



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0069259
(43) 공개일자 2011년06월23일

(51) Int. Cl.

G06F 17/00 (2006.01) G06T 17/00 (2006.01)

(21) 출원번호 10-2009-0125919

(22) 출원일자 2009년12월17일

심사청구일자 2009년12월17일

(71) 출원인

한국과학기술원

대전 유성구 구성동 373-1

이화여자대학교 산학협력단

서울 서대문구 대현동 11-1 이화여자대학교내

(72) 발명자

윤성의

대전 유성구 구성동 한국과학기술원 전산과

김덕수

대전 유성구 구성동 한국과학기술원 전산학과
3443호

(뒷면에 계속)

(74) 대리인

김성호

전체 청구항 수 : 총 4 항

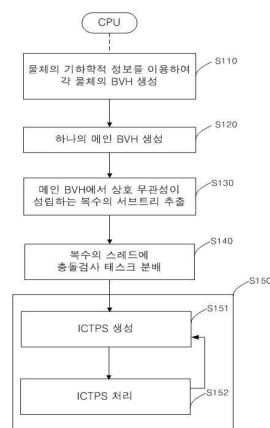
(54) 상호충돌검사 기반 병렬충돌검사 방법 및 컴퓨터 판독가능 매체

(57) 요약

본 발명은 영상처리에 관한 것으로서, 보다 구체적으로는 영상에서 물체의 충돌 감지를 위한 태스크 분배 방법에 관한 것이다.

본 발명에 따른 충돌 검출 방법은 컴퓨터 시스템에서 수행되는 씬(scene)에 포함된 물체 사이의 충돌을 검출하기 위한 방법으로서, 각 물체들의 기하학적 정보에 기초하여 각 물체들에 대한 바운딩 볼륨 체계(bounding volume hierarchies, BVH)를 생성하는 단계, 각 물체들의 BVH들에 기초한 메인 BVH를 생성하는 단계, 메인 BVH에서 상호 무관성이 성립하는 노드들을 루트로 하는 복수의 서브트리를 추출하는 단계, 복수의 서브트리에 구성된 노드들을 복수의 스레드(thread)에 각각 분배하는 단계, 및 복수의 서브트리에 구성된 노드들을 자식 노드로 하는 부모 노드에 대한 물체 간 충돌검사를 수행하기 위하여, 복수의 서브트리에 구성된 노드 쌍의 집합인 ICTPS(inter collision test pair set)를 생성하는 단계를 포함한다.

대표도 - 도1



(72) 발명자

허재필

대전 유성구 구성동 한국과학기술원 전산학과 3443

허재혁

대전 유성구 구성동 한국과학기술원 전산학과 3443

김동준

대전 유성구 구성동 한국과학기술원 전산학과 3443

김영준

서울 서대문구 대현동 이화여자대학교
컴퓨터학과(120-750)

특허청구의 범위

청구항 1

컴퓨터 시스템에서 수행되는, 씬(scene)에 포함된 물체 사이의 충돌을 검출하기 위한 방법으로서,
 각 물체들의 기하학적 정보에 기초하여 상기 각 물체들에 대한 바운딩 볼륨 체계(bounding volume hierarchies, BVH)를 생성하는 단계;
 상기 각 물체들의 BVH들에 기초한 메인 BVH를 생성하는 단계;
 상기 메인 BVH에서 상호 무관성이 성립하는 노드들을 루트로 하는 복수의 서브트리를 추출하는 단계;
 상기 복수의 서브트리에 구성된 노드들을 복수의 스레드(thread)에 각각 분배하는 단계; 및
 상기 복수의 서브트리에 구성된 노드들을 자식 노드로 하는 부모 노드에 대한 물체 간 충돌검사를 수행하기 위하여, 상기 복수의 서브트리에 구성된 노드 쌍의 집합인 ICTPS(inter collision test pair set)를 생성하는 단계를 포함하는 충돌 검출 방법.

청구항 2

제1항에 있어서,
 상기 메인 BVH에 포함된 상기 기초 다각형은 삼각형(triangle)인, 충돌 검출 방법.

청구항 3

제1항에 있어서,
 상기 복수의 스레드 중 요청 스레드는 배분 스레드에게 태스크 재분배 요청을 하고,
 상기 배분 스레드는 상기 태스크 재분배 요청에 의해, 자신에게 할당된 충돌 검사 대상 기초 다각형의 정보를 상기 요청 스레드에게 주며,
 상기 요청 스레드는,
 상기 복수의 스레드 중 자신의 태스크를 끝내고 더 이상 태스크가 없는 스레드를 의미하고,
 상기 배분 스레드는,
 복수의 스레드 중 태스크를 가지고 있는 스레드를 의미하는, 충돌 검사 방법.

청구항 4

컴퓨터에 의해 실행 가능한 프로그램을 탑재하는 컴퓨터 판독가능 매체로서,
 상기 프로그램은,
 컴퓨터 시스템에서, 씬(scene)에 포함된 물체 사이의 충돌을 검출하는 방법으로서 제1항 내지 제3항 중 어느 한 항에 따른 충돌 검출 방법을 실행하기 위한 프로그램인, 컴퓨터 판독가능 매체.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 영상처리에 관한 것으로서, 보다 구체적으로는 영상에서 물체의 충돌 감지에 관한 것이다.

배경 기술

[0002] 물체의 충돌 감지란 기하학 데이터로 이루어진 가상세계에서 기하학 데이터로 표현된 두 물체 사이의 충돌(물체

간 충돌, Inter-collision) 또는 한 물체의 두 부분의 충돌(자기 충돌, Intra-collision or self-collision)을 검출하는 것을 의미한다. 일반적으로, 효율적인 충돌 검사를 위해 바운딩 볼륨 체계(bounding volume hierarchies, BVH), KD-Tree 등의 계층적인 자료 구조를 이용하며, 물체를 구성하는 기초 다각형(elementary polygonal)간 충돌여부 확인을 위해 수학적 방법을 사용한다. 충돌 검사 방법은 이산적 그리고 연속적 검출 방법으로 나눌 수 있다. 정확한 충돌 검사를 위해서는 연속적 충돌 검사를 수행해야 하지만 많은 연산량을 요구한다는 단점을 가지고 있다.

[0003] 이러한 연속적 충돌 검사의 단점을 극복하기 위해서 많은 방법들이 제시되었으며, 멀티코어 중앙처리장치(CPU) 또는 그래픽처리장치(GPU)를 이용하는 병렬처리 알고리즘들도 제시되어 있다. 하지만, 기존 병렬처리 알고리즘들은 멀티코어 CPU 또는 GPU 중 하나만 이용하는 방법으로서 실시간 처리를 요구하는 분야에 사용하기에는 성능(속도) 면에서 충분하지 못하다.

[0004] BVH 등의 계층적 자료구조를 이용하는 환경에서 여러 개의 코어를 동시에 사용해서 충돌 검사를 수행할 경우, 자료구조를 갱신 및 탐색하는 동안 둘 이상의 스레드(thread)가 같은 데이터에 접근하는 경우가 발생한다. 이와 같은 잘못된 데이터 접근 및 갱신을 방지하기 위하여 록킹(Locking) 메커니즘을 사용한다. 록킹 메커니즘이란 한 스레드가 어떤 데이터에 접근할 때 다른 스레드의 접근을 막는 것을 의미하며, 이런 경우 접근이 막힌 스레드는 접근이 풀릴 때까지 다른 태스크(task)를 처리하지 않고 기다리게 된다.

[0005] 기존의 멀티코어를 이용한 충돌 검사 방법들은 이런 록킹 메커니즘을 사용하기 때문에 사용 가능한 코어를 최대한으로 사용하지 못한다는 문제점을 가지고 있다. 이러한 문제점은 충돌은 한 부분에 집중적으로 발생하는 경향을 의미하는 충돌의 국지 발생 특성(Locality)에 의해, 그리고 멀티 코어를 이용함에 따라 더욱더 심화된다.

발명의 내용

해결 하고자하는 과제

[0006] 본 발명은 속도가 향상되고 CPU의 스레드 개수가 증가하는 경우에 충돌 검사 방법의 높은 확장성을 보장할 수 있는 충돌 검사 방법을 제공하는 것을 목적으로 한다.

[0007] 또한, 본 발명은 CPU의 스레드 개수가 증가할수록 충돌 검사의 속도가 향상되는 충돌 검사 방법을 제공하는 것을 목적으로 한다.

[0008] 또한, 본 발명은 복수의 스레드를 이용하여 물체 충돌 검사를 수행함에 있어, 효율적인 일 분배 방법을 통해 록킹을 최소화할 수 있는 충돌 검사 방법을 제공하는 것을 목적으로 한다.

과제 해결수단

[0009] 청구항 1의 발명은 컴퓨터 시스템에서 수행되는 씬(scene)에 포함된 물체 사이의 충돌을 검출하기 위한 방법으로서, 각 물체들의 기하학적 정보에 기초하여 상기 각 물체들에 대한 바운딩 볼륨 체계(bounding volume hierarchies, BVH)를 생성하는 단계, 상기 각 물체들의 BVH들에 기초한 메인 BVH를 생성하는 단계, 상기 메인 BVH에서 상호 무관성이 성립하는 노드들을 루트로 하는 복수의 서브트리를 추출하는 단계, 상기 복수의 서브트리에 구성된 노드들을 복수의 스레드(thread)에 각각 분배하는 단계, 및 상기 복수의 서브트리에 구성된 노드들을 자식 노드로 하는 부모 노드에 대한 물체 간 충돌검사를 수행하기 위하여, 상기 복수의 서브트리에 구성된 노드 쌍의 집합인 ICTPS(inter collision test pair set)를 생성하는 단계를 포함한다.

[0010] 청구항 2의 발명은 상기 메인 BVH에 포함된 상기 기초 다각형은 삼각형(triangle)인 것을 특징으로 한다.

[0011] 청구항 3의 발명은 상기 복수의 스레드 중 요청 스레드는 배분 스레드에게 태스크 재분배 요청을 하고, 상기 배분 스레드는 상기 태스크 재분배 요청에 의해, 자신에게 할당된 충돌 검사 대상 기초 다각형의 정보를 상기 요청 스레드에게 주며, 상기 요청 스레드는 상기 복수의 스레드 중 자신의 태스크를 끝내고 더 이상 태스크가 없는 스레드를 의미하고, 상기 배분 스레드는 복수의 스레드 중 태스크를 가지고 있는 스레드를 의미하는 것을 특징으로 한다.

[0012] 청구항 4의 발명은 컴퓨터에 의해 실행 가능한 프로그램을 탑재하는 컴퓨터 판독가능 매체로서, 상기 프로그램

은, 컴퓨터 시스템에서 수행되는 씬(scene)에 포함된 물체 사이의 충돌을 검출하는 방법으로서 청구항 1 내지 3의 발명 중 어느 한 발명에 따른 충돌 검출 방법을 실행하기 위한 프로그램이다.

효 과

- [0013] 본 발명에 의하면, 충돌 검사 작업의 속도를 향상시킬 수 있고, CPU의 스레드 개수가 증가하는 경우에 충돌 검사 방법의 높은 확장성을 보장할 수 있다.
- [0014] 또한, 본 발명에 의하면 CPU의 스레드 개수가 증가할수록 충돌 검사의 속도가 향상된다.
- [0015] 또한, 복수의 스레드를 갖는 CPU를 이용하여 물체 충돌 검사를 수행함에 있어 핵심 루프에서 록킹을 사용하지 않도록 해주며, 효과적인 태스크 분배를 가능하다.

발명의 실시를 위한 구체적인 내용

- [0016] 본 발명에서는 계층적 자료구조 갱신 및 탐색 작업을 효율적으로 CPU 스레드에게 분배하는 새로운 태스크 분배 방법을 포함한다.
- [0017] 본 발명에서는 기존 기술의 문제점을 해결하기 위해 다음과 같은 방법을 사용할 수 있다.
- [0018] 1) 계층적 자료구조의 생성 및 갱신과 탐색을 위해서 멀티코어 CPU를 사용할 수 있다.
- [0019] 2) 계층적 자료구조의 갱신 및 탐색 작업을 여러 개의 CPU 스레드가 록킹 메커니즘 없이 동시에 수행할 수 있도록 태스크를 나누고, 각 스레드에 할당해 줄 수 있다.
- [0020] 3) 여러 개의 CPU 코어가 모두 최대로 이용되게 하기 위해서 태스크가 많은 스레드와 태스크가 적은 스레드들 사이에 태스크를 적은 비용으로 주고 받을 수 있게 할 수 있다.
- [0021]
- [0022] 이하, 첨부된 도면을 참조하여 본 발명을 상세히 설명한다. 본 발명을 설명함에 있어 주지기술이 본 발명의 구성이 일부인 경우 그 주지기술에 대해서는 상세한 설명을 생략하도록 한다. 이렇게 하더라도 당업자는 본 발명을 정확하게 이해하고 용이하게 재현할 수 있을 것이다.

[실시예]

- [0023]
- [0024] 도 1은 본 발명의 실시예에 따른 충돌 검사 방법의 순서도이다. 본 발명의 실시예에 따른 충돌 검사 방법은, 컴퓨터 시스템에서 수행되는, 씬(scene)에 포함된 두 물체 사이의 충돌 또는 한 물체내의 두 부분의 충돌을 검출하는 방법에 관한 것이며, 보다 구체적으로 물체 간 충돌 검사를 위한 태스크 분배 방법에 관한 것이다. 여기서 컴퓨터 시스템은 CPU를 포함할 수 있다.
- [0025] 먼저 첫 번째 단계 S110에서, 씬(scene)에 포함된 물체의 기하학적 정보로부터 각 물체의 바운딩 볼륨 체계(bounding volume hierarchies, 이하 BVH라고 함)를 생성한다. BVH는 연속충돌 검사에서 가속화를 위하여 사용되는 기법으로서 바운딩 볼륨(bounding volume, BV)을 계층적(예를 들어 트리 구조로)으로 나타낸 데이터 구조이다. 바운딩 볼륨은 물체의 부피 일부 또는 전부를 감싸는 영역을 의미한다. 여러 종류의 바운딩 볼륨이 공지되어 있으며, 본 발명에서는 그 중 가장 널리 사용되고 있는 축-정렬 바운딩 박스(axis-aligned bounding boxes, AABBs)를 사용하는 경우를 가정하고 이후의 설명을 계속한다. AABB가 아니더라도 구(spheres), 오리엔티드 바운딩 박스(oriented bounding boxes, OBBs 등의 BV를 이용해서 본 발명을 실시할 수도 있다. 정리하면, BVH는 씬 내에 포함된 물체들을 바운딩 볼륨으로 구분하고, 그 바운딩 볼륨을 또 다시 복수(예를 들어 2개)의 바운딩 볼륨으로 쪼개는 것을 반복하여 최종적으로 바운딩 볼륨을 기초 다각형으로 쪼갠 것을 계층화된 데이터 구조로 나타낸 것이다. BVH가 트리 구조인 경우, 말단 노드에 기초 다각형들이 존재하게 된다.
- [0026] 그 다음 단계 S120에서, 각 물체의 BVH로부터 하나의 메인 BVH를 생성한다. 각 BVH들의 루트(root) 노드들을 묶는 상위 노드들을 추가하여 하나의 메인 BVH를 생성할 수 있다. 이렇게 함으로써, 여러 물체들을 하나의 물체로 취급하면서 물체 간 충돌을 물체 내부의 충돌과 마찬가지로 취급할 수 있다.
- [0027] 한편, 도시하지 않았으나, 물체 모델이 변형(deforming)되고 있는 경우에는 BVH를 업데이트할 수 있다. BVH를

업데이트하는 방법으로는 재생성이 필요한 부분만 재생성하고 그렇지 않은 부분은 갱신만 하는 선택적 재구성(selective reconstruction) 방법을 사용할 수 있다. 그 구체적인 방법은, [YCM07] YOON S., CURTIS S., MANOCHA D.: Ray tracing dynamic scenes using selective restructuring. Eurographics Symp. on Rendering (2007), 73-84. 3에 개시된 방법을 이용할 수 있다. BVH 전체 또는 일부를 수시로 업데이트할 수도 있고, 필요에 따라 선택적으로 BVH를 재구성할 수도 있다.

[0028] 그 다음 단계 S130에서, 하나의 메인 BVH에서 서로 상호 무관성이 성립하는 복수의 서브트리를 추출한다. 이를 위해 먼저, 물체 내 충돌 검사를 물체 간 충돌 검사로 바꾸어 준다. 이를 표현하기 위하여, 도 2에 도시된 바와 같이, 한 물체를 표현한 계층적 자료구조에서 한 부모 노드 N이 없다고 가정하면, 그 부모 노드 N의 바로 하위에 존재하는 자식 노드 A, B는 각각의 물체가 되고, 그 자식 노드 A, B 사이의 충돌은 물체 간 충돌 검사와 같게 된다는 성질을 갖게 된다. 이와 같이, 물체 간 충돌 검사를 위해 가정한 자식 노드 A, B는 하나의 서브트리 가 되며, 서로 상호 무관성(Disjoint property)을 갖는다.

[0029] 상호 무관성은 두 상호 충돌 검사를 구성하는 물체들이 서로 계층적 자료구조 상에서 부모-자식 관계가 없는 경우 충돌 검사를 수행하는 동안 방문하는 노드들은 겹치는 경우가 없다는 성질이다. 예를 들어 도 3에 도시된 바와 같이, 물체 C, D, E 및 F는 서로 부모-자식 관계가 없으므로, (C, D)를 루트 노드로 갖는 서브트리와 (E, F)를 루트 노드로 갖는 서브트리 사이의 물체 간 충돌 검사 동안에는 서로 접근하는 노드가 완전히 독립적이라는 것이다. 다른 예로, 도 4에 도시된 노드 n4, n5, n6, n7은 서로 상호 무관성이 성립한다. 만약, 할당된 노드들 사이에 상호무관 성질이 성립한다면 록킹 메커니즘을 사용하지 않고 레이지 BV 재구성(Lazy BVH reconstruction)을 사용하면서 BVH를 탐색할 수 있게 된다. 레이지 BV 재구성이란, 노드의 상태를 현재 상태에 맞추어 업데이트하는 작업이, 계층적(hierarchy) 탐색 중 해당 노드가 처음 방문될 때 수행된다는 의미이다. 노드의 업데이트 작업은 write operation이 필요하기 때문에 록킹이 필요하지만, 본 발명에서 제안하는 상호 무관성을 이용한 테스트 분배 방법에 따르면, 하나의 노드를 여러 스레드가 방문하는 경우가 없기 때문에 록킹 메커니즘을 사용할 필요가 없다.

[0030] 최초 할당되는 노드들 사이에 상호 무관성이 있도록 하기 위해서, 넓이우선(breadth-first) 탐색을 수행해서 이 성질이 만족하는 노드를 사용 가능한 스레드 수만큼 추출할 수 있다. 예를 들어 도 4에 나타낸 바와 같이, 사용 가능한 스레드 수가 4개인 경우, BVH의 루트에서 같은 깊이를 갖는 노드의 개수가 4가 되는 노드들을 선택할 수 있다. 이렇게 선택한 노드들(n4, n5, n6, n7)과 그 자식 노드들을 로우-레벨(low-level) 노드라고 칭하며, 나머지는 하이-레벨(high-level) 노드라고 한다. 즉, 도 4의 예에서 n1, n2, n3가 하이-레벨 노드이고 나머지는 로우-레벨 노드이다.

[0031] 그 다음 단계 S140에서, 복수의 서브트리에 구성된 노드들을 자식 노드로 하는 부모 노드 각각에 대하여 물체 간 충돌 검사를 처리하기 위해, 태스크 분배가 이루어진다. 이때, 복수의 서브트리에 구성된 노드들을 복수의 CPU 스레드에 각각 할당한다. 예를 들어 도 4를 참조하면, 추출한 서브트리들이 n4, n5, n6, n7 루트 노드로 구성되어 있을 경우, 각각의 노드들은 CPU 스레드(Thread 1, Thread 2, Thread 3, Thread 4)에 각각 할당될 수 있다. 즉, n4는 Thread 1으로, n5는 Thread 2로, n6는 Thread 3로, n7은 Thread 4로 각각 할당될 수 있다. 이러한 과정은 충돌검사 태스크의 초기 분배 과정에 해당하며, 이후, 동적 태스크 재분배가 추가로 이루어 질 수 있다. 동적 태스크 재분배에 대해서는 후술하도록 한다.

[0032] 그 다음 단계 S150에서, 부모 노드 사이의 충돌검사를 수행하기 위하여, 자식 노드 쌍들로 구성된 집합인 ICTPS(inter collision test pair set)를 생성한다. 도 4를 참조하면, 부모 노드 n2에 대하여 ICTPS를 처리(ICTPS(n2))(S151)한다고 할 경우, ICTPS(n2)는 n2의 자식 노드 n4와 n5를 루트로 가지는 서브트리 사이의 충돌 검사이므로, n4 또는 n5 노드로 구성된 노드 쌍들의 집합을 생성(S152)할 수 있다. 또한, ICTPS(n3)의 경우(S151), n3의 자식 노드 n6 또는 n7로 구성된 노드 쌍들의 집합을 생성(S152)할 수 있다. 이러한 ICTPS은 여러 개의 CPU 스레드에서 동시에 처리될 수 있다.

[0033] 이상의 단계는 CPU에서 처리될 수 있다.

[0034] 이후, GPU를 통해 TCTPS를 기반으로 충돌검사를 수행할 수 있다.

[0035] 이와 같이 복수의 CPU 스레드를 이용하여 상호 무관성이 성립하는 복수의 서브트리를 개별적으로 처리함으로써 CPU의 스레드 개수가 증가할수록 충돌 검사의 속도가 향상될 수 있다.

[0036] **동적 태스크 재분배(dynamic task reassignment)**

[0037] 복수의 스레드를 이용하는 경우 다이나믹 태스크 재분배를 이용하여 CPU 자원을 가장 효율적으로 이용할 수 있다.

[0038] 복수의 스레드 중 자신의 태스크를 빨리 끝내고 더 이상 태스크가 없는 스레드를 '요청 스레드'라고 하고, 태스크를 충분히 가지고 있는 스레드를 '배분 스레드'라고 부르기로 한다. 태스크 유닛 큐(task unit queue)의 프론트(front)(여기서 상호 무관성이 성립하는 노드들로 구성된 서브트리)에 있는 노드(이하, 프론트 노드라고 함)들 중 태스크가 많은 스레드를 배분 스레드로 선택한다. 이에 따라 요청 스레드는 태스크 재분배 요청을 한다.

[0039] 다음, 배분 스레드가 선택되면, 배분 스레드는 요청 스레드의 태스크 재분배 요청에 의해 프론트 노드를 요청 스레드에게 줌으로써 동적 태스크 재분배가 이루어진다. 이러한 동적 태스크 분배 방법의 이론적 근거는 태스크 유닛 큐 내의 모든 노드들은 상호 무관성을 만족한다는 것이다. 그리고 태스크 유닛 큐의 프론트 노드를 요청 스레드에 주는 이유는 프론트 노드의 서브트리에 가장 많은 태스크가 있다고 볼 수 있기 때문이다. 이와 같은 동적 태스크 재분배를 이용하여 8개의 CPU 코어를 이용해서 최대 7배의 성능향상을 얻을 수 있었다.

[0040] 요청 스레드는 분배 스레드의 스케줄링 큐에 태스크 요청 메시지를 입력한다. 분배 스레드의 스케줄링 큐에 접근하기 위해서는 분배 스레드의 록킹이 필요하지만 여러 스레드가 같은 스레드의 스케줄링 큐에 접근하는 경우는 드물기 때문에 성능에 큰 영향을 미치지 않는다. 요청 메시지를 분배 스레드의 스케줄링 큐에 입력한 후에 요청 스레드는 슬립(sleep) 상태가 된다.

[0041] 각 스레드는 하나의 충돌 검사 작업을 끝낸 후에 자신의 스케줄링 큐를 확인한다. 만약 태스크 분배 요청이 있다면 태스크 유닛 큐의 프론트 노드를 요청 노드에 제공한다. 그 다음 슬립 상태인 요청 스레드에게 웨이크업(wake up) 메시지를 전송하고, 요청 스레드는 웨이크업 메시지를 받으면 다시 충돌 검사 작업을 재개할 수 있다.

[0042] 본 발명에 의하면 충돌 검사 작업의 속도를 향상시킬 수 있고, CPU의 스레드 개수가 증가하는 경우에 충돌 검사 방법의 높은 확장성을 보장할 수 있다.

[0043] 또한, CPU의 스레드 개수가 증가할수록 충돌 검사의 속도가 향상될 수 있다.

[0044] 또한, 복수의 스레드를 갖는 CPU를 이용하여 물체 충돌 검사를 수행함에 있어 핵심 루프에서 록킹을 사용하지 않도록 해주며, 효과적인 태스크 분배를 가능하다.

[0045] 이상에서 설명한 충돌 검출 방법은, 컴퓨터에 의해 그 방법을 실행할 수 있는 프로그램이 탑재된 컴퓨터 판독가능 매체로서 구현될 수 있다. 따라서, 상술한 충돌 검출 방법을 실행할 수 있는 프로그램이 포함된 플로피 디스크, CD-ROM, RAM, EEPROM, 마이크로 프로세서, 광자기 디스크, 하드 디스크 등의 프로그램 저장 디바이스 또한 본 발명의 권리 범위에 속하는 것으로 이해되어야 한다.

[0046] 이상에서 보는 바와 같이, 본 발명이 속하는 기술 분야의 당업자는 본 발명이 그 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다.

[0047] 그러므로 이상에서 기술한 실시 예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로서 이해해야만 하고, 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 등가개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

도면의 간단한 설명

[0048] 도 1은 본 발명의 실시예에 따른 충돌 검사 방법의 순서도.

[0049] 도 2는 물체 내 충돌 검사를 물체 간 충돌 검사로 바꾸는 방법을 개념적으로 나타낸 도면.

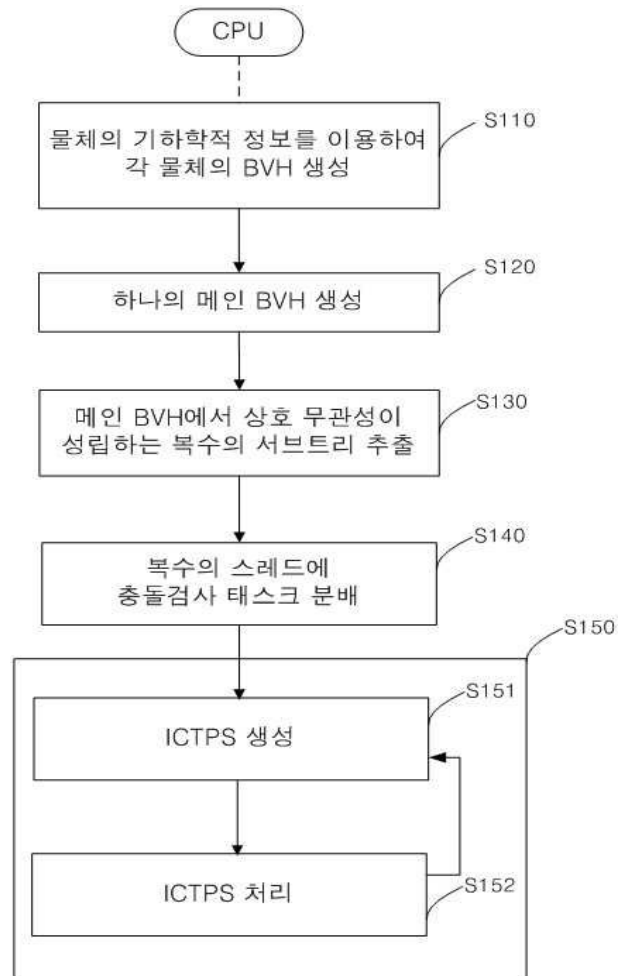
[0050] 도 3은 노드 간 상호 무관성을 설명하기 위해 나타낸 도면.

[0051] 도 4는 BVH 트리 구조와 초기 태스크를 분배하는 방법을 설명하기 위한 도면.

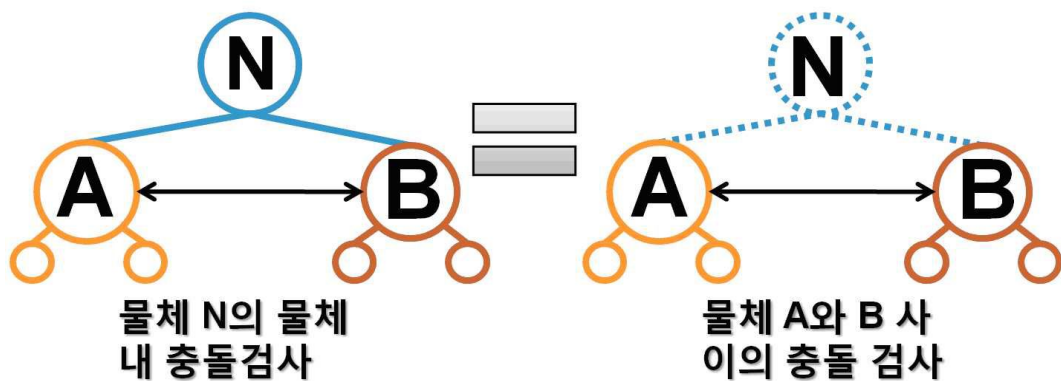
[0052] 도 5는 본 발명에 따른 충돌 검사 방법이 수행되는 컴퓨터 시스템의 블록도와 그 구성요소간 데이터 전송 경로를 나타낸 도면.

도면

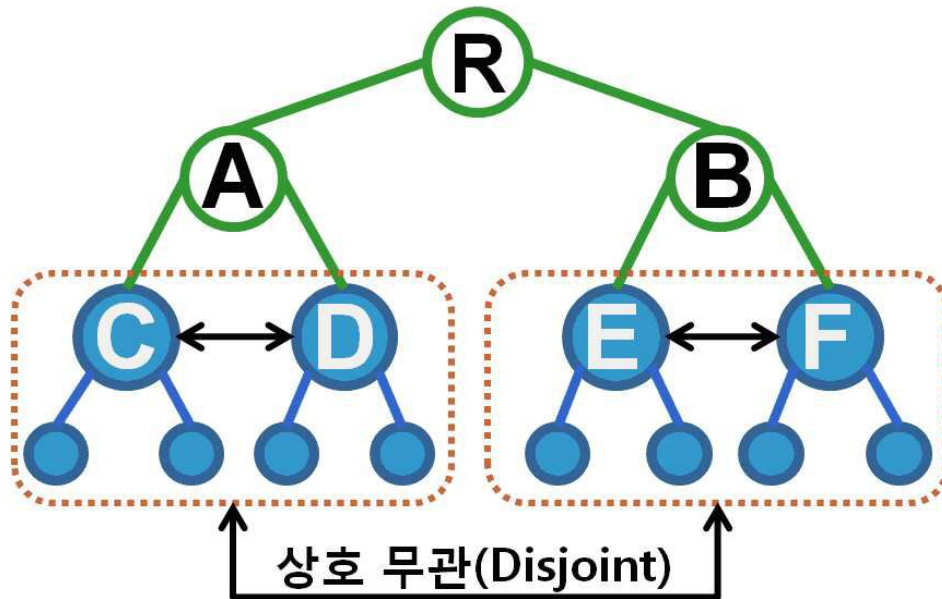
도면1



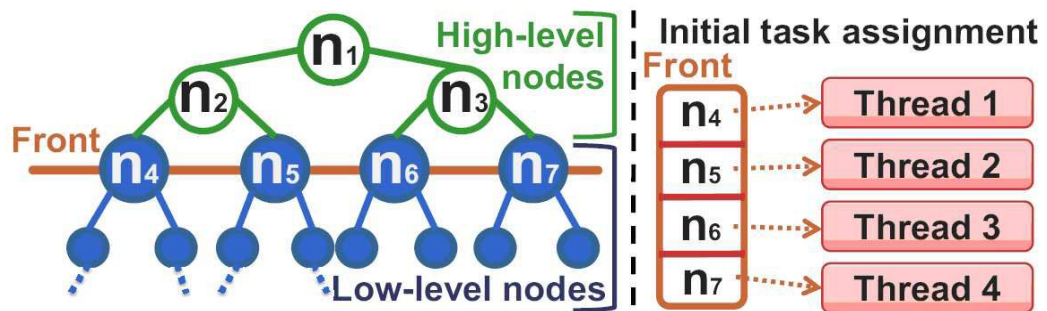
도면2



도면3



도면4



도면5

