(K-Dimensional Tree)를 적용할 수 있다. 예를 들어, 상기 복잡도는 BVH 또는 KD 트리의 구조에 대한 리프노드(Leaf Node)의 최대 프리미티브 수(Maximum Primitive Number) 및/또는 트리 깊이(Tree Depth)에 따라 변경될 수 있다. 일 실시예에서, 상기 레이 트레이싱 코어는 프리미티브 씬(Primitive Scene)을 상기 트리 빌드 유닛에 제공하는 프리미티브 캐시(Primitive Cache) 및 상기 트리 빌드 유닛으로부터 상기 프리미티브 씬에 대한 처리 결과를 전송받는 AS 결과 버퍼(Acceleration Structure Result Buffer)를 더 포함할 수 있다. 다른 일 실시예에서, 상기 레이 트레이싱 코어는 상기 트리 빌드 유닛에 의해 처리된 프리미티브 씬에 대한 트리구축정보를 상기 레이 트레이싱 유닛에 제공하는 AS 캐시(Acceleration Structure Cache)와, 텍스처(Texture)를 상기 레이 트레이싱 유닛에 제공하는 텍스처 캐시 및 상기 레이 트레이싱 유닛으로부터 상기 트리구축정보 및 텍스처에 대한 처리 결과를 전송받는 컬러 결과 버퍼(Color Result Buffer)를 더 포함할 수 있다.

[0006] 실시예들 중에서, 레이 트레이싱 처리 방법은 레이 트레이싱 유닛(Ray Tracing Unit, RTU) 및 트리 빌드 유닛(Tree Build Unit, TBU)의 부하 상태를 모니터링하며, 상기 레이 트레이싱 유닛의 부하가 상기 트리 빌드 유닛의 부하보다 큰 경우에는 상기 공간 분할 구조체(Spatial Partitioning Tree)의 복잡도를 낮추도록 조절하고, 상기 레이 트레이싱 유닛의 부하가 상기 트리 빌드 유닛의 부하보다 작은 경우에는 상기 공간 분할 구조체의 복잡도를 높이도록 조절하며, 상기 조절된 복잡도를 상기 트리 빌드 유닛에 제공한다. 예를 들어, 상기 레이 트레이싱 유닛은 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray Tracing)을 수행할 수 있고, 상기 트리 빌드 유닛은 공간 분할 구조체를 구축할 수 있다. 일 실시예에서, 레이 트레이싱 처리 방법은 레이 트레이싱 유닛의 프레임 처리 속도를 확인할 수 있고, 트리 빌드 유닛의 프레임 처리 속도를 확인할 수 있으며, 단위 시간당 상기 레이 트레이싱 유닛에서 처리되는 프레임과 상기 트리 빌드 유닛에서 처리되는 프레임의 비율을 산출할 수 있고, 상기 산출된 프레임 비율(Frame Rate)을 기초로 부하 상태를 판단할 수 있다. 예를 들어, 상기 공간 분할 구조체는 BVH(Bounding Volume Hierarchy) 또는 KD 트리(K-Dimensional Tree)를 포함할 수 있고, 상기 복잡도는 BVH 또는 KD 트리의 구조에 대한 리프노드(Leaf Node)의 최대 프리미티브 수(Maximum Primitive Number) 및/또는 트리 깊이(Tree Depth)에 따라 변경될 수 있다. 일 실시예에서, 레이 트레이싱 처리 방법은 상기 레이 트레이싱 유닛의 부하가 상기 트리 빌드 유닛의 부하보다 큰 경우에는 최대 프리미티브 수의 증가 및/또는 트리 깊이(Tree Depth)의 감소로 상기 공간 분할 구조체의 복잡도를 낮출 수 있다. 다른 일 실시예에서, 레이 트레이싱 처리 방법은 상기 레이 트레이싱 유닛의 부하가 상기 트리 빌드 유닛의 부하보다 작은 경우에는 최대 프리미티브 수의 감소 및/또는 트리 깊이(Tree Depth)의 증가로 상기 공간 분할 구조체의 복잡도를 높일 수 있다.

[0007] 실시예들 중에서, 레이 트레이싱 처리 장치는 CPU(Central Processing Unit), 시스템 메모리, DRTX(Dynamic Ray Tracing Accelerator)를 포함한다. CPU는 3차원 어플리케이션을 실행한다. 시스템 메모리는 상기 3차원 어플리케이션에 필요한 그래픽 데이터 정보를 저장한다. 상기 DRTX는 상기 그래픽 데이터 정보를 기초로 공간 분할 구조체를 구축하고, 상기 생성된 공간 분할 구조체를 기초로 레이 트레이싱을 수행하며, 상기 수행된 레이 트레이싱의 결과를 상기 CPU로 전송하고, 상기 레이 트레이싱 속도를 모니터링하여 상기 그래픽 데이터 정보에 대한 공간 분할 구조체를 재구축한다. 예를 들어, 상기 레이 트레이싱 처리 장치는 상기 DRTX와 연동하며, 상기 어플리케이션에 필요한 그래픽 데이터에 따라 구축된 공간 분할 구조체의 가속 구조(Acceleration Structure)를 저장하고, 상기 공간 분할 구조체의 가속 구조를 상기 DRTX로 제공하는 외부 메모리를 더 포함할 수 있다. 일 실시예에서, 상기 시스템 메모리는 상기 3차원 어플리케이션에 필요한 고정 씬(Scene) 정보를 저장하는 PSS(Primitive Static Scene) 영역과, 상기 3차원 어플리케이션에 필요한 동적 씬 정보를 저장하는 PDS(Primitive Dynamic Scene) 영역 및 텍스처 매핑을 위한 맵(MIP-MAP)을 저장하는 텍스처 맵(Texture Map) 영역을 포함할 수 있다. 다른 일 실시예에서, 상기 DRTX는 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 고정 씬 및/또는 동적 씬에 대한 레이 트레이싱(Ray Tracing)을 수행하는 레이 트레이싱 유닛(Ray Tracing Unit, RTU)과, 상기 레이 트레이싱 유닛의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산하는 컨트롤 유닛 및 동적 씬에 대하여 상기 계산된 복잡도를 가지는 공간 분할 구조체를 구축하는 트리 빌드 유닛(Tree Build Unit, TBU)을 포함할 수 있다. 예를 들어, 상기 레이 트레이싱 처리 장치는 상기 트리 빌드 유닛에서 구축된 해당 고정 씬 및/또는 동적 씬에 따라 구축된 공간 분할 구조체의 가속 구조(Acceleration

Structure)를 저장하고, 상기 공간 분할 구조체의 가속 구조를 상기 레이 트레이싱 유닛으로 제공하는 외부 메모리를 더 포함할 수 있다.

도면의 간단한 설명

- [0008] 도 1은 개시된 기술의 일 실시예에 따른 레이 트레이싱 코어를 설명하는 블록도이다.
- 도 2는 도 1의 레이 트레이싱 코어를 포함하는 레이 트레이싱 장치를 설명하는 블록도이다.
- 도 3은 도 1에서 수행되는 레이 트레이싱 처리 방법을 설명하는 흐름도이다.
- 도 4는 도 3의 구체적인 예를 설명하는 순서도이다.
- 도 5는 레이 트레이싱의 과정을 설명하는 도면이다.
- 도 6은 개시된 기술에서 사용되는 가속 구조와 기하학적 데이터를 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0009] 개시된 기술에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 개시된 기술의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 개시된 기술의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다.
- [0010] 한편, 본 출원에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.
- [0011] “제1”, “제2” 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.
- [0012] “및/또는”의 용어는 하나 이상의 관련 항목으로부터 제시 가능한 모든 조합을 포함하는 것으로 이해되어야 한다. 예를 들어, “제1 항목, 제2 항목 및/또는 제3 항목”의 의미는 제1, 제2 또는 제3 항목뿐만 아니라 제1, 제2 또는 제3 항목들 중 2개 이상으로부터 제시될 수 있는 모든 항목의 조합을 의미한다.
- [0013] 어떤 구성요소가 다른 구성요소에 “연결되어” 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 “직접 연결되어” 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 “~사이에”와 “바로 ~사이에” 또는 “~에 이웃하는”과 “~에 직접 이웃하는” 등도 마찬가지로 해석되어야 한다.
- [0014] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, “포함하다” 또는 “가지다” 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0015] 각 단계들에 있어 식별부호(예를 들어, a, b, c, ...)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0016] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 개시된 기술이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.
- [0017] 도 1은 개시된 기술의 일 실시예에 따른 레이 트레이싱 코어를 설명하는 블록도이다.
- [0018] 도 1을 참조하면, 레이 트레이싱 코어(100)는 레이 트레이싱 유닛(Ray Tracing Unit, RTU)(110), 컨트롤 유닛(120) 및 트리 빌드 유닛(Tree Build Unit, TBU)(130)을 포함한다.
- [0019] 레이 트레이싱 유닛(110)은 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray

Tracing)을 수행한다. 레이 트레이싱을 수행하는 방법은 하기에 보다 상세히 설명하기로 한다.

- [0020] 컨트롤 유닛(120)은 레이 트레이싱 유닛의 부하 상태를 모니터링하여 공간 분할 구조체의 복잡도를 계산한다. 예를 들어, 부하 상태는 레이 트레이싱 유닛(110) 및/또는 트리 빌드 유닛(130)에서 처리되는 프레임 비율(Frame Rate)을 기초로 결정될 수 있다.
- [0021] 트리 빌드 유닛(130)은 컨트롤 유닛(120)에서 계산된 복잡도를 가지는 공간 분할 구조체(Spatial Partitioning Structure)를 구축한다. 예를 들어, 공간 분할 구조체는 BVH(Bounding Volume Hierarchy) 또는 KD-Tree(K-Dimensional Tree)로 구현될 수 있다. 일 실시예에서, BVH 또는 KD-Tree를 적용하였을 때, 복잡도는 BVH 또는 KD-Tree의 구조에 대한 리프노드(Leaf Node)의 최대 프리미티브 수(Maximum Primitive Number) 및/또는 트리 깊이(Tree Depth)에 따라 변경될 수 있다. BVH 또는 KD-Tree에 대한 설명은 하기에 보다 상세히 설명하기로 한다. 예를 들어, 리프 노드의 최대 프리미티브 수를 늘리거나 전체 트리 깊이를 낮추면 가속 구조(AS)의 품질이 저하되지만 트리 빌드 유닛(130)의 처리 속도는 향상될 수 있고 결과적으로 레이 트레이싱의 성능은 저하될 수 있다. 다른 예로, 리프 노드의 최대 프리미티브 수를 줄이거나 전체 트리 깊이를 깊게하면 가속 구조(AS)의 품질이 향상되어 결과적으로 레이 트레이싱의 성능은 높아질 수 있지만 트리 빌드 유닛(130)의 처리 속도는 느려질 수 있다. 따라서, 트리 빌드 유닛(130)의 성능이 저하되면 처리 속도를 향상시키도록 제어할 수 있고, 트리 빌드 유닛(130)의 성능이 충분한 경우에는 품질을 향상시키도록 제어할 수 있다.
- [0022] 도 5는 레이 트레이싱의 과정을 설명하는 도면이다.
- [0023] 도 5를 참조하면, 각 픽셀(Pixel) 당 카메라(510) 위치로부터 최초 레이(Primary Ray)(P)를 생성하여 레이(P)와 만나는 물체(520)를 찾기 위한 계산을 수행한다. 해당 레이(P)와 만나게 된 물체가 굴절의 성질을 갖는 물체(520)나 반사의 성질을 갖는 물체(531)(532)인 경우에는, 해당 레이(P)와 물체가 만난 위치에서 굴절효과를 위한 굴절 레이(Refraction Ray)(F) 및/또는 반사 효과를 위한 반사 레이(Reflection Ray)(R)를 생성하고, 라이트(550) 방향으로 그림자 레이(Shadow Ray)(S)를 생성할 수 있다. 일 실시예에서, 그림자 레이(S)가 다른 물체(540)와 만나면 해당 그림자 레이(S)가 생성된 지점에 그림자가 생성될 수 있다.
- [0024] 도 6은 개시된 기술에서 사용되는 가속 구조와 기하학적 데이터를 설명하는 도면이다.
- [0025] 도 6에서, 가속 구조(Acceleration Structure, AS)는 KD-tree를 사용한다고 가정하였다. KD-tree는 공간 분할 구조체(Spatial Partitioning Structure)의 일종으로, 레이-삼각형 교차 테스트(Ray-Triangle Intersection Test)를 위하여 사용될 수 있다. 예를 들어, KD-tree는 박스 노드(Box Node)(610), 내부 노드(Inner Node)(620) 및 리프 노드(Leaf Node)(630)를 포함할 수 있다. 일 실시예에서, 리프 노드(630)는 기하학적 데이터에 포함된 적어도 하나의 삼각형 정보를 포인팅 하기 위한 삼각형 리스트를 포함할 수 있다. 예를 들어, 삼각형 정보는 삼각형의 세 점에 대한 정점 좌표, 법선 벡터 및/또는 텍스처 좌표를 포함할 수 있다. 일 실시예에서, 기하학적 데이터에 포함된 삼각형 정보가 배열로 구현된 경우에는 리프 노드에 포함된 삼각형 리스트는 배열 인덱스에 상응할 수 있다.
- [0026] 도 3은 도 1에서 수행되는 레이 트레이싱 처리 방법을 설명하는 흐름도이다.
- [0027] 도 3에서, 컨트롤 유닛(120)은 레이 트레이싱 유닛(Ray Tracing Unit, RTU)(110) 및 트리 빌드 유닛(Tree Build Unit, TBU)(130)의 부하 상태를 모니터링할 수 있다(단계 S310). 예를 들어, 레이 트레이싱 유닛(110)은 공간 분할 구조체(Spatial Partitioning Structure)를 기초로 레이 트레이싱(Ray Tracing)을 수행하고, 트리 빌드 유닛(130)은 공간 분할 구조체를 구축할 수 있다.
- [0028] 컨트롤 유닛(120)은 레이 트레이싱 유닛(110)의 부하가 트리 빌드 유닛(130)의 부하보다 큰 경우에는, 공간 분할 구조체(Spatial Partitioning Structure)의 복잡도를 낮추도록 조절하고, 레이 트레이싱 유닛(110)의 부하가 트리 빌드 유닛(130)의 부하보다 작은 경우에는 공간 분할 구조체의 복잡도를 높이도록 조절할 수 있다(단계 S320).
- [0029] 컨트롤 유닛(120)은 조절된 복잡도를 트리 빌드 유닛(130)에 제공할 수 있다(단계 S330). 일 실시예에서, 트리 빌드 유닛(130)은 컨트롤 유닛(120)에서 조절되어 제공된 복잡도에 상응하여 공간 분할 구조체를 재구성할 수 있다.
- [0030] 도 2는 도 1의 레이 트레이싱 코어를 포함하는 레이 트레이싱 장치를 설명하는 블록도이다.
- [0031] 도 2를 참조하면, 레이 트레이싱 장치(200)는 CPU(Central Processing Unit) (210), 시스템 메모리(System Memory)(220), DRTX(Dynamic Ray Tracing Accelerator) 및 외부 메모리(External Memory)(240)를 포함할 수 있

다.

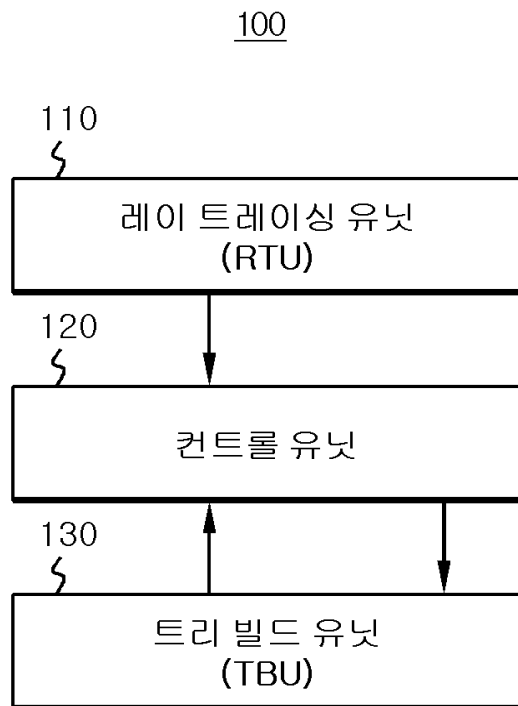
- [0032] CPU(210)는 3차원 어플리케이션을 처리할 수 있으며, 3차원 게임 엔진 등의 어플리케이션(211), API(Application Programming Interface)(212) 및/또는 씬 매니저(Scene Manager)를 포함할 수 있다.
- [0033] 시스템 메모리(220)는 3차원 어플리케이션에 필요한 그래픽 데이터 정보를 저장할 수 있으며, 원시 고정 씬(Primitive Static Scene)을 저장하는 PPS 영역(221), 프리미티브(원시) 동작 씬(Primitive Dynamic Scene)을 저장하는 PDS 영역(222) 및/또는 텍스처 매핑을 위한 밍맵(MIP-MAP)을 저장하는 텍스처 맵(Texture Map) 영역(223)을 포함할 수 있다.
- [0034] DRTX(230)는 도 1의 레이 트레이싱 코어(100)를 포함하며, 버스 인터페이스 유닛(Bus Interface Unit)(231), 가속 구조 결과 버퍼(AS Result Buffer)(232), 프리미티브 캐시(Primitive Cache)(233), 작업 메모리(Working Memory)(234), 가속 구조 캐시(AS Cache)(235), 텍스처 캐시(Texture Cache)(236), 컬러 결과 버퍼(Color Result Buffer)(237) 및/또는 스택 메모리(Stack Memory)(238)를 더 포함할 수 있다. 일 실시예에서, 그래픽 데이터 정보를 기초로 공간 분할 구조체를 구축하고, 생성된 공간 분할 구조체를 기초로 레이 트레이싱을 수행하며, 수행된 레이 트레이싱의 결과를 CPU(210)로 전송하고, 레이 트레이싱 속도를 모니터링하여 그래픽 데이터 정보에 대한 공간 분할 구조체를 재구축할 수 있다.
- [0035] 외부 메모리(240)는 DRTX(230)에서 처리되는 정보를 임시로 저장할 수 있으며, 기하학적 정보 저장 영역(241), 고정 씬 가속 구조 저장 영역(242), 동적 씬 가속 구조 저장 영역(243), 텍스처 맵 저장 영역(244) 및/또는 컬러 정보 저장 영역(245)을 포함할 수 있다.
- [0036] 도 4는 도 3의 구체적인 예를 설명하는 순서도이다.
- [0037] 도 4에서 도 2의 구성을 참조하여 설명하면, CPU(210)의 어플리케이션(211)이 구동되면 씬 매니저(213)는 API(212)를 통해 고정 씬, 동적 씬, 텍스처 맵을 시스템 메모리(220)의 각 영역에 저장하는 전처리과정을 수행할 수 있다. 일 실시예에서, CPU(210)와 시스템 메모리(220) 및 DRTX(230)는 고속 버스(High Speed Bus)를 통해 데이터를 전송할 수 있다.
- [0038] DRTX(230)는 버스 인터페이스 유닛(231)을 통해 시스템 메모리(220)에 저장된 고정 씬을 트리 빌드 유닛(130)으로 전송하고, 트리 빌드 유닛(130)은 고정 씬에 대한 트리를 구축하며(단계 S401), 구축된 고정 씬 트리 구조는 가속 구조 결과 버퍼(232)를 통해 외부 메모리(240)의 고정 씬 가속 구조 저장 영역(242)에 저장할 수 있다.
- [0039] DRTX(230)는 버스 인터페이스 유닛(231)을 통해 시스템 메모리(220)에 저장된 동적 씬을 트리 빌드 유닛(130)으로 전송하고, 트리 빌드 유닛(130)은 동적 씬에 대한 트리를 구축하며(단계 S402), 구축된 동적 씬 트리 구조는 가속 구조 결과 버퍼(232)를 통해 외부 메모리(240)의 동적 씬 가속 구조 저장 영역(243)에 저장할 수 있다.
- [0040] DRTX(230)는 고정 씬 트리 구조와 동적 씬 트리 구조를 외부 메모리(240)에 저장하면서 각 구조의 기하학적 정보를 외부 메모리(240)의 기하학적 정보 저장 영역(241)에 저장할 수 있다. 또한, DRTX(230)는 각 씬의 트리 구조를 생성하는 과정에서 작업 메모리(234)를 연동할 수 있다.
- [0041] 레이 트레이싱 유닛(110)은 가속 구조 캐시(235)를 통해 외부 메모리(240)에 저장된 고정 씬 트리 구조와 동적 씬 트리 구조를 호출하여 레이 트레이싱을 수행할 수 있다(단계 S403). 예를 들어, 레이 트레이싱 유닛(110)은 스택 메모리(238)를 이용하여 레이 트레이싱을 수행할 수 있다. 일 실시예에서, 시스템 메모리(220)의 텍스처 맵 영역(223)에 저장된 텍스처 맵은 외부 메모리(240)의 텍스처 맵 저장 영역(244)에 저장된 후, 필요에 따라 텍스처 캐시(236)를 통해 레이 트레이싱 유닛(110)으로 전송될 수 있다.
- [0042] 3차원 어플리케이션이 종료되지 않고 레이 트레이싱이 지속적으로 수행되는 경우(단계 S404), 컨트롤 유닛(120)은 레이 트레이싱 유닛(110)과 트리 빌드 유닛(130)의 부하 상태를 모니터링할 수 있다(단계 S405). 일 실시예에서, 컨트롤 유닛(120)은 레이 트레이싱 유닛(110)의 프레임 처리 속도와 트리 빌드 유닛(130)의 프레임 처리 속도를 확인하고, 단위 시간당 레이 트레이싱 유닛(110)에서 처리되는 프레임과 트리 빌드 유닛(130)에서 처리되는 프레임의 비율을 산출하며, 산출된 프레임 비율(Frame Rate)을 기초로 부하 상태를 판단할 수 있다.
- [0043] 컨트롤 유닛(120)은 레이 트레이싱 유닛(110)의 성능이 트리 빌드 유닛(130)의 성능보다 높은 경우에는(단계 S406), 트리 빌드 유닛(130)에 부하가 많이 걸리는 것으로 판단할 수 있고, 트리 빌드 유닛(130)의 부하를 줄이기 위하여 공간 분할 구조체 구조의 복잡도를 낮추도록 트리 빌드 유닛(130)을 제어할 수 있다(단계 S407). 일 실시예에서, 공간 분할 구조체 구조의 복잡도를 낮추는 트리 빌드 유닛(130)의 제어는, 최대 프리미티브 수를

증가시키거나 트리 깊이(Tree Depth)를 감소시킴으로써 수행할 수 있다.

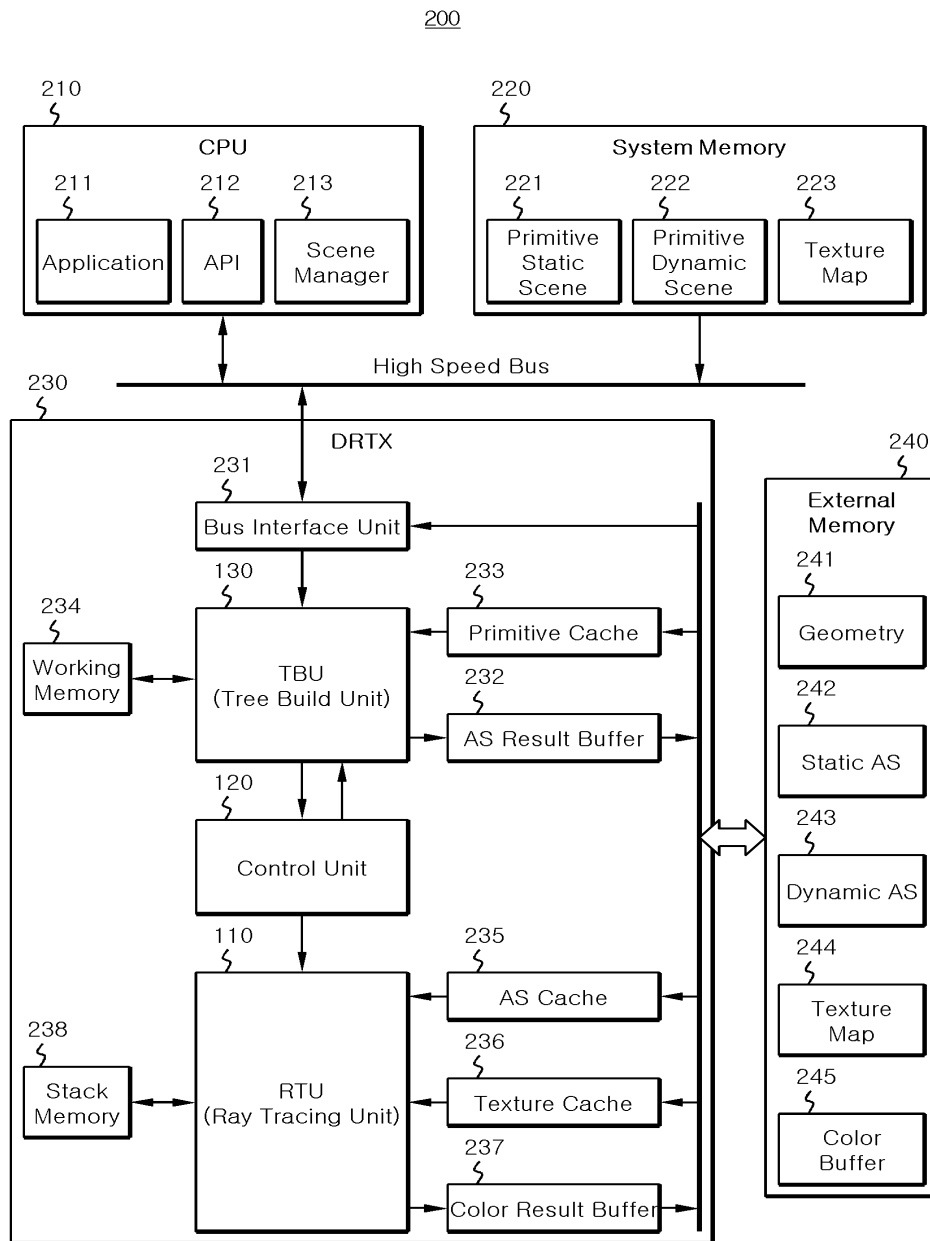
- [0044] 컨트롤 유닛(120)은 레이 트레이싱 유닛(110)의 성능이 트리 빌드 유닛(130)의 성능보다 낮은 경우에는(단계 S406), 트리 빌드 유닛(130)의 부하에 여유가 있는 것으로 판단할 수 있고, 가속 구조(AS)의 품질을 향상시키기 위하여 공간 분할 구조체 구조의 복잡도를 높이도록 트리 빌드 유닛(130)을 제어할 수 있다(단계 S408). 일 실시예에서, 공간 분할 구조체 구조의 복잡도를 높이는 트리 빌드 유닛(130)의 제어는, 최대 프리미티브 수를 감소시키거나 트리 깊이(Tree Depth)를 증가시킴으로써 수행할 수 있다.
- [0045] 트리 빌드 유닛(130)은 컨트롤 유닛(120)의 제어 하에 동적 씬의 트리 구조를 재구축할 수 있다(단계 S409).
- [0046] 레이 트레이싱 유닛(110)과 컨트롤 유닛(120) 및 트리 빌드 유닛(130)에 의한 단계 "405" 내지 단계 "409"는 3차원 어플리케이션의 요청 또는 3차원 어플리케이션의 종료가 이루어지는 시점까지 반복하여 수행될 수 있다.
- [0047] 개시된 기술은 다음의 효과를 가질 수 있다. 다만, 특정 실시예가 다음의 효과를 전부 포함하여야 한다거나 다음의 효과만을 포함하여야 한다는 의미는 아니므로, 개시된 기술의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.
- [0048] 일 실시예에 따른 레이 트레이싱 코어 및 레이 트레이싱 처리 방법은 레이 트레이싱 성능을 개선할 수 있다. 레이 트레이싱 유닛 및 트리 빌드 유닛을 모니터링하고, 트리 빌드 유닛을 제어함으로써, 각 유닛에서 요구되는 자원의 부족이나 낭비를 방지할 수 있기 때문이다.
- [0049] 또한, 일 실시예에 따른 레이 트레이싱 코어 및 레이 트레이싱 처리 방법은 3차원 영상을 처리하는 장치의 성능을 향상시킬 수 있다. 레이 트레이싱의 처리 속도를 향상시킴으로써, 전역 조명(Global Illumination)에 따른 렌더링(Rendering) 방식의 영상처리를 실시간으로 수행할 수 있기 때문이다.
- [0050] 또한, 일 실시예에 따른 레이 트레이싱 코어 및 레이 트레이싱 처리 방법은 현재 개발되고 있는 3차원 영상 처리 장치는 물론, 사용되고 있는 3차원 영상 처리장치에도 쉽게 적용할 수 있다. 기존의 장치에서 개시된 기술에 따른 레이 트레이싱 코어만을 교체하거나, 프로그램을 업데이트하는 것만으로도 개시된 기술을 수행할 수 있기 때문이다.
- [0051] 상기에서는 본 출원의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 출원의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 출원을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

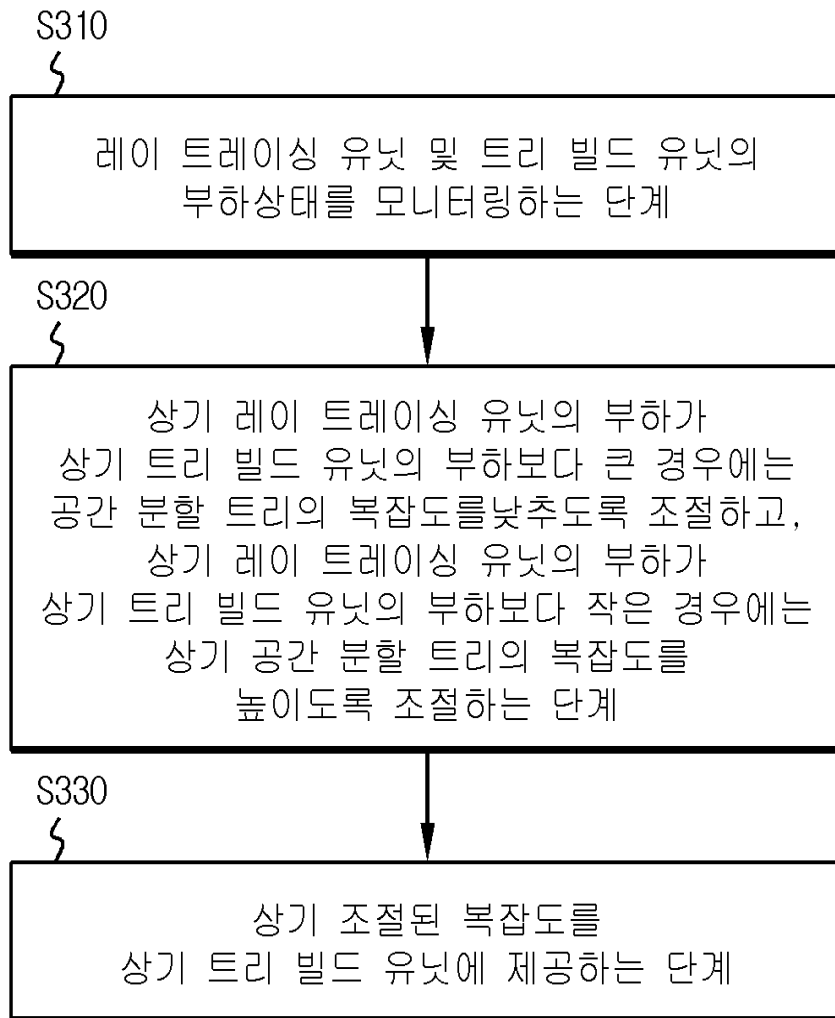
도면1



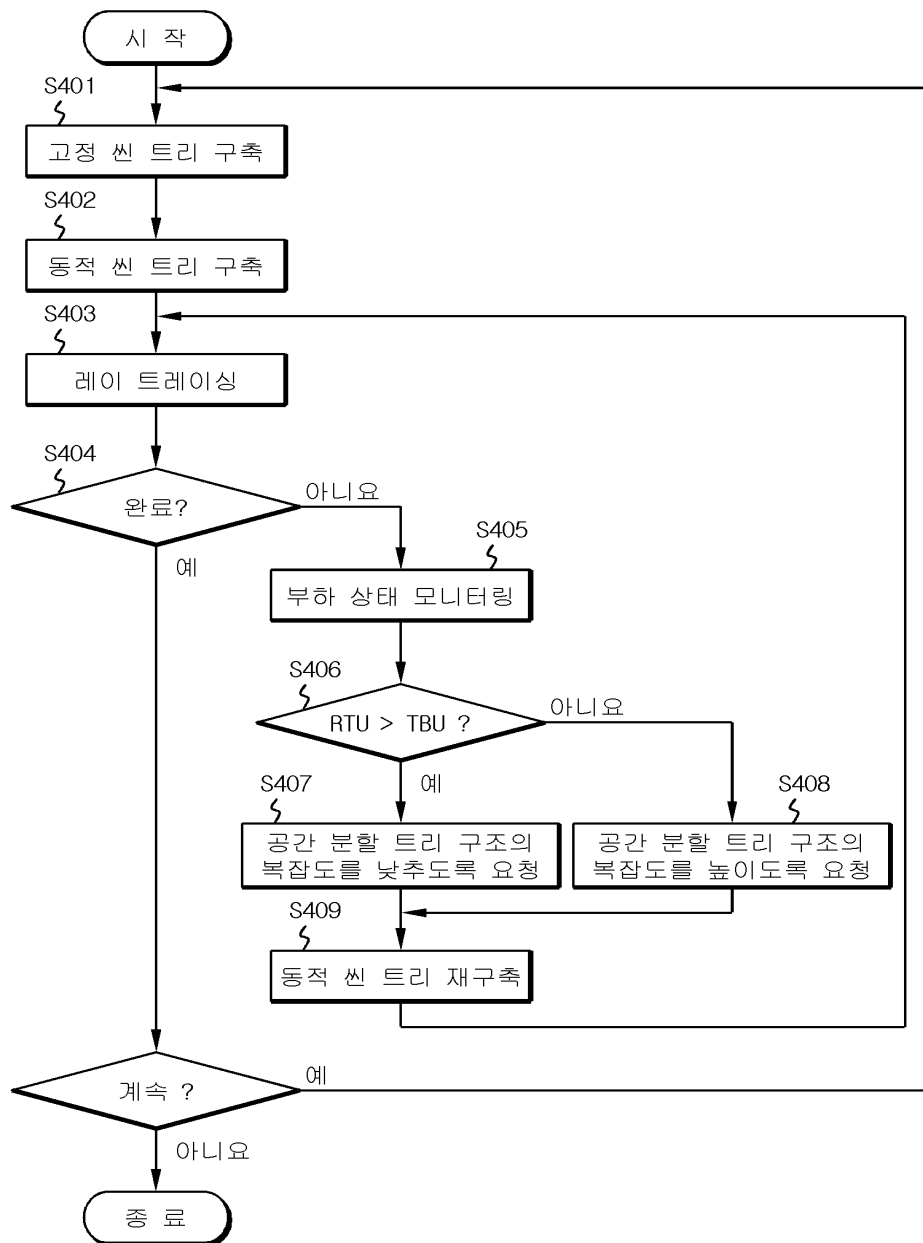
도면2



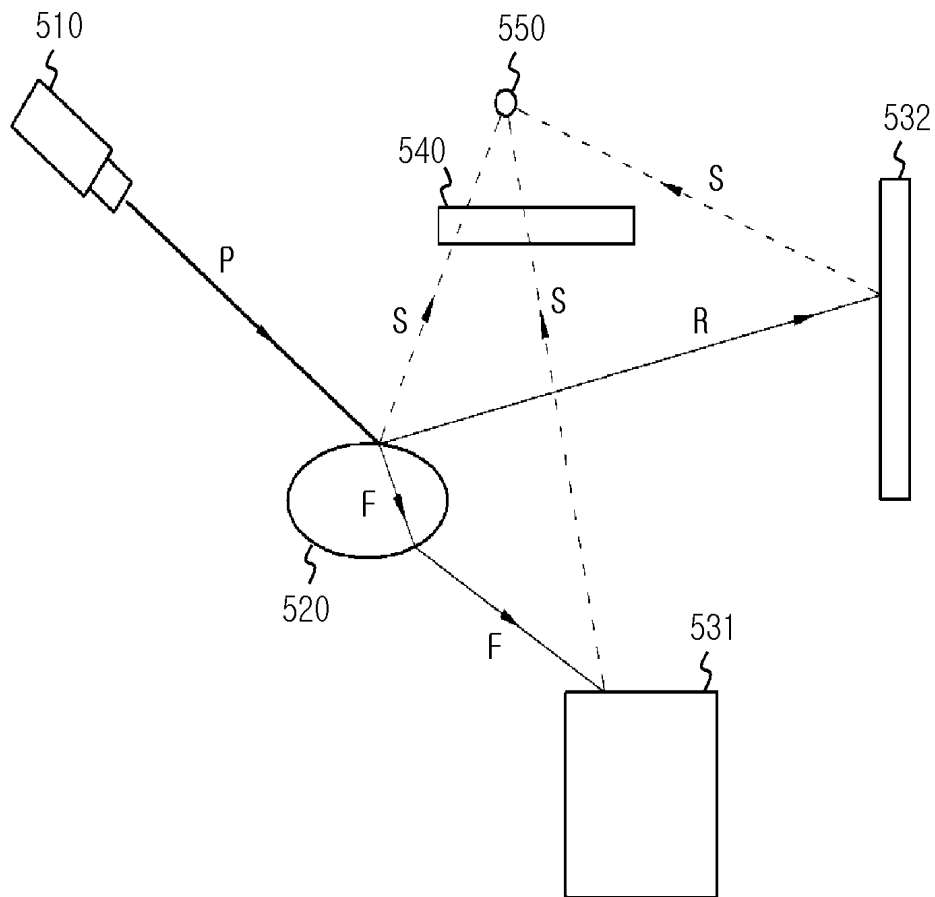
도면3



도면4



도면5



도면6

