



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0077879
(43) 공개일자 2010년07월08일

(51) Int. Cl.

G06F 9/44 (2006.01) G06F 17/10 (2006.01)
G06F 9/00 (2006.01)

(21) 출원번호 10-2008-0135952

(22) 출원일자 2008년12월29일

심사청구일자 2008년12월29일

(71) 출원인

이화여자대학교 산학협력단

서울 서대문구 대현동 11-1 이화여자대학교내

(72) 발명자

김영준

서울특별시 마포구 창전동 442번지 서강 한화 오벨리스크 102동 1104호

탕민

서울 서대문구 대현동 11-1 이화여자대학교 미관 321호

(74) 대리인

특허법인우인

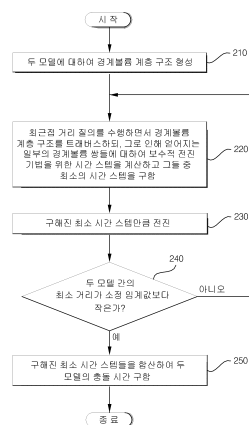
전체 청구항 수 : 총 24 항

(54) 보수적 전진 기법을 이용한 실시간 충돌 검사 방법 및 시스템

(57) 요약

보수적 전진 기법을 이용한 실시간 충돌 검사 방법 및 시스템이 개시된다. 본 발명에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 방법은 (a) 충돌을 검사하고자 하는 두 모델에 대하여 경계볼륨 계층구조를 형성하는 단계; (b) 최근접 거리 질의를 수행하면서 상기 형성된 경계볼륨 계층구조를 트래버스하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고, 그들 중 최소의 시간 스텝을 구하는 단계; (c) 상기 구해진 최소의 시간 스텝에 따라 상기 두 모델 중 적어도 한 모델을 전진시킨 후 상기 (b) 단계를 반복하는 단계; 및 (d) 상기 반복되는 (b) 단계를 통하여 구해진 시간 스텝들을 합계하여 상기 두 모델의 충돌 시간을 구하는 단계를 포함하는 것을 특징으로 한다. 이러한 본 발명에 의하면 연속적 충돌 검사를 실시간으로 수행할 수 있으며, 처리 속도가 보다 빠르고 폴리곤 수프 모델에 대한 연속적 충돌 검사 속도를 개선할 수 있다.

대표도 - 도2



특허청구의 범위

청구항 1

보수적 전진 기법을 이용한 실시간 충돌 검사 방법에 있어서,

- (a) 충돌을 검사하고자 하는 두 모델에 대하여 경계볼륨 계층구조를 형성하는 단계;
- (b) 최근접 거리 질의를 수행하면서 상기 형성된 경계볼륨 계층구조를 트래버스하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고, 그들 중 최소의 시간 스텝을 구하는 단계;
- (c) 상기 구해진 최소의 시간 스텝에 따라 상기 두 모델 중 적어도 한 모델을 전진시킨 후 상기 (b) 단계를 반복하는 단계; 및
- (d) 상기 반복되는 (b) 단계를 통하여 구해진 시간 스텝들을 합계하여 상기 두 모델의 충돌 시간을 구하는 단계를 포함하는 것을 특징으로 하는 충돌 검사 방법.

청구항 2

제1항에 있어서,

상기 (b) 단계는, 상기 최근접 거리 질의를 수행하면서 얻어지는 경계볼륨 쌍의 거리에 따라서 상기 경계볼륨 계층구조의 트래버스를 터미네이트하고 그때의 경계볼륨 쌍의 시간 스텝을 계산하는 것을 특징으로 하는 충돌 검사 방법.

청구항 3

제2항에 있어서,

상기 (b) 단계는, 해당 경계볼륨 쌍의 거리와 앞서 얻어진 현재 최근접 거리를 비교하고 그 결과에 따라서 상기 경계볼륨 계층구조의 트래버스를 터미네이트하고 그때의 경계볼륨 쌍의 시간 스텝을 계산하는 것을 특징으로 하는 충돌 검사 방법.

청구항 4

제2항에 있어서,

적어도 초기 일부 반복의 상기 (b) 단계에서, 해당 경계볼륨 쌍의 거리와 앞서 얻어진 현재 최근접 거리보다 작게 수정된 수정 현재 최근접 거리를 비교하고 그 결과에 따라서 상기 경계볼륨 계층구조의 트래버스를 터미네이트하고 그때의 경계볼륨 쌍의 시간 스텝을 계산하는 것을 특징으로 하는 충돌 검사 방법.

청구항 5

제4항에 있어서,

상기 해당 경계볼륨 쌍의 거리가 상기 수정 현재 최근접 거리보다 작은 경우에 상기 경계볼륨 계층구조의 트래버스를 터미네이트하는 것을 특징으로 하는 충돌 검사 방법.

청구항 6

제4항에 있어서,

상기 수정 현재 최근접 거리는 상기 앞서 얻어진 현재 최근접 거리에 0에서 1 사이의 소정 값을 곱함으로써 얻어지는 것을 특징으로 하는 충돌 검사 방법.

청구항 7

제4항에 있어서,

상기 (b) 단계에서, 상기 앞서 얻어진 현재 최근접 거리가 소정 임계값보다 작은 경우, 상기 수정 현재 최근접 거리를 이용하지 않고 상기 해당 경계볼륨 쌍의 거리와 상기 앞서 얻어진 현재 최근접 거리를 비교하는 것을 특

정으로 하는 충돌 검사 방법.

청구항 8

제4항에 있어서,

상기 (b) 단계에서, 상기 (b) 단계의 반복의 회수가 소정 임계값보다 큰 경우, 상기 수정 현재 최근접 거리를 이용하지 않고 상기 해당 경계볼륨 쌍의 거리와 상기 앞서 얻어진 현재 최근접 거리를 비교하는 것을 특징으로 하는 충돌 검사 방법.

청구항 9

제1항에 있어서,

상기 (b) 단계에서 상기 계산된 시간 스텝이 소정 값보다 큰 경우 상기 경계볼륨 계층구조에서 해당하는 노드들 및 그 자식 노드들을 전지하는 것을 특징으로 하는 충돌 검사 방법.

청구항 10

제8항에 있어서,

상기 소정 값은 상기 (b) 단계의 처음 실행에서 1로 설정되고, 다음 반복들에서는 다음 수학식에 따라 설정되는 것을 특징으로 하는 충돌 검사 방법.

$$\Lambda_k = 1 - \sum_{i=1}^{k-1} \Delta t_i$$

여기서, k는 상기 반복의 회수를 나타내고, Δt_i 는 i 번째 반복에서 구해진 상기 최소의 시간 스텝이다.

청구항 11

제1항에 있어서,

상기 경계볼륨은 swept sphere volume인 것을 특징으로 하는 충돌 검사 방법.

청구항 12

제11항에 있어서,

상기 보수적 전진 기법을 위한 시간 스텝은 상기 경계볼륨 쌍 사이의 가장 가까운 거리에 대한 하계와 단위 시간당 상기 가장 가까운 거리로 사영되는 한 경계볼륨의 모선의 상계인 모선 바운드에 기초하여 계산되는 것을 특징으로 하는 충돌 검사 방법.

청구항 13

제12항에 있어서,

상기 한 경계볼륨 a 가 병진 속도 v 와 회전 속도 ω 를 가지는 강체 변환 M(t)를 따를 때 $\mathbf{p}_i$ 를 a 상의 점이라 하고, $\mathbf{n}$ 을 상기 a와 다른 경계볼륨 β 사이의 가장 가까운 방향이라 하고 $\mathbf{r}_i$ 를 원점으로부터 상기 $\mathbf{p}_i$ 로의 벡터라 하면,

상기 모선 바운드는, $\mathbf{c}(t) = \mathbf{R}^{-1}(t)\mathbf{n} \times \omega$ 가 형성하는 평면 S 상으로의 상기 $\mathbf{r}_i$ 의 최대 투사에 의해 얻어지는 것을 특징으로 하는 충돌 검사 방법.

청구항 14

제13항에 있어서,

상기 모션 바운드는 다음 수학적식에 따라 얻어지는 것을 특징으로 하는 충돌 검사 방법.

$$\mu \leq |\mathbf{v} \cdot \mathbf{n}| + |\boldsymbol{\omega} \times \mathbf{n}| (r + \max(|\mathbf{c}_i^\perp|))$$

여기서, $|\mathbf{c}_i^\perp| = \frac{|\mathbf{c}_i \times \boldsymbol{\omega}|}{|\boldsymbol{\omega}|}$ 이고, r 은 SSV를 형성하기 위해 사용되는 구의 반지름이며, $\mathbf{c}_i$ 는 SSV의 생성 프리미티브의 끝점(endpoint)이다.

청구항 15

제1항에 있어서,

상기 (b) 단계는 상기 최근접 거리 질의에 따라 얻어지는 상기 두 모델 간의 최근접 거리가 소정 임계값보다 작을 때까지 반복되는 것을 특징으로 하는 충돌 검사 방법.

청구항 16

제1항 내지 제 15항 중 어느 한 항에 기재된 보수적 전진 기법을 이용한 실시간 충돌 검사 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

청구항 17

보수적 전진 기법을 이용한 실시간 충돌 검사 시스템에 있어서,

충돌을 검사하고자 하는 두 모델에 대하여 경계볼륨 계층구조를 형성하는 경계볼륨 계층구조 형성부;

최근접 거리 질의를 수행하면서 상기 형성된 경계볼륨 계층구조를 트래버스하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고 그들 중 최소의 시간 스텝을 구하며, 모델 전진부에 의해 전진된 모델을 가지고 이를 반복 수행하는 시간 스텝 계산부,

상기 구해진 최소 시간 스텝만큼 상기 두 모델 중 적어도 한 모델을 전진시키는 모델 전진부;

상기 시간 스텝 계산부를 통하여 구해진 시간 스텝들을 합계하여 상기 두 모델의 충돌 시간을 구하는 충돌 시간 계산부를 포함하는 것을 특징으로 하는 충돌 검사 시스템.

청구항 18

제17항에 있어서,

상기 시간 스텝 계산부는, 상기 최근접 거리 질의를 수행하면서 얻어지는 경계볼륨 쌍의 거리에 따라서 상기 경계볼륨 계층구조의 트래버스를 터미네이트하고 그때의 경계볼륨 쌍의 시간 스텝을 계산하는 것을 특징으로 하는 충돌 검사 시스템.

청구항 19

제18항에 있어서,

상기 시간 스텝 계산부는, 해당 경계볼륨 쌍의 거리와 앞서 얻어진 현재 최근접 거리를 비교하고 그 결과에 따라서 상기 경계볼륨 계층구조의 트래버스를 터미네이트하고 그때의 경계볼륨 쌍의 시간 스텝을 계산하는 것을 특징으로 하는 충돌 검사 시스템.

청구항 20

제18항에 있어서,

상기 시간 스텝 계산부의 적어도 초기 일부의 반복에서, 해당 경계볼륨 쌍의 거리와 앞서 얻어진 현재 최근접 거리보다 작게 수정된 현재 최근접 거리를 비교하고 그 결과에 따라서 상기 경계볼륨 계층구조의 트래버스를 터미네이트하고 그때의 경계볼륨 쌍의 시간 스텝을 계산하는 것을 특징으로 하는 충돌 검사 시스템.

청구항 21

보수적 전진 기법을 이용한 실시간 충돌 검사 방법에 있어서,

최근접 거리 질의를 수행하면서 보수적 전진을 위한 시간 스텝을 구하는 과정을 반복하되, 상기 시간 스텝을 구하는 과정을 위한 입력으로서, 두 모델의 경계볼륨 계층구조의 두 노드와, 앞서 구해진 현재 최근접 거리와, 앞서 구해진 현재 시간 스텝을 입력받고,

상기 시간 스텝을 구하는 과정은,

(a) 상기 두 노드가 리프 노드라면 해당 경계볼륨 쌍의 거리를 구하고 상기 구해진 거리를 상기 현재 최근접 거리와 비교하여 그 결과에 따라 상기 현재 최근접 거리를 업데이트하고 상기 해당 경계볼륨 쌍에 대하여 보수적 전진을 위한 시간 스텝을 계산하고 상기 계산된 시간 스텝을 상기 현재 시간 스텝과 비교하여 그 결과에 따라 상기 현재 시간 스텝을 업데이트하는 단계; 및

(b) 상기 두 노드 중 적어도 하나가 리프 노드가 아니라면, 상기 적어도 하나의 노드의 자식 노드 및 다른 노드에 해당하는 경계볼륨 쌍의 거리를 구하고 상기 구해진 거리를 상기 현재 최근접 거리와 비교하고, 그 결과에 따라 선택적으로, 상기 자식 노드 및 상기 다른 노드를 입력으로 하여 재귀적으로 상기 시간 스텝을 구하는 과정을 수행하거나 상기 자식 노드 및 상기 다른 노드에 해당하는 경계볼륨 쌍에 대하여 보수적 전진을 위한 시간 스텝을 계산하는 단계를 포함하는 것을 특징으로 하는 충돌 검사 방법.

청구항 22

제21항에 있어서,

상기 시간 스텝을 구하는 과정의 적어도 일부의 초기 일부 반복에서, 상기 구해진 거리를 상기 현재 최근접 거리 대신에 상기 현재 최근접 거리보다 작도록 수정된 수정 현재 최근접 거리와 비교하는 것을 특징으로 하는 충돌 검사 방법.

청구항 23

제21항에 있어서,

상기 (b) 단계에서 상기 계산된 시간 스텝이 소정 값보다 큰 경우 상기 경계볼륨 계층구조에서 해당하는 노드들 및 그 자식 노드들을 전지하는 단계를 더 포함하는 것을 특징으로 하는 충돌 검사 방법.

청구항 24

제21항 내지 제 23항 중 어느 한 항에 기재된 보수적 전진 기법을 이용한 실시간 충돌 검사 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 실시간 충돌 검사 방법 및 시스템에 관한 것으로, 특히 보수적 전진 기법을 이용한 실시간 충돌 검사 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 충돌과 거리 질의는 로봇틱스, 컴퓨터 그래픽, 그리고 기하학적 모델링 등에서 중요한 문제이다. 특히 신뢰성 있고 빠른 충돌 검사 알고리즘은 로봇 모션 플래닝(robot motion planning)과 비침투(non-penetration) 제약 조건을 필요로 하는 동적 시뮬레이션을 위해 요구된다. 충돌 검사 문제는 그 분야의 많은 연구자들에 의해 광범위하게 연구되어 왔다. 충돌 검사에 대한 초기의 접근 대부분은 질의된 오브젝트들이 공간 내에서 정적으로 놓여지는 정적 장면(static scenes)에 초점을 맞추었다. 최근에 연구자들은 오브젝트들이 움직임 가능한 동적 장면(dynamic scenes)에 초점을 맞추고 있다. 몇 가지 동적 시나리오에서는 오브젝트의 전체 모션 궤도(motion trajectory)는 시간의 함수로서 미리 알려질 수 있다. 대부분의 실제 시나리오들에서는 오브젝트의 모션은 선형적으로 알려지지 않고, 몇 개의 샘플된 위치들만이 알려진다. 예를 들어, 샘플링 기반의 로봇 모션 플래닝에서는 로봇의 충돌 없는 컨피규레이션들을 샘플링하고, 그것들을 초기로부터 최종 컨피규레이션까지의 연속적인 충

돌 없는 경로를 계산하는 데 사용한다.

[0003] 넓은 수준에서 동적 충돌 검사 알고리즘은 불연속과 연속의 두 카테고리로 나누어질 수 있다. 불연속 알고리즘들은 정적 충돌 검사 알고리즘을 사용하여 오직 샘플링된 컨피규레이션들에서 충돌을 검사한다. 그 결과, 정적 충돌 검사 알고리즘은 두 잇따른 컨피규레이션들 사이에서 충돌을 놓칠 수 있다. 이것은 터널링 문제(tunnelling problem)라고도 알려져 있다. 반면에, 연속적 충돌 검사(Continuous Collision Detection, CCD) 알고리즘은, 첫 번째로, 잇따른 컨피규레이션들 사이에서 연속적인 모션 보간(motion interpolation)을 수행하고, 보간된 모션을 사용하여 충돌을 검사함으로써 터널링 문제를 해결할 수 있다. 따라서 CCD는 컨피규레이션들 사이의 어떠한 발생 가능한 충돌이라도 놓치지 않는다. 그러나 CCD 알고리즘의 주된 약점은 불연속 알고리즘보다 일반적으로 느리다는 것이다. 그리고 연속적 충돌 검사에서 대부분의 앞선 연구들은 다면체 모델(polyhedral model) 또는 관절 연결 모델(articulated model)에 제한되며, 이들은 커넥티비티 정보(connectivity information) 또는 토폴로지 정보(topological information)를 고려한다. 이러한 접근 방식은 커넥티비티 정보가 없는 트라이앵글 수프 모델(triangle-soup model)과 같은 폴리곤 수프 모델(polygon-soup model)로 표현되는 일반적인 모델 상에서는 처리가 불가능하다.

발명의 내용

해결 하고자하는 과제

[0004] 본 발명이 이루고자 하는 기술적 과제는 연속적 충돌 검사를 실시간으로 수행할 수 있으며, 처리 속도가 빠르고 폴리곤 수프 모델에도 적용 가능한, 보수적 전진 기법을 이용한 실시간 충돌 검사 방법 및 시스템, 그리고 상기 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공하는 데 있다.

과제 해결수단

[0005] 상기 기술적 과제를 해결하기 위하여 본 발명의 일 태양에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 방법은, (a) 충돌을 검사하고자 하는 두 모델에 대하여 경계볼륨 계층구조를 형성하는 단계; (b) 최근접 거리 질의를 수행하면서 상기 형성된 경계볼륨 계층구조를 트래버스하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고, 그들 중 최소의 시간 스텝을 구하는 단계; (c) 상기 구해진 최소의 시간 스텝에 따라 상기 두 모델 중 적어도 한 모델을 전진시킨 후 상기 (b) 단계를 반복하는 단계; 및 (d) 상기 반복되는 (b) 단계를 통하여 구해진 시간 스텝들을 합계하여 상기 두 모델의 충돌 시간을 구하는 단계를 포함하는 것을 특징으로 한다.

[0006] 상기 기술적 과제를 해결하기 위하여 본 발명의 일 태양에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 시스템은, 충돌을 검사하고자 하는 두 모델에 대하여 경계볼륨 계층구조를 형성하는 경계볼륨 계층구조 형성부; 최근접 거리 질의를 수행하면서 상기 형성된 경계볼륨 계층구조를 트래버스하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고 그들 중 최소의 시간 스텝을 구하며, 모델 전진부에 의해 전진된 모델을 가지고 이를 반복 수행하는 시간 스텝 계산부; 상기 구해진 최소 시간 스텝만큼 상기 두 모델 중 적어도 한 모델을 전진시키는 모델 전진부; 및 상기 시간 스텝 계산부를 통하여 구해진 시간 스텝들을 합계하여 상기 두 모델의 충돌 시간을 구하는 충돌 시간 계산부를 포함하는 것을 특징으로 한다.

[0007] 상기 기술적 과제를 해결하기 위하여 본 발명의 다른 일 태양에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 방법은, 최근접 거리 질의를 수행하면서 보수적 전진을 위한 시간 스텝을 구하는 과정을 반복하되, 상기 시간 스텝을 구하는 과정을 위한 입력으로서, 두 모델의 경계볼륨 계층구조의 두 노드와, 앞서 구해진 현재 최근접 거리와, 앞서 구해진 현재 시간 스텝을 입력받고, 상기 시간 스텝을 구하는 과정은, (a) 상기 두 노드가 리프 노드라면 해당 경계볼륨 쌍의 거리를 구하고 상기 구해진 거리를 상기 현재 최근접 거리와 비교하여 그 결과에 따라 상기 현재 최근접 거리를 업데이트하고 상기 해당 경계볼륨 쌍에 대하여 보수적 전진을 위한 시간 스텝을 계산하고 상기 계산된 시간 스텝을 상기 현재 시간 스텝과 비교하여 그 결과에 따라 상기 현재 시간 스텝을 업데이트하는 단계; 및 (b) 상기 두 노드 중 적어도 하나가 리프 노드가 아니라면, 상기 적어도 하나의 노드의 자식 노드 및 다른 노드에 해당하는 경계볼륨 쌍의 거리를 구하고 상기 구해진 거리를 상기 현재 최근접 거리와 비교하고, 그 결과에 따라 선택적으로, 상기 자식 노드 및 상기 다른 노드를 입력으로 하여 재귀적으로 상기 시간 스텝을 구하는 과정을 수행하거나 상기 자식 노드 및 상기 다른 노드에 해당하는 경계볼륨 쌍에 대하여 보수적 전진을 위한 시간 스텝을 계산하는 단계를 포함하는 것을 특징으로 한다.

[0008] 또한, 상기 기술적 과제를 해결하기 위하여 상기된 본 발명에 따른, 보수적 전진 기법을 이용한 실시간 충돌 검

사 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공한다.

효 과

[0009] 상기된 본 발명에 의하면, 연속적 충돌 검사를 실시간으로 수행할 수 있으며, 처리 속도가 보다 빠르고 폴리곤 수프 모델에 대한 연속적 충돌 검사 속도를 개선할 수 있다.

발명의 실시를 위한 구체적인 내용

[0010] 이하에서는 도면을 참조하여 본 발명의 바람직한 실시예들을 상세히 설명한다. 이하 설명 및 첨부된 도면들에서 실질적으로 동일한 구성요소들은 각각 동일한 부호들로 나타냄으로써 중복 설명을 생략하기로 한다. 또한 본 발명을 설명함에 있어 관련된 공지기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그에 대한 상세한 설명은 생략하기로 한다.

[0011] 먼저, 본 발명의 일 실시예에서 채용하는 보수적 전진(conservative advancement) 기법에 관하여 설명하기로 한다. 도 1은 보수적 전진 기법을 설명하기 위한 참고도로서, line swept sphere를 예로 든 것이다. $\mathcal{A}$ 와 $\mathcal{B}$ 를 3D 내의 두 폴리곤-수프 모델을 감싸는 컨벡스 경계 볼륨이라고 하고, 여기서 $\mathcal{A}$ 는 강체 변환(rigid transformation) $M(t)$ 하에서 움직일 수 있고, $\mathcal{B}$ 는 고정되어 있다고 하자. $\mathcal{B}$ 가 고정되어 있다고 하더라도 일반성을 잃지 않는다. 그리고 $\mathcal{A}$ 의 초기와 마지막 컨피규레이션이 시간 $t=0$ 와 시간 $t=1$ 에서 각각 $\mathbf{q}_0$ 와 $\mathbf{q}_1$ 으로 주어진다고 가정하고, $\mathcal{A}(t) = M(t)\mathcal{A}$ 라 정의한다. 그러면 연속적 충돌 검사의 문제는 다음 수학적 1 이 비공백(non-empty)인지 검사하는 것으로 공식화될 수 있다.

수학적 1

$$\{t \in [0, 1] \mid \mathcal{A}(t) \cap \mathcal{B} \neq \emptyset\}$$

[0013] 나아가, 만일 상기 수학적 1이 비공백이라면, 이식을 만족시키는 t 의 최소값인 최초 접촉 시간(time of contact, TOC) τ 를 계산한다.

[0014] 보수적 전진 기법은 두 컨벡스(convex) 오브젝트들 사이에서 충돌을 피하는 동안에 반복적으로 $\mathcal{A}$ 를 Δt_i 만큼 $\mathcal{B}$ 를 향하여 전진시킴으로써 τ 의 하계(lower bound)를 계산하는 단순한テクニック이다. 도 1을 참조하면, Δt_i 는 $\mathcal{A}$ 와 $\mathcal{B}$ 사이의 가장 가까운 거리 $d(\mathcal{A}(t), \mathcal{B})$ 에 대한 하계와 단위 시간당 $d(\mathcal{A}(t), \mathcal{B})$ 로 사영되는 $\mathcal{A}(t)$ 의 모션의 상계(upper bound) μ 에 기초하여 계산된다. 즉, 다음 수학적식과 같다.

수학적 2

$$\Delta t_i \leq \frac{d(\mathcal{A}(t), \mathcal{B})}{\mu}$$

[0016] 그러면, 최초 접촉 시간은, $\mathcal{A}(t)$ 와 $\mathcal{B}$ 간의 최소 거리인 $d(\mathcal{A}(t), \mathcal{B})$ 가 어떤 사용자 지정된 거리 임계값보다 작을 때까지, 전진 시간 스텝(advancement time-step)을 반복하고 그것을 합계함으로써(즉, $\tau = \sum \Delta t_i$) 구해진다.

[0017] 도 2는 본 발명의 일 실시예에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 방법의 흐름도이다. 본 실시예에서는, $\mathcal{A}$ 의 초기 및 최종 컨피규레이션만이 $\mathbf{q}_0$ 와 $\mathbf{q}_1$ 으로 주어지는 것으로 가정한다. 이 컨피규레이션들을 이용하여 $\mathbf{q}_0$ 와 $\mathbf{q}_1$ 을 보간하기 위하여, 일정한 병진 속도(translational velocity)와 회전 속도(rotational velocity)를 가지는 연속 모션 $M(t)$ 를 생성한다. 이에 관하여는 문헌 [X. Zhang, M. Lee, and Y. J. Kim, "Interactive continuous collision detection for non-convex polyhedra," The Visual Computer,

pp. 749??760, 2006.]에 자세히 설명되어 있다.

- [0018] 도 2를 참조하면, 전처리로서 210단계에서, 충돌을 검사하고자 하는 두 모델에 대하여 경계볼륨 계층구조(bounding volume hierarchy, BVH)를 형성한다. 일 실시예로서, swept sphere volume(SSV) 계층구조를 사용한다. SSV는 이른바 point swept sphere(PSS), line swept sphere(LSS) 그리고 rectangle swept sphere(RSS) 중 하나로 구성되는 경계볼륨이다. SSV 계층구조는 SSV를 노드로 사용하면서 재귀적으로 형성된다. 도 3은 두 모델에 대하여 형성된 경계볼륨 계층구조의 간단한 예를 나타낸다. SSV 계층구조의 형성과 그것을 사용한 거리 및 근접 계산에 관하여는 문헌 [E. Larsen, S. Gottschalk, M. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," Department of Computer Science, University of North Carolina, Tech. Rep. TR99-018, 1999.]에 자세히 설명되어 있다.
- [0019] 220단계에서는, 상기 형성된 BVH를 가지고, 최근접 거리 질의(closest distance queries)를 수행하면서 BVH를 트래버스(traverse)하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고 그들 중 최소의 시간 스텝을 구한다.
- [0020] 본 단계에서, 보수적 전진의 시간 스텝의 계산은 BVH 내의 노드들에 선택적으로 적용된다. 상기 수학식 2에서 알 수 있듯이, 보수적 전진 기법은 가장 가까운 거리와 모션 바운드의 두 가지 성분을 필요로 한다. 따라서 SSV들 간의 가장 가까운 거리와 모션 바운드를 계산할 필요가 있다. SSV들 간의 가장 가까운 거리는 최근접 거리 질의(closest distance query) 수행 동안에 부산물로서 얻어지나, SSV에 대한 모션의 밀착 상한(tight upper bound) μ 를 계산해야 한다. 이에 관하여는 후술하기로 한다.
- [0021] 또한, 본 단계에서 중요한 계산은 계층구조 내의 어떠한 노드들이 보수적 전진의 시간 스텝의 계산을 위한 후보가 될 것인지 결정하는 것이다. 한 가지 선택은 최근접 질의 동안에 BVH의 트래버스를 멈추는 노드-이들을 프론트 노드(front nodes)라 부른다-로 하는 것이다. 왜냐하면 이 노드들이 다른 노드들보다 τ 를 실현하기에 더 높은 확률을 가지기 때문이다. 그러나 계층구조 내의 프론트 노드들의 깊이는 비교적 깊고 따라서 프론트 노드들의 개수 또한 많다. 연속적 충돌 검사에서는 이 노드들에 반복적으로 보수적 전진 기법을 적용하는 것을 필요로 하기 때문에 프론트 노드들의 깊이 또는 개수는 알고리즘의 성능에 영향을 준다. 따라서 일 실시예에서, 보수적 전진의 반복 동안에 프론트 노드들의 깊이를 조절함으로써 성능을 향상시키는 기법을 제시한다. 이에 관하여도 후술하기로 한다.
- [0022] 도 2를 참조하면, 220단계에서 최소의 시간 스텝이 구해지면 230단계에서는 그 최소 시간 스텝만큼 두 모델 중 어느 한 모델(또는 두 모델을 각각)을 전진시킨다. 그리고 240단계에서 두 모델 간의 최근접 거리(이 거리는 상기 220단계의 최근접 거리 질의를 통해 구해진다)가 사용자 지정된 소정 임계값보다 작은지 검사하고, 그렇지 않다면 다시 220단계로 돌아가서 전진된 모델을 가지고 220단계를 수행한다.
- [0023] 220 내지 240단계를 다수 회 반복하고 난 뒤에, 240단계에서 두 모델 간의 거리가 상기 소정 임계값보다 작은 것으로 검사되면 250단계로 진행하여 반복된 220단계들을 통하여 구해진 시간 스텝들을 합계하여 두 모델의 충돌 시간(TOC)을 구한다.
- [0024] 이하, 본 발명의 일 실시예에 따라 SSV에 대한 모션의 밀착 상한 μ 를 계산하는 과정을 설명한다. 여기에서, 경계볼륨과 삼각형 프리미티브에 관한 모션 바운드 μ 를 구하는 방법을 제시한다. 이 프리미티브들은 일정한 병진 속도 v 와 회전 속도 ω 를 가지는 강체 변환 $M(t)$ 를 따르는 것으로 가정한다.
- [0025] 먼저, 경계볼륨에 대한 모션 바운드 μ 를 구하는 방법을 설명한다. 본 실시예에서 경계볼륨의 선택은 SSV이다. SSV는 PSS, LSS, 그리고 RSS 로 구성되는데, 이들은 3D 내에서 구(sphere)와 점(point), 직선(line), 그리고 직사각형(rectangle)의 민코우스키 합(Minkowski sum)들로 각각 정의된다. 도 4는 PSS, LSS, 그리고 RSS를 나타낸 도면이다. α 를 모션 바운드를 계산하고자 하는 $\mathcal{A}$ 를 구성하는 경계볼륨 또는 삼각형 프리미티브라 하고 β 를 $\mathcal{B}$ 를 구성하는 경계볼륨 또는 삼각형 프리미티브라 하자. 나아가, $\mathbf{p}_i$ 를 α 상의 점이라 하고, $\mathbf{n}$ 을 α , β 사이의 가장 가까운 방향이라 하고, $\mathbf{r}_i$ 를 $\mathcal{A}$ 가 속한 로컬 바디 프레임의(local body frame)의 원점으로 부터 $\mathbf{p}_i$ 로의 벡터라 하자. 그러면 $\mathbf{n}$ 위로 사영되는 α 의 최대 궤도 길이(또는 모션 바운드)는 다음 수학식과 같다.

수학식 3

$$\begin{aligned}\mu &= \max_i \int_0^1 |\dot{\mathbf{p}}_i(t) \cdot \mathbf{n}| dt \\ &\leq |\mathbf{v} \cdot \mathbf{n}| + \int_0^1 \max_i |\boldsymbol{\omega} \times \mathbf{n} \cdot \mathbf{r}_i(t)| dt \\ &\leq |\mathbf{v} \cdot \mathbf{n}| + \int_0^1 \max_i |\mathbf{c}(t) \cdot \mathbf{r}_i| dt\end{aligned}$$

[0026]

여기서, $\mathbf{c}(t) = \mathbf{R}^{-1}(t)\mathbf{n} \times \boldsymbol{\omega}$ 이고, $\mathbf{R}$ 은 $\mathbf{M}(t)$ 의 회전 성분이다. 상기 수학식 3에서, 첫 번째 항은 상수이므로, 두 번째 항을 최대화해야 한다. 여기서, $\mathbf{c}(t)$ 는 동일 평면 상에 있고(coplanar), 노멀(normal)이 $\boldsymbol{\omega}$ 인 평면 S를 형성하게 된다. 상기 수학식 3에서 두 번째 항의 극대치는 아래 수학식에 보여지는 바와 같이 평면 S 상으로의 $\mathbf{r}_i$ 의 최대 투사(maximal projection)에 의해 얻어질 수 있다. 즉, α 를 SSV라 하면, α 의 모션 바운드 μ 는 다음 수학식과 같다.

[0027]

수학식 4

$$\mu \leq |\mathbf{v} \cdot \mathbf{n}| + |\boldsymbol{\omega} \times \mathbf{n}| (r + \max(|\mathbf{c}_i^\perp|))$$

[0028]

여기서, $|\mathbf{c}_i^\perp| = \frac{|\mathbf{c}_i \times \boldsymbol{\omega}|}{|\boldsymbol{\omega}|}$ 이고, r 은 SSV를 형성하기 위해 사용되는 구의 반지름이며, $\mathbf{c}_i$ 는 SSV의 생성 프리미티브(PSS, LSS 및 RSS의 생성 프리미티브는 각각 점, 직선, 직사각형이다)의 끝점(endpoint)이다.

[0029]

나아가, 모든 세 타입의 SSV에 대하여 $\mathbf{r}_i = \mathbf{r} + \mathbf{k}$ 라 할 수 있다. 여기서 $\mathbf{k}$ 는 $\mathcal{A}$ 의 바디 프레임의 원점으로 부터 SSV의 생성 프리미티브 상의 한 점으로의 벡터이다. 그러면 다음 수학식을 얻을 수 있다.

[0030]

수학식 5

$$|\mathbf{c}(t) \cdot \mathbf{r}_i| = |\mathbf{c}(t) \cdot \mathbf{r}_i^\perp| \leq |\mathbf{c}(t)| \cdot |\mathbf{r}_i^\perp| \leq |\mathbf{c}(t)| r + |\mathbf{c}(t)| |\mathbf{k}^\perp|$$

[0031]

여기서, $\mathbf{k}^\perp$ 는 $\mathbf{k}$ 의 평면 S로의 사영이다.

[0032]

사영 기하 이론에 따르면, 점의 평면으로의 사영은 점이고, 직선의 평면으로의 사영은 직선 또는 점이며, 직사각형의 평면으로의 사영은 사각형 또는 직선이다. 따라서 $\mathbf{k}$ 의 평면 S로의 사영은 SSV의 생성 프리미티브의 평면 S로의 사영 내에 존재하여야 하고, 다음 수학식을 얻을 수 있다.

[0033]

수학식 6

$$|\mathbf{k}^\perp| \leq \max(|\mathbf{c}_i^\perp|), \quad |\mathbf{c}_i^\perp| = \frac{|\mathbf{c}_i \times \boldsymbol{\omega}|}{|\boldsymbol{\omega}|}$$

[0034]

상기된 수학식 3, 5, 6에 의해 상기 수학식 4가 증명된다.

[0035]

이제 삼각형 프리미티브에 대한 모션 바운드 μ 를 구하는 방법을 설명한다. 삼각형 프리미티브의 모션 바운드는 삼각형의 정점들(vertices)을 사영의 범위의 경계로 사영함으로써 얻을 수 있다. 즉, α 를 삼각형 프리미티브라 하면, α 의 모션 바운드 μ 는 다음 수학식과 같다.

[0036]

수학식 7

$$\mu \leq |\mathbf{v} \cdot \mathbf{n}| + |\boldsymbol{\omega} \times \mathbf{n}| (\max(|\mathbf{c}_i^\perp|))$$

여기서, $|\mathbf{c}_i^\perp| = \frac{|\mathbf{c}_i \times \boldsymbol{\omega}|}{|\boldsymbol{\omega}|}$ 이며, $\mathbf{c}_i$ 는 α 의 정점들이다. 증명은 상기 수학적 4와 유사하므로 생략한다.

이제, 본 발명의 일 실시예에서, 보수적 전진의 반복 동안에 프론트 노드들의 깊이를 조절함으로써 성능을 향상시키는 기법에 관하여 설명한다. 기본적인 아이디어는 초기 몇 번의 보수적 전진의 반복들 동안에는 가장 가까운 거리를 정확하게 계산할 필요가 없다는 것이다. 다만 보수적 전진의 마지막 반복을 향해서는 정확한 거리를 필요로 한다. 거리 근사 정도의 선택이 프론트 노드들의 깊이 또는 그들의 개수를 결정하게 된다.

본 실시예의 설명에 앞서, 실시간 충돌 검사를 수행하기 위한 코스트 함수를 분석하기로 한다. 실시간 충돌 검사의 성능을 측정하기 위한 간단한 공식을 다음 수학적식과 같이 나타낼 수 있다.

수학적식 8

$$T = 2T_{BVH} + N_\tau \times N_{BV} \times T_{CA}$$

여기서, T는 실시간 충돌 검사에 대한 전체 코스트 함수이며, T_{BVH} 는 각 모델에 대하여 BVH 를 형성하기 위한 코스트이고, N_τ 는 τ 를 찾기 위한 보수적 전진 반복의 총 회수이고, N_{BV} 는 보수적 전진이 적용되는, 즉 시간 스텝을 계산하는 경계볼륨 쌍(pair)들의 개수이고, T_{CA} 는 상기 수학적식 2에 의한 보수적 전진 수식을 계산하기 위한 코스트이다. 일반적으로, T_{BVH} 와 T_{CA} 는 런타임 동안에는 상수이므로, 성능을 향상시키기 위해서는 N_τ 와 N_{BV} 를 줄일 필요가 있다.

이미 설명한 바와 같이, 본 실시예에서는 보수적 전진 기법을 최근접 거리 질의 동안에 발견되는 프론트 노드들에 적용한다. 따라서 N_{BV} 는 이 프론트 노드들의 개수에 해당한다. N_{BV} 를 줄이기 위해서, 프론트 노드들의 깊이를 줄이거나 조절할 필요가 있는데, 본 실시예에서는 최근접 거리 질의 동안에 BVH의 트래버스를 일찍 터미네이트함으로써 그렇게 한다.

도 5는 다른 깊이 값들을 가지는 프론트 노드들을 도식화한 그림으로서, 근사적 및 정확한 거리 값을 가지고 최근접 거리 질의 동안에 재귀호출(recursion)을 일찍 터미네이트함으로써 서로 다른 프론트 노드가 얻어짐을 보여 준다. 도 5를 참조하면, 이른 터미네이션의 결과, 일반적으로 실제 최근접 거리보다 작은 단지 근사적인 거리가 도출된다. 따라서 $d(\mathcal{A}(t), \mathcal{B})$ 가 원래 나올 값보다 더 작기 때문에 상기 수학적식 2에 기초한 전진 시간 스텝 Δt_i 는 역시 덜 엄격할 것이다.

다만 몇 번의 반복에 대하여 N_{BV} 를 줄임으로써 결국 더 많은 보수적 전진의 반복을 필요로 할 수 있다. 따라서 N_τ 와 N_{BV} 를 모두 동시에 줄이는 것은 거의 불가능할 수 있다. 그러므로 코스트 함수를 최소화시키기 위해 N_τ 와 N_{BV} 의 균형을 맞출 필요가 있다. 일 실시예로서, 과도한 보수적 전진의 반복 회수를 방지하기 위하여, 각 반복에 대하여 근사적인 거리가 어떤 임계값보다 작은지 또는 현재까지의 반복 회수 N_τ 가 너무 크게 되었는지를 검사한다. 만일 그렇다면 BVH 트래버스의 이른 터미네이션을 하지 않고 보통의 최근접 거리 질의 방식으로 돌아간다.

최근접 거리 질의 동안에 이른 터미네이션을 수행하기 위한 방법으로서 일 실시예에서, 거리 질의를 위한 재귀 호출(recursive call) 동안에, 앞서 얻어진 현재 최근접 거리보다 작게 수정된 거리(이하, 수정 현재 최근접 거리)를 현재 최근접 거리처럼 가장하여 재귀 함수에게 제공하는 것이다. 이것은 그 재귀가 가장된 현재 최근접 거리 값을 향상시킬 수 없을 때 재귀적인 거리 질의가 일찍 터미네이트되도록 유발한다. 마지막으로, 모든 재귀가 끝나면, 재귀가 중단된 프론트 노드들을 수집하고 그들에 대하여 보수적 전진 작업을 수행한다. 즉, 프론트 노드들에 해당하는 경계볼륨 쌍들에 대하여 보수적 전진을 위한 시간 스텝을 계산한다.

도 6은 상기 220단계의 구체적인 일 실시예의 슈도코드(pseudocode)를 나타낸다. 도시된 슈도코드를 편의상 C^2A 라 칭하기로 한다. 도시된 슈도코드는 입력으로서, BVH의 노드들 $n_{\mathcal{A}}$ 와 $n_{\mathcal{B}}$, 현재 최근접 거리 d_{cur} , 현재

전진 시간 스텝 Δt_{cur} , 그리고 재귀의 터미네이션을 조절하기 위한 변수 값 w 를 입력으로 한다. 출력으로는 업데이트된 d_{cur} 및 Δt_{cur} 가 출력되며, 출력된 d_{cur} 및 Δt_{cur} 는 다음 반복을 위한 입력이 된다. 처음 실행에서는 BVH의 루트 노드 $BVH_{\mathcal{A}}.root$ 와 $BVH_{\mathcal{B}}.root$, 현재 최근접 거리로서 ∞ , 현재 전진 시간 스텝으로서 1, 그리고 w ($w < 1$)가 입력된다. 변수 값 w 는 사용자에게 의해 미리 지정될 수 있다.

[0048] 도 6을 참조하면, 우선 라인 1 내지 3에서, 보수적 전진의 반복 회수, 즉 220단계의 반복 회수가 소정 임계값 R_{max} 보다 크거나, 또는 앞선 반복에서 얻어진 현재 최근접 거리 d_{cur} 가 소정 임계값 $D_{threshold}$ 보다 작은지 검사하고, 그렇다면 $w=1$ 로 리셋한다. 보수적 전진의 반복이 진행될수록 그 반복 회수는 늘어나고 d_{cur} 는 작아진다. 따라서 보수적 전진의 초기 일부 반복에서는 w 가 1보다 작도록 설정되고, 그 후에 보수적 전진 반복의 마지막을 향하여는 w 가 1로 리셋된다. w 가 1보다 작도록 설정되면 후술할 알고리즘 내에서 이른 터미네이션이 일어나고, w 가 1로 설정되면 후술할 알고리즘 내에서 보통의 최근접 거리 질의가 수행된다.

[0049] 다음으로 라인 4 내지 10에 관하여 설명한다.

[0050] 노드 $n_{\mathcal{A}}$ 와 $n_{\mathcal{B}}$ 가 만일 리프 노드(leaf node)라면 $n_{\mathcal{A}}$ 와 $n_{\mathcal{B}}$ 사이의 거리 d 를 계산한다. 이때 PQP(Proximity Query Package) 라이브러리를 사용할 수 있다. 그리고 d 가 d_{cur} 보다 작다면 d_{cur} 를 d 로 업데이트한다. 그리고 수식식 2를 사용하여 d , $n_{\mathcal{A}}$ 와 $n_{\mathcal{B}}$ 를 가지고 Δt 를 구한다(라인 7) 다음, Δt 가 Δt_{cur} 보다 작다면 Δt_{cur} 를 Δt 로 업데이트한다. 라인 7이 수행될 때 $n_{\mathcal{A}}$ 와 $n_{\mathcal{B}}$ 는 프론트 노드가 된다. 마지막으로 d_{cur} 와 Δt_{cur} 값을 리턴한다.

[0051] 다음으로 라인 11 내지 19에 관하여 설명한다.

[0052] 만일 $n_{\mathcal{A}}$ 가 리프 노드가 아니라면(라인 11), $n_{\mathcal{A}}$ 의 왼쪽 자식 노드를 A로 하고, $n_{\mathcal{A}}$ 의 오른쪽 자식 노드를 B로 하고, $n_{\mathcal{B}}$ 를 C와 D로 한 다음, A와 C의 거리 d_1 (즉, $n_{\mathcal{A}}$ 의 왼쪽 자식 노드에 해당하는 경계볼륨과 $n_{\mathcal{B}}$ 간의 거리)과 B와 D와의 거리 d_2 (즉, $n_{\mathcal{A}}$ 의 오른쪽 자식 노드에 해당하는 경계볼륨과 $n_{\mathcal{B}}$ 간의 거리)를 계산한다. 만일 $n_{\mathcal{A}}$ 가 리프 노드라면(라인 15), $n_{\mathcal{B}}$ 가 리프 노드가 아닐 것이므로, $n_{\mathcal{A}}$ 를 A와 B로 하고, $n_{\mathcal{B}}$ 의 왼쪽 자식 노드를 C로 하고 $n_{\mathcal{B}}$ 의 오른쪽 자식 노드를 D로 한 다음, A와 C의 거리 d_1 과 B와 D와의 거리 d_2 를 계산한다.

[0053] 다음으로 라인 20 내지 46에 관하여 설명한다.

[0054] 만일 d_2 가 d_1 보다 작다면, 후술할 라인 21 내지 32가 수행되고, 그렇지 않다면, 즉 d_2 가 d_1 보다 작지 않다면 후술할 라인 34 내지 45가 수행된다. 즉, 도 5를 참조하면, 먼저 d_2 와 d_1 중 작은 값을 가지고 wd_{cur} 와 비교하여 그 결과에 따라 C^2A 의 재귀호출을 하던지 보수적 전진을 위한 시간 스텝을 계산한 후, 다음으로 d_2 와 d_1 중 큰 값을 가지고 마찬가지로 wd_{cur} 와 비교하여 그 결과에 따라 C^2A 의 재귀호출을 하던지 보수적 전진을 위한 시간 스텝을 계산한다. 여기서, wd_{cur} 가 ($w < 1$ 일 때) 상기된 수정 현재 최근접 거리가 된다.

[0055] d_2 가 d_1 보다 작을 때 수행되는 라인 21 내지 32에 관하여 설명한다.

[0056] d_2 가 wd_{cur} 보다 작은지 검사하고(라인 21), 그렇다면 $C^2A(B, D, d_{cur}, \Delta t_{cur}, w)$ 를 리턴하는 재귀를 수행한

다. 즉, B와 D, 앞서 얻어진 현재 최근접 거리 d_{cur} , 앞서 얻어진 현재 시간 스텝 Δt_{cur} , 그리고 w 를 입력으로 하여 C^2A 를 수행하고 업데이트된 출력값 d_{cur} 및 Δt_{cur} 을 리턴한다(라인 22). d_2 가 $w d_{cur}$ 보다 작은지 검사했는데, 그렇지 않다면 d_2 와 B, D를 가지고 시간 스텝 Δt 를 계산(라인 24)한 후, Δt 가 앞서 얻어진 현재 시간 스텝 Δt_{cur} 보다 작다면 Δt_{cur} 를 Δt 로 업데이트한다(라인 25). 여기서, d_2 와 $w d_{cur}$ 를 비교하게 되는데, 만일 w 가 1보다 작은 값이라면(즉, 이른 터미네이션) w 가 1인 경우보다 라인 24가 일찍 수행되므로 재귀호출의 이른 터미네이션, 즉 계층구조 트래버스의 이른 터미네이션이 일어나게 된다. 라인 24가 수행될 때 B와 D가 프론트 노드가 된다. w 가 1이라면, 이른 터미네이션을 하지 않고 보통의 최근접 거리 질의가 된다.

[0057] 이제 d_1 이 $w d_{cur}$ 보다 작은지 검사하고(라인 27), 그렇다면 $C^2A(A, C, d_{cur}, \Delta t_{cur}, w)$ 를 리턴하는 재귀를 수행하고(라인 28), 그렇지 않다면 d_1 과 A, C를 가지고 시간 스텝 Δt 를 계산(라인 30)한 후, Δt 가 앞서 얻어진 현재 시간 스텝 Δt_{cur} 보다 작다면 Δt_{cur} 를 Δt 로 업데이트한다(라인 31). 라인 27 내지 32의 취지는 라인 21 내지 26의 취지와 동일하다.

[0058] d_2 가 d_1 보다 작지 않을 때 수행되는 라인 34 내지 45는 d_1 과 d_2 의 순서만 바뀌었을 뿐 상기된 라인 21 내지 32에 관한 설명과 동일하므로 그 설명은 생략하기로 한다.

[0059] 또한, 일 실시예에서 충돌 검사 성능을 보다 향상시키기 위하여 다음과 같은 간단한 기법을 더 채용할 수 있다. 보수적 전진의 i 번째 반복에서, 경계볼륨 쌍에 해당하는 어떤 노드 n 을 방문하였을 때, 그 경계볼륨 쌍의 시간 스텝 Δt_i^n 가 어떤 상계(upper bound) 값 Λ_i 보다 크다면 해당 노드와 그 자식 노드들을 전지(prune)하는 것

이다. 이때 초기적으로 $\Lambda_1 = 1$ 로 설정하고, k 번째 반복에서 $\Lambda_k = 1 - \sum_{i=1}^{k-1} \Delta t_i$ 로 설정할 수 있다. 왜냐하면 $\sum \Delta t_i > 1$ 이라는 것은 $[0,1]$ 동안에 그 노드 n 에 대하여 충돌이 없음을 의미하고, k 번째 반복에서 구해진

시간 스텝이 1에서 $(k-1)$ 번째 반복까지 더해진 시간 스텝 $\sum_{i=1}^{k-1} \Delta t_i$ 을 뺀 값보다 크다면 $t=1$ 이 되더라도 충돌이 없을 것이기 때문이다. 따라서 그러한 노드와 그 자식 노드들은 더 이상 검사할 필요가 없다.

[0060] 도 7은 본 발명의 일 실시예에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 시스템의 블록도이다. 본 실시예에 따른 충돌 검사 시스템은 경계볼륨 계층구조 형성부(710), 시간 스텝 계산부(720), 모델 전진부(730), 충돌 시간 계산부(740)를 포함하여 이루어진다. 도시하지는 않았지만 충돌 검사 시스템은, 충돌을 검사하고자 하는 입력되는 두 모델과, 후술할 모델 전진부(730)에 의해 전진된 모델, 경계볼륨 계층구조, 시간 스텝 계산부(720)에 의해 구해진 시간 스텝 등이 저장되는 메모리를 더 포함할 수 있다.

[0061] 경계볼륨 계층구조 형성부(710)는 충돌을 검사하고자 하는 두 모델을 입력받고, 그에 대하여 경계볼륨 계층구조를 형성한다. 경계볼륨 계층구조 형성부(710)의 동작은 도 2의 210단계에 관련된 설명과 동일하므로 구체적인 설명은 생략한다.

[0062] 시간 스텝 계산부(720)는 두 모델에 대하여 최근접 거리 질의를 수행하면서 경계볼륨 계층구조를 트래버스하되, 그로 인해 얻어지는 일부의 경계볼륨 쌍들에 대하여 보수적 전진 기법을 위한 시간 스텝을 계산하고 그들 중 최소의 시간 스텝을 구한다. 최근접 거리 질의로 얻어지는 두 모델 간의 현재 최근접 거리 d_{cur} 및 구해진 현재 시간 스텝 Δt_{cur} 가 시간 스텝 계산부(720)의 출력이 되며, Δt_{cur} 는 후술할 모델 전진부(730)와 충돌 시간 계산부(740)로 입력되고, d_{cur} 및 Δt_{cur} 는 다음 반복을 위하여 시간 스텝 계산부(720)로 다시 입력된다. 시간 스텝 계산부(720)의 동작은 도 2의 220단계에 관련된 설명과 동일하므로 구체적인 설명은 생략한다.

[0063] 모델 전진부(730)는 시간 스텝 계산부(720)에 의해 구해진 시간 스텝 Δt_{cur} 만큼 두 모델 중 어느 한 모델(또는

두 모델을 각각)을 전진시킨다. 전진된 모델은 시간 스텝 계산부(720)로 입력되고, 시간 스텝 계산부(720)는 전진된 모델을 가지고 반복적으로 상기된 동작을 수행하여 d_{cur} 및 Δt_{cur} 를 구한다.

[0064] 시간 스텝 계산부(720)와 모델 전진부(730)의 반복적인 동작은 두 모델 간의 최근접 거리가 사용자 지정된 소정 임계값보다 작을 때까지 수행된다. 두 모델 간의 최근접 거리가 상기 소정 임계값보다 작은 것으로 검사되면 충돌 시간 계산부(740)는 그때까지 시간 스텝 계산부(720)로부터 얻은 시간 스텝들을 합계하여 두 모델의 충돌 시간(TOC)을 구한다.

[0065] 상술한 실시예에 의하면, SSV에 대한 모션 바운드를 효율적으로 계산할 수 있고, 프론트 노드들의 깊이 또는 개수를 조절함으로써 연속적 충돌 검사의 성능을 가속화할 수 있다. 따라서 폴리곤 수프 모델들의 연속적 충돌 검사 속도를 획기적으로 개선할 수 있으며, 이를 실시간 속도로 수행할 수 있고, 모델이 테셀레이트되어(tessellated) 있거나 하면, 모델의 기하(geometry)나 위상(topology)에 제약이 없는 연속적 충돌 검사를 가능하게 하며, 종전의 다면체 모델(polyhedral models)에 특화된 연속적 충돌 검사 방법에 비해서도 더 빠른 성능을 가질 수 있다.

[0066] 한편, 상술한 본 발명의 실시예들은 컴퓨터에서 실행될 수 있는 프로그램으로 작성가능하고, 컴퓨터로 읽을 수 있는 기록매체를 이용하여 상기 프로그램을 동작시키는 범용 디지털 컴퓨터에서 구현될 수 있다. 상기 컴퓨터로 읽을 수 있는 기록매체는 마그네틱 저장매체(예를 들면, 롬, 플로피 디스크, 하드 디스크 등), 광학적 판독 매체(예를 들면, 시디롬, 디브이디 등) 및 캐리어 웨이브(예를 들면, 인터넷을 통한 전송)와 같은 저장매체를 포함한다.

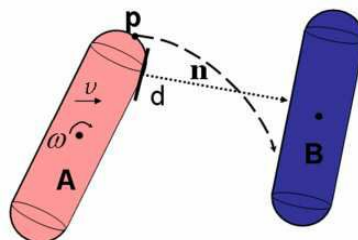
[0067] 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

도면의 간단한 설명

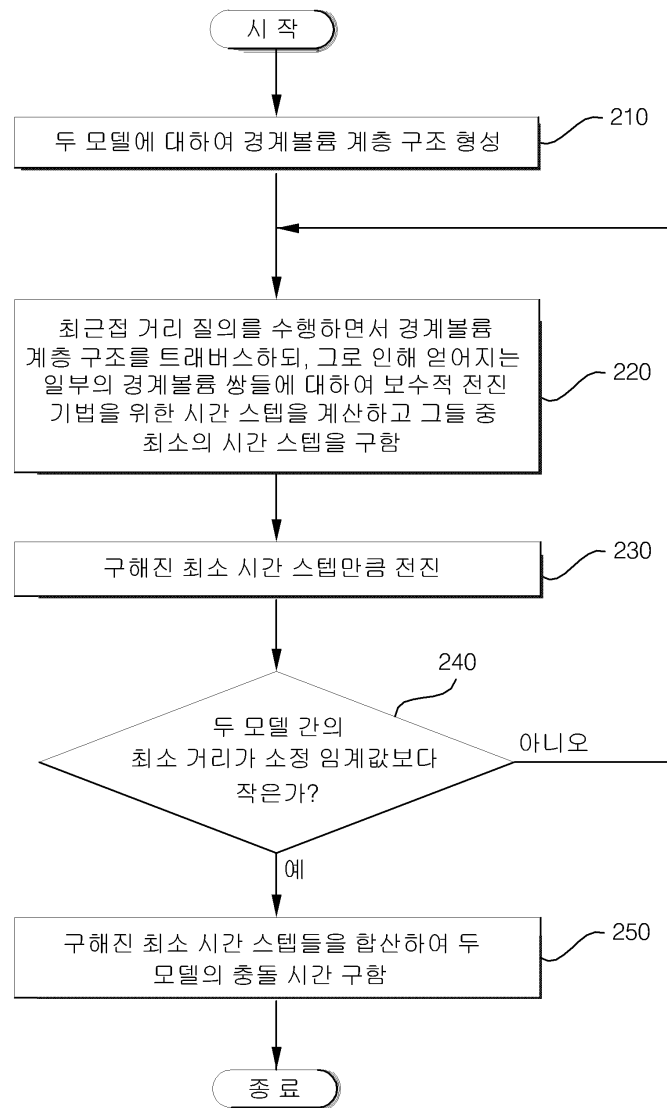
- [0068] 도 1은 보수적 전진 기법을 설명하기 위한 참고도이다.
- [0069] 도 2는 본 발명의 일 실시예에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 방법의 흐름도이다.
- [0070] 도 3은 두 모델에 대하여 형성된 경계볼륨 계층구조의 간단한 예를 나타낸다.
- [0071] 도 4는 PSS, LSS, 그리고 RSS를 나타낸 도면이다.
- [0072] 도 5는 다른 깊이 값들을 가지는 프론트 노드들을 도식화한 그림이다.
- [0073] 도 6은 220단계의 구체적인 일 실시예의 수도코드(pseudocode)를 나타낸다.
- [0074] 도 7은 본 발명의 일 실시예에 따른 보수적 전진 기법을 이용한 실시간 충돌 검사 시스템의 블록도이다.

도면

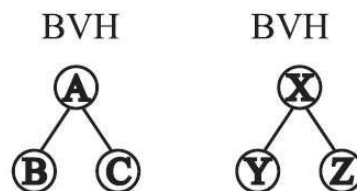
도면1



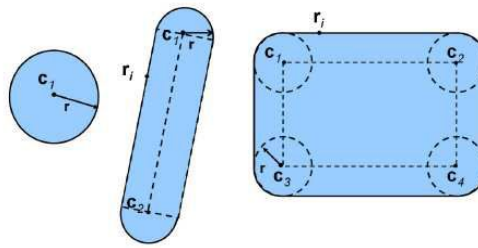
도면2



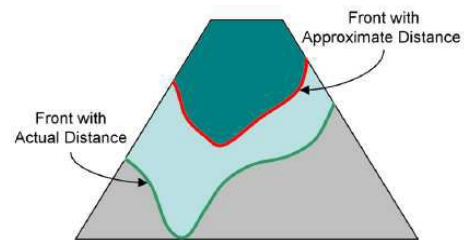
도면3



도면4



도면5



도면6

Algorithm 1 C²A: Controlled Conservative Advancement

Input: BVH nodes $n_{\mathcal{A}}, n_{\mathcal{B}}$, current closest distance d_{cur} , current advancement step Δt_{cur} , CA controlling variable w

Output: Updated d_{cur} and Δt_{cur}

{Initially, C²A($BVH_{\mathcal{A}}.root, BVH_{\mathcal{B}}.root, \infty, 1, w$) is called and $w < 1$. }

```

1: if # of CA iterations >  $R_{max}$  or  $d_{cur} < D_{threshold}$  then
2:    $w = 1$ ;
3: end if
4: if  $n_{\mathcal{A}}$  and  $n_{\mathcal{B}}$  are leaf nodes then
5:    $d = \text{Distance}(n_{\mathcal{A}}, n_{\mathcal{B}})$ ; {Using the PQP library}
6:   if( $d < d_{cur}$ )  $d_{cur} = d$ ;
7:    $\Delta t = \text{CalculateCAStep}(d, n_{\mathcal{A}}, n_{\mathcal{B}})$ ; {Using Eq. 2}
8:   if( $\Delta t < \Delta t_{cur}$ )  $\Delta t_{cur} = \Delta t$ ;
9:   return  $d_{cur}, \Delta t_{cur}$ 
10: end if
11: if  $n_{\mathcal{A}}$  is not a leaf node then
12:    $A = n_{\mathcal{A}}.leftchild$ ;  $B = n_{\mathcal{A}}.rightchild$ ;  $C = D = n_{\mathcal{B}}$ ;
13:    $d_1 = \text{Distance}(A, C)$ ;
14:    $d_2 = \text{Distance}(B, D)$ ;
15: else
16:    $A = B = n_{\mathcal{B}}$ ;  $C = n_{\mathcal{A}}.leftchild$ ;  $D = n_{\mathcal{A}}.rightchild$ ;
17:    $d_1 = \text{Distance}(A, C)$ ;
18:    $d_2 = \text{Distance}(B, D)$ ;
19: end if
20: if  $d_2 < d_1$  then
21:   if  $d_2 < wd_{cur}$  then
22:     return C2A( $B, D, d_{cur}, \Delta t_{cur}, w$ );
23:   else
24:      $\Delta t = \text{CalculateCAStep}(d_2, B, D)$ ;
25:     if( $\Delta t < \Delta t_{cur}$ )  $\Delta t_{cur} = \Delta t$ ;
26:   end if
27:   if  $d_1 < wd_{cur}$  then
28:     return C2A( $A, C, d_{cur}, \Delta t_{cur}, w$ );
29:   else
30:      $\Delta t = \text{CalculateCAStep}(d_1, A, C)$ ;
31:     if( $\Delta t < \Delta t_{cur}$ )  $\Delta t_{cur} = \Delta t$ ;
32:   end if
33: else
34:   if  $d_1 < wd_{cur}$  then
35:     return C2A( $A, C, d_{cur}, \Delta t_{cur}, w$ );
36:   else
37:      $\Delta t = \text{CalculateCAStep}(d_1, A, C)$ ;
38:     if( $\Delta t < \Delta t_{cur}$ )  $\Delta t_{cur} = \Delta t$ ;
39:   end if
40:   if  $d_2 < wd_{cur}$  then
41:     return C2A( $B, D, d_{cur}, \Delta t_{cur}, w$ );
42:   else
43:      $\Delta t = \text{CalculateCAStep}(d_2, B, D)$ ;
44:     if( $\Delta t < \Delta t_{cur}$ )  $\Delta t_{cur} = \Delta t$ ;
45:   end if
46: end if

```

도면7

