

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2014-0092536 (43) 공개일자 2014년07월24일
(51) 국제특허분류(Int. Cl.) G06T 17/00 (2006.01)	(71) 출원인 계명대학교 산학협력단 대구광역시 달서구 달구벌대로 1095 (신당동)	
(21) 출원번호 10-2013-0004653	(72) 발명자 성만규 대구광역시 달서구 달구벌대로 1095 계명대학교성서캠퍼스	
(22) 출원일자 2013년01월16일 심사청구일자 2013년01월16일	이준재 대구광역시 달서구 달구벌대로 1095 계명대학교성서캠퍼스	
	(74) 대리인 김건우	

전체 청구항 수 : 총 19 항

(54) 발명의 명칭 **깊이맵 센서를 이용한 3차원 캐릭터 가상공간 내비게이션 동작 합성 및 제어 방법**

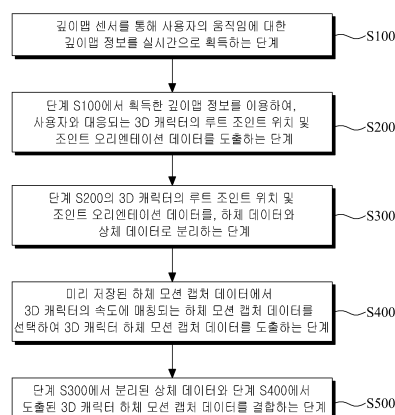
(57) 요약

본 발명은 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치에 관한 것으로서, 보다 구체적으로는 (1) 깊이맵 센서를 통해 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 단계; (2) 상기 단계 (1)에서 획득한 깊이맵 정보를 이용하여, 상기 사용자와 대응되는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 도출하는 단계; (3) 상기 단계 (2)의 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를, 하체 데이터와 상체 데이터로 분리하는 단계; (4) 미리 저장된 하체 모션 캡처 데이터에서 상기 3D 캐릭터의 속도에 매칭 되는 하체 모션 캡처 데이터를 선택하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계; 및 (5) 상기 단계 (3)에서 분리된 상체 데이터와 상기 단계 (4)에서 도출된 3D 캐릭터 하체 모션 캡처 데이터를 결합하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

본 발명에서 제안하고 있는 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치에 따르면, 깊이맵 센서를 통해 깊이맵 정보를 실시간으로 획득하여 상체 움직임을 반영하되, 하체 움직임은 미리 저장된 하체 모션 캡처 데이터에서 3D 캐릭터 속도에 매칭 되는 하체 모션 캡처 데이터를 선택 도출하여 합성하는 구성을 채택함으로써, 깊이맵 센서를 이용하여 사용자의 움직임을 충실히 반영함과 동시에 장소의 한계를 극복할 수 있다.

또한, 본 발명에 따르면, 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출하여 제자리걸음 동작을 인식함과 동시에, 속도를 자동으로 검색하고 미리 입력되어 있는 다양한 속도의 걷기 모션 중 적절한 모션을 선택하여 블렌딩하고, 이를 상체 데이터와 결합함으로써, 발 움직임을 인식하여 간단하게 사용자의 내비게이션 의도를 파악하고, 제자리걸음을 자연스러운 실제 걸음으로 바꾸어 가상공간을 내비게이션 할 수 있도록 한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 2012-D-0037-010107

부처명 교육과학기술부

연구사업명 LINC 산학공동기술개발과제

연구과제명 3차원 행동인식 노인성 재활 S/W 개발

기 여 율 1/1

주관기관 계명대학교

연구기간 2012.07.01 ~ 2013.01.31

특허청구의 범위

청구항 1

3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법으로서,

- (1) 깊이맵 센서를 통해 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 단계;
- (2) 상기 단계 (1)에서 획득한 깊이맵 정보를 이용하여, 상기 사용자와 대응되는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 도출하는 단계;
- (3) 상기 단계 (2)의 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를, 하체 데이터와 상체 데이터로 분리하는 단계;
- (4) 미리 저장된 하체 모션 캡처 데이터에서 상기 3D 캐릭터의 속도에 매칭 되는 하체 모션 캡처 데이터를 선택하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계; 및
- (5) 상기 단계 (3)에서 분리된 상체 데이터와 상기 단계 (4)에서 도출된 3D 캐릭터 하체 모션 캡처 데이터를 결합하는 단계를 포함하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 2

제1항에 있어서, 상기 단계 (2)에서는,

상기 깊이맵 정보로부터 3D 캐릭터를 연동하기 위하여 계층적 캐릭터 스켈레톤 포맷을 이용하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 3

제2항에 있어서, 상기 계층적 캐릭터 스켈레톤은,

루트 조인트를 포함하여 15개 조인트로 구성되는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 4

제1항에 있어서, 상기 단계 (2)의 3D 캐릭터의 루트 조인트 위치 데이터는,

상기 깊이맵 정보로부터 얻은 상기 사용자의 루트 조인트 위치 정보를, 상기 사용자와 상기 3D 캐릭터의 크기 값 차이를 계산하여 스케일링한 값인 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 5

제4항에 있어서, 상기 사용자와 상기 3D 캐릭터의 크기 값 차이의 계산은, 하기 수학적식 1 및 2에 의해 수행하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

[수학적식 1]

$$r = \frac{(P^0 - P^i)^2 + (P^0 - P^r)^2}{(p^0 - p^i)^2 + (p^0 - p^r)^2}$$

[수학식 2]

$$p_t^0 = p_0^0 + r(P_0^0 - P_t^0)$$

(p_t^0 = t프레임에서의 캐릭터 위치, p_0^0 = 캐릭터의 초기 위치, p^l = 캐릭터의 왼발 위치, p^r = 캐릭터의 오른발 위치, P_t^0 = t프레임에 들어오는 깊이맵 상의 사용자 루트 조인트 위치, P_0^0 = 깊이맵 상의 사용자의 초기 위치, P^l = 깊이맵 상의 사용자 왼발 위치, P^r = 깊이맵 상의 사용자 오른발 위치, r = 스케일 상수)

청구항 6

제1항에 있어서, 상기 단계 (3) 이전에,

(2-1) 상기 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 상기 3D 캐릭터의 발이 땅에 닿는 프레임(이하, ‘제약 프레임’이라 한다.)을 검출하는 단계; 및

(2-2) 상기 단계 (2-1)에서 검출한 왼발의 제약 프레임이 일어나는 시작 시각과 오른발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되면 제자리걸음 동작으로 인식하는 단계를 더 포함하고,

상기 단계 (3) 내지 단계 (5)는 상기 제자리걸음 동작이 인식되는 경우에 수행되는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 7

제6항에 있어서, 상기 단계 (2-1)는,

상기 왼발의 속도 값 및 상기 오른발의 속도 값을 하기 수학식 3 및 4에 따라 계산하고,

상기 해당 포즈에서의 왼발의 위치 및 속도 값, 또는 상기 오른발의 위치 및 속도 값이 미리 정해진 임계값 p_{max} 및 v_{max} 보다 작은 경우, 상기 해당 프레임을 제약 프레임으로 인식하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

[수학식 3]

$$v^l = u\left(\frac{\|p_t^l - p_{t-1}^l\|}{\Delta t}, n\right)$$

$$u(v, n) = rdxl(v, n)$$

[수학식 4]

$$v^r = u\left(\frac{\|p_t^r - p_{t-1}^r\|}{\Delta t}, n\right)$$

$$u(v, n) = rdxl(v, n)$$

(v^l = 왼발의 속도 값, v^r = 오른발의 속도 값, $\|p_t^l - p_{t-1}^l\|$ = t와 t-1 프레임 사이의 발이 움직인 거리, Δt = 프레임 사이의 시간 차이, $rdxl(v, n)$ 함수 = n번째 숫자에서 반올림되는 값을 구해주는 함수)

청구항 8

제7항에 있어서, 상기 단계 (2-1)에서, 상기 임계값 p_{max} 및 v_{max} 는,

상기 3D 캐릭터의 왼발 및 오른발에 대한 위치 및 속도 값이 모두 초깃값으로 주어진 임계값보다 큰 값을 가지

는 경우, 자동으로 상기 임계값이 미리 정해진 값만큼 수정되는 적응적 임계값 설정 방법을 적용하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 9

제8항에 있어서, 상기 단계 (2-1)에서, 상기 $p_{\max}$ 및 $v_{\max}$ 는,

하기 수학적 식 5에 따라 설정되는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

[수학적 식 5]

$$\begin{aligned} p_{\max} &= \min(p^r, p^l) + \delta \\ v_{\max} &= \min(v^r, v^l) + \delta \end{aligned}$$

($\delta = 0.1$)

청구항 10

제6항에 있어서, 상기 단계 (2-2)에서, 상기 미리 정해진 시간은,

1.5초 내지 2.5초 내에서 설정되는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 11

제6항에 있어서, 상기 단계 (2-2)는,

상기 3D 캐릭터의 왼발 및 오른발의 위치가 미리 정해진 높이 이상 땅으로부터 떨어져 있으면 뛰는 동작으로 인식하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 12

제1항에 있어서, 상기 단계 (4)에서 미리 저장된 모션 캡처 데이터는,

상기 3D 캐릭터와 동일한 조인트 개수를 가지고, 복수의 걷기 속도를 가지는 것으로서, 오른발로 시작하는 걷기 모션과 왼발로 시작하는 걷기 모션이 구분되어 저장되는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 13

제1항에 있어서, 상기 단계 (3) 내지 단계 (5)는,

한걸음 단위(walking-cycle-motion)로 각각 수행되는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 14

제1항에 있어서, 상기 단계 (4)는,

(4-1) 미리 저장된 하체 모션 캡처 데이터를 한걸음 단위로 쪼갠 후 속도 및 방향 값으로 매개화하는 단계;

(4-2) 상기 매개화된 하체 모션 캡처 데이터에 대하여 해당 3D 캐릭터의 속도를 반영하여 방사형 기저함수(RBF, Radial Basis Function) 기반으로 가중치를 구하는 단계; 및

(4-3) 상기 매개화된 하체 모션 캡처 데이터 중에서 선택된 데이터를 상기 가중치를 적용하여 모션 블렌딩하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계를 더 포함하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 15

제14항에 있어서,

상기 3D 캐릭터의 속도는 하기 수학식 6을 통해 구하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

[수학식 6]

$$v = D / \Delta t$$

(v = 제자리걸음 속도, Δt = 제약 프레임 간의 시간적 차이, D = 제약 프레임 간의 시간 동안 발끝의 움직이는 거리 값)

청구항 16

제14항에 있어서,

상기 단계 (4-2)의 가중치는 하기 수학식 7을 통해 구하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

[수학식 7]

$$w_m(v) = a_m A(v) + \sum_{j=1}^n r_{mj} R_j(v)$$

($w_m(v)$ = 모션 m의 속도 v에 대한 가중치, $A(v)$ = 선형 기저 함수, a_m = 선형 기저 함수의 계수, $R_j(v)$ = 방사형 기저 함수, r_{mj} = 방사형 기저 함수의 계수, n = 모션의 개수)

청구항 17

제14항에 있어서, 상기 단계 (4-3)에서 모션 블렌딩하기 이전에,

상기 선택된 데이터들을 동일한 시간 축에 정렬하는 시간 정규화 단계를 더 포함하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법.

청구항 18

3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어장치로서,

사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 깊이맵 센서;

3D 캐릭터의 실시간 포즈별 조인트 위치 및 오리엔테이션 데이터를 도출하는 3D 캐릭터 조인트 데이터 출력 모듈;

상기 3D 캐릭터의 하체 모션 캡처 데이터를 저장하는 하체 모션 캡처 데이터 저장 모듈; 및

상기 3D 캐릭터 조인트 데이터 출력 모듈을 통해 도출된 데이터 중 상체 데이터와 상기 하체 모션 캡처 데이터

저장 모듈에 저장된 하체 모션 캡처 데이터 중에서 선택된 데이터를 합성하는 모션 스플라이싱 모듈을 포함하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어장치.

청구항 19

제18항에 있어서, 상기 모션 스플라이싱 모듈은,

3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 상기 제약 프레임을 검출하고, 검출한 왼발의 제약 프레임이 일어나는 시작 시각과 오른발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되어 제자리걸음 동작으로 인식되면, 상기 모션 스플라이싱을 수행하는 것을 특징으로 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어장치.

명세서

기술분야

[0001] 본 발명은 3D 캐릭터 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치에 관한 것으로서, 보다 구체적으로는 깊이맵 센서를 이용한 3D 캐릭터 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치에 관한 것이다.

배경기술

[0002] 가상현실 시스템에서는 사용자와 실제/가상 작업 공간이 하드웨어로 상호 연결되고, 사용자의 동작의 변화를 감지하여 그에 대응하는 가상 캐릭터가 동작하도록 하는 가상공간 내비게이션이 수행된다. 전통적인 방법으로 사용되는 가상공간 내비게이션은 주로 마우스나 키보드를 통해 제어되는데, 입력제어를 사용자로부터 받으면, 미리 키 프레임(key frame)방식을 통해 세밀하게 만들어진 모션 클립들을 자연스럽게 합성(blending)하여 원하는 어느 가상공간에 자신의 캐릭터를 이동시킬 수 있다. 전통적인 이 방법은 구현이 간단한 반면에, 아바타와 사용자의 일체감 측면에서는 단점이 있다.

[0003] 최근, 마이크로소프트사는 2010년 11월 자사의 XBOX용 인터페이스 장치인 키넥트를 발표함으로써 새로운 내추럴 사용자 인터페이스를 시도하였다. 이 장치는 8개월 만에 만개나 팔림으로써 기네스북에 오를 만큼 대성공을 거두었다. 키넥트 센서는 실공간상의 오브젝트와 센서와의 거리 값을 실시간으로 측정하는 깊이맵 센서로서, 구조 광(structured light) 방식의 IR(Infrared) 빛을 공간상에 투사한 후, 오브젝트에 반사되어 돌아오는 패턴을 분석하여, 얼마나 원래의 패턴으로부터 이동되어 있는지를 계산함으로써 거리 값을 측정한다. 이 센서를 이용하여 취득한 깊이맵 정보를 토대로, 오픈소스 프로젝트인 OpenNI 미들웨어와 키넥트의 원 개발자인 프라임센서사의 드라이버를 이용하면 이 3차원 깊이맵으로부터 15개 조인트의 전역위치와 전역오리엔테이션 값을 구할 수 있으며, 이 데이터를 이용해 3차원 캐릭터를 연동시킬 수 있다.

[0004] 키넥트와 같은 깊이맵 센서를 이용하면, 사용자의 동작을 모션캡처한 후 충실하게 아바타에게 표현함으로써 사용자의 동작을 충실히 재현할 수 있는 장점이 있지만, 사용자는 항상 감지영역 안에 존재하여야 하므로, 가상공간 내비게이션을 통해 캐릭터를 이동하는 데 한계가 생긴다는 문제가 있다. 즉, 사용자는 항상 센서를 정면으로 마주본 상태에서 일정거리 이상 떨어져서 동작을 취하여야 하며, 되도록 감지영역을 벗어나서는 안 되는 단점이 있다. 이 단점은 컴퓨터 게임에서 가장 대표적으로 요구되는 기능인 가상공간 내비게이션을 어렵게 만드는 주요한 원인이 되고 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 기존에 제안된 방법들의 상기와 같은 문제점들을 해결하기 위해 제안된 것으로서, 깊이맵 센서를 통해 깊이맵 정보를 실시간으로 획득하여 상체 움직임을 반영하되, 하체 움직임은 미리 저장된 하체 모션 캡처 데이터에서 3D 캐릭터 속도에 매칭 되는 하체 모션 캡처 데이터를 선택 도출하여 합성하는 구성을 채택함으로써, 깊이맵 센서를 이용하여 사용자의 움직임을 충실히 반영함과 동시에 장소의 한계를 극복할 수 있는, 깊이맵 센

서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치를 제공하는 것을 그 목적으로 한다.

[0006] 또한, 본 발명은, 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출하여 제자리걸음 동작을 인식함과 동시에, 속도를 자동으로 검색하고 미리 입력되어 있는 다양한 속도의 걷기 모션 중 적절한 모션을 선택하여 블렌딩하고, 이를 상체 데이터와 결합함으로써, 발 움직임을 인식하여 간단하게 사용자의 내비게이션 의도를 파악하고, 제자리걸음을 자연스러운 실제 걸음으로 바꾸어 가상공간을 내비게이션 할 수 있도록 하는, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치를 제공하는 것을 또 다른 목적으로 한다.

과제의 해결 수단

[0007] 상기한 목적을 달성하기 위한 본 발명의 특징에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법은,

[0008] (1) 깊이맵 센서를 통해 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 단계;

[0009] (2) 상기 단계 (1)에서 획득한 깊이맵 정보를 이용하여, 상기 사용자와 대응되는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 도출하는 단계;

[0010] (3) 상기 단계 (2)의 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를, 하체 데이터와 상체 데이터로 분리하는 단계;

[0011] (4) 미리 저장된 하체 모션 캡처 데이터에서 상기 3D 캐릭터의 속도에 매칭 되는 하체 모션 캡처 데이터를 선택하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계; 및

[0012] (5) 상기 단계 (3)에서 분리된 상체 데이터와 상기 단계 (4)에서 도출된 3D 캐릭터 하체 모션 캡처 데이터를 결합하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

[0013] 바람직하게는, 상기 단계 (2)에서는,

[0014] 상기 깊이맵 정보로부터 3D 캐릭터를 연동하기 위하여 계층적 캐릭터 스켈레톤 포맷을 이용할 수 있다.

[0015] 더욱 바람직하게는, 상기 계층적 캐릭터 스켈레톤은,

[0016] 루트 조인트를 포함하여 15개 조인트로 구성될 수 있다.

[0017] 바람직하게는, 상기 단계 (2)의 3D 캐릭터의 루트 조인트 위치 데이터는,

[0018] 상기 깊이맵 정보로부터 얻은 상기 사용자의 루트 조인트 위치 정보를, 상기 사용자와 상기 3D 캐릭터의 크기 값 차이를 계산하여 스케일링한 값일 수 있다.

[0019] 더욱 바람직하게는, 상기 사용자와 상기 3D 캐릭터의 크기 값 차이의 계산은, 하기 수학식 1 및 2에 의해 수행할 수 있다.

[0020] [수학식 1]

$$r = \frac{(P^0 - P^i)^2 + (P^0 - P^r)^2}{(p^0 - p^i)^2 + (p^0 - p^r)^2}$$

[0021]

[0022] [수학식 2]

$$p_t^0 = p_0^0 + r(P_0^0 - P_t^0)$$

[0023]

[0024] (p_t^0 = t프레임에서의 캐릭터 위치, p_0^0 = 캐릭터의 초기 위치, p^l = 캐릭터의 왼발 위치, p^r = 캐릭터의 오른발 위치, P_t^0 = t프레임에 들어오는 깊이맵 상의 사용자 루트 조인트 위치, P_0^0 = 깊이맵 상의 사용자의 초기 위치, P^l = 깊이맵 상의 사용자 왼발 위치, P^r = 깊이맵 상의 사용자 오른발 위치, r = 스케일 상수)

[0025] 바람직하게는, 상기 단계 (3) 이전에,

[0026] (2-1) 상기 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 상기 3D 캐릭터의 발이 땅에 닿는 프레임(이하, '제약 프레임'이라 한다.)을 검출하는 단계; 및

[0027] (2-2) 상기 단계 (2-1)에서 검출한 왼발의 제약 프레임이 일어나는 시작 시각과 오른발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되면 제자리걸음 동작으로 인식하는 단계를 더 포함하고,

[0028] 상기 단계 (3) 내지 단계 (5)는 상기 제자리걸음 동작이 인식되는 경우에 수행될 수 있다.

[0029] 더욱 바람직하게는, 상기 단계 (2-1)는,

[0030] 상기 왼발의 속도 값 및 상기 오른발의 속도 값은 하기 수학식 3 및 4에 따라 계산하고,

[0031] 상기 해당 포즈에서의 왼발의 위치 및 속도 값, 또는 상기 오른발의 위치 및 속도 값이 미리 정해진 임계값 p_{max} 및 v_{max} 보다 작은 경우, 상기 해당 프레임을 제약 프레임으로 인식하는 것일 수 있다.

[0032] [수학식 3]

$$v^l = u\left(\frac{\|p_t^l - p_{t-1}^l\|}{\Delta t}, n\right)$$

$$u(v, n) = rdxl(v, n)$$

[0033]

[0034] [수학식 4]

$$v^r = u\left(\frac{\|p_t^r - p_{t-1}^r\|}{\Delta t}, n\right)$$

$$u(v, n) = rdxl(v, n)$$

[0035]

[0036] (v^l = 왼발의 속도 값, v^r = 오른발의 속도 값, $\|p_t^l - p_{t-1}^l\|$ = t와 t-1 프레임 사이의 발이 움직인 거리, Δt = 프레임 사이의 시간 차이, $rdxl(v, n)$ 함수 = n번째 숫자에서 반올림 되는 값을 구해주는 함수)

[0037] 더더욱 바람직하게는, 상기 단계 (2-1)에서, 상기 임계값 p_{max} 및 v_{max} 는,

[0038] 상기 3D 캐릭터의 왼발 및 오른발에 대한 위치 및 속도 값이 모두 초깃값으로 주어진 임계값보다 큰 값을 가지는 경우, 자동으로 상기 임계값이 미리 정해진 값만큼 수정되는 적응적 임계값 설정 방법을 적용하는 것일 수 있다.

[0039] 더더욱 바람직하게는, 상기 단계 (2-1)에서, 상기 p_{max} 및 v_{max} 는,

[0040] 하기 수학식 5에 따라 설정되는 것일 수 있다.

[0041] [수학식 5]

$$p_{\max} = \min(p^r, p^l) + \delta$$

$$v_{\max} = \min(v^r, v^l) + \delta$$

[0042]

[0043] ($\delta = 0.1$)

[0044] 더욱 바람직하게는, 상기 단계 (2-2)에서, 상기 미리 정해진 시간은,

[0045] 1.5초 내지 2.5초 내에서 설정될 수 있다.

[0046] 더욱 바람직하게는, 상기 단계 (2-2)는,

[0047] 상기 3D 캐릭터의 왼발 및 오른발의 위치가 미리 정해진 높이 이상 땅으로부터 떨어져 있으면 뛰는 동작으로 인식하는 것일 수 있다.

[0048] 바람직하게는, 상기 단계 (4)에서 미리 저장된 모션 캡처 데이터는,

[0049] 상기 3D 캐릭터와 동일한 조인트 개수를 가지고, 복수의 걷기 속도를 가지는 것으로서, 오른발로 시작하는 걷기 모션과 왼발로 시작하는 걷기 모션이 구분되어 저장되는 것일 수 있다.

[0050] 바람직하게는, 상기 단계 (3) 내지 단계 (5)는,

[0051] 한걸음 단위(walking-cycle-motion)로 각각 수행되는 것일 수 있다.

[0052] 바람직하게는, 상기 단계 (4)는,

[0053] (4-1) 미리 저장된 하체 모션 캡처 데이터를 한걸음 단위로 쪼갠 후 속도 및 방향 값으로 매개화하는 단계;

[0054] (4-2) 상기 매개화된 하체 모션 캡처 데이터에 대하여 해당 3D 캐릭터의 속도를 반영하여 방사형 기저함수(RBF, Radial Basis Function) 기반으로 가중치를 구하는 단계; 및

[0055] (4-3) 상기 매개화된 하체 모션 캡처 데이터 중에서 선택된 데이터를 상기 가중치를 적용하여 모션 블렌딩하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계를 더 포함할 수 있다.

[0056] 더욱 바람직하게는,

[0057] 상기 3D 캐릭터의 속도는 하기 수학식 6을 통해 구할 수 있다.

[0058] [수학식 6]

[0059]
$$v = D / \Delta t$$

[0060] (v = 제자리걸음 속도, Δt = 제약 프레임 간의 시간적 차이, D = 제약 프레임 간의 시간 동안 발끝의 움직이는 거리 값)

[0061] 더욱 바람직하게는,

[0062] 상기 단계 (4-2)의 가중치는 하기 수학식 7을 통해 구할 수 있다.

[0063] [수학식 7]

$$w_m(v) = a_m A(v) + \sum_{j=1}^n r_{mj} R_j(v)$$

[0064]

[0065] ($w_m(v)$ = 모션 m의 속도 v에 대한 가중치, $A(v)$ = 선형 기저 함수, a_m = 선형 기저 함수의 계수, $R_j(v)$ = 방사형 기저 함수, r_{mj} = 방사형 기저 함수의 계수, n = 모션의 개수)

[0066] 더욱 바람직하게는, 상기 단계 (4-3)에서 모션 블렌딩하기 이전에,

[0067] 상기 선택된 데이터들을 동일한 시간 축에 정렬하는 시간 정규화 단계를 더 포함할 수 있다.

[0068] 상기한 목적을 달성하기 위한 본 발명의 특징에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 장치는,

[0069] 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 깊이맵 센서;

[0070] 3D 캐릭터의 실시간 포즈별 조인트 위치 및 오리엔테이션 데이터를 도출하는 3D 캐릭터 조인트 데이터 출력 모듈;

[0071] 상기 3D 캐릭터의 하체 모션 캡처 데이터를 저장하는 하체 모션 캡처 데이터 저장 모듈; 및

[0072] 상기 3D 캐릭터 조인트 데이터 출력 모듈을 통해 도출된 데이터 중 상체 데이터와 상기 하체 모션 캡처 데이터 저장 모듈에 저장된 하체 모션 캡처 데이터 중에서 선택된 데이터를 합성하는 모션 스플라이싱 모듈을 포함하여 구성될 수 있다.

[0073] 바람직하게는, 상기 모션 스플라이싱 모듈은,

[0074] 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 상기 제약 프레임을 검출하고, 검출한 왼발의 제약 프레임이 일어나는 시작 시각과 오른발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되어 제자리걸음 동작으로 인식되면, 상기 모션 스플라이싱을 수행하는 것일 수 있다.

발명의 효과

[0075] 본 발명에서 제안하고 있는 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법 및 그 장치에 따르면, 깊이맵 센서를 통해 깊이맵 정보를 실시간으로 획득하여 상체 움직임을 반영하되, 하체 움직임은 미리 저장된 하체 모션 캡처 데이터에서 3D 캐릭터 속도에 매칭 되는 하체 모션 캡처 데이터를 선택 도출하여 합성하는 구성을 채택함으로써, 깊이맵 센서를 이용하여 사용자의 움직임을 충실히 반영함과 동시에 장소의 한계를 극복할 수 있다.

[0076] 또한, 본 발명에 따르면, 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출하여 제자리걸음 동작을 인식함과 동시에, 속도를 자동으로 검색하고 미리 입력되어 있는 다양한 속도의 걷기 모션 중 적절한 모션을 선택하여 블렌딩하고, 이를 상체 데이터와 결합함으로써, 발 움직임을 인식하여 간단하게 사용자의 내비게이션 의도를 파악하고, 제자리걸음을 자연스러운 실제 걸음으로 바꾸어 가상공간을 내비게이션 할 수 있도록 한다.

도면의 간단한 설명

[0077] 도 1은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법의 흐름을 도시한 도면.

도 2는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에 이용되는 15개 조인트를 가진 계층적 스켈레톤 구조를 도시한 도면.

도 3은 본 발명의 다른 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법의 흐름을 도시한 도면.

도 4는 본 발명의 다른 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 제약 프레임을 표시한 도면.

도 5는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에 의해 3D 캐릭터가 가상공간 내비게이션 동작 합성된 결과를 도시한 도면.

도 6은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 모션 스플라이싱 단계(단계 S300 내지 S500)을 도시한 도면.

도 7은 본 발명의 다른 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 시간 정규화 단계를 도시한 도면.

도 8 및 도 9는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 루트 조인트 위치의 2D 변환 과정을 도시한 도면.

도 10은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 사용자 움직임과 3D 캐릭터의 실시간 연동 모습을 도시한 도면.

도 11은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 제약 조건 파악을 통한 제자리걸음 인식을 도시한 도면.

도 12 및 도 13은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 모션 스플라이싱을 통한 가상공간 내비게이션 동작을 도시한 도면.

도 14는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 상체 움직임과 하체 움직임이 합성된 가상공간 내비게이션 동작을 도시한 도면.

발명을 실시하기 위한 구체적인 내용

[0078] 이하에서는 첨부된 도면을 참조하여 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명을 용이하게 실시할 수 있도록 바람직한 실시예를 상세히 설명한다. 다만, 본 발명의 바람직한 실시예를 상세하게 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다. 또한, 유사한 기능 및 작용을 하는 부분에 대해서는 도면 전체에 걸쳐 동일 또는 유사한 부호를 사용한다.

[0079] 덧붙여, 명세서 전체에서, 어떤 부분이 다른 부분과 ‘연결’되어 있다고 할 때, 이는 ‘직접적으로 연결’되어 있는 경우뿐만 아니라, 그 중간에 다른 소자를 사이에 두고 ‘간접적으로 연결’되어 있는 경우도 포함한다. 또한, 어떤 구성요소를 ‘포함’한다는 것은, 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있다는 것을 의미한다.

[0080] 도 1은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법의 흐름을 도시한 도면이다. 도 1에 도시된 바와 같이, 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법은, 깊이맵 센서를 통해 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 단계(S100), 단계 S100에서 획득한 깊이맵 정보를 이용하여, 사용자와 대응되는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 도출하는 단계(S200), 단계 S200의 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를, 하체 데이터와 상체 데이터로 분리하는 단계(S300), 미리 저장된 하체 모션 캡처 데이터에서 3D 캐릭터의 속도에 매칭 되는 하체 모션 캡처 데이터를 선택하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계(S400), 및 단계 S300에서 분리된 상체 데이터와 단계 S400에서 도출된 3D 캐릭터 하체 모션 캡처 데이터를 결합하는 단계(S500)를 포함하여 구현될 수 있다.

[0081] 단계 S100은, 깊이맵 센서를 통해 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 단계로서, 실공간

간상의 오브젝트와 센서와의 거리 값을 실시간으로 측정하여 사용자의 움직임을 인식하도록 할 수 있다. 깊이 맵 센서에는 대표적으로 마이크로소프트사가 개발한 키넥트 센서가 있고, 최근에는 Activate 3D사에서 개발한 Intelligent Character motion(ICM)기술, 즉, 깊이맵 카메라를 이용하여 팔의 움직임을 기준으로 실시간 캐릭터 내비게이션을 제어하는 기술이 개발되었다. 그러나 이와 같은 종래의 깊이맵을 이용한 가상공간 캐릭터 내비게이션 동작 및 제어는, 사용자가 깊이맵 센서 또는 카메라를 정면으로 바라보아야하고, 제자리에서 수행할 수 있는 동작 정도로만 모션이 제한되는 문제가 있었다. 이에 본 발명자들은 깊이맵 센서를 이용하여 사용자 움직임을 인식하되, 하체 데이터는 분리하여 미리 저장된 하체 모션 캡처 데이터를 이용하여 모션 스플라이싱(motion splicing)함으로써, 센서를 이용하여 사용자의 움직임을 충실히 반영함과 동시에 장소의 한계를 극복할 수 있는 새로운 방법을 제안하고자 한다.

[0082]

단계 S200에서는, 단계 S100에서 획득한 깊이맵 정보를 이용하여, 사용자와 대응되는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 도출할 수 있다. 즉, 3D 캐릭터의 조인트 오리엔테이션을 계산하고, 루트 조인트 위치를 설정하는 단계이다. 실시예에 따라서는, 깊이맵 정보로부터 3D 캐릭터를 연동하기 위하여 계층적 캐릭터 스켈레톤 포맷을 이용할 수 있고, 상용 3D제작 소프트웨어에서 사용되는 다양한 상용 포맷이 가능하다. 도 2는 본 발명의 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에 이용되는 15개 조인트를 가진 계층적 스켈레톤 구조를 도시한 도면이다. 도 2에 도시된 바와 같이, 계층적 캐릭터 스켈레톤은 루트 조인트를 포함하여 15개 조인트로 구성되는 것일 수 있다.

[0083]

보다 구체적으로는, 바이오비전사의 BVH(Biovision Hierarchy) 포맷을 이용할 수 있고, 사용자의 조인트 오리엔테이션 값을 받기 위해 OpenNI 미들웨어를 사용할 수 있다. OpenNI에서는 15개 조인트에 대한 전역 위치 및 전역 오리엔테이션 값을 제공한다. BVH포맷의 계층적 스켈레톤에서는 루트 조인트를 제외하고는 모두 로컬 오리엔테이션(local orientation)을 이용하기 때문에, 전역 오리엔테이션 값을 로컬 값으로 변환할 필요가 있다. 스켈레톤의 포즈 M은 하나의 벡터 $M=\{p^0, q^0, q^1, q^2, q^3, \dots, q^n\}$ 로 표현할 수 있는데($p \in R^3, q \in S^3$ (사원수)), p^0 는 루트 조인트의 전역 위치를 말하며 q^0 는 루트 조인트의 전역 오리엔테이션, $q^1, \dots, q^n$ 은 루트 조인트를 제외한 n개의 조인트의 로컬 오리엔테이션 값을 나타낸다. 특히, p^0 는 계층적 구조에서 루트 조인트에 해당하는 값으로서, 가상공간상의 캐릭터의 전체적인 위치를 나타내는 3차원 벡터값이며, q^0 는 가상공간상의 전체적인 캐릭터의 방향을 나타낸다. 만약, 깊이맵 센서를 통해 얻은 n개의 조인트에 대한 오리엔테이션 값을 $Q^0, Q^1, \dots, Q_n$ 이라고 하고, 캐릭터의 특정 조인트 i의 현재 로컬 오리엔테이션을 q^i , 조인트 i의 부모 조인트 j의 전역 오리엔테이션을 q^j 라고 하면, $q^i = q^{j-1} * Q^i$ 로 표현될 수 있다. 즉, Q^i 와 부모 조인트의 전역 오리엔테이션 값 q^j 와의 차이 값을 구한 후(q^{-1} =사원수의 역수), 이값을 현재 조인트 i의 로컬 오리엔테이션 값으로 설정하면 된다. 루트 조인트의 경우 로컬오리엔테이션과 전역오리엔테이션에 동일한 값이 들어간다. 이때, OpenNI에서 지원해주는 신뢰값(confidence)을 측정하여 센서로부터 받은 데이터의 정확도를 확인한 후에, 일정 값 이상의 신뢰도를 보일 때에만 설정하도록 하여, 조인트의 꼬임 및 떨림 현상을 줄일 수 있다.

[0084]

또한, 3D 캐릭터의 루트 조인트 위치 데이터는, 깊이맵 정보로부터 얻은 상기 사용자의 루트 조인트 위치 정보를, 사용자와 3D 캐릭터의 크기 값 차이를 계산하여 스케일링한 값일 수 있다. 즉, 캐릭터의 루트 조인트의 위치를 나타내는 p^0 는 전체적인 캐릭터의 가상공간상의 위치를 나타내는 중요한 정보인데, 본 발명에서는 깊이맵 센서로부터 얻은 루트 조인트 위치 정보를 적절하게 스케일링하여 캐릭터가 땅속으로 들어가거나 공중에 떠있지 않고, 항상 땅위에 서 있도록 하며, 사용자와 캐릭터가 연동되어 움직일 때도 캐릭터가 항상 적절한 위치로 이동하도록 할 수 있다. 캐릭터의 위치인 p^0 는 센서 앞에 동작하는 사용자의 깊이맵 상의 전역 위치 값 P^0 정보를 이용하여 설정하되, 현재 연동하고 있는 캐릭터와 사용자의 사이즈 차이를 계산하여 자동으로 스케일링한다. 사용자와 캐릭터의 크기 값 차이를 구하기 위해, 루트 조인트 전역위치와 양 발(Foot) 조인트의 전역 위치 간 거리 값을 이용할 수 있는데, 깊이맵 센서를 통해 얻는 사용자 루트 조인트의 전역위치 및 오른쪽 발 위치와 왼쪽 발 위치를 각각 P^0, P^r, P^l 이라고 하고, 가상공간상에서 나타나 있는 캐릭터의 루트 조인트 및 오른쪽, 왼쪽

발 위치를 각각 P^0 , P^r , P^l 이라고 하면, 스케일 상수 r 은 하기 수학식 1에 의해 결정된다. 또한, 스케일 상수 r 을 이용하여, t 프레임에 들어오는 깊이맵 상의 사용자 루트 조인트 위치를 P_t^0 라고 하고 초기 위치를 P_o^0 이라고 하면, t 프레임에서의 캐릭터 위치인 p_t^0 는 하기 수학식 2와 같이 계산된다. 즉, 사용자와 3D 캐릭터의 크기 값 차이의 계산은, 하기 수학식 1 및 2에 의해 수행할 수 있다.

수학식 1

$$r = \frac{(P^0 - P^r)^2 + (P^0 - P^l)^2}{(p^0 - p^r)^2 + (p^0 - p^l)^2}$$

[0085]

수학식 2

$$p_t^0 = p_o^0 + r(P_o^0 - P_t^0)$$

[0086]

[0087] (p_t^0 = t 프레임에서의 캐릭터 위치, p_o^0 = 캐릭터의 초기 위치, p^l = 캐릭터의 왼발 위치, p^r = 캐릭터의 오른발 위치, P_t^0 = t 프레임에 들어오는 깊이맵 상의 사용자 루트 조인트 위치, P_o^0 = 깊이맵 상의 사용자의 초기 위치, P^l = 깊이맵 상의 사용자 왼발 위치, P^r = 깊이맵 상의 사용자 오른발 위치, r = 스케일 상수)

[0088] 단계 S300에서는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 하체 데이터와 상체 데이터로 분리할 수 있고, 단계 S400에서는 미리 저장된 하체 모션 캡처 데이터에서 3D 캐릭터의 속도에 매칭 되는 하체 모션 캡처 데이터를 선택하여 3D 캐릭터 하체 모션 캡처 데이터를 도출할 수 있으며, 단계 S500에서는, 단계 S300에서 분리된 상체 데이터와 단계 S400에서 도출된 3D 캐릭터의 하체 모션 캡처 데이터를 결합할 수 있다.

[0089] 깊이맵 센서를 이용하면, 3D 캐릭터는 사용자의 움직임을 실시간으로 따라하며, 그대로 3D 캐릭터는 가상공간상으로 이동하게 된다. 하지만, 가상공간을 내비게이션 하는 중에 사용자가 감지영역을 벗어나게 되면 조인트의 위치 및 오리엔테이션에 대한 신뢰도 값이 급속도로 떨어지면서 스켈레톤의 꼬임이 발생하게 된다. 따라서 본 발명에서는, 상체 움직임에 대해서는 깊이맵 정보를 그대로 이용하되, 하체 데이터는 분리하여 미리 저장된 하체 모션 캡처 데이터에서 선택된 데이터를 모션 합성을 통해 새로운 내비게이션 모션을 생성하고, 이를 다시 상체 데이터와 결합하는 모션 스플라이싱 과정을 포함함으로써, 사용자는 걷는 도중에 자유롭게 상체를 사용할 수 있으며, 이와 동시에 하체 동작은 모션 캡처 데이터를 그대로 이용하므로 자연스러운 걷기 동작을 생성할 수 있다. 모션 스플라이싱을 위해서는 사용자의 내비게이션 의도를 파악할 필요가 있는바, 이는 이후 단계 S210 및 S220과 관련하여 상세하게 설명하기로 한다.

[0090] 도 3은 본 발명의 다른 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법의 흐름을 도시한 도면이다. 도 3에 도시된 바와 같이, 단계 S300 이전에, 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출하는 단계(S210) 및 단계 S210에서 검출한 오른발의 제약 프레임이 일어나는 시작 시각과 왼발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되면 제자리걸음 동작으로 인식하는 단계(S220)를 더 포함하여 구현될 수 있고, 단계 S300 내지 단계 S500은 단계 S220에서 제자리걸음 동작이 인식되는 경우에 수행될 수 있다.

[0091] 즉, 본 발명에서는, 사용자의 가상공간 내비게이션 의도를 실시간으로 인식하기 위하여 제자리걸음 동작 인식 (gesture recognition)단계(S210 및 S220)가 수행될 수 있다. 제자리걸음 동작 인식 단계에서는 3D 캐릭터의 발이 가상공간의 땅에 닿는 시간 프레임을 제약프레임(constrained frame)으로 정의한 후, 이 제약프레임을 실시간으로 자동 검출한다. 이때, 땅에 닿는지를 확인하기 위해서는 임계값(threshold) 설정이 필요한데, 본 논문에서는 이를 자동으로 설정하기 위한 새로운 적응적 임계값(adaptive threshold)을 제안한다. 이 방법을 통하여 사용자는 정확한 임계값을 설정하지 않아도, 알고리즘이 스스로 필요에 따라 임계값을 수정함으로써 더욱 정확한 검출을 가능하게 한다. 검출된 제약프레임은 시공간적 차이를 분석하여 최종적으로 제자리 걷기 동작인지의 여부를 인식한다.

[0092] 사용자의 가상공간 내비게이션 의도를 파악하기 위한 제스처 인식 기술은 컴퓨터 비전 및 기계 학습 연구자들 사이에 많은 연구가 되어 오고 있으나, 복잡하고 구현이 어려워 실제 사용이 어려운 것이 대부분이다. 이에 본 발명에서는 일반적인 동작 인식이 아닌, 본 발명의 목적인 3D 캐릭터 내비게이션을 위한 새롭고 간단한 제자리 걸음 인식방법을 제시하고자 한다.

[0093] 단계 S210에서는, 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출할 수 있다. 검출된 제약 프레임들은 시공간적 간격 및 순서를 점검하여 최종적으로 제자리걸음을 인식할 수 있다. 제약 프레임을 자동으로 찾는 방법은, 그 제약 프레임을 깊이맵 센서를 통해 나오는 데이터를 이용한 센서 도메인 안에서 찾는 방법과, 이 데이터를 이용하여 3D 캐릭터를 연동한 후 3D 캐릭터가 속해져 있는 가상공간 도메인 안에서 찾는 방법이 있다. 첫 번째 방법의 경우, IR(Infrared) 센서의 특성상 조명에 상당히 민감하여 센서의 공간상의 위치에 따라 전혀 다른 데이터가 나올 수 있으므로, 3D 캐릭터가 속해져 있는 가상공간 도메인 안에서 찾는 방법을 선택하는 것이 바람직하다.

[0094] 제약 프레임인지를 확인하기 위해서는 3D 캐릭터의 발 위치 및 속도 값을 이용할 수 있다. 현재 캐릭터의 왼쪽 발의 3차원 전역 위치를 $p^l(\in R^3)$ 이라고 하면, 스칼라값으로 정의되는 속도 v^l 은 하기 수학적 식 3에 의해 결정된다.

수학적 식 3

$$v^l = u\left(\frac{\|p_t^l - p_{t-1}^l\|}{\Delta t}, n\right)$$

$$u(v, n) = rdxl(v, n)$$

[0095]

[0096] $\|p_t^l - p_{t-1}^l\|$ 는 t와 t-1프레임 사이의 발이 움직인 거리, Δt 는 프레임 사이의 시간차이를 의미하며, $rdxl(v, n)$ 함수는 n번째 숫자에서 반올림 되는 값을 구해주는 함수를 의미한다. 라운드 함수를 쓰는 이유는 값의 민감성을 줄이기 위한 일종의 댐핑(damping) 역할을 위해서다. 오른쪽 발의 경우 같은 방식에 의해 다음 수학적 식 4와 같이 계산될 수 있다.

수학적 식 4

$$v^r = u\left(\frac{\|p_t^r - p_{t-1}^r\|}{\Delta t}, n\right)$$

$$u(v, n) = rdxl(v, n)$$

[0097]

[0098] 실시예에 따라서는, 계산된 p^l 과 v^l 이 동시에 각각 주어진 임계값 p_{max} 와 v_{max} 안에 동시에 들어오면 해당 프레임을 제약 프레임으로 간주하여 제약 프레임을 검출할 수 있다. 즉, 해당 포즈에서의 왼발의 위치(p^l) 및 속도 값

(v^l), 또는, 오른발의 위치(p^r) 및 속도 값(v^r)이 미리 정해진 임계값 p_{max} 및 v_{max} 보다 작은 경우, 해당 프레임을 제약 프레임으로 인식할 수 있다.

[0099] 따라서, 정확한 p_{max} 와 v_{max} 를 설정하는 것은 제약 프레임 검출에 상당히 중요한 작업이다. 하지만, 이값은 애플리케이션에 따라 다르고, 캐릭터마다 세밀한 조정이 필요한 시간적 노력이 요구된다.

[0100] 본 발명의 일실시예에 따르면, 임계값 p_{max} 와 v_{max} 를 자동으로 설정하는 적응적 임계값(adaptive threshold)방법을 제시한다. 3D 캐릭터의 왼발 및 오른발에 대한 위치 및 속도 값이 모두 초깃값으로 주어진 임계값보다 큰 값을 가지는 경우, 자동으로 임계값이 미리 정해진 값만큼 수정되는 적응적 임계값 설정 방법을 적용할 수 있다. 이 방법은 임의의 초깃값으로 주어진 임계값을 제약조건 판별함수를 실행하면서 매 프레임마다 필요에 따라 알고리즘을 통해 자동으로 수정되는 방식으로 사용자의 세밀한 조정이나 시간적 노력을 덜어줄 수 있다. 이 방법을 가능케 하는 기본 아이디어는 사람이 걷는 동작을 수행 시 반드시 한 발 이상은 땅에 항상 닿아야 한다는 것이다. 그러므로 현재 주어진 임계값을 이용하여 왼발과 오른발에 대한 제약조건 프레임 판별을 수행 후, 두 발 모두 임계값을 벗어나 제약조건이 아닌 것으로 판별이 된다면, 이 임계값은 조정이 되어야 한다는 것을 알 수 있다.

[0101] 한편, 실시예에 따라서는, p^l 과 p^r 을 각각 왼발과 오른발의 위치, 그리고 v^l 과 v^r 을 각각의 속도라고 가정하고, 왼쪽오른쪽 발에 대한 제약조건 판별 후에 이 두 발 모두 임계값보다 큰 값을 가져 제약 프레임이 아닌 걸로 판별된다면 새로운 임계값 p_{max} 와 v_{max} 는 하기 수식 5를 통해 설정될 수 있고, δ 는 0.1로 설정할 수 있다. 이 방법을 통해 임계값은 자동으로 필요에 따라 업데이트되며, 이에 따라 제약조건 프레임은 자동으로 판별된다.

수식 5

$$p_{max} = \min(p^r, p^l) + \delta$$

$$v_{max} = \min(v^r, v^l) + \delta$$

[0102]

[0103] 단계 S220에서는, 단계 S210에서 검출한 왼발의 제약 프레임이 일어나는 시작 시각과 오른발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되면 제자리걸음 동작으로 인식할 수 있다. 사람의 동작이 제자리 걷기로 판별되기 위해서는 제약 프레임의 생성 순서를 확인한 후, 이 제약 프레임 사이의 시간적 차이를 확인하여 동작 인식을 할 수 있다.

[0104] 도 4는 본 발명의 다른 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 제약 프레임을 표시한 도면이다. 도 4에 도시된 바와 같이, 사람의 걷기 동작을 포함하는 일반적인 동작 수행 시 제약조건 프레임은 한꺼번에 여러 개의 프레임이 연속적으로 나타나게 된다. 제자리걸음이 이루어지기 위해서는 캐릭터의 양 발이 번갈아 땅에 닿아야 하며, 두 사이의 간격이 일정시간 미만이어야만 한다. 이 조건을 이용하기 위해 제안된 알고리즘에서는 오른쪽 제약조건 프레임 묶음과 왼쪽 제약조건 프레임 묶음이 일어나는 시작 시간(t_r , t_l)을 기록해 두며, 이 둘 사이의 시간 차이가 t 초(미리 정해진 시간) 차이 미만으로 판정되면 제자리걸음으로 인식하였다. 바람직하게는, 본 단계에서의 미리 정해진 시간은, 1.5초 내지 2.5초 내에서 설정될 수 있다. 그러나 이는 실시예에 따른 것일 뿐으로서, 3D 캐릭터나 가상공간의 환경 등에 따라 다양하게 응용되어 적용될 수 있다.

[0105] 한편, 본 발명에서는 두 발 중 한 발은 항상 땅에 닿아 있는 것으로 간주하여 제자리 걷기 동작에서 적합하게 적용될 수 있으나, 실시예에 따라서는, 3D 캐릭터의 왼발 및 오른발의 위치가 미리 정해진 높이 이상 땅으로부

터 떨어져 있으면 뛰는 동작으로 인식할 수 있다. 즉, 뛰는 동작을 수행 시, 두 발이 땅에 닿지 않는 프레임이 존재하게 되므로, 이와 같은 특성을 이용하여 걷는 동작뿐만 아니라 뛰는 동작도 반영이 가능하다.

[0106] 제자리걸음 동작 인식 단계가 수행된 후에는, 모션 스플라이싱을 이용한 모션 합성(synthesis) 단계(단계 S300 내지 S500)가 수행될 수 있다. 제자리걸음 동작 인식 단계의 수행으로 사용자의 내비게이션 의도를 파악한 다음 단계로서, 3D 캐릭터가 가상공간을 자연스럽게 내비게이션 하기 위한 모션 스플라이싱이 수행될 수 있다. 현재 깊이맵 센서를 통해 사용자와 연동되어 있는 캐릭터 모션은 제자리걸음(walking-in-place)이므로, 실제 내비게이션을 위한 걷기 모션과는 물리적으로 다른 모션이다. 이와 같은 제자리 걷기 모션을 실제 걷기 동작으로 자연스럽게 변화시키기 위해 본 발명에서는 새로운 모션 스플라이싱 기법을 제안한다. 모션 스플라이싱 기술은 사용자의 제자리걸음을 그대로 아바타의 가상공간 내비게이션에 이용하지 못하므로 사용자의 동작을 상체(upper body)와 하체(lower body)로 분리한 후에, 미리 입력받은 다양한 모션캡처 데이터와의 분석 및 매칭을 통해 상하체 모션을 결합하여, 이를 3D 캐릭터에 반영함으로써 자연스럽게 가상공간 상을 내비게이션 하는 3D 캐릭터의 모션을 합성하는 기술이다. 이 기법을 이용하면 상체는 깊이맵 센서의 실시간 동작 데이터를 그대로 이용하여 3D 캐릭터와 사용자의 일체감을 그대로 유지함과 동시에 모션 캡처를 통해 미리 저장된 걷기 모션을 하체에 적용함으로써 자연스러운 움직임 동작을 생성할 수 있다.

[0107] 도 5는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에 의해 3D 캐릭터가 가상공간 내비게이션 동작 합성된 결과를 도시한 도면이다. 본 발명에 따르면, 도 5에 도시된 바와 같이, 깊이맵 센서의 장점인 실시간 모션캡처를 이용하여 상체의 모션을 그대로 캐릭터에 반영함과 동시에, 감지영역의 한계 때문에 가능하지 못한 3D 캐릭터의 자연스러운 가상공간 내비게이션 모션을 생성할 수 있는 장점이 있으므로, 이를 이용한 다양한 게임에 적용 가능하게 된다.

[0108] 3D 캐릭터의 모션을 생성하기 위해 제약조건 기반 동작 인식 기법을 통해 사용자의 내비게이션 의도를 파악한 후, 이 제자리 모션을 일반적인 걷기 모션으로 변환하여 캐릭터에 적용하는 작업이 필요하다. 이 작업은, 상체 모션은 깊이맵을 통한 실시간 연동 모션을 그대로 이용하고, 하체 모션만을 일반적인 걷기 모션으로 바꾸는 모션 스플라이싱을 통해 이루어진다. 도 6은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 모션 스플라이싱 단계(단계 S300 내지 S500)를 도시한 도면이다. 도 6에 도시된 바와 같이, 깊이맵을 통해 연동된 캐릭터의 원 포즈를 $M=\{p^0, q^0, q^1, q^2, q^3, \dots, q^n\}$ 벡터로 표현한다면, 이 중에 하체에 해당하는 $L=\{p^0, q^0, q^1, \dots, q^k\}$ 를 모션캡처를 통해 미리 생성해놓은 $L'=\{p^0, q^0, q^1, \dots, q^k\}$ 로 교체하여 새로운 포즈 $M'=\{L', q^{k+1}, q^{k+2}, \dots, q^n\}$ 을 생성할 수 있다.

[0109] 단계 S400에서는 3D 캐릭터 하체 모션 캡처 데이터를 도출할 수 있는데, 이를 위해서는 전처리된 하체 모션 캡처 데이터가 미리 저장되어야 한다. 본 발명의 일실시예에 따르면, 땅이 발에 닿는 순간 제자리걸음을 인식하기 때문에, 모든 모션 스플라이싱은 한걸음 단위(walking-cycle-motion)로 각각 수행될 수 있다. 전처리 단계에서는 현재 연동되고 있는 3D 캐릭터와 동일한 조인트 개수를 가지며 복수의 걷기 속도를 가진 하체 모션 캡처 데이터들을 미리 사이클 단위로 쪼개어 메모리 상에 한꺼번에 읽어 들일 수 있다. 이때 구현의 편의성을 위해 오른발로 시작하는 사이클 걷기 모션과, 왼발로 시작하는 사이클 걷기 모션을 따로 구분하여 저장할 수 있다.

[0110] 본 발명의 다른 실시예에 따르면, 도 3에 도시된 바와 같이, 단계 S400은, 미리 저장된 하체 모션 캡처 데이터를 한걸음 단위로 쪼개 후 속도 및 방향 값으로 매개화하는 단계(S410), 매개화된 하체 모션 캡처 데이터에 대하여 해당 3D 캐릭터의 속도를 반영하여 방사형 기저함수(RBF, Radial Basis Funtion)기반으로 가중치를 구하는 단계(S420), 및 매개화된 하체 모션 캡처 데이터 중에서 선택된 데이터를 가중치를 적용하여 모션 블렌딩하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계(S420)를 더 포함하여 구현될 수 있다.

[0111] 걸음 동작에 있어서, 같은 걸음에도 다양한 속도의 모션이 존재하므로, 모션 스플라이싱을 수행할 때도 제자리

걸음의 속도에 정확히 일치하는 걷는 모션을 합쳐하여야 한다. 만약 속도가 일치하지 않는 하체 모션을 이용할 시 상당히 어색한 걸음걸이가 나타나므로, 본 발명의 일실시예에 따르면, 속도 값(v)을 가중치로 사용한 주기 단위의 예제 모션캡처 데이터 모션 블렌딩 기법을 적용할 수 있다. 제자리걸음 속도인 v 는 프레임 간의 시간적 차이인 Δt 와 그 시간 동안 발끝의 움직이는 거리 값인 D 를 이용하여 하기 수학식 6으로 구할 수 있다.

수학식 6

$$v = D / \Delta t$$

[0112]

주어진 속도 v 와 필요로 하는 주기의 유형(오른쪽, 왼쪽 걷기 주기)을 기준으로, n 개의 모션에 대해 각각의 가중치(weight)를 계산할 수 있다. 실시예에 따라서는, 가중치를 계산하는 방법으로 Sloan이 제안한 산란 데이터 보간법(scattered data interpolation)을 이용할 수 있다. 구체적으로, 단계 S420에서 해당 3D 캐릭터의 속도란, 하기 수학식 7을 통해 구할 수 있다.

수학식 7

$$w_m(v) = a_m A(v) + \sum_{j=1}^n r_{mj} R_j(v)$$

[0114]

이 공식에서 $w_m(v)$ 는 모션 m 의 속도 v 에 대한 가중치를 나타낸다. $A(v)$ 와 a_m 은 선형 기저 함수(linear basis function) 및 계수, $R_j(v)$ 및 r_{mj} 는 방사형기저(radial basis function) 함수 및 계수, 그리고 n 은 모션의 개수를 나타낸다. 이때 예제 모션에 대한 데이터보간(interpolation)이 이루어지기 위해서는 가중치 값을 계산할 때 모션 m 자신이 가진 v 에서는 1, 그 외의 모션에 대해서는 0의 가중치를 가져야 한다. 즉, $i=m$ 인 모션에 대해서는 $w_m(v_i)=1$ 이어야 하고, 그 외의 $i \neq m$ 인 모션에 대해서는 $w_m(v_i)=0$ 을 가져야 한다. a_m 과 $R_j(v)$ 는 Park이 제안한 방법을 각각 이용하여 계산할 수 있다.

[0115]

한편, 실시예에 따라서는, 단계 S430에서 모션 블렌딩하기 이전에, 선택된 데이터들을 동일한 시간 축에 정렬하는 시간 정규화 단계를 더 포함하여 구현될 수 있다. 즉, n 개의 예제 모션에 대한 가중치를 구한 후, 이 가중치에 따라 모션 블렌딩을 수행하기 전에, 이 예제 모션 사이클을 동일한 시간 축에 정렬하는 작업이 필요하다. 이를 시간 정규화라고 하며, 이렇게 정렬되어 있는 시간을 정규(generic) 시간이라고 한다. 주어진 실시간을 f 라고 할 때, 정규 시간인 $g(f)$ 는 하기 수학식 8에서와 같이 구할 수 있다.

수학식 8

$$g(f) = \left((m-1) \left(\frac{f - k_m}{k_{m+1} - k_m} \right) \right) \frac{1}{n_{k-1}}$$

[0117]

수학식 8에서 k_m 은 땅에 처음 닿은 프레임의 의미하며 m 은 $f > k_m$ 이 되는 가장 큰 인덱스, 그리고 n_k 는 전체 프레임 수를 나타낸다. 이 계산을 통해 모든 시간은 $[0 \ 1]$ 사이로 정규화된다. 반대로 정규 시간 $g(f)$ 로부터 원 시간인 f 는 다음 수학식 9에 의해 구할 수 있다.

수학식 9

$$f(g) = ((n_k - 1) g(f) - (m - 1)(k_{m+1} - k_m) + k_m$$

[0119]

[0120] 도 7은 본 발명의 다른 실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 시간 정규화 단계를 도시한 도면이다. 도 7에 나타난 바와 같이, 정규화 시간 모션을 대상으로 제자리 걷기 모션의 속도인 v 를 이용하여, 현재 프레임인 f 와 바로 앞 프레임인 $f-1$ 에서의 정규 시간을 t_f 와 t_{f-1} 이라고 하면 $t_f = t_{f-1} + \Delta t_{f-1}$ 로 표현할 수 있다. 이 Δt_{f-1} 을 이용하여 각 모션의 새로운 정규시간인 $g(f)$ 와 대한 원래 시간인 $f(g)$ 를 구하고, $f(g)$ 포즈에 대해 수식식 7에서 얻은 가중치 값을 블렌딩하여 새로운 포즈를 합성할 수 있다. 즉, 하체의 조인트 $L' = \{p^0, q^0, q^1, \dots, q^k\}$ 의 각 조인트 앵글 q^i , $0 \leq i \leq k$ 는 가중치인 w_m 을 이용하여 k 개의 사원수 값을 SLERP(Spherical Linear Interpolation)을 연속적으로 수행하여 구할 수 있다.

[0121] 아울러, 본 발명의 일실시예에서 제자리걸음 동작 인식 방법은, 땅에 닿은 제약 프레임을 이용하기 때문에 인식 자체는 연속적(continuous)으로 하지 않고, 이산적인(discrete) 특징이 있는바, 이를 해결하기 위하여 제자리 걷기 속도인 v 값을 이용하여 일단 제자리걸음이 시작된 걸로 인식이 되면 v 가 일정 속도 범위 안에 있으면 계속 걷기동작으로 간주하도록 변경하여 더욱 부드러운 모션 합성을 수행할 수도 있다.

[0122] 도 8 및 도 9는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 루트 조인트 위치의 2D 변환 과정을 도시한 도면이다. 사이클 단위로 수행한 예제 모션 블렌딩은 캐릭터의 가상공간상 전역위치와 방향을 나타내는 p^0 와 q^0 를 제외한 모든 로컬 조인트 오리엔테이션을 대상으로 한다. 캐릭터의 전체적인 위치를 나타내는 p^0 값은 제자리걸음의 경우 거의 변화가 없으나, 실제 걸음의 경우, 루트 조인트 위치가 걷기 모션 수행에 따라 변하므로 가상공간상의 실제 위치가 시간 축에 따라 반드시 변화가 있어야 한다. 입력된 모션캡처 데이터의 경우, 데이터 내부에 전역위치인 p^0 데이터가 미리 들어 있기 때문에 이를 현재 가상공간에 맞게끔 2D 변환시킨 후 캐릭터에 적용할 필요가 있다. 도 8에 도시된 바와 같이, 사이클 단위로 이루어진 하체 모션 캡처 데이터의 p^0 값을 현재 가상공간 아바타 캐릭터의 모션에 2D 변환 후에 적용할 수 있다. 이 2D 변환에 사용되는 회전 행렬은 현재 센서와 사용자와의 각도차이를 이용해 구할 수 있다. 도 9에 도시된 바와 같이, 사용자의 루트 조인트 위치와 센서를 잇는 벡터와 현재 캐릭터의 몸체가 가리키고 있는 벡터 사이의 각도 값을 이용하여 2D 회전 행렬을 구할 수 있다.

[0123] 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법을 검증하기 위하여 마이크로소프트의 키넥트 센서를 이용하여 실시간 3D 캐릭터 연동을 실험하였다. 모션 스플라이싱을 위해서는 10개의 다양한 속도를 가진 걷기 동작 모션 캡처 데이터를 사이클 단위로 나눈 후에, 오른발로 시작하는 사이클 걷기 모션 5개와 왼발로 시작하는 사이클 걷기 모션 5개를 나눴다. 단위 시간 동안 발끝의 움직이는 거리를 통한 평균 모션 속도를 계산하면 하기 표 1과 같다.

표 1

번호	모션 사이클 타입	속도(unit/sec)
1	오른발	7.57
2	오른발	8.05
3	오른발	9.21
4	오른발	10.02
5	오른발	12.21
6	왼발	7.54
7	왼발	8.06
8	왼발	9.22
9	왼발	10.00
10	왼발	12.01

[0124]

[0125] 도 10은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 사용자 움직임과 3D 캐릭터의 실시간 연동 모습을 도시한 도면이다. 도 10에 도시된 바와 같이, 사용자가 제자리 걷기 모션을 취하면 3D 캐릭터도 사용자의 모션을 그대로 반영하였다. 제자리걸음에 대한 동작 인식을 확인하기 위하여 3D 캐릭터의 발이 땅에 닿는 순간 발 조인트의 색깔을 다르게 표시하였다. 도 11은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 제약 조건 파악을 통한 제자리걸음 인식을 도시한 도면이다. 도 12 및 도 13은 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 모션 스플라이싱을 통한 가상공간 내비게이션 동작을 도시한 도면이다. 도 14는 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어 방법에서 상체 움직임과 하체 움직임이 합성된 가상공간 내비게이션 동작을 도시한 도면이다. 도 10 내지 도 14에 도시된 바와 같이, 상체 부분은 실시간 연동이 계속되고 있으므로 사용자는 걷는 도중에 자유롭게 상체를 사용할 수 있으며 이와 동시에 하체 동작은 하체 모션 캡처 데이터를 그대로 이용하므로 자연스러운 걷기 동작을 생성할 수 있다.

[0126] 본 발명의 다른 측면은, 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어장치를 제공한다. 본 발명의 일실시예에 따른 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어장치는, 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 깊이맵 센서, 3D 캐릭터의 실시간 포즈별 조인트 위치 및 오리엔테이션 데이터를 도출하는 3D 캐릭터 조인트 데이터 출력 모듈, 3D 캐릭터의 하체 모션 캡처 데이터를 저장하는 하체 모션 캡처 데이터 저장 모듈, 및 3D 캐릭터 조인트 데이터 출력 모듈을 통해 도출된 데이터 중 상체 데이터와 하체 모션 캡처 데이터 저장 모듈에 저장된 하체 모션 캡처 데이터 중에서 선택된 데이터를 합성하는 모션 스플라이싱 모듈을 포함하여 구성될 수 있다. 실시예에 따라서는, 모션 스플라이싱 모듈은, 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출하고, 검출한 왼발의 제약 프레임이 일어나는 시작 시각과 오른발의 제약 프레임이 일어나는 시작 시각의 차이가 미리 정해진 시간 미만으로 판정되어 제자리걸음 동작으로 인식되면, 모션 스플라이싱을 수행하는 것일 수 있다. 각 구성에 대해서는 앞서 깊이맵 센서를 이용한 3D 캐릭터의 가상공간 내비게이션 동작 합성 및 제어방법과 관련하여 설명한 바와 유사하므로 상세한 설명은 생략하기로 한다.

[0127] 이상 설명한 본 발명은 본 발명이 속한 기술분야에서 통상의 지식을 가진 자에 의하여 다양한 변형이나 응용이 가능하며, 본 발명에 따른 기술적 사상의 범위는 아래의 특허청구범위에 의하여 정해져야 할 것이다.

부호의 설명

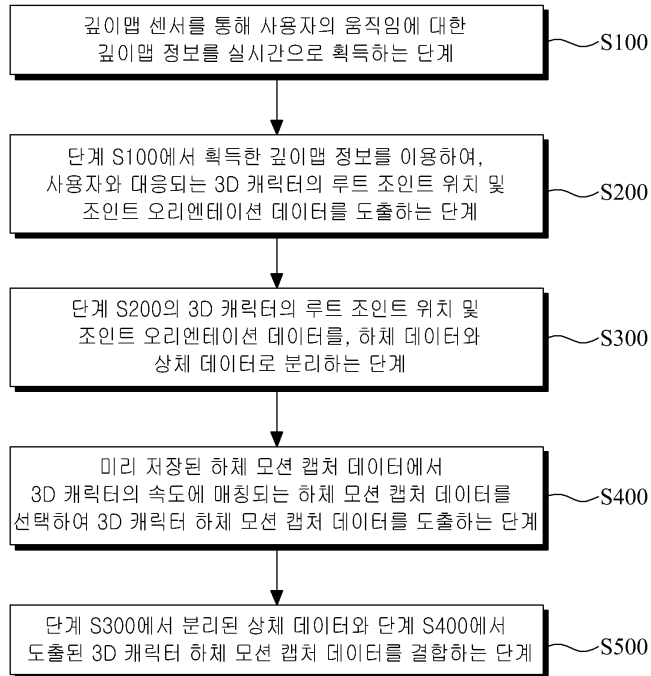
[0128] S100: 깊이맵 센서를 통해 사용자의 움직임에 대한 깊이맵 정보를 실시간으로 획득하는 단계
 S200: 단계 S100에서 획득한 깊이맵 정보를 이용하여, 사용자와 대응되는 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를 도출하는 단계
 S210: 3D 캐릭터의 해당 포즈에서의 왼발과 오른발 위치 및 속도 값을 이용하여, 3D 캐릭터의 발이 땅에 닿는 프레임(제약 프레임)을 검출하는 단계
 S220: 단계 S210에서 검출한 왼발의 제약 프레임이 일어나는 시작 시간과 오른발의 제약 프레임이 일어나는 시작 시간의 차이가 미리 정해진 시간미만으로 판정되면 제자리걸음 동작으로 인식하는 단계
 S300: 단계 S200의 3D 캐릭터의 루트 조인트 위치 및 조인트 오리엔테이션 데이터를, 하체 데이터와 상체 데이터로 분리하는 단계
 S400: 미리 저장된 하체 모션 캡처 데이터에서 3D 캐릭터의 속도에 매칭되는 하체 모션 캡처 데이터를 선택하여 3D 캐릭터 하체 모션 캡처 데이터를 도출하는 단계
 S410: 미리 저장된 하체 모션 캡처 데이터를 한걸음 단위로 쪼갬 후 속도 및 방향 값으로 매개화하는 단계
 S420: 매개화된 하체 모션 캡처 데이터에 대하여 해당 3D 캐릭터의 속도를 반영하여 방사형 기저함수(RBF, Radial Basis Function) 기반으로 가중치를 구하는 단계
 S430: 매개화된 하체 모션 캡처 데이터 중에서 선택된 데이터를 가중치를 적용하여 모션 블렌딩하여 최종 하체

모션 캡처 데이터를 도출하는 단계

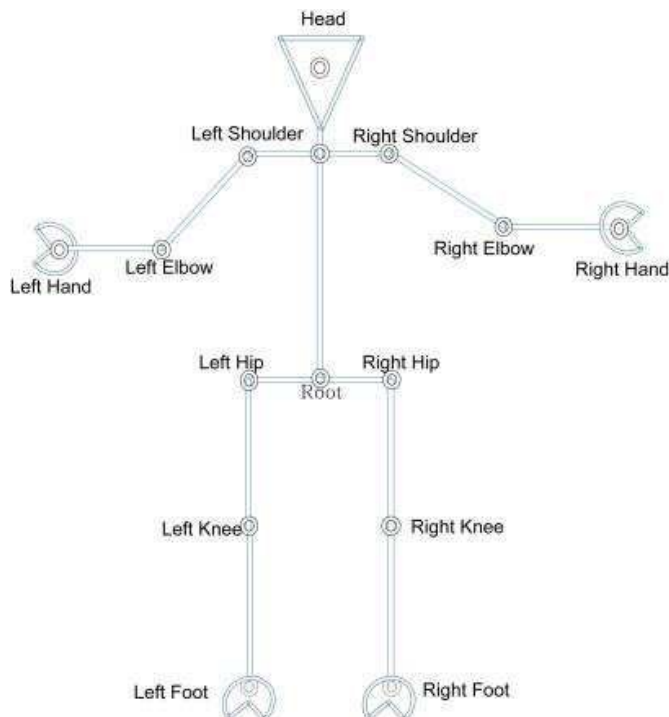
S500: 단계 S300에서 분리된 상체 데이터와 단계 S400에서 도출된 3D 캐릭터 하체 모션 캡처 데이터를 결합하는 단계

도면

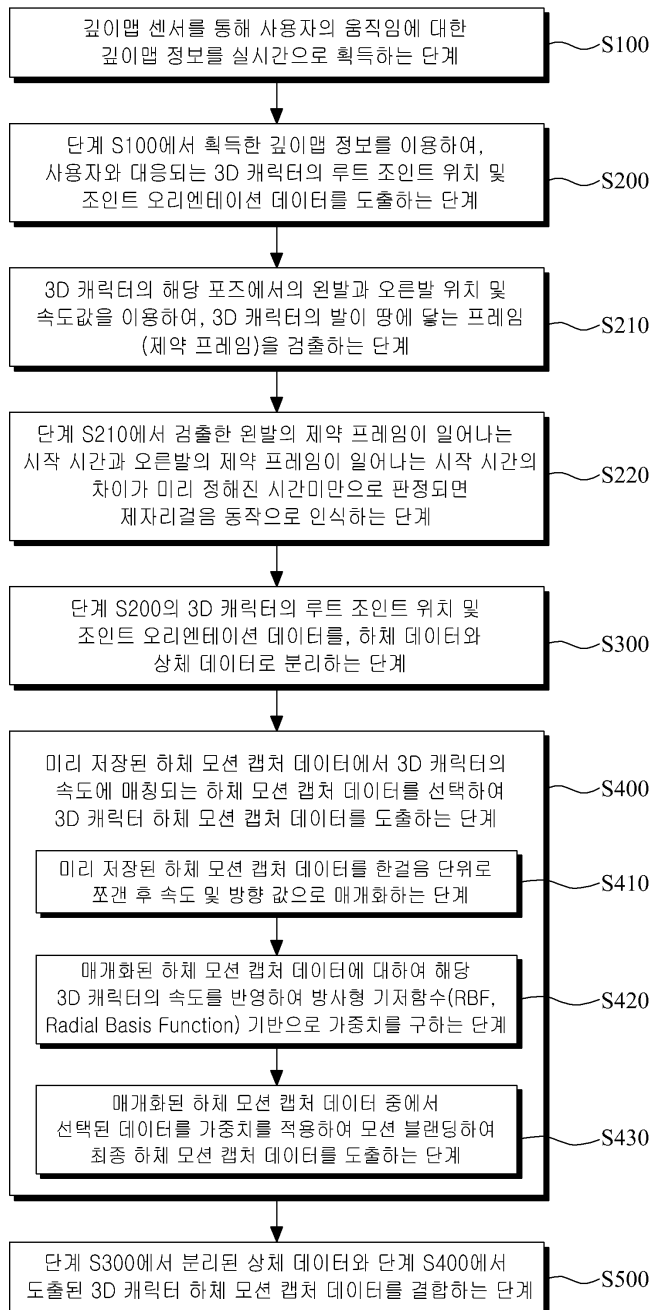
도면1



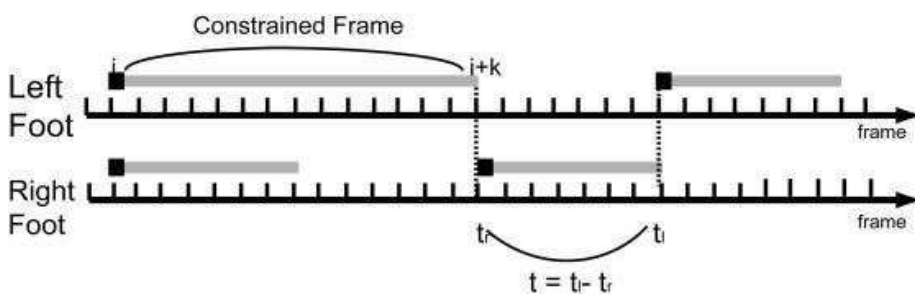
도면2



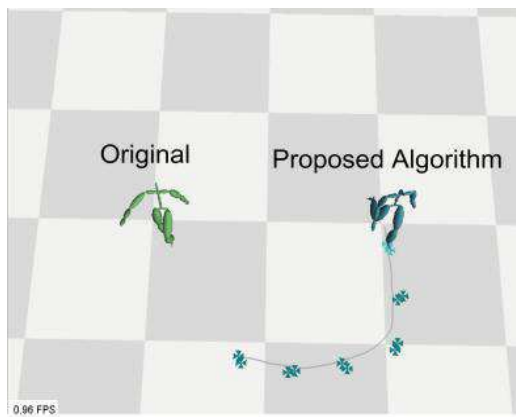
도면3



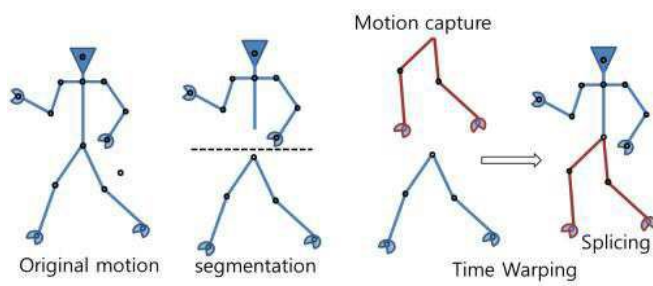
도면4



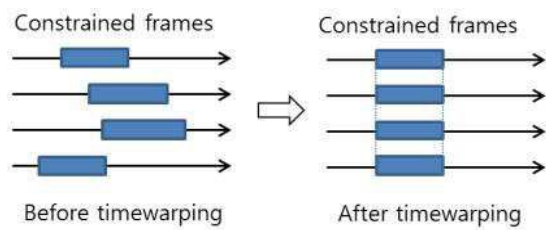
도면5



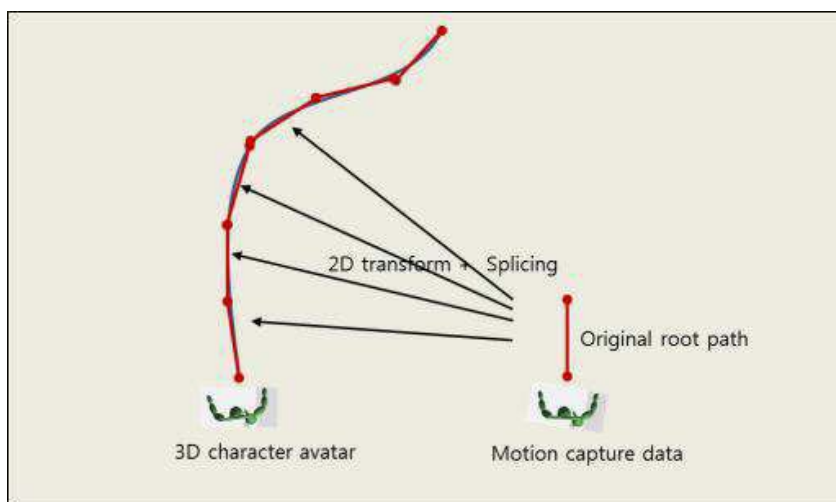
도면6



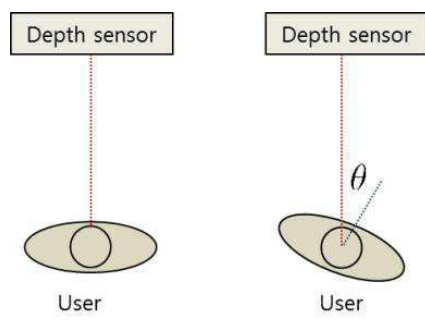
도면7



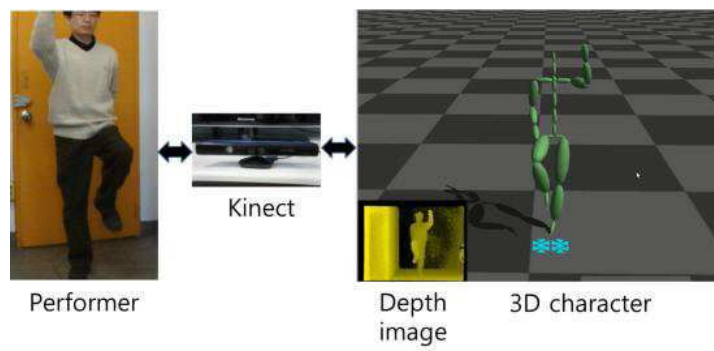
도면8



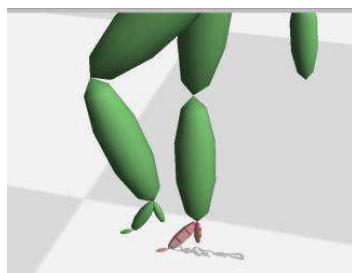
도면9



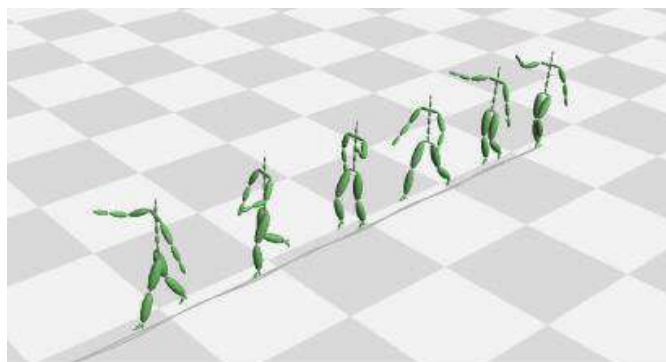
도면10



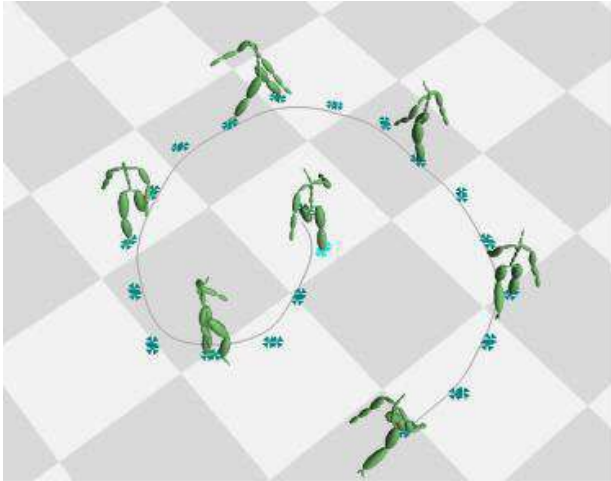
도면11



도면12



도면13



도면14

