



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0041498
(43) 공개일자 2022년04월01일

(51) 국제특허분류(Int. Cl.)
G06T 7/20 (2017.01) G06T 7/33 (2017.01)
G06T 7/593 (2017.01) G06T 7/90 (2017.01)
(52) CPC특허분류
G06T 7/20 (2013.01)
G06T 7/33 (2017.01)
(21) 출원번호 10-2020-0124784
(22) 출원일자 2020년09월25일
심사청구일자 2020년09월25일
기술이전 회망 : 기술양도

(71) 출원인
인천대학교 산학협력단
인천광역시 연수구 아카데미로 119 (송도동)
(72) 발명자
최욱
인천광역시 연수구 송도문화로28번길 28, 106동
2802호(송도동, 송도글로벌캠퍼스푸르지오)
(74) 대리인
특허법인충정

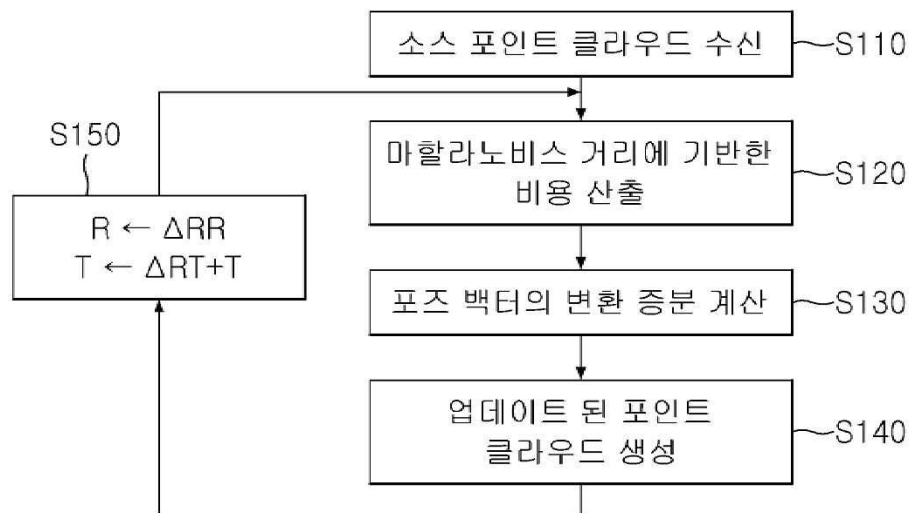
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 색상을 반영한 포인트 클라우드의 재구성을 위한 반복 최근접 포인트 탐색 방법

(57) 요약

본 발명은 색상을 반영한 포인트 클라우드의 재구성을 위한 반복 최근접 포인트 탐색 방법에 관한 것으로서, 본 발명의 영상 처리 시스템에서 소스 포인트 클라우드를 참조 포인트 클라우드와 정합하기 위한 반복 최근접 포인트 탐색 방법은, 소스 포인트 클라우드와 참조 포인트 클라우드 간의 회전 벡터와 병진 벡터를 포함하는 포즈 벡터를 이용하여, 미리 정해진 수의 네이버들에 대해 매칭 확률을 할당하는 반복 K-최근접 포인트 알고리즘에 따라, 상기 포즈 벡터와 연관되고 소스 포인트와 참조 포인트 간의 마할라노비스 거리에 기초한 비용을 산출하는 단계, 및 상기 회전 벡터의 회전 증분 및 상기 병진 벡터의 병진 증분을 이용하여 상기 포즈 벡터를 업데이트하는 과정을 복수회 반복 과정을 수행하여, 상기 소스 포인트 클라우드가 상기 포즈 벡터에 의해 업데이트된 포인트 클라우드를 생성하는 단계를 포함한다.

대표도 - 도1a



(52) CPC특허분류

G06T 7/593 (2017.01)

G06T 7/90 (2017.01)

G06T 2207/10028 (2013.01)

G06T 2210/56 (2013.01)

G06T 2211/416 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711117140
과제번호	GK20P0200
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	범부처GigaKOREA사업(R&D)
연구과제명	4D 복원 및 동적 변형 거동 모델 기반의 초실감 서비스 기술 개발
기 여 율	1/2
과제수행기관명	전자부품연구원
연구기간	2017.04.01 ~ 2020.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711116148
과제번호	IITP-2020-2018-0-01431
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	대학ICT연구센터지원사업
연구과제명	MR-IoT융합기반의 재난대응 인공지능 응용기술
기 여 율	1/2
과제수행기관명	아주대학교 산학협력단
연구기간	2018.06.01 ~ 2021.12.31

명세서

청구범위

청구항 1

영상 처리 시스템에서 소스 포인트 클라우드를 참조 포인트 클라우드와 정합하기 위한 반복 최근접 포인트 탐색 방법에 있어서,

소스 포인트 클라우드와 참조 포인트 클라우드 간의 회전 벡터와 병진 벡터를 포함하는 포즈 벡터를 이용하여, 미리 정해진 수의 네이버들에 대해 매칭 확률을 할당하는 반복 K-최근접 포인트 알고리즘에 따라, 상기 포즈 벡터와 연관되고 소스 포인트와 참조 포인트 간의 마할라노비스 거리에 기초한 비용을 산출하는 단계; 및

상기 회전 벡터의 회전 증분 및 상기 병진 벡터의 병진 증분을 이용하여 상기 포즈 벡터를 업데이트하는 과정을 복수회 반복 과정을 수행하여, 상기 소스 포인트 클라우드가 상기 포즈 벡터에 의해 업데이트된 포인트 클라우드를 생성하는 단계

를 포함하는 반복 최근접 포인트 탐색 방법.

청구항 2

제1항에 있어서,

상기 비용은 상기 소스 포인트와 참조 포인트 간의 점대 점 또는 점대 평면 거리에 대한 제곱에 가중치를 적용한 후 합산한 값에 대응되는 반복 최근접 포인트 탐색 방법.

청구항 3

제1항에 있어서,

상기 반복 K-최근접 포인트 알고리즘에 따른 상기 포즈 벡터의 탐색은 상기 소스 포인트와 참조 포인트 각각의 3차원 좌표 벡터와 3차원 색상 벡터를 결합한 6차원 벡터 공간에서 이루어지는 반복 최근접 포인트 탐색 방법.

청구항 4

제1항에 있어서,

상기 반복 과정에서 상기 비용의 변화율(gradient)이 0이 되는 증분을 계산하고 상기 회전 증분 및 병진 증분의 크기가 임계치 이하가 될 때까지 미리 정해진 최대 반복 횟수 이내에서 상기 반복 과정을 수행하며,

상기 회전 벡터 R 의 상기 회전 증분 β_R 및 상기 병진 벡터 T 의 병진 증분 β_T 에 따라, 상기 반복 과정에서 $R \leftarrow \Delta R R$, $T \leftarrow \Delta R T + \Delta T$ 에 따라 상기 포즈 벡터가 업데이트되는 반복 최근접 포인트 탐색 방법.

청구항 5

제1항에 있어서,

상기 상기 비용을 산출하는 단계 전에,

상기 포즈 벡터를 상수로 하고, 3차원 카메라로부터 획득된 상기 소스 포인트의 깊이값에 대하여, 상기 반복 K-최근접 포인트 알고리즘에 따라, 상기 마할라노비스 거리에 기초한 비용이 최소가 되도록 상기 깊이값을 업데이트하는 과정을 복수회 수행하는, 깊이 벡터 탐색 단계; 및

상기 업데이트된 깊이값을 상기 비용을 산출하는 단계의 상기 소스 포인트 클라우드에 적용하는 단계

를 더 포함하는 반복 최근접 포인트 탐색 방법.

청구항 6

제1항에 있어서,

멀티-뷰 포인트 클라우드들을 병합하는 단계를 더 포함하고,

상기 병합에서, 업데이트된 상기 포즈 벡터에 따라, 상기 멀티-뷰 포인트 클라우드들 중 가장 가까운 포인트 클라우드로부터 가장 먼 포인트 클라우드 순서로 병합하는 반복 최근접 포인트 탐색 방법.

청구항 7

제5항에 있어서,

상기 병합에서, 업데이트된 상기 깊이값에 따라, 상기 멀티-뷰 포인트 클라우드들 중 가장 먼 포인트 클라우드로부터 가장 가까운 포인트 클라우드 순서로 병합하고, 이후 라운드에서 가장 먼 것과 가장 가까운 것의 처리 순서를 교대로 반전시키는 반복 최근접 포인트 탐색 방법.

청구항 8

영상 처리 시스템에서 소스 포인트 클라우드를 참조 포인트 클라우드와 정합하기 위한 반복 최근접 포인트 탐색을 처리하는, 컴퓨터가 판독 가능한 코드를 기록한 기록 매체에 있어서,

소스 포인트 클라우드와 참조 포인트 클라우드 간의 회전 벡터와 병진 벡터를 포함하는 포즈 벡터를 이용하여, 미리 정해진 수의 네이버들에 대해 매칭 확률을 할당하는 반복 K-최근접 포인트 알고리즘에 따라, 상기 포즈 벡터와 연관되고 소스 포인트와 참조 포인트 간의 마할라노비스 거리에 기초한 비용을 산출하는 기능; 및

상기 회전 벡터의 회전 증분 및 상기 병진 벡터의 병진 증분을 이용하여 상기 포즈 벡터를 업데이트하는 과정을 복수회 반복 과정을 수행하여, 상기 소스 포인트 클라우드가 상기 포즈 벡터에 의해 업데이트된 포인트 클라우드를 생성하는 기능

을 구현하기 위한 기록 매체.

발명의 설명

기술 분야

[0001] 본 발명은 ICP(Iterative Closest Point, 반복 최근접 포인트) 탐색 방법에 관한 것으로서, 특히, 사람 등 3D 이미지를 재구성하기 위하여, 반복적인 K-최근접 포인트 알고리즘(Iterative K-Closest Point Algorithms)을 이용해, 소스 포인트 클라우드의 포인트가 K-최근접 포인트에 의해 참조 포인트 클라우드로 자연스럽게 매칭되고 비용을 최소화할 수 있는 반복 최근접 포인트 탐색 방법에 관한 것이다.

배경 기술

[0002] RGB-D (RGB-Depth) 카메라는 최근 비용 절감과 사용 편의성으로 인해 로봇 공학 및 3D 모델링의 많은 응용 분야에서 사용되고 있다. 환경의 RGB-D 비디오를 촬영하고 RGB-D 이미지에서 얻은 포인트 클라우드를 정합하여 조밀한 3D 맵을 만들 수 있다. 물체의 3D 모델은 여러 대의 보정된 RGB-D 카메라로 구성된 멀티뷰 시스템을 사용하여 얻을 수 있다.

[0003] ICP (Iterative Closest Point) 알고리즘은 정합 및 교정 오류의 보상에서 중요한 역할을 한다. ICP 알고리즘은 대응관계 검색과 포즈(pose) 추정 단계를 번갈아 가며 두 포인트 클라우드를 정합한다. 지난 30 년 동안 연구자들은 알고리즘의 문제를 극복하기 위한 알고리즘을 개발해 왔다. 3D 위치에만 의존하는 대응관계를 찾는 모호성을 줄이기 위해 대응관계 검색 단계는 3D 위치와 색상을 모두 활용하여 고차원 공간에서 가장 가까운 포인트를 찾도록 발전했다. 포인트 클라우드의 제한된 해상도로 인해 어느 포인트에 정확히 대응되는 포인트를 찾지 못할 수 있다. 이는 포인트가 다른 포인트 클라우드의 모든 포인트와 일치하도록 허용하고 모든 대응에 일치 확률을 할당함으로써 해결될 수 있다. 또는 점대 점 거리를 점대 평면 또는 평면 대 평면 거리로 대체하여 비용 함수를 개선하기도 한다. 포인트 간 대응의 가정을 완화함으로써 알고리즘이 더욱 정확하고 강력해진다.

[0004] 그러나, 종래의 ICP 알고리즘은 자연스러운 이미지 재구성이 부족하며 비용이 많이 들므로 이에 대한 개선이 필요한 실정이다.

발명의 내용

해결하려는 과제

- [0005] 따라서, 본 발명은 상술한 문제점을 해결하기 위하여 안출된 것으로, 본 발명의 목적은, 사람 등 3D 이미지를 재구성하는 데 있어서, 포인트 클라우드의 포인트가 K-최근접 포인트 그룹에 자연스럽게 매칭되고 비용을 최소화할 수 있도록 하기 위하여, 반복적인 K-최근접 포인트 알고리즘을 이용하되 계산 속도와 정확도 사이의 균형을 맞추기 위해 포인트가 K-최근접 포인트와 일치하도록 허용하고 소스/참조 포인트간 일치 확률을 K 쌍들에만 할당하며, 포즈 탐색은 3D 좌표와 포인트의 색상을 결합하여 6D 벡터 공간에서 수행하는, 반복 최근접 포인트 탐색 방법을 제공하는 데 있다.

과제의 해결 수단

- [0006] 먼저, 본 발명의 특징을 요약하면, 상기의 목적을 달성하기 위한 본 발명의 일면에 따른 영상 처리 시스템에서 소스 포인트 클라우드를 참조 포인트 클라우드와 정합하기 위한 반복 최근접 포인트 탐색 방법은, 소스 포인트 클라우드와 참조 포인트 클라우드 간의 회전 벡터와 병진 벡터를 포함하는 포즈 벡터를 이용하여, 미리 정해진 수의 네이버들에 대해 매칭 확률을 할당하는 반복 K-최근접 포인트 알고리즘에 따라, 상기 포즈 벡터와 연관되고 소스 포인트와 참조 포인트 간의 마할라노비스 거리에 기초한 비용을 산출하는 단계; 및 상기 회전 벡터의 회전 증분 및 상기 병진 벡터의 병진 증분을 이용하여 상기 포즈 벡터를 업데이트하는 과정을 복수회 반복 과정을 수행하여, 상기 소스 포인트 클라우드가 상기 포즈 벡터에 의해 업데이트된 포인트 클라우드를 생성하는 단계를 포함한다.
- [0007] 상기 비용은 상기 소스 포인트와 참조 포인트 간의 점대 점 또는 점대 평면 거리에 대한 제곱에 가중치를 적용한 후 합산한 값에 대응된다.
- [0008] 상기 반복 K-최근접 포인트 알고리즘에 따른 상기 포즈 벡터의 탐색은 상기 소스 포인트와 참조 포인트 각각의 3차원 좌표 벡터와 3차원 색상 벡터를 결합한 6차원 벡터 공간에서 이루어진다.
- [0009] 상기 반복 과정에서 상기 비용의 변화율(gradient)이 0이 되는 증분을 계산하고 상기 회전 증분 및 병진 증분의 크기가 임계치 이하가 될 때까지 미리 정해진 최대 반복 횟수 이내에서 상기 반복 과정을 수행하며, 상기 회전 벡터 R 의 상기 회전 증분 ΔR 및 상기 병진 벡터 T 의 병진 증분 ΔT 에 따라, 상기 반복 과정에서 $R \leftarrow \Delta R R$, $T \leftarrow \Delta R T + \Delta T$ 에 따라 상기 포즈 벡터가 업데이트된다.
- [0010] 상기 반복 최근접 포인트 탐색 방법은, 상기 상기 비용을 산출하는 단계 전에, 상기 포즈 벡터를 상수로 하고, 3차원 카메라로부터 획득된 상기 소스 포인트의 깊이값에 대하여, 상기 반복 K-최근접 포인트 알고리즘에 따라, 상기 마할라노비스 거리에 기초한 비용이 최소가 되도록 상기 깊이값을 업데이트하는 과정을 복수회 수행하는, 깊이 벡터 탐색 단계; 및 상기 업데이트된 깊이값을 상기 비용을 산출하는 단계의 상기 소스 포인트 클라우드에 적용하는 단계를 더 포함한다.
- [0011] 상기 반복 최근접 포인트 탐색 방법은, 멀티-뷰 포인트 클라우드들을 병합하는 단계를 더 포함하고, 상기 병합에서, 업데이트된 상기 포즈 벡터에 따라, 상기 멀티-뷰 포인트 클라우드들 중 가장 가까운 포인트 클라우드로부터 가장 먼 포인트 클라우드 순서로 병합할 수 있다.
- [0012] 상기 병합에서, 업데이트된 상기 깊이값에 따라, 상기 멀티-뷰 포인트 클라우드들 중 가장 먼 포인트 클라우드로부터 가장 가까운 포인트 클라우드 순서로 병합하고, 이후 라운드에서 가장 먼 것과 가장 가까운 것의 처리 순서를 교대로 반전시킬 수 있다.
- [0013] 그리고, 본 발명의 다른 일면에 따른 영상 처리 시스템에서 소스 포인트 클라우드를 참조 포인트 클라우드와 정합하기 위한 반복 최근접 포인트 탐색을 처리하는, 컴퓨터가 판독 가능한 코드를 기록한 기록 매체는, 소스 포인트 클라우드와 참조 포인트 클라우드 간의 회전 벡터와 병진 벡터를 포함하는 포즈 벡터를 이용하여, 미리 정해진 수의 네이버들에 대해 매칭 확률을 할당하는 반복 K-최근접 포인트 알고리즘에 따라, 상기 포즈 벡터와 연관되고 소스 포인트와 참조 포인트 간의 마할라노비스 거리에 기초한 비용을 산출하는 기능; 및 상기 회전 벡터의 회전 증분 및 상기 병진 벡터의 병진 증분을 이용하여 상기 포즈 벡터를 업데이트하는 과정을 복수회 반복 과정을 수행하여, 상기 소스 포인트 클라우드가 상기 포즈 벡터에 의해 업데이트된 포인트 클라우드를 생성하는 기능을 구현할 수 있다.

발명의 효과

[0014] 본 발명에 따른 반복 최근접 포인트 탐색 방법에 따르면, 반복적인 K-최근접 포인트 알고리즘을 이용하되 계산 속도와 정확도 사이의 균형을 맞추기 위해 포인트가 K-최근접 포인트와 일치하도록 허용하고 소스/참조 포인트 간 일치 확률을 K 쌍들에만 할당하되, 포즈 탐색은 3D 좌표와 포인트의 색상을 결합하여 6D 벡터 공간에서 수행함으로써, 사람 등 3D 이미지를 재구성하는 데 있어서, 소스 포인트 클라우드의 포인트가 K-최근접 포인트에 의해 참조 포인트 클라우드에 자연스럽게 매칭되고 비용을 최소화할 수 있다.

[0015] 본 발명에서, 모든 중요한 결과를 통합하는 비용 함수는 안정성과 정확도를 향상시키기 위해 점대 점 및 점대 평면 거리에 대한 제곱의 가중치 합으로 정의된다. 본 발명에서는, 1) ICP 알고리즘을 통해 비용을 최소화하는 포즈 파라미터를 찾아 비용을 최소화할 수 있으며, 2) RGB-D 이미지에서 포인트 클라우드를 얻었다고 가정하고 RGB-D 카메라로 측정한 깊이 값에 오류가 있음을 고려하여 측정된 깊이 값을 변수로 사용하고, 포즈 파라미터를 상수로 처리해 포인트 클라우드들의 포인트간 정합이 더욱 긴밀하게 이루어지도록 깊이 값을 탐색해 비용을 최소화할 수 있다.

도면의 간단한 설명

[0016] 본 발명에 관한 이해를 돕기 위해 상세한 설명의 일부로 포함되는 첨부도면은, 본 발명에 대한 실시예를 제공하고 상세한 설명과 함께 본 발명의 기술적 사상을 설명한다.

도 1a는 본 발명의 일 실시예에 따른 영상 처리 시스템에서의 포인트 클라우드의 재구성 방법을 설명하기 위한 흐름도이다.

도 1b는 본 발명의 일 실시예에 따른 영상 처리 시스템에서의 깊이 벡터 탐색을 위한 반복 K-최근접 포인트 알고리즘을 설명하기 위한 흐름도이다.

도 2는 (a) 깊이 이미지를 획득하기 위한, RGB 컬러 카메라, 두 개의 적외선(IR) 스테레오 카메라 및 하나의 IR 프로젝터를 포함하는 RGB-D 카메라 시스템을 예시하며, (b) 깊이 값 Z에 따라 그 표준 편차 σ_z 의 예시이다.

도 3은 멀티-뷰 이미지 획득을 위한 RGB-D 카메라의 위치 설정의 예시이다.

도 4는 RGB-D 카메라로 획득한 샘플 RGB-D 이미지를 보여준다.

도 5는 종래의 방법과 본 발명의 병합된 포인트 클라우드들을 비교하기 위한 렌더링된 이미지들의 예이다.

도 6은 본 발명의 일 실시예에 따른 포인트 클라우드의 재구성을 처리하기 위한 영상 처리 시스템의 구현 방법의 일례를 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0017] 이하에서는 첨부된 도면들을 참조하여 본 발명에 대해서 자세히 설명한다. 이때, 각각의 도면에서 동일한 구성 요소는 가능한 동일한 부호로 나타낸다. 또한, 이미 공지된 기능 및/또는 구성에 대한 상세한 설명은 생략한다. 이하에 개시된 내용은, 다양한 실시 예에 따른 동작을 이해하는데 필요한 부분을 중점적으로 설명하며, 그 설명의 요지를 흐릴 수 있는 요소들에 대한 설명은 생략한다. 또한 도면의 일부 구성요소는 과장되거나 생략되거나 또는 개략적으로 도시될 수 있다. 각 구성요소의 크기는 실제 크기를 전적으로 반영하는 것이 아니며, 따라서 각각의 도면에 그려진 구성요소들의 상대적 크기나 간격에 의해 여기에 기재되는 내용들이 제한되는 것은 아니다.

[0018] 본 발명의 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 본 발명의 실시 예들을 기술하기 위한 것이며, 결코 제한적이지는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0019] 또한, 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되는 것은 아니며, 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만

사용된다.

[0020] 도 1a는 본 발명의 일 실시예에 따른 영상 처리 시스템에서의 포인트 클라우드의 재구성 방법을 설명하기 위한 흐름도이다.

[0021] 도 1a를 참조하면, 본 발명의 ICP, 즉, 반복적인 K-최근접 포인트 알고리즘(Iterative K-Closest Point Algorithms)을 통한, 반복 최근접 포인트 탐색을 수행하는 본 발명의 영상 처리 시스템은, 카메라(RGB-D 카메라 또는 스테레오 카메라 등)에 기초한 2차원 좌표값 이외에 깊이 값을 포함하는 3차원 캡처 이미지, 즉, 포인트 클라우드를 수신한다(S110). 예를 들어, 포인트 클라우드는 3차원 좌표계의 포인트들 (X, Y, Z)에서의 색상 및 계조값 등을 포함하는 포인트 데이터로 이루어질 수 있다. 본 발명의 영상 처리 시스템은, 수신되는 소스 포인트 클라우드 $S_s = \{\mathbf{X}_i^{(s)}\}_{i=1}^{N_s}$ 를 데이터베이스의 참조 포인트 클라우드 $S_r = \{\mathbf{X}_i^{(r)}\}_{i=1}^{N_r}$ 와 비교하고 유사도 연관성을 탐색하여 그와 정합하기 위한 포인트 클라우드로 변환(재구성)한다. 이외에도 본 발명에서 반복적인 K-최근접 포인트 알고리즘에 따라 업데이트된 포즈 벡터나 깊이 벡터는 멀티-뷰의 병합이나 정합에도 이용될 수 있다.

[0023] <포즈 벡터 탐색을 위한 반복 K-최근접 포인트 알고리즘>

[0024] 본 발명의 영상 처리 시스템은, 소스 포인트 클라우드와 참조 포인트 클라우드 간의 회전 행렬 R (예, 3X3 행렬), 병진 벡터 T(예, 3차원)를 포함하는 포즈(pose) 파라미터(포즈 벡터)를 탐색한다(S120). 소스 포인트 클라우드 S_s 와 참조 포인트 클라우드 S_r 의 대응되는 포인트 쌍 $X_i^{(s)}$ 와 $X_j^{(r)}$ 의 관계는 [수학식1]과 같이 포즈 파라미터를 이용하여 나타낼 수 있다. 소스 포인트 클라우드 S_s 의 포인트 $X_i^{(s)}$ 는 [수학식1]에 따라 $\hat{\mathbf{X}}_i^{(s)} = \mathbf{R}\mathbf{X}_i^{(s)} + \mathbf{T}$ 와 같이 포즈 벡터 R, T에 의해 변환될 수 있다.

[0025] [수학식1]

$$\mathbf{X}_j^{(r)} = \mathbf{R}\mathbf{X}_i^{(s)} + \mathbf{T}$$

[0026]

[0027] 이와 같은 변환을 위한 본 발명의 영상 처리 시스템은, 각 반복 과정에서 소스/참조 포인트간 매칭 확률을 K-최근접 포인트에 의한 유사도가 가장 가까운 K 쌍들의 네이버들에만(K는 양의 정수) 할당해 탐색한다. 탐색은 소스 포인트의 3D 좌표 $\hat{\mathbf{X}}_i^{(s)}$ 와 3D 색상 벡터 $\mathbf{C}_i^{(s)}$ 을 결합하여 6D 벡터 공간에서 이루어진다. 참조 포인트 클라우드의 6D 벡터 포인트들은 3D 좌표 $X_j^{(r)}$ 와 색상(예, RGB, Red Green Blue) 벡터 $\mathbf{C}_j^{(r)}$ 을 결합하여 $\{(\mathbf{X}_j^{(r)}, \beta \circ \mathbf{C}_j^{(r)})\}_{j=1}^{N_r}$ 와 같이 나타낸다($\circ$ 는 하다마드곱(Hadamard product)). 소스 포인트 클라우드의 6D 벡터 포인트 $\{\hat{\mathbf{X}}_i^{(s)}, \beta \circ \mathbf{C}_i^{(s)}\}$ ($i=1 \sim N_s$)에 대한 ICP 알고리즘에 따른 K-최근접 네이버의 세트 N_i 에서의 K-최근접 포인트 $X_j^{(r)}$ ($j \in N_i$)의 탐색이 이루어진다. 여기서, β 는 서로 다른 색상 모델 간(예, RGB-D / YIQ)의 밸런싱(balancing)을 위해 곱해진다. 이때, $\hat{\mathbf{X}}_i^{(s)}$ 와 $X_j^{(r)}$ 사이의 공간 상의 거리 차이 벡터 d_{ij} 는 [수학식2]와 같이 정의되고, $\hat{\mathbf{X}}_i^{(s)}$ 와 $X_j^{(r)}$ 사이의 6D 차이 벡터 c_{ij} 는 [수학식3]과 같이 정의된다.

[0028] [수학식2]

$$\mathbf{d}_{i,j} = \mathbf{X}_j^{(r)} - \hat{\mathbf{X}}_i^{(s)} = \mathbf{X}_j^{(r)} - \mathbf{R}\mathbf{X}_i^{(s)} - \mathbf{T}$$

[0029]

[0030] [수학식3]

$$\mathbf{c}_{i,j} = (\mathbf{X}_j^{(r)}, \beta \circ \mathbf{C}_j^{(r)}) - (\hat{\mathbf{X}}_i^{(s)}, \beta \circ \mathbf{C}_i^{(s)})$$

[0031]

[0032] K-최근접 네이버의 세트 N_i 는 c_{ij} 의 크기 ($\|c_{ij}\|$)의 오름 차순으로 정렬된다. N_i 의 가장 가까운 인덱스에서의

첫번째 원소는 a_i 로 나타내기로 한다.

[0033] 포즈 벡터 탐색 단계에서, 본 발명의 영상 처리 시스템은, [수학식4]와 같이 포즈 벡터와 연관된 비용함수 E 를 계산해, 이를 최소화하는 회전 행렬 포즈 벡터, 즉, R (예, 3×3 행렬), 병진 벡터 T (예, 3차원)를 탐색해 나간다(S120).

[0034] [수학식4]

$$E = \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{d}_{i,j}^T \mathbf{M}_{i,j} \mathbf{d}_{i,j}$$

[0035]

[0036] [수학식4]에서, $p_{i,j}$ 는 $X_i^{(s)}$ 와 $X_j^{(r)}$ 사이의 매칭 확률이다. 여기서, $(j \in \mathcal{N}_i)$ 에서 $p_{i,j}=0$ 또는 $\sum_{j \in \mathcal{N}_i} p_{i,j} = 1$ 이다. $M_{i,j}$ 는 임의의 형태에서 소스 포인트와 참조 포인트 간의 마할라노비스(Mahalanobis) 거리로서, 비용함수를 정의하기 위한 행렬(예, 3×3 행렬)이다. 또한, $1/2$ 은 E 의 계산 상에서 2차 함수가 구분되어 2번 곱해진 팩터를 제거하기 위함이다.

[0037] 대응하는 포인트 $\hat{X}_i^{(s)}$ 와 $X_j^{(r)}$ 가 3D 위치와 색상에서 유사도가 가깝다고(네이버) 가정할 때, 매칭 확률 $p_{i,j}$ 는 [수학식5]와 같이, 나타낼 수 있다. 여기서, $\forall_i$ 는 모든 $p_{i,j}$ 가 0이 아니라면 $p_{i,j}$ 의 합이 1이 되게 하는 정규화 계수이고, τ_c 는 ICP 알고리즘 상의 거리 임계치이다.

[0038] [수학식5]

$$p_{i,j} = \begin{cases} \gamma_i \exp\left(-\frac{\|\mathbf{c}_{i,j}\|^2}{2\sigma_c^2}\right), & \text{if } \|\mathbf{c}_{i,j}\| < \tau_c, \\ 0, & \text{otherwise,} \end{cases}$$

[0039]

[0040] 본 발명에서, 초기 포즈 벡터 R^0, T^0 에 대해, $\{\|\mathbf{c}_{i,a_i}\|\}_{i=1}^{N_s}$ 의 평균이 τ_c 보다 크면, τ_c 는 해당 평균으로 대체된다. 이는 충분히 많은 대응 포인트들이 계산되도록 한다. $\{\|\mathbf{c}_{i,a_i}\|\}_{i=1}^{N_s}$ 의 표준편차 σ_c 는 τ_c 로 설정될 수 있다. 초기 포즈 벡터 R^0, T^0 는 소정의 알고리즘을 통해 추정되는 것도 가능하고, 미리 주어진 것으로 가정할 수도 있다.

[0041] 종래의 평면대 평면 거리에 대한 것과 달리, 소스 포인트와 참조 포인트 간의 점대 평면(또는 점대 점) 공간상의 거리에 대하여, $M_{i,j}$ 는 참조 포인트들에만 의존하며, 모든 대응관계의 탐색 후 재 계산없이 [수학식6]과 같이 한번만 계산된다. 여기서, $\mathbf{n}_j$ 는 $X_j^{(r)}$ 의 3차원 표면에서의 표면 법선 벡터(surface normal vector)이다. $\mathbf{n}_j \mathbf{n}_j^T$ 의 랭크(rank)는 1이고, 따라서 이는 비가역적(noninvertible)이다.

[0042] [수학식6]

$$\mathbf{M}_{i,j} = \mathbf{n}_j \mathbf{n}_j^T$$

[0043]

[0044] 본 발명에서 수식적 안정화를 증가시키도록, $M_{i,j}$ 를 단위행렬 I 와 계수 ϵ (예, 0.001)을 이용해 [수학식7]과 같이 정의한다. 여기서, $M_{i,j}$ 는 [수학식4]와 같이 점대 점 및 점대 평면 거리에 대한 제곱에 가중치를 적용한 후 합산한 값에 해당하는 비용 함수가 계산되도록 한다.

[0045] [수학식7]

$$\mathbf{M}_{i,j} = \mathbf{M}_j^{(r)} = \epsilon I + \mathbf{n}_j \mathbf{n}_j^T$$

[0046]

[0047] [수학식4]와 같은 비용함수 E 를 최소화하기 위하여, 본 발명에서 ICP 알고리즘에 가우스-뉴턴(Gauss-Newton) 알

고리즘을 적용한다. 매칭 확률 p_{ij} 는 포즈 파라미터에 따라서는 변할 수 있지만, 고정 변수로 간주된다.

[0048] 본 발명의 영상 처리 시스템은, 반복 과정에서, 포즈 벡터의 변환 증분을 계산한다(S130). 회전 증분 (incremental rotation) ΔR 은 [수학식8]과 같고, $\Delta \theta_X$, $\Delta \theta_Y$, $\Delta \theta_Z$ 는 XYZ 직각 좌표계에서의 해당 각각의 축에 대한 회전각이다. 병진 증분 $\Delta T=(\Delta T_X, \Delta T_Y, \Delta T_Z)$ 과 같이 나타낼 수 있고, 변환 증분 파라미터들은 벡터 $\delta=(\Delta \theta_X, \Delta \theta_Y, \Delta \theta_Z, \Delta T_X, \Delta T_Y, \Delta T_Z)^T$ 로 나타낼 수 있다.

[0049] [수학식8]

$$\Delta \mathbf{R} = \begin{bmatrix} 1 & -\Delta \theta_Z & \Delta \theta_Y \\ \Delta \theta_Z & 1 & -\Delta \theta_X \\ -\Delta \theta_Y & \Delta \theta_X & 1 \end{bmatrix}$$

[0051] 이에 따라, 포즈 벡터들에 의해 계산된 증분 δ 에서의 $d_{i,j}$ 는 [수학식9], [수학식10]과 같이 나타낼 수 있고(J_i 는 자코비안 행렬), 비용함수 E 의 변화율(gradient) ∇E 은 [수학식11]과 같이 나타낼 수 있다.

[0052] [수학식9]

$$\mathbf{d}_{i,j}(\delta) \approx \mathbf{d}_{i,j}(\mathbf{0}) + \mathbf{J}_i \delta$$

[0054] [수학식10]

$$\mathbf{J}_i = \begin{bmatrix} 0 & -\hat{Z}_i^{(s)} & \hat{Y}_i^{(s)} & -1 & 0 & 0 \\ \hat{Z}_i^{(s)} & 0 & -\hat{X}_i^{(s)} & 0 & -1 & 0 \\ -\hat{Y}_i^{(s)} & \hat{X}_i^{(s)} & 0 & 0 & 0 & -1 \end{bmatrix}$$

[0056] [수학식11]

$$\nabla E = \sum_{i=1}^{N_s} \sum_{j \in \mathcal{N}_i} p_{i,j} \left(\mathbf{J}_i^T \mathbf{M}_{i,j} \mathbf{J}_i \delta + \mathbf{J}_i^T \mathbf{M}_{i,j} \mathbf{d}_{i,j}(\mathbf{0}) \right)$$

[0058] 이때 비용이 최소가 되도록 $\nabla E=0$ 을 만족하는 δ 는 [수학식12]와 같이 계산될 수 있으며, 증분 δ 가 적용된 회전 및 병진 벡터는 [수학식13](' $\Delta R * R$ '에 의해 새로운 R 로 업데이트됨), [수학식14](' $\Delta R * T + \Delta T$ '에 의해 새로운 T 로 업데이트됨)와 같이 업데이트되어 적용될 수 있다.

[0059] [수학식12]

$$\delta = - \left(\sum_{i=1}^{N_s} \mathbf{J}_i^T \left(\sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{M}_{i,j} \right) \mathbf{J}_i \right)^{-1} \left(\sum_{i=1}^{N_s} \mathbf{J}_i^T \left(\sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{M}_{i,j} \mathbf{d}_{i,j}(\mathbf{0}) \right) \right)$$

[0061] [수학식13]

$$\mathbf{R} \leftarrow \Delta \mathbf{R} \mathbf{R}$$

[0063] [수학식14]

$$\mathbf{T} \leftarrow \Delta \mathbf{R} \mathbf{T} + \Delta \mathbf{T}$$

[0065] 본 발명의 영상 처리 시스템은, 상기 변경된 회전 및 병진 벡터를 이용하여 $X_i^{(s)}$ 가 상기 변경된 회전 및 병진 벡터에 의해 그만큼 회전되고 병진된 업데이트된 포인트 클라우드가 생성되고(S140), 가우스-뉴턴 알고리즘 적용이 수렴될 때까지 상기 변경된 회전 및 병진 벡터가 비용 산출에 이용되어 위와 위와 같은 과정은 반복 수행된다(S150). 이와 같은 소스 포인트로부터 참조 포인트로 정합을 위한 반복 과정은, 회전 및 병진 증분의 크기가 임계치(예, 회전 0.001° , 병진 0.001mm) 이하가 될 때까지 반복하고 이는 최대 반복 횟수(예, 80회)에 도달 시까지 그 이내에서 반복될 수 있다.

[0066] 위와 같은 소스 포인트 클라우드 S_s 를 참조 포인트 클라우드 S_r 로 정합하기 위한 ICP 알고리즘에서, 포즈 벡터를 탐색하는 과정은, 하기의 Algorithm1에 요약되어 있다.

[0067] 초기 포즈 벡터 R^0, T^0 는 설정은 정확한 포즈 벡터를 획득하도록 위와 같은 계산을 수렴시키는데 필요하다. 처음에는 다소 거칠게(coarse) 복셀(voxel)(예, 크기 4cm*2cm*1cm)들에 대한 소스 포인트로부터 참조 포인트로 정합을 위한 포즈 벡터들을 탐색하기 위한 위와 같은 반복 과정을 수행하면서, 점점 정확한 해가 획득되어 가고 비용함수가 최소화된다. 거리 임계치 τ_c 는 ($\sqrt{2}$ *복셀크기)로 설정될 수 있다.

Algorithm 1: Iterative K-closest point algorithm for pose refinement

Input: S_s, S_r, τ_c , initial pose parameters R^0, T^0
Output: Refined pose parameters R, T

```

1  $R \leftarrow R^0, T \leftarrow T^0$ 
2 Adjust  $\tau_c$  if the initial registration error is large
3 for  $i \leftarrow 1$  to  $N_r$  do
4   Compute the surface normal vector  $n_i$  for  $X_i^{(r)}$ 
5    $M_i^{(r)} \leftarrow \epsilon I + n_i n_i^T$ 
6 end
7 while not converged do
8   for  $i \leftarrow 1$  to  $N_s$  do
9     Compute  $\hat{X}_i^{(s)}$  and find the K-nearest neighbors to  $(\hat{X}_i^{(s)}, \beta \circ C_i^{(s)})$ 
10     $M_i \leftarrow O$ 
11    for  $j \in N_i$  do
12      Compute  $p_{i,j}$ 
13       $M'_{i,j} \leftarrow p_{i,j} M_j^{(r)}$ 
14       $M_i \leftarrow M_i + M'_{i,j}$ 
15    end
16  end
17  while not converged do
18     $A \leftarrow O, b \leftarrow 0$ 
19    for  $i \leftarrow 1$  to  $N_s$  do
20      Compute  $\hat{X}_i^{(s)}, J_i$ 
21       $e_i \leftarrow 0$ 
22      for  $j \in N_i$  do
23        Compute  $d_{i,j}(0)$ 
24         $e_i \leftarrow e_i + M'_{i,j} d_{i,j}(0)$ 
25      end
26       $A \leftarrow A + J_i^T M_i J_i$ 
27       $b \leftarrow b + J_i^T e_i$ 
28    end
29     $\delta \leftarrow -A^{-1}b$ 
30    Update  $R, T$  using Equations (13) and (14)
31  end
32 end

```

[0068]

[0070] <깊이 벡터 탐색을 위한 반복 K-최근접 포인트 알고리즘>

[0071] 도 1b는 본 발명의 일 실시예에 따른 영상 처리 시스템에서의 깊이 벡터 탐색을 위한 반복 K-최근접 포인트 알고리즘을 설명하기 위한 흐름도이다.

[0072] 반복 K-최근접 포인트 알고리즘에 따른 반복 과정에서, 3차원 카메라로부터 획득된 깊이값을 탐색함으로써 소스 포인트와 참조 포인트 간의 정합이 이루어질 수 있다. [수학식4]의 비용함수를 최소화하도록, 포인트 $X_i^{(s)}$ 의 측정한 깊이값 $Z_i^{(s)}$ 에 대한 해를 구할 수 있다. 포즈 벡터 R, T 를 상수로 간주하고, 비용 함수 E 는 [수학식15] 및 [수학식16]과 같이 독립된 비용 함수들 E_i 의 합으로 볼 수 있다(S210).

[0073] [수학식15]

$$E = \sum_{i=1}^{N_s} E_i$$

[0074]

[0075] [수학식16]

$$E_i = \frac{1}{2} \sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{d}_{i,j}^T \mathbf{M}_{i,j} \mathbf{d}_{i,j}$$

[0076]

[0077] $X_i^{(s)}$ 에 대응되는 정규화된 좌표 벡터를 x_i 로 나타내면, $X_i^{(s)} = Z_i^{(s)} x_i$ 로 되고, $d_{i,j}$ 의 변화율(gradient) $\nabla d_{i,j}$ 은 [수학식17]과 같다.

[0078] [수학식17]

$$\nabla \mathbf{d}_{i,j} = -\mathbf{R} \mathbf{x}_i$$

[0079]

[0080] 체인 룰을 적용하면, E_i 의 $Z_i^{(s)}$ 에 대한 도함수는 [수학식18]과 같고, 비용이 최소가 되도록 $\frac{dE_i}{dZ_i^{(s)}} = 0$ 을 만족하는 $Z_i^{(s)} = Z_{i,opt}^{(s)}$ 은 [수학식19]와 같이 계산된다.

[0081] [수학식18]

$$\frac{dE_i}{dZ_i^{(s)}} = \sum_{j \in \mathcal{N}_i} p_{i,j} \nabla^T \mathbf{d}_{i,j} \mathbf{M}_{i,j} \mathbf{d}_{i,j} = -\mathbf{x}_i^T \mathbf{R}^T \left(\sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{M}_{i,j} \left(\mathbf{x}_j^{(r)} - \mathbf{T} - Z_i^{(s)} \mathbf{R} \mathbf{x}_i \right) \right)$$

[0082]

[0083] [수학식19]

$$Z_{i,opt}^{(s)} = \frac{\mathbf{x}_i^T \mathbf{R}^T \sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{M}_{i,j} \left(\mathbf{x}_j^{(r)} - \mathbf{T} \right)}{\mathbf{x}_i^T \mathbf{R}^T \left(\sum_{j \in \mathcal{N}_i} p_{i,j} \mathbf{M}_{i,j} \right) \mathbf{R} \mathbf{x}_i}$$

[0084]

[0085] 위와 같은 소스 포인트 클라우드 S_s 를 참조 포인트 클라우드 S_r 로 정합하기 위한 ICP 알고리즘에서, 깊이 벡터를 탐색하는 과정은, 하기의 Algorithm2에 요약되어 있다. 공간 상의 거리 임계치 τ_d 는 4cm 정도로 설정되어 공간 적으로 먼 포인트가 $X_i^{(s)}$ 에 대한 네이버(이웃)가 되는 것을 방지하도록 한다.

Algorithm 2: Iterative K-closest point algorithm for depth refinement

Input: $\mathcal{S}_s, \mathcal{S}_r, \tau_d$, pose parameters $\mathbf{R}, \mathbf{T}$
Output: Refined point set $\mathcal{S}_s$

```

1 for  $i \leftarrow 1$  to  $N_r$  do
2   Compute the surface normal vector  $\mathbf{n}_i$  for  $\mathbf{X}_i^{(r)}$ 
3    $\mathbf{M}_i^{(r)} \leftarrow \epsilon \mathbf{I} + \mathbf{n}_i \mathbf{n}_i^T$ 
4 end
5 for  $i \leftarrow 1$  to  $N_s$  do
6   Compute  $\hat{\mathbf{X}}_i^{(s)}$  and find the  $K$ -nearest neighbors to  $\hat{\mathbf{X}}_i^{(s)}$ 
7    $\mathbf{M}_i \leftarrow \mathbf{O}, \mathbf{e}_i \leftarrow \mathbf{0}, p_i \leftarrow 0$ 
8   for  $j \in \mathcal{N}_i$  do
9     Compute  $p_{i,j}$ 
10     $p_i \leftarrow p_i + p_{i,j}$ 
11     $\mathbf{M}'_{i,j} \leftarrow p_{i,j} \mathbf{M}_j^{(r)}$ 
12     $\mathbf{M}_i \leftarrow \mathbf{M}_i + \mathbf{M}'_{i,j}$ 
13     $\mathbf{e}_i \leftarrow \mathbf{e}_i + \mathbf{M}'_{i,j} (\mathbf{X}_j^{(r)} - \mathbf{T})$ 
14  end
15  if  $p_i > 0$  then
16    if  $\mathbf{x}_i^T \mathbf{R}^T \mathbf{M}_i \mathbf{R} \mathbf{x}_i > \eta$  then
17       $Z_{i,opt}^{(s)} \leftarrow \frac{\mathbf{x}_i^T \mathbf{R}^T \mathbf{e}_i}{\mathbf{x}_i^T \mathbf{R}^T \mathbf{M}_i \mathbf{R} \mathbf{x}_i}$ 
18    end
19    else
20       $p_i \leftarrow 0$ 
21    end
22  end
23 end

```

[0087]

[0088]

[수학식5]의 가중치 함수에 따라 색상이 유사한 포인트 쌍들은 높은 매칭 확률 $p_{i,j}$ 을 갖는다. [수학식5]에서 색 상 제외 3D 위치에 대하여는, $\|\mathbf{c}_{i,j}\|$ 는 $\|\mathbf{d}_{i,j}\|$ 로 대체되어, 조건 $\|\mathbf{c}_{i,j}\| < \tau_c$ 대신에 $\|\mathbf{d}_{i,j}\| < \tau_d$ 로 대체된다.

[0089]

Algorithm2에서, $\eta=0.1$ 과 같이 계산상의 수식적 안정화를 위해 양의 상수로 설정된다. 어떤 포인트에서 $|Z_{i,opt}^{(s)} - Z_i^{(s)}|$ 는 τ_d 보다 클 수 있고, 이때 $Z_{i,opt}^{(s)}$ 는 $Z_i^{(s)} + \tau_d$ 또는 $Z_i^{(s)} - \tau_d$ 로 설정될 수 있다.

[0090]

모든 소스 포인트들에 대해 임계치 내에서 최근접 포인트들이 발견될 수 있는 것은 아니며, 이때 Algorithm2에 의해 해당 소스 포인트들이 변경될 수 있다. 변경되지 않는 포인트들은, 개선되도록 탐색된 포인트들이 되도록 하며, [수학식20]과 같이, $Z_i^{(s)}$ 를 $Z_{i,opt}^{(s)}$ 로 대체하지 않고 비용 함수 F 를 최소화할 수 있다. 여기서, p_i 는 $Z_i^{(s)}$ 에 대한 $Z_{i,opt}^{(s)}$ 의 식별자로서, 0 또는 1의 값을 갖는다. $\Delta Z_{i,m}$ 은 F 를 최소화하기 전의 초기 차이 $Z_i^{(s)} - Z_m^{(s)}$, R_i 는 이미지에서 $\mathbf{X}_i^{(s)}$ 의 복셀(또는 픽셀) 위치 주위의 네이버들의 인덱스(예, 8개)이다.

[0091]

[수학식20]

$$F(\{Z_i^{(s)}\}_{i=1}^{N_s}) = \sum_{i=1}^{N_s} p_i (Z_i^{(s)} - Z_{i,opt}^{(s)})^2 + \frac{\lambda}{2} \sum_{i=1}^{N_s} \sum_{m \in R_i} (Z_i^{(s)} - Z_m^{(s)} - \Delta Z_{i,m})^2$$

[0092]

[0093]

상기 F 의 최소화에 대한 자코비안 방법을 적용하면, [수학식21]과 같은 업데이트된 식을 획득할 수 있다. 여기서, λ 는 서로 다른 두 텀(term) 간의 밸런싱을 위한 계수(예, 0.2)이고, $|R_i|$ 은 R_i 의 원소들의 개수이다. 여기서, p_i 가 0인 $Z_i^{(s)}$ 에 대하여, 이웃 업데이트된 $Z_m^{(s)}$ 은 초기 상대 차이를 보유하는 $\Delta Z_{i,m}$ 만큼 $Z_i^{(s)}$ 를 $Z_m^{(s)}$ 로 끌어

당긴다.

[0094] [수학식21]

$$Z_i^{(s)} \leftarrow \frac{p_i Z_{i,opt}^{(s)} + \lambda \sum_{m \in \mathcal{R}_i} (Z_m^{(s)} + \Delta Z_{i,m})}{p_i + \lambda |\mathcal{R}_i|}$$

[0095]

[0096] [수학식21]과 같은 업데이트된 깊이값이 되도록 소정의 횟수(예,6회) 깊이 벡터 탐색 단계가 반복될 수 있다(S220). 이러한 최대 반복 카운트를 작게 하는 것은, 소스 부분(fragment)이 참조 부분의 적절한 부분 집합에 해당하면, 적당하다 할 수 있다. 이오 같이 깊이 벡터 탐색 단계에서 업데이트된 깊이값은 포즈 벡터 탐색 단계의 소스 포인트 클라우드에 적용된다(S230). 또한, 이와 같은 방식은 후술하는 바와 같이 병합 포인트 클라우드(merged point cloud)에 적용되는 것이 바람직하다.

[0098] <멀티-뷰 포인트 클라우드 정합>

[0099] 여러 시점(멀티-뷰)의 포인트 클라우드가 주어지면, 상기 포즈 벡터 및 상기 깊이값을 업데이트하는 위에서 기술한 바와 같은 방법을 이용하여, 인접한 포인트 클라우드를 반복적으로 병합하여 결국 모든 멀티-뷰 포인트 클라우드들을 병합한다. 포즈-그래프 최적화를 기반으로 하는 고급 알고리즘이 있지만 쌍별 정합 알고리즘의 견고성을 평가하는 데 적합한 더 간단한 방법을 제안한다. 단일 쌍의 오류가 후속 쌍의 오류로 전파되기 때문에, 본 발명의 이러한 방법에서는 쌍별 정합의 견고성이 중요하다.

[0100] 참조 포인트 클라우드 $S_r = S_0$ 및 인접 포인트 클라우드 S_s 가 주어지면, 위에서 기술한 바와 같은 본 발명의 ICP 알고리즘을 이용한 포즈 벡터 탐색 방법(Section 3.1)으로 S_s 를 S_r 에 정합할 수 있다. 병합된 포인트 클라우드 S 를 얻기 위해 추정된 포즈 파라미터들을 사용해 S_s 는 $\hat{S}_s$ 로 변환되고, S 는 S_r 과 $\hat{S}_s$ 의 합성으로 계산된다. S 를 새로운 참조 포인트 클라우드로 간주하면 새로운 인접 포인트 클라우드가 S 에 병합된다. 각 뷰의 포인트 클라우드에 대하여, 업데이트된 상기 포즈 벡터를 이용하여 거리(수학식2,3 참조)가 가장 가까운 포인트 클라우드 $S_r = S_0$ 로부터 가장 먼 포인트 클라우드 $S_r = S_f$ 가 병합될 때까지 이와 같은 각 대응 쌍의 병합이 반복된다.

[0101] 순차 병합 절차가 오류 누적을 방지할 수 없지만, 오류를 줄일 수 있도록 다음과 같은 처리할 수 있다. 한 라운드의 병합 후 S 가 처음의 S_0 과 $\hat{S}_f$ (S_f 가 S_0 좌표계로 변환된 것)의 합성으로 설정되어있는 S 와 함께 역순으로 절차를 시작한다.

[0102] 쌍별 병합 알고리즘은 대칭이 아니기 때문에 소스 포인트와 참조 포인트가 반전되면 다른 솔루션 결과가 나올 수 있다. 순서를 반대로 하면 알고리즘을 통해 반전된 입력 쌍에 대해 현재 포즈 파라미터를 업데이트되도록 탐색할 기회를 얻는다. 현재 포즈 파라미터가 정확하지 않은 경우 반전된 입력은 로컬 최소값에서 벗어나는 데 도움이 될 수 있다. S_0 에서 S_f 까지의 단일 순방향 경로가 정확하면 S_f 의 포즈 파라미터가 정확할 수 있다. 이 경우 잘못된 포인트 클라우드를 정확한 위치와 방향으로 S_f 에 병합하면 다른 역방향 경로의 오류를 줄일 수 있다. 그러나 S_f 에 대한 경로가 하나뿐이면 이러한 오류 수정을 기대하기 어렵다.

[0103] 하기의 알고리즘 3에는 제안된 멀티-뷰 포인트 클라우드 병합 알고리즘이 요약되어 있다. 일반성을 잃지 않고 0 번째 뷰를 참조 뷰로 가정한다. $ITER_{max}$ 는 본 발명에서 5로 설정된다. 포인트 클라우드를 병합한 후 위에서 기술한 바와 같은 깊이 벡터 탐색 방법(Section 3.2)을 적용하여 병합 오류를 더욱 줄일 수 있다. 이 단계에서 S_s 의 깊이 값을 조정하여 S_s 를 $S - \hat{S}_s$ 에 병합한다. 상기 깊이 벡터 탐색 프로세스를 기초로, 업데이트된 상기 깊이 값을 이용함으로써, 거리(수학식2,3 참조)가 가장 먼 포인트 클라우드 S_f 에서 시작하여 첫 번째 라운드에서 참조 포인트 클라우드 S_0 으로 끝난다. 이 순서는 가장 먼 포인트 클라우드가 가장 오류가 많다는 가정을 기반으로 한다. 멀티-뷰 포인트 클라우드들 중 포인트 클라우드를 병합하는 두 번째 이후 라운드에서는 처리 순서가 교대로 반전된다. 오류의 크기에 따라 이 순서는 바뀔 수 있다.

Algorithm 3: Multi-view point cloud merging algorithm

Input: Point clouds $\{S_i\}_{i=0}^L$ and their pose parameters $\{R_i, T_i\}_{i=1}^L$ with respect to S_0
Output: Merged point cloud S , refined pose parameters $\{R_i, T_i\}_{i=1}^L$

- 1 Sort view indices $i = 1, \dots, L$ from the nearest to the farthest from S_0 to attain an index set $\mathcal{I}$
- 2 Remove from $\mathcal{I}$ the last element f and reverse the order to attain $\mathcal{J}$
- 3 **for** $iter \leftarrow 1$ to $ITER_{max}$ **do**
- 4 $S \leftarrow S_0, \mathcal{G} \leftarrow \mathcal{I}$
- 5 **if** $iter$ is even **then**
- 6 $S \leftarrow S \cup \hat{S}_f, \mathcal{G} \leftarrow \mathcal{J}$
- 7 **end**
- 8 **for** $s \in \mathcal{G}$ **do**
- 9 Apply the method in Section 3.1 to S_s and S to refine R_s and T_s
- 10 Compute $\hat{S}_s$ using R_s and T_s
- 11 $S \leftarrow S \cup \hat{S}_s$
- 12 **end**
- 13 **end**

[0105]

[0106]

알고리즘 4에는 이와 같은 제안된 알고리즘이 요약되어 있다. 여기서 $ITER_{max}$ 는 본 발명에서 2로 설정된다.

Algorithm 4: Multi-view depth refinement algorithm

Input: Point clouds $\{S_i\}_{i=0}^L$ and their pose parameters $\{R_i, T_i\}_{i=1}^L$ with respect to S_0
Output: Merged point cloud S with refined depth values

- 1 Sort view indices $i = 1, \dots, L$ from the nearest to the farthest from S_0 to attain an index set $\mathcal{I}$
- 2 Reverse the order of $\mathcal{I}$ to attain $\mathcal{J}$
- 3 Append 0 to $\mathcal{J}$
- 4 **for** $iter \leftarrow 1$ to $ITER_{max}$ **do**
- 5 $\mathcal{G} \leftarrow \mathcal{J}$
- 6 **if** $iter$ is even **then**
- 7 $\mathcal{G} \leftarrow \mathcal{I}$
- 8 **end**
- 9 **for** $s \in \mathcal{G}$ **do**
- 10 $S' \leftarrow$ the union of $\{\hat{S}_i : i \neq s\}_{i=0}^L$
- 11 Apply the method in Section 3.2 to S_s and S' to refine the depth values of S_s
- 12 Compute $\hat{S}_s$
- 13 **end**
- 14 **end**
- 15 $S \leftarrow$ the union of $\{\hat{S}_i\}_{i=0}^L$

[0107]

[0109]

<합성 멀티-뷰 RGB-D 데이터 세트>

[0110]

먼저, 포즈에 따라 달라지는 인간 모델 데이터 세트의 그래픽 모델을 렌더링하여 합성 멀티-뷰 RGB-D 데이터 세트를 구축하여 메모리 등 데이터베이스에 저장하였다. 즉, 남성 (모델 0-199)과 여성 (모델 6800-6999) 외모를 사용하고 각 외모에서 서로 다른 포즈의 20 개의 메시 모델을 정기적으로 샘플링하였으며, 메시 모델의 높이는 1.7m로 설정했다. RGB-D 이미지 데이터는 도 2와 같이 RGB 컬러 카메라 이외에 두 개의 적외선(IR) 스테레오 카메라와 하나의 IR 프로젝터를 포함하는 시스템에 의해 수집되고 깊이 값이 획득된다고 가정한다. 이때, 깊이 값을 획득하기 위한 스테레오 매칭에 의해 얻어지는 불일치의 오차는 표준편차 $\sigma_d=0.5$ 픽셀 0-평균 가우시안 분포를 따른다고 가정한다. 깊이 오차는 그 표준 편차인 [수학식22]의 σ_z 가 깊이 값 Z 에 따라 달라지는 0-평균 가우스 분포를 대략 따른다.

[0111] [수학식22]

$$\sigma_Z = \frac{Z^2}{f_d B_d} \sigma_d$$

[0112]

[0113] 여기서 f_d 와 B_d 는 스테레오 카메라의 초점 거리와 기선(baseline) 길이이며 하기의 실제 매개 변수를 반영하여 780 및 26cm로 설정되었다. 깊이 이미지의 너비와 높이는 1468 및 1228 픽셀이고 이미지 중심 좌표 벡터는 (734, 614) 이었다.

[0114] 도 2는 (a) 깊이 이미지를 획득하기 위한, RGB 컬러 카메라, 두 개의 적외선(IR) 스테레오 카메라 및 하나의 IR 프로젝터를 포함하는 RGB-D 카메라 시스템을 예시하며, (b) 깊이 값 Z 에 따라 그 표준 편차 σ_z 의 예시이다.

[0115] 노이즈가 있는 깊이 이미지는 RGB-D 이미지를 형성하기 위해 내장 및 외부 카메라 매개 변수를 사용함에 의해 컬러 카메라 좌표계에서 앞으로 휘다(forward-warped). 컬러 카메라의 초점 거리는 900 픽셀이고 기준 IR 카메라와의 거리는 6.5cm이다. 컬러 이미지는 깊이 이미지와 너비, 높이 및 이미지 중심이 동일하다. 실제 깊이 이미지는 메쉬 모델을 컬러 이미지 좌표계에 직접 렌더링하여 얻는다.

[0116] 도 3은 멀티-뷰 이미지 획득을 위한 RGB-D 카메라의 위치 설정의 예시이다.

[0117] 도 3을 참조하면, RGB-D 카메라는 타원 상에서 30도 간격으로 배치되어 12 위치에서 이미지를 캡처할 수 있으며, 반장축 상의 3m, 반단축 상의 1.5m 거리에 배치될 수 있다.

[0118] 도 4는 RGB-D 카메라로 획득한 샘플 RGB-D 이미지를 보여준다. 도 4에서, 제1행과 제2행은 남성 모델의 컬러 이미지와 그 깊이 이미지, 제3행과 제4행은 여성 모델의 컬러 이미지와 그 깊이 이미지를 나타낸다. 여기서, 제 1,2,3,4컬럼은 도 3의 View 0, 3, 6, 9 위치의 이미지들이다.

[0119] 도 4와 같이, 깊이 이미지는 실제 깊이 이미지에서 관찰되는 것처럼 자기 가림(self-occlusions)으로 인해 누락된 데이터가 있다. 이외에도, 포인트 해상도는 모델과의 거리가 다르기 때문에 뷰들에서 고르지 않음을 알 수 있다.

[0120] RGB-D 카메라의 실질적인(gt, ground truth) 포즈 파라미터 $\{\mathbf{R}_{i,gt}, \mathbf{T}_{i,gt}\}_{i=1}^L$ 은 본 발명의 알고리즘에 입력으로 사용되는 초기 부정확한 포즈 파라미터를 생성하기 위해 조정된다. 회전 벡터(행렬) $\mathbf{R}_{i,gt}$ 를 조정하기 위해 [수학식23]과 같이 $\mathbf{R}_{i,gt}$ 에 임의의 회전 축에 대한 회전 행렬 βQ 를 미리 곱한 회전 벡터 $\mathbf{R}_i$ 를 이용한다. 역시 [수학식24]와 같이 실질적인 병진 벡터(행렬) $\mathbf{T}_{i,gt}$ 에 임의의 3D 벡터 βU 를 더한 병진 벡터 $\mathbf{T}_i$ 를 이용한다.

[0121] [수학식23]

$$\mathbf{R}_i = \Delta \mathbf{Q} \mathbf{R}_{i,gt}$$

[0122]

[0123] [수학식24]

$$\mathbf{T}_i = \mathbf{T}_{i,gt} + \Delta \mathbf{U}$$

[0124]

[0125] 메쉬 모델에서 얻은 각 RGB-D 이미지 세트에 5 개의 서로 다른 회전 및 병진 벡터들의 이와 같은 조정이 각각 회전 각도 2도 ~ 10도 및 병진 길이 5cm ~ 25cm로 적용되었다. 따라서 각 세트는 실험에서 10 가지 다른 방식으로 조정된다.

[0126] 도 5는 종래의 방법과 본 발명의 병합된 포인트 클라우드들을 비교하기 위한 렌더링된 이미지들의 예이다. 도 5에서, 제1, 3 행은 병합된 포인트 클라우드를 나타내고, 제2, 4행은 손 부분을 확대한 이미지이다.

[0127] 도 5와 같이, 종래의 점대 평면 비용 함수를 이용하는 ICP 알고리즘은 Ground Truth 포즈와 완전히 차이가 나며 시각적 품질이 떨어진다. 종래의 컬러 6D GICP 및 본 발명의 포즈 탐색 알고리즘(Ours (pose))은 Ground Truth 포즈와 시각적으로 유사한 이미지 정합을 달성한다. 나아가 본 발명의 깊이 탐색 알고리즘(Ours (pose+depth))은, 노이즈가 많은 포인트들을 표면쪽으로 이동하여 시각적으로 더 가깝게 정합이 이루어짐으로써, 시각적 품질을 향상시키고 정합 오류를 줄일 수 있음을 확인하였다.

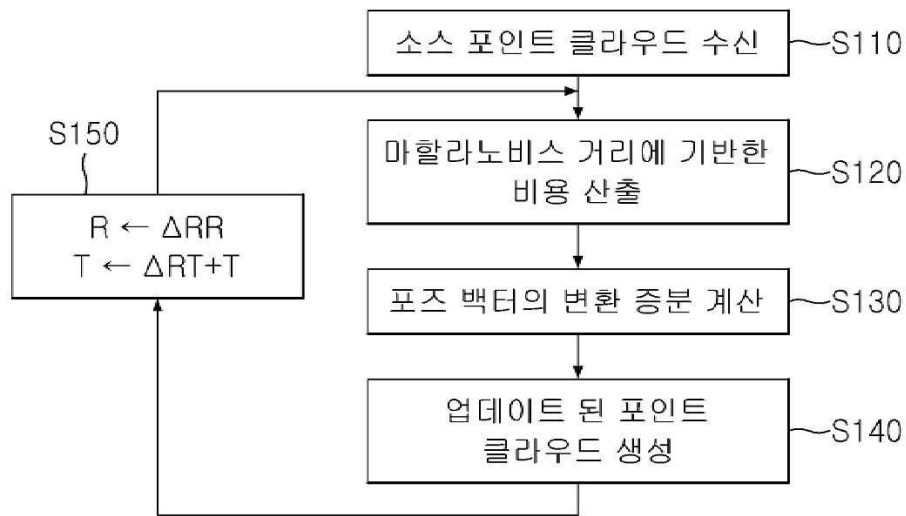
[0128] 도 6은 본 발명의 일 실시예에 따른 포인트 클라우드의 재구성을 처리하기 위한 상기 영상 처리 시스템의 구현

방법의 일례를 설명하기 위한 도면이다.

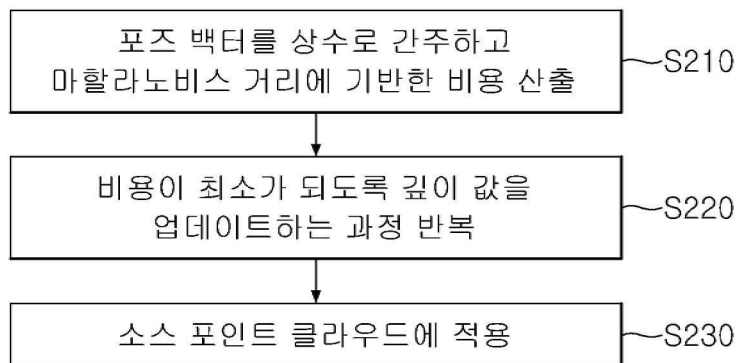
- [0129] 본 발명의 일 실시예에 따른 포인트 클라우드의 재구성을 처리하는 상기 영상 처리 시스템은, 하드웨어, 소프트웨어, 또는 이들의 결합으로 이루어질 수 있다. 예를 들어, 본 발명의 상기 영상 처리 시스템은 위와 같은 기능/단계/과정들을 수행하기 위한 적어도 하나의 프로세서를 갖는 도 6과 같은 컴퓨팅 시스템(1000) 또는 인터넷 상의 서버 형태로 구현될 수 있다.
- [0130] 컴퓨팅 시스템(1000)은 버스(1200)를 통해 연결되는 적어도 하나의 프로세서(1100), 메모리(1300), 사용자 인터페이스 입력 장치(1400), 사용자 인터페이스 출력 장치(1500), 스토리지(1600), 및 네트워크 인터페이스(1700)를 포함할 수 있다. 프로세서(1100)는 중앙 처리 장치(CPU) 또는 메모리(1300) 및/또는 스토리지(1600)에 저장된 명령어들에 대한 처리를 실행하는 반도체 장치일 수 있다. 메모리(1300) 및 스토리지(1600)는 다양한 종류의 휘발성 또는 불휘발성 저장 매체를 포함할 수 있다. 예를 들어, 메모리(1300)는 ROM(Read Only Memory)(1310) 및 RAM(Random Access Memory)(1320)을 포함할 수 있다.
- [0131] 또한, 네트워크 인터페이스(1700)는 스마트폰, 노트북 PC, 데스크탑 PC 등 사용자 단말에서의 유선 인터넷 통신이나 WiFi, WiBro 등 무선 인터넷 통신, WCDMA, LTE 등 이동통신을 지원하는 모뎀 등의 통신 모듈이나, 근거리 무선 통신 방식(예, 블루투스, 지그비, 와이파이 등)의 통신을 지원하는 모뎀 등의 통신모듈을 포함할 수 있다.
- [0132] 따라서, 본 명세서에 개시된 실시예들과 관련하여 설명된 방법 또는 알고리즘의 단계는 프로세서(1100)에 의해 실행되는 하드웨어, 소프트웨어 모듈, 또는 그 2 개의 결합으로 직접 구현될 수 있다. 소프트웨어 모듈은 RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터, 하드 디스크, 착탈형 디스크, CD-ROM과 같이 컴퓨터 등 장치로 판독 가능한 저장/기록 매체(즉, 메모리(1300) 및/또는 스토리지(1600))에 상주할 수도 있다. 예시적인 저장 매체는 프로세서(1100)에 커플링되며, 그 프로세서(1100)는 저장 매체로부터 정보(코드)를 판독할 수 있고 저장 매체에 정보(코드)를 기입할 수 있다. 다른 방법으로, 저장 매체는 프로세서(1100)와 일체형일 수도 있다. 프로세서 및 저장 매체는 주문형 집적회로(ASIC) 내에 상주할 수도 있다. ASIC는 사용자 단말기 내에 상주할 수도 있다. 다른 방법으로, 프로세서 및 저장 매체는 사용자 단말기 내에 개별 컴포넌트로서 상주할 수도 있다.
- [0133] 상술한 바와 같이, 본 발명에 따른 포인트 클라우드의 재구성을 처리하기 위한 상기 영상 처리 시스템은, 반복적인 K-최근접 포인트 알고리즘을 이용하되 계산 속도와 정확도 사이의 균형을 맞추기 위해 포인트가 K-최근접 포인트와 일치하도록 허용하고 소스/참조 포인트간 일치 확률을 K 쌍들에만 할당하되, 포즈 탐색은 3D 좌표와 포인트의 색상을 결합하여 6D 벡터 공간에서 수행함으로써, 사람 등 3D 이미지를 재구성하는 데 있어서, 소스 포인트 클라우드의 포인트가 K-최근접 포인트에 의해 참조 포인트 클라우드에 자연스럽게 매칭되고 비용을 최소화할 수 있다. 본 발명에서, 모든 중요한 결과를 통합하는 비용 함수는 안정성과 정확도를 향상시키기 위해 점대 점 및 점대 평면 거리에 대한 제곱의 가중치 합으로 정의된다. 본 발명에서는, 1) ICP 알고리즘을 통해 비용을 최소화하는 포즈 파라미터를 찾아 비용을 최소화할 수 있으며, 2) RGB-D 이미지에서 포인트 클라우드를 얻었다고 가정하고 RGB-D 카메라로 측정된 깊이 값에 오류가 있음을 고려하여 측정된 깊이 값을 변수로 사용하고, 포즈 파라미터를 상수로 처리해 포인트 클라우드들의 포인트간 정합이 더욱 긴밀하게 이루어지도록 깊이 값을 탐색해 비용을 최소화할 수 있다.
- [0134] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

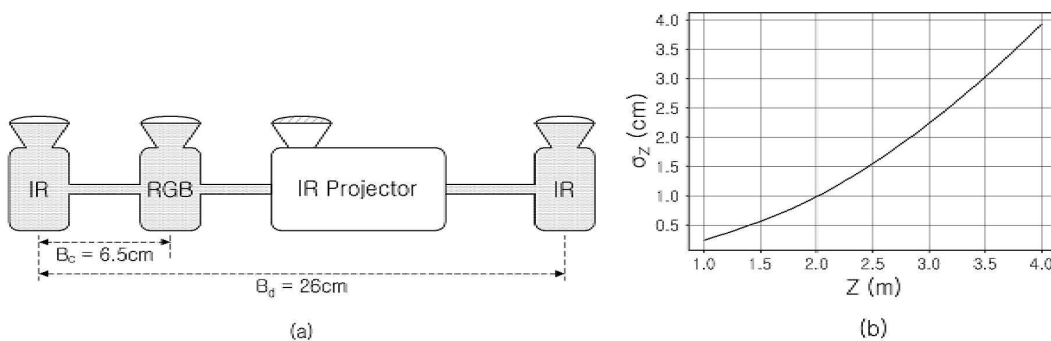
도면1a



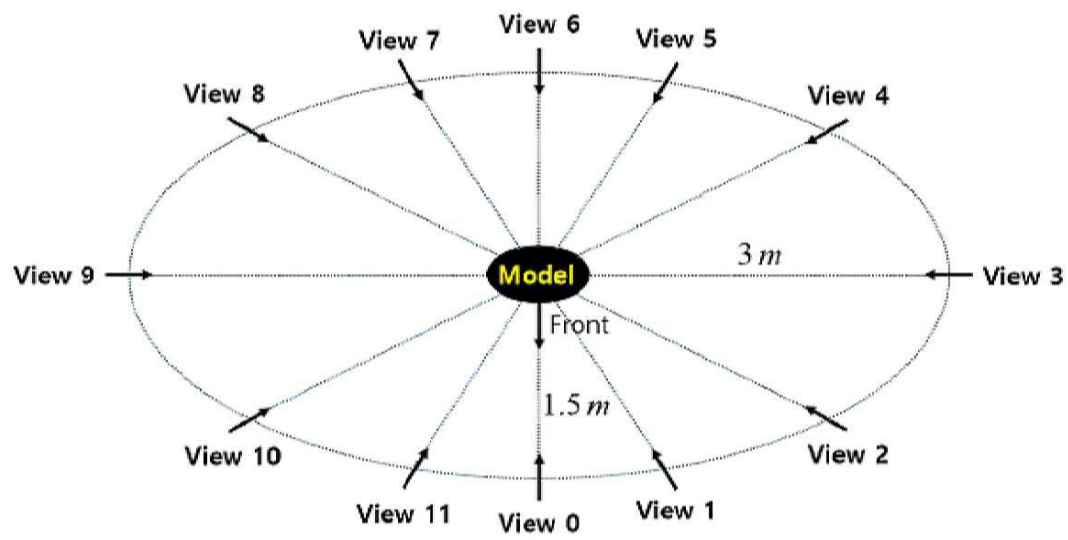
도면1b



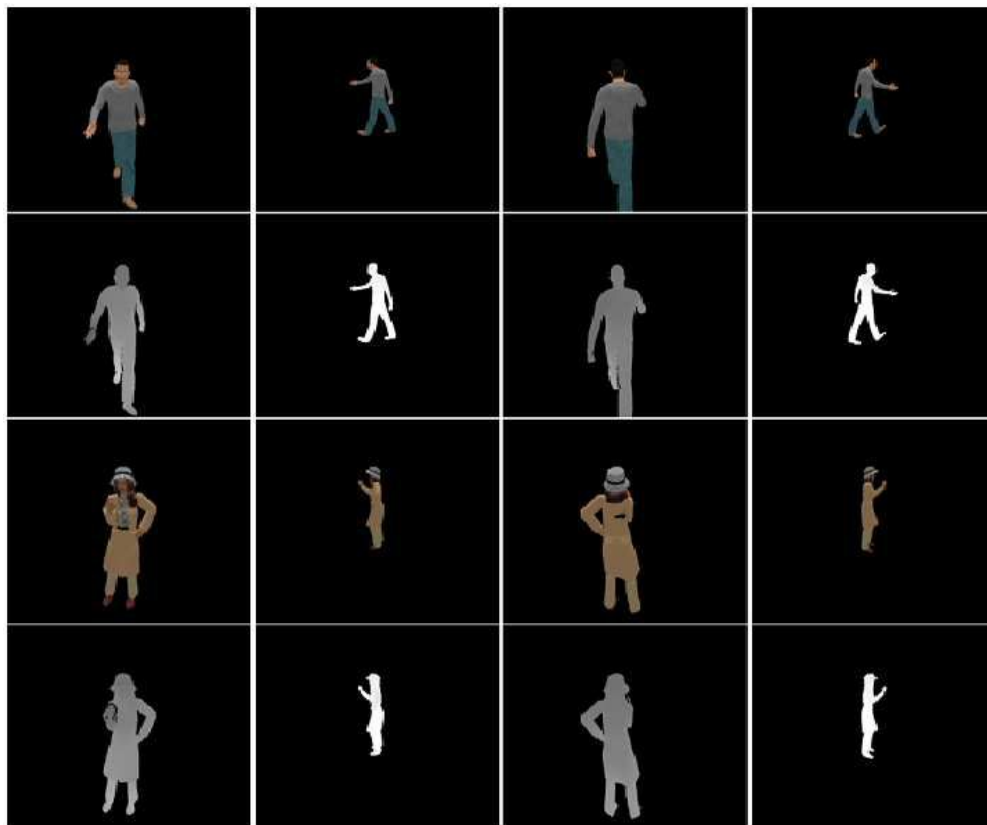
도면2



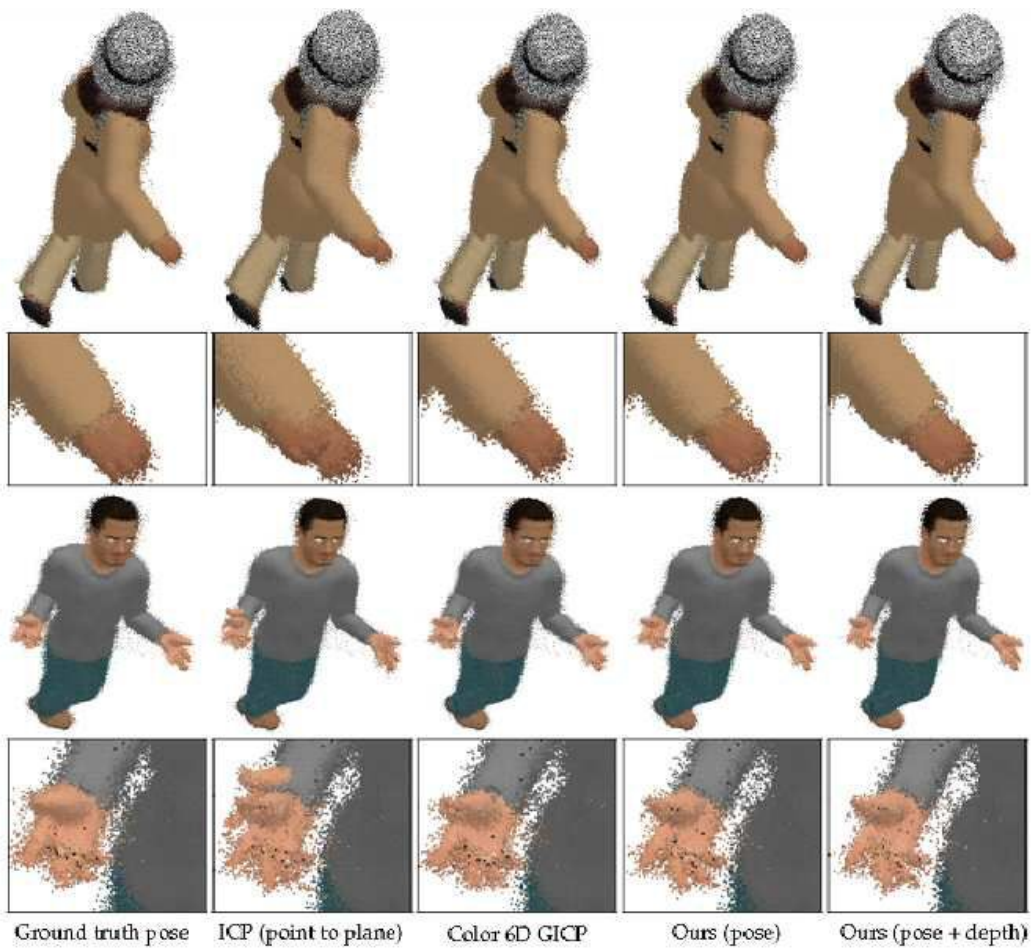
도면3



도면4



도면5



도면6

