



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0020210
(43) 공개일자 2017년02월22일

(51) 국제특허분류(Int. Cl.)
G06T 17/20 (2006.01) H04N 5/262 (2006.01)
(52) CPC특허분류
G06T 17/205 (2013.01)
H04N 5/2627 (2013.01)
(21) 출원번호 10-2016-0058779
(22) 출원일자 2016년05월13일
심사청구일자 없음
(30) 우선권주장
2015134345 2015년08월14일 러시아(RU)

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
삼수아로프 아르템 겐나디예비치
러시아, 모스크바 109044, 크루티스키 발, 3, 에프엘.291
류뎀트세바 마리아 미하일로프나
러시아, 모스크바 107076, 부호보스토바 2가, 7, 에프엘.24
(뒷면에 계속)
(74) 대리인
리엔목특허법인

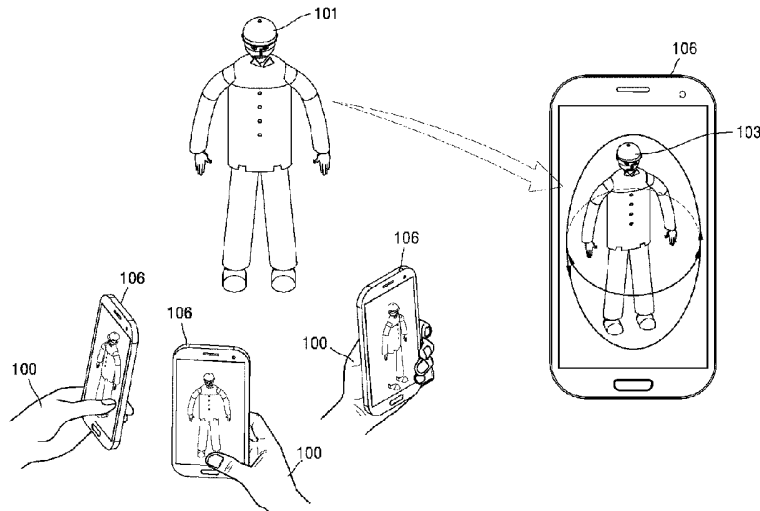
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 오브젝트의 3차원 모델을 구성하는 방법 및 장치

(57) 요약

오브젝트의 3D 모델을 구성하는 방법 및 장치는, 오브젝트를 스캐닝함으로써 오브젝트의 이미지들을 캡처하고, 캡처된 이미지들 각각에 대응되는 스캐닝 모듈의 위치들을 계산하고, 계산된 위치들에 대응되는 깊이 맵들을 추정하고, 깊이 맵들을 합성함으로써 오브젝트의 표면 메쉬를 생성하고, 표면 메쉬에 텍스처들을 매핑함으로써 오브젝트의 3D 모델을 구성하여 표시한다.

도면도



(52) CPC특허분류

G06T 2207/20221 (2013.01)

G06T 2215/16 (2013.01)

(72) 발명자

무라토프 올레그 파닐레비치

러시아, 모스크바 115408, 보리소프스키에 프루디
스트리트, 6, 빌딩, 1, 에프엘.144

체르노프 비탈리 블라디미로비치

러시아, 모스크바 123154, 카라미셰프스카야 냅,
56, 에프엘.67

슬린코 유리 바체슬라보비치

러시아, 모스크바 141411, 험키, 쿠킨스코예 쇼스,
7, 에프엘.137

부차 빅터 발렌티노비치

러시아, 모스크바 125581, 라보츠키나 스트리트,
34, 에프엘.314

명세서

청구범위

청구항 1

오브젝트의 3차원(3D) 모델을 구성하는 방법에 있어서,

상기 오브젝트 주위의 궤도를 따라 상기 오브젝트를 스캐닝함으로써 상기 오브젝트의 이미지들을 캡처하는 단계;

상기 캡처된 이미지들 각각에 대응되는 스캐닝 모듈의 위치들을 추정하는 단계;

상기 궤도 상의 적어도 둘 이상의 위치들에 기초하여 상기 스캐닝 모듈의 상기 추정된 위치들을 미세조정하는 단계;

상기 스캐닝 모듈의 상기 미세조정된 위치들에 대응되는 깊이 맵들을 추정하는 단계;

상기 추정된 깊이 맵들을 합성함으로써, 상기 오브젝트의 표면 메쉬를 생성하는 단계; 및

상기 생성된 표면 메쉬에 텍스처들을 매핑함으로써, 상기 오브젝트의 상기 3D 모델을 구성하여 표시하는 단계를 포함하는, 방법.

청구항 2

제 1 항에 있어서,

상기 궤도는 폐 루프(closed loop) 또는 개방 루프(opened loop)이고,

상기 궤도 상의 적어도 둘 이상의 상기 위치들은 상기 궤도의 루프 결합(loop closure)을 위한 루프 결합 점들(loop closure points)에 대응하는 상기 궤도의 시작 점 및 끝 점을 포함하는, 방법.

청구항 3

제 1 항에 있어서,

상기 이미지들을 캡처하는 단계는

SLAM(Simultaneous Localization and Mapping)을 이용하여 상기 오브젝트를 360도 스캐닝함으로써 상기 이미지들을 캡처하는, 방법.

청구항 4

제 1 항에 있어서,

상기 스캐닝 모듈의 위치들을 추정하는 단계는

상기 캡처된 이미지들 중 제 1 이미지를 이용하여 맵 초기화를 수행하는 단계;

코너 검출(corner detection)을 수행하는 FAST(Features from Accelerated Segment Test) 알고리즘을 이용하여 상기 제 1 이미지로부터 키폰트들을 추출하는 단계; 및

상기 추출된 키폰트들로 상기 캡처된 이미지들 각각에 대응되는 상기 스캐닝 모듈의 위치들을 추정하는 단계를 포함하는, 방법.

청구항 5

제 4 항에 있어서,

상기 스캐닝 모듈의 위치들을 추정하는 단계는

상기 캡처된 이미지들 각각이 키프레임에 해당되는지 여부를 판단하는 단계; 및

상기 키프레임으로 판단된 이미지를 상기 스캐닝 모듈의 상기 추정된 위치에 대응시켜 메모리에 저장하는 단계

를 더 포함하는, 방법.

청구항 6

제 5 항에 있어서,

상기 추정된 위치들을 미세조정하는 단계는

상기 키프레임에 대하여, 궤도 상의 루프 점들(loop points)를 획득하는 단계;

상기 루프 점들이 루프 결합 점들(loop closure points)인지 여부를 검사하는 단계; 및

상기 루프 결합 점들로 검사된 경우, 상기 루프 결합 점들을 이용하여 루프 결합을 수행한 결과에 기초하여 상기 추정된 위치들을 미세조정하는 단계를 포함하는, 방법.

청구항 7

제 1 항에 있어서,

상기 스캐닝 모듈의 위치들에 대한 상기 미세조정은

SLAM을 이용하여 상기 오브젝트가 스캐닝될 때 발생한 드리프트의 영향을 보상하기 위하여 수행되는, 방법.

청구항 8

제 1 항에 있어서,

상기 깊이 맵들을 추정하는 단계는

인접한 키프레임들을 포함하는 스테레오 쌍을 선택하는 단계; 및

저해상도 샘플링의 결과를 더 미세한(finer) 해상도의 샘플링에 의해 업-샘플링하는 피라미드 방식(pyramidal approach)을 이용하여, 상기 선택된 스테레오 쌍으로부터 깊이 맵들을 추정하는 단계를 포함하는, 방법.

청구항 9

제 1 항에 있어서,

상기 표면 메쉬를 생성하는 단계는

3D 복셀 TSDF(truncated signed distance function)를 이용하여 상기 추정된 깊이 맵들을 합성함으로써, 상기 오브젝트에 대한 복셀-투-옥트리(voxel-to-octree) 표현에 대응되는 상기 표면 메쉬를 생성하는, 방법.

청구항 10

제 1 항에 있어서,

상기 3D 모델을 구성하여 표시하는 단계는

상기 미세조정된 위치들 각각에서 상기 표면(surface) 메쉬의 각 면(face)이 가시적 또는 비가시적인지 검사하는 단계;

상기 검사 결과에 따라 상기 표면 메쉬의 가시적 면을 상기 캡처된 이미지들 중 하나와 대응되는 투영(projection)으로 텍스처링하는 단계; 및

상기 텍스처링된 메쉬를 생성함으로써, 상기 3D 모델을 구성하여 표시하는 단계를 포함하는, 방법.

청구항 11

오브젝트의 3차원(3D) 모델을 구성하는 장치에 있어서,

상기 오브젝트 주위의 궤도를 따라 상기 오브젝트를 스캐닝함으로써 상기 오브젝트의 이미지들을 캡처하는 스캐닝 모듈;

상기 캡처된 이미지들 각각에 대응되는 스캐닝 모듈의 위치들을 추정하고, 상기 궤도 상의 적어도 둘 이상의 위치들에 기초하여 상기 스캐닝 모듈의 상기 추정된 위치들을 미세조정하고, 상기 스캐닝 모듈의 상기 미세조정된

위치들에 대응되는 깊이 맵들을 추정하고, 상기 추정된 깊이 맵들을 합성함으로써 상기 오브젝트의 표면 메쉬를 생성하고, 상기 생성된 표면 메쉬에 텍스처들을 매핑함으로써 상기 오브젝트의 상기 3D 모델을 구성하는 적어도 하나의 프로세서; 및

상기 오브젝트의 상기 3D 모델을 표시하는 디스플레이를 포함하는, 장치.

청구항 12

제 11 항에 있어서,

상기 궤도는 폐 루프(closed loop) 또는 개방 루프(opened loop)이고,

상기 궤도 상의 적어도 둘 이상의 상기 위치들은 상기 궤도의 루프 결합(loop closure)을 위한 루프 결합 점들(loop closure points)에 대응하는 상기 궤도의 시작 점 및 끝 점을 포함하는, 장치.

청구항 13

제 11 항에 있어서,

상기 스캐닝 모듈은

SLAM(Simultaneous Localization and Mapping)을 이용하여 상기 오브젝트를 360도 스캐닝함으로써 상기 이미지들을 캡처하는, 장치.

청구항 14

제 11 항에 있어서,

상기 적어도 하나의 프로세서는

상기 캡처된 이미지들 중 제 1 이미지를 이용하여 맵 초기화를 수행하고, 코너 검출(corner detection)을 수행하는 FAST(Features from Accelerated Segment Test) 알고리즘을 이용하여 상기 제 1 이미지로부터 키폰트들을 추출하고, 상기 추출된 키폰트들로 상기 캡처된 이미지들 각각에 대응되는 상기 스캐닝 모듈의 위치들을 추정하는, 방법.

청구항 15

제 14 항에 있어서,

상기 적어도 하나의 프로세서는

상기 캡처된 이미지들 각각이 키프레임에 해당되는지 여부를 판단하고, 상기 키프레임으로 판단된 이미지를 상기 스캐닝 모듈의 상기 추정된 위치에 대응시켜 메모리에 저장하는, 장치.

청구항 16

제 15 항에 있어서,

상기 적어도 하나의 프로세서는

상기 키프레임에 대하여, 궤도 상의 루프 점들(loop points)를 획득하고, 상기 루프 점들이 루프 결합 점들(loop closure points)인지 여부를 검사하고, 상기 루프 결합 점들로 검사된 경우, 상기 루프 결합 점들을 이용하여 루프 결합을 수행한 결과에 기초하여 상기 추정된 위치들을 미세조정하는, 장치.

청구항 17

제 11 항에 있어서,

상기 스캐닝 모듈의 위치들에 대한 상기 미세조정은

SLAM을 이용하여 상기 오브젝트가 스캐닝될 때 발생된 드리프트의 영향을 보상하기 위하여 수행되는, 장치.

청구항 18

제 1 항에 있어서,

상기 적어도 하나의 프로세서는

인접한 키프레임들을 포함하는 스테레오 쌍을 선택하고, 저해상도 샘플링의 결과를 더 미세한(finier) 해상도의 샘플링에 의해 업-샘플링하는 피라미드 방식(pyramidal approach)을 이용하여 상기 선택된 스테레오 쌍으로부터 깊이 맵들을 추정하는, 장치.

청구항 19

제 11 항에 있어서,

상기 적어도 하나의 프로세서는

3D 복셀 TSDF(truncated signed distance function)를 이용하여 상기 추정된 깊이 맵들을 합성함으로써, 상기 오브젝트에 대한 복셀-투-옥트리(voxel-to-octree) 표현에 대응되는 상기 표면 메쉬를 생성하는, 장치.

청구항 20

제 11 항에 있어서,

상기 적어도 하나의 프로세서는

상기 미세조정된 위치들 각각에서 상기 표면(surface) 메쉬의 각 면(face)이 가시적 또는 비가시적인지 검사하고, 상기 검사 결과에 따라 상기 표면 메쉬의 가시적 면을 상기 캡처된 이미지들 중 하나와 대응되는 투영(projection)으로 텍스처링하고, 상기 텍스처링된 메쉬를 생성함으로써 상기 3D 모델을 구성하는, 장치.

발명의 설명

기술 분야

[0001] 이미지 프로세싱에 관한 것으로서, 특히 오브젝트를 360도로 스캐닝하여 생성된 이미지들을 이용하여 오브젝트를 3차원(3D) 모델로 구성하는 방법 및 장치에 관한다.

배경 기술

[0002] 3D 재구성(3D reconstruction)(또는 3D 구성)은, 서로 다른 시점들(points of view)로부터 획득된 오브젝트의 2차원 투영들(2D projections)을 이용하여, 오브젝트의 3D 표면에 관한 수학적 표현(mathematical representation)을 계산하는 프로세스를 의미한다.

[0003] 일반적으로, 3D 재구성은 이미지 촬영(capture)에 사용되는 다양한 광학 시스템에 기반한다. 광학 시스템은 예를 들어, 능동형 깊이 센서(active depth sensor) 또는 스테레오 카메라와 같은 깊이 카메라로 구현되거나, 또는 단안 카메라(monocular camera)로 구현될 수 있다.

[0004] 깊이 카메라는, 오브젝트 또는 오브젝트 주변 환경에 대한 깊이 맵을 실시간으로 획득할 수 있다. 이때, 능동형 깊이 센서의 경우에는 즉시 깊이 맵이 획득될 수 있고, 스테레오 카메라의 경우에는 한 쌍의 정류된 스테레오 이미지들(rectified stereo images)을 처리한 후에 깊이 맵이 획득될 수 있다. 깊이 맵에서 각각의 픽셀은, 오브젝트의 어느 한 점으로부터 카메라까지의 거리에 대응될 수 있다. 하지만, 깊이 맵의 획득을 위하여 깊이 카메라가 디바이스 내에 구비되기 위해서는, 깊이 센서와 관련된 많은 하드웨어 자원이 요구되므로, 디바이스의 하드웨어 구성이 복잡해 진다는 문제가 있다.

[0005] 실루엣 기반(silhouette-based)의 3D 재구성 방식(approach)은 단안 카메라 기반 방식들(monocular camera-based approaches)을 활용할 수 있다. 실루엣 기반 방식은, 오브젝트의 실루엣들을 추출한 후 오브젝트의 볼륨 구조를 형성한다. 실루엣 기반 방식에 따르면, 컬러 텍스처를 입힌 다각형 메쉬(polygonal mesh)가 생성된다. 하지만, 실루엣 기반 방식은, 모바일 디바이스와 같이 낮은 처리 능력을 갖는 디바이스 내에서 실루엣들의 추출 및 볼륨 데이터의 합성(fusion)과 관련된 복잡한 계산을 처리해야 된다는 한계를 갖는다. 또한 실루엣 기반 방식은, 실루엣 이미지들에서는 보이지 않는 오목한 부분들(concavities)을 복원(recover)할 수 없다.

[0006] 단안 카메라 기반 하드웨어를 이용한 다른 방식으로는, SfM(structure-from-motion) 기반 3D 재구성 방식을 예로 들 수 있다. SfM(structure-from-motion) 기반 방식은, 추출된 특징 점들(feature points)의 궤도들

(trajectories)을 이용하여, 카메라 움직임 및 오브젝트의 3D 위치들을 재구성한다. 하지만, SfM 기반 방식은 텍스처를 입히지 않고 3D 포인트 클라우드(3D point cloud) 또는 다각형 메쉬로만 오브젝트를 재구성하므로, 재구성된 3D 모델의 시각적인 외형(visual appearance)이 불충분하게 된다. 또한, SfM 기반 방식은, 정확성을 높이기 위해서는 많은 수의 특징 점들이 처리되어야 하므로, 많은 계산량이 요구될 수 있다는 문제가 있다.

[0007] 앞서 설명된 3D 재구성 방식들은, 많은 복잡한 계산들을 요구하므로, 스마트폰, 태블릿 등과 같이 제한된 처리 성능을 갖는 모바일 플랫폼에서 구현되는 것은 불가능할(infeasible) 수 있다. 한편, 모바일 플랫폼의 처리 성능의 한계를 극복하고자 클라우드 기반 데이터 프로세싱(cloud-based data processing)이 사용될 수 있으나, 모바일 플랫폼이 언제나 네트워크에 접근 가능하여야 한다는 불편함이 있을 수 있고, 또한 앞서 설명된 방식들보다 더 많은 시간이 소요될 수 있다는 문제가 있다.

[0008] 그러므로, 보다 적은 시간 및 적절한 계산량으로 추가적인 하드웨어 자원의 도움 없이 오브젝트의 3D 재구성을 효율적으로 수행할 수 있는 기술이 요구된다.

발명의 내용

해결하려는 과제

[0009] 오브젝트의 3차원 모델을 구성하는 방법 및 장치를 제공하는데 있다. 또한, 이 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체를 제공하는 데 있다. 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 이하의 실시예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제의 해결 수단

[0010] 일 측면에 따르면, 오브젝트의 3차원(3D) 모델을 구성하는 방법은, 상기 오브젝트 주위의 궤도를 따라 상기 오브젝트를 스캐닝함으로써 상기 오브젝트의 이미지들을 캡처하는 단계; 상기 캡처된 이미지들 각각에 대응되는 스캐닝 모듈의 위치들을 추정하는 단계; 상기 궤도 상의 적어도 둘 이상의 위치들에 기초하여 상기 스캐닝 모듈의 상기 추정된 위치들을 미세조정하는 단계; 상기 스캐닝 모듈의 상기 미세조정된 위치들에 대응되는 깊이 맵들을 추정하는 단계; 상기 추정된 깊이 맵들을 합성함으로써, 상기 오브젝트의 표면 메쉬를 생성하는 단계; 및 상기 생성된 표면 메쉬에 텍스처들을 매핑함으로써, 상기 오브젝트의 상기 3D 모델을 구성하여 표시하는 단계를 포함한다.

[0011] 다른 측면에 따르면, 오브젝트의 3차원(3D) 모델을 구성하는 장치는, 상기 오브젝트 주위의 궤도를 따라 상기 오브젝트를 스캐닝함으로써 상기 오브젝트의 이미지들을 캡처하는 스캐닝 모듈; 상기 캡처된 이미지들 각각에 대응되는 스캐닝 모듈의 위치들을 추정하고, 상기 궤도 상의 적어도 둘 이상의 위치들에 기초하여 상기 스캐닝 모듈의 상기 추정된 위치들을 미세조정하고, 상기 스캐닝 모듈의 상기 미세조정된 위치들에 대응되는 깊이 맵들을 추정하고, 상기 추정된 깊이 맵들을 합성함으로써 상기 오브젝트의 표면 메쉬를 생성하고, 상기 생성된 표면 메쉬에 텍스처들을 매핑함으로써 상기 오브젝트의 상기 3D 모델을 구성하는 적어도 하나의 프로세서; 및 상기 오브젝트의 상기 3D 모델을 표시하는 디스플레이를 포함한다.

도면의 간단한 설명

- [0012] 도 1a는 일 실시예에 따라 디바이스에 표시된 오브젝트의 3D 모델을 설명하기 위한 도면이다.
- 도 1b는 일 실시예에 따라 3D 재구성을 위하여 디바이스로 오브젝트를 스캐닝하는 것을 설명하기 위한 도면이다.
- 도 2a는 일 실시예에 따른 3D 재구성 프로세스를 설명하기 위한 도면이다.
- 도 2b는 일 실시예에 따른 3D 재구성 프로세스의 흐름도이다.
- 도 3은 일 실시예에 따라 카메라 위치를 추정하는 것을 설명하기 위한 도면이다.
- 도 4는 일 실시예에 따른 새로운 이미지를 프로세싱하여 카메라 위치를 추정하는 것을 설명하기 위한 도면이다.
- 도 5는 일 실시예에 따른 스페이스 이미지 정렬을 설명하기 위한 도면이다.
- 도 6은 일 실시예에 따른 스페이스 특징 정렬을 설명하기 위한 도면이다.

- 도 7은 일 실시예에 따른 로컬 번들 조정을 설명하기 위한 도면이다.
- 도 8은 일 실시예에 따른 깊이 필터들의 업데이트를 설명하기 위한 도면이다.
- 도 9는 일 실시예에 따른 카메라 위치 미세조정을 설명하기 위한 도면이다.
- 도 10은 일 실시예에 따른 깊이 맵 추정을 설명하기 위한 도면이다.
- 도 11은 일 실시예에 따른 깊이 맵 합성을 설명하기 위한 도면이다.
- 도 12는 일 실시예에 따른 표면 재구성을 설명하기 위한 도면이다.
- 도 13은 일 실시예에 따른 텍스처 매핑을 설명하기 위한 도면이다.
- 도 14는 일 실시예에 따른 z-버퍼 면 가시성 검사(z-buffer face visibility check)를 설명하기 위한 도면이다.
- 도 15는 일 실시예에 따른 면 라벨링(face labeling)을 설명하기 위한 도면이다.
- 도 16은 일 실시예에 따른 홀 필링(hole filling)을 설명하기 위한 도면이다.
- 도 17은 일 실시예에 따른 블록의 구조(structure of block)를 도시한 도면이다.
- 도 18은 일 실시예에 따른 디바이스의 하드웨어 구성을 도시한 도면이다.
- 도 19는 일 실시예에 따른 오브젝트의 3차원 모델을 재구성하는 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0013] 본 실시예들에서 사용되는 용어에 대해 간략히 설명하고, 본 실시예들에 대해 구체적으로 설명하기로 한다.
- [0014] 본 실시예들에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 관례, 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 실시예들의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 실시예들에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 실시예들의 전반에 걸친 내용을 토대로 정의되어야 한다.
- [0015] 본 실시예들 전체에서 어떤 부분이 어떤 구성요소를 “포함” 한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있음을 의미한다. 또한, 실시예들에 기재된 “...부”, “모듈” 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다.
- [0016] 본 실시예들에서 사용되는 “구성된다” 또는 “포함한다” 등의 용어는 명세서 상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.
- [0017] 이하 첨부된 도면을 참조하면서 오로지 예시를 위한 실시예들을 상세히 설명하기로 한다. 하기 실시예들은 발명을 구체화하기 위한 것일 뿐 발명의 권리 범위를 제한하거나 한정하는 것이 아님은 물론이다. 상세한 설명 및 실시예들로부터 발명이 속하는 기술분야의 전문가가 용이하게 유추할 수 있는 것은 발명의 권리범위에 속하는 것으로 해석된다.
- [0018] 도 1a는 일 실시예에 따라 디바이스에 표시된 오브젝트의 3D 모델을 설명하기 위한 도면이다.
- [0019] 도 1a를 참고하면, 사용자(100)는 사람 모형의 오브젝트(101)에 대한 3D 모델을 재구성하기 위하여, 사람 모형의 오브젝트(101)의 외형 이미지들을, 디바이스(106)를 이용하여 캡처한다. 사용자(100)는 디바이스(106)의 스캐닝 모듈(예를 들어, 카메라)이 오브젝트(101)를 향하도록 하며 오브젝트(101)의 전, 후, 좌, 우로 이동하면서, 오브젝트(101)의 이미지들을 획득한다.
- [0020] 디바이스(106)는 내부의 프로세서들(예를 들어, CPU(central processing unit) 또는 GPU(graphic processing unit))을 이용하여, 획득된 이미지들로부터 오브젝트 데이터를 추출하고, 추출된 오브젝트 데이터에 대한 재구성 프로세스를 수행함으로써, 3D 모델(103)을 생성한다. 생성된 3D 모델(103)은 실제 오브젝트(101)와

유사하게, 텍스처링된 메쉬(textured mesh)의 표현(representation)으로 재구성될 수 있다. 한편, 본 실시예들에서 "재구성(reconstruction)"의 용어는 실제하는 어느 오브젝트를 디스플레이 화면에 표시 가능한 가상의 3D 모델 또는 텍스처링된 메쉬 등으로 처리하는 것을 의미하고, 3D 모델 또는 텍스처링된 메쉬의 "생성(generation)" 또는 "구성(construction)"과 같은 유사한 용어들로도 대체되어 호칭될 수 있다.

- [0021] 재구성된 3D 모델(103)은 디바이스(106)의 디스플레이 화면에 표시된다. 사용자(100)는 디스플레이 화면을 터치하여, 표시된 3D 모델(103)을 상하좌우로 둘러면서 오브젝트(101)의 재구성된 3D 모델(103)을 확인할 수 있다.
- [0022] 도 1b는 일 실시예에 따라 3D 재구성을 위하여 디바이스로 오브젝트를 스캐닝하는 것을 설명하기 위한 도면이다.
- [0023] 도 1b를 참고하면, 디바이스(106)의 사용자(100)가 사람 모형의 오브젝트(102)에 대한 3D 모델을 재구성하고자 하는 경우를 가정하여 설명하도록 한다.
- [0024] 3D 재구성을 위한 프로세스는, 디바이스(106)의 스캐닝 모듈에 의해 오브젝트(102)를 스캐닝하는 것으로 시작된다. 스캐닝 모듈은, 모바일 전화기, 태블릿과 같은 휴대용(hand-held) 디바이스(106)에 구비된 카메라에 해당될 수 있다. 스캐닝 모듈은, 예를 들어 단안(monocular) 카메라, 멀티뷰(multi-view) 스테레오 카메라, 깊이 센서, 또는 이들의 조합으로 구현될 수 있으나, 스캐닝 모듈의 종류는 이에 제한되지 않는다.
- [0025] 오브젝트(102)는 스캐닝의 편의를 위하여 받침대(104) 위에 놓여질 수 있다. 오브젝트(102)가 스캔되는 동안, 디바이스(106)의 스캐닝 모듈은 항상 오브젝트(102) 쪽을 향하고 있는 것이 바람직하다. 이때, 사용자(100)는 디바이스(106)의 스캐닝 모듈이 오브젝트(102)를 향하도록 하면서 궤도(trajectory)(108)를 따라 움직인다. 비록, 도 1b에서는 궤도(108)가 반시계 방향의 원형 궤도인 것으로 도시되어 있으나, 궤도(108)는 이에 제한되지 않고 달라질 수 있다. 예를 들어, 궤도(108)는 폐 루프(closed loop) 형태이거나, 또는 개방 루프(opened loop) 형태일 수 있다. 또한 궤도(108)는 원형(circular shape)이거나 또는 다른 임의의 형태일 수 있다. 사용자(100)는 오브젝트(102)를 스캐닝하는 동안, 시계 방향으로 움직이거나, 반시계 방향으로 움직이거나, 또는 시계 방향 및 반시계 방향을 함께 움직이면서 궤도(108)를 형성할 수 있다.
- [0026] 오브젝트(102)의 스캐닝이 완료된 후에, 사용자(100)는 오브젝트(102)의 이미지들을 캡처하는 것을 멈춘다. 즉, 사용자(100)가 궤도(108)를 한 바퀴 돈 후에, 디바이스(106)는 오브젝트(102)의 스캐닝을 멈춘다. 이미지들의 캡처가 완료되면, 디바이스(106)는 캡처된 이미지들을 처리하기 위한 소프트웨어를 실행하여 후처리 스테이지(post-processing stage)를 수행한다. 후처리 스테이지의 수행 결과는 텍스처링된 메쉬(textured mesh)로서, 오브젝트(102)의 3D 모델일 수 있다. 후처리 스테이지에서 수행되는 연산들에 대해서는 이하 해당 부분에서 보다 상세하게 설명하도록 한다. 최종적으로, 텍스처링된 메쉬(즉, 3D 재구성된 3D 모델)는 디바이스(106)에 저장된다.
- [0027] 도 2a는 일 실시예에 따른 3D 재구성 프로세스를 설명하기 위한 도면이다.
- [0028] 도 2a를 참고하면, 3D 재구성 프로세스는 실시간(real-time) 스테이지(21) 및 후처리(post-processing) 스테이지(22)를 포함한다.
- [0029] 실시간 스테이지(21)에서, 디바이스(106)는 오브젝트(102)의 주변의 궤도를 따라 사용자에 의해 이동되면서 오브젝트(102)의 이미지들(201)을 캡처한다. 캡처된 이미지들(201) 각각에는, 오브젝트(102) 및 배경에 대한 정보가 포함되어 있다.
- [0030] 후처리 스테이지(22)에서, 디바이스(106)는 우선, 이미지들(201)에 포함된 오브젝트(102)에 관한 깊이 맵(205)을 생성한다. 여러 장의 이미지들(201)은 서로 다른 카메라 방향(orientation)에서 캡처되었으므로, 깊이 맵(205)은 각 카메라 방향에서의 오브젝트(102)의 깊이에 대한 정보를 포함하고 있을 수 있다. 다음으로, 디바이스(106)는 각 카메라 방향에 대응되는 각 깊이 맵(205)을 합성함으로써, 오브젝트(102)에 관한 3D 메쉬(211)의 표현을 재구성한다. 마지막으로, 디바이스(106)는 이미지들(201)로부터 획득된 오브젝트(102)의 텍스처 정보에 기초하여 3D 메쉬(211)를 텍스처링함으로써, 텍스처링된 3D 메쉬(214)를 생성한다. 텍스처링된 3D 메쉬(214)는 오브젝트의 3D 재구성된 3D 모델의 표현이다.
- [0031] 도 2b는 일 실시예에 따른 3D 재구성 프로세스의 흐름도이다.
- [0032] 도 2b를 참고하면, 3D 재구성 프로세스는 실시간 스테이지(21) 및 후처리 스테이지(22)의 2개의 메인 스테이지들을 포함할 수 있다.

- [0033] 실시간 스테이지(21)는 도 1b에서 설명되었던 오브젝트(102)에 대한 이미지 캡처를 수행하는 서브 스테이지(200) 및 캡처된 이미지를 이용하여 카메라 위치들을 추정하는 서브 스테이지(202)를 포함한다.
- [0034] 추정된 카메라 위치는 서로 다른 파라미터들의 세트로 구체화될 수 있다. 파라미터들의 세트는, 공간에서의 카메라 좌표들 및 카메라 방향(orientation)을 포함할 수 있다. 카메라 방향은 요(yaw), 피치(pitch) 및 롤(roll) 각도들로 정의될 수 있다. 서브 스테이지(202)는 SLAM(Simultaneous Localization and Mapping) 시스템을 이용하여 수행될 수 있다. SLAM 시스템에 의하면, 카메라 좌표들 및 카메라 방향에 관한 파라미터들을 이용하여, 카메라의 궤도뿐만 아니라 매 순간의 카메라의 방향을 추정하는 것이 가능하게 된다. SLAM 시스템을 이용하여, 변환(translation), 회전(rotation), 흔들림(shaking movements) 등과 같은 어떠한 캡처 조건에서도 강인한(robust) 카메라 위치를 추정할 수 있게 된다. 실시간 스테이지(21)는 디바이스(106)의 CPU에 의해 수행될 수 있다.
- [0035] 실시간 스테이지(21)가 완료된 후에는, 후처리 스테이지(22)가 수행된다. 후처리 스테이지(22)는 디바이스(106)의 CPU 및 GPU 모두에서 수행될 수 있고, 대개 수 초가 소요될 수 있다.
- [0036] 앞서 설명된 SLAM 시스템은 보통, 궤도 상에서 급격한 드리프트(significant drift)를 할 수 있는 것으로 알려져 있다. 이에 따라, 완전한(complete) 원형 궤도가 불완전한(incomplete) 원호인 것으로 추정될 수도 있다. 따라서, 서브 스테이지(204)에서는, 추정된 카메라 위치들(카메라의 궤도 및 방향)의 정확도를 높이기 위하여, 추정된 카메라 위치들이 미세조정(refine)된다. 미세조정을 위하여, 궤도 상의 시작 점(start point) 및 끝 점(end point)에 관련된 모든 이용 가능한(available) 정보들이 사용될 수 있다.
- [0037] 카메라 위치들은, 단계적 방식(coarse-to-fine manner)을 이용하여 두 단계들로 추정될 수 있다. 거친 추정(coarse estimation)은 오브젝트(102)를 스캐닝하는 동안 SLAM 시스템을 이용하여 수행된다. 이후에, SLAM 시스템의 설정 특성들(configuration peculiarities)에 기인한 드리프트에 대한 보상(compensation)을 위하여, 궤도의 시작과 끝에 관한 정보들에 기초하여, 거친-추정된 카메라 위치들이 미세조정됨으로써, 미세 추정(fine estimation)이 수행된다.
- [0038] 서브 스테이지(206)에서, 미세조정된 카메라 위치들과 캡처된 이미지들에 기초하여, 깊이 맵들이 생성된다.
- [0039] 서브 스테이지(208)에서, 서로 다른 카메라 위치들로부터 획득된 깊이 맵들이 합성(fuse)된다.
- [0040] 서브 스테이지(210)에서, 깊이 맵들이 합성됨으로써 표면 메쉬(surface mesh)(즉, 표면 재구성(surface reconstruction))가 생성된다. 표면 메쉬는 오브젝트(102)의 3D 표현(3D representation)이다.
- [0041] 서브 스테이지(212)에서, 표면 메쉬 위에 텍스처들이 매핑된다. 표면 메쉬 상에 텍스처들이 매핑됨으로써, 오브젝트(102)의 3D 재구성 결과인 텍스처링된 메쉬(texturized mesh)(또는 3D 모델)(214)가 생성된다. 텍스처링된 메쉬는 디바이스(106)의 디스플레이를 통해 출력될 수 있다.
- [0042] 고속의(fast) 3D 모델 재구성은 디바이스(106)의 CPU 및 GPU에 의해서, 고속으로(high-speed) 깊이 맵을 추출하고 깊이 맵을 합성함으로써 수행될 수 있다. 또한, 고품질의 3D 모델 재구성은, 정확한 표면 재구성(surface reconstruction) 및 텍스처 매핑에 의해 제공될 수 있다.
- [0043] 이하에서는 3D 재구성 프로세스에서의 서브 스테이지들(200 내지 212) 각각에 대하여 보다 상세하게 설명하도록 한다.
- [0044] 도 3은 일 실시예에 따라 카메라 위치를 추정하는 것을 설명하기 위한 도면이다. 도 3에 도시된 카메라 위치 추정은 도 2b의 서브 스테이지(202)와 관련될 수 있다.
- [0045] 카메라 위치 추정은, 제 1 이미지(300)를 이용한 맵 초기화(map initialization)(302)를 수행함으로써 시작될 수 있다. 맵 초기화(302)는 카메라 위치 추정의 초기화를 의미한다. 맵 초기화(302)를 위한 월드 레퍼런스 프레임(world reference frame)은 제 1 이미지(300)의 레퍼런스 프레임과 동일하게 할당된다. 제 1 이미지(300)로부터 키포인트들(keypoints)을 추출하기 위하여, 코너 검출(corner detection)을 수행하는 FAST(Features from Accelerated Segment Test) 알고리즘이 이용될 수 있다. FAST 알고리즘에 대해서는, 예를 들어 Edward Rosten 및 Tom Drummond의 "Fusing points and lines for high performance tracking", In Computer Vision, 2005, ICCV 2005, Tenth IEEE International Conference, volume 2, pages 1508-1515, IEEE, 2005에서 설명되어 있다. 추출된 키포인트들 각각에 대해서는 평균 장면 깊이(mean scene depth)가 할당된다. 맵 데이터베이스(304)는 FAST 알고리즘에 의해 추출된 키포인트들과 함께, 할당된 평균 장면을 저장한다. 맵 데이터베이스(30

4)에서 키포인트들은, 제 1 이미지(300)의 레퍼런스 프레임과 동일한 월드 레퍼런스 프레임에 저장된다.

- [0046] 디바이스(106)의 카메라(즉, 스캐닝 모듈)에 의해 캡처된 모든 일련의 이미지들(306)은, 데이터베이스(304)의 맵들과 함께, 어느 프레임들(또는 이미지들)이 추적(tracking) 및 3D 재구성을 위한 키프레임들(keyframes)에 해당되는지를 판단하기 위하여 사용된다. 키프레임들에 대응되는 카메라 위치들은, 데이터베이스(304)의 맵들을 업데이트함으로써 추정될 수 있다(308). 도 2b의 서브 스테이지(202)의 결과는 키프레임들 및 추정된 카메라 위치들(310)이다.
- [0047] 도 4는 일 실시예에 따른 새로운 이미지(404)를 프로세싱하여 카메라 위치를 추정하는 것을 설명하기 위한 도면이다.
- [0048] 도 4를 참고하면, 카메라 위치 추정의 알고리즘은, 움직임 추정(motion estimation) 스테이지(41) 및 검사(checking) 스테이지(42)를 포함할 수 있다. 한편, 도 3에서 설명된 SLAM 시스템은 도 4의 카메라 위치 추정의 알고리즘에도 사용될 수 있다. 카메라 위치 추정의 알고리즘을 통해, 카메라의 궤도뿐만 아니라 매 순간에서의 카메라 방향이 추정될 수 있다.
- [0049] 움직임 추정 스테이지(41)는, 스파스 이미지 정렬(sparse image alignment)을 위한 서브 스테이지(400), 스파스 특징 정렬(sparse feature alignment)을 위한 서브 스테이지(402) 및 로컬 번들 조정(local bundle adjustment)을 위한 서브 스테이지(408)를 포함한다.
- [0050] 스파스 이미지 정렬을 위한 서브 스테이지(400)에서, 맵 데이터베이스(304)에 저장된 맵의 키포인트들에 기초하여, 새로운 이미지(404)에 대응되는 카메라 위치가 추정된다. 맵의 키포인트들은 새로운 이미지(404)에 투영되고, 새로운 이미지(404)에 대응되는 카메라 위치는 Direct Image Alignment 알고리즘을 이용하여 추정된다.
- [0051] 스파스 특징 정렬을 위한 서브 스테이지(402)에서, 새로운 이미지(404) 및 그에 대응되는 카메라 위치가 이용된다. 맵의 키포인트들이 새로운 이미지(404)에 투영되고, Lucas Kanade tracker (LKT 또는 KLT)를 이용하여 새로운 이미지(404)에서의 각 포인트의 위치가 미세조정된다. LKT는 기울기 하강(gradient descent)을 이용하여 포인트의 투영(projection) 주위의 패치(patch)(사각형 영역)를 추적한다. 예를 들어, Lucas Kanade tracker (LKT)에 대해서는, Simon Baker 및 Iain Matthews의 "Lucas-kanade 20 years on: A unifying framework", International journal of computer vision, 56(3):221-255, 2004에 설명되어 있다.
- [0052] 로컬 번들 조정을 위한 서브 스테이지(408)에서, 새로운 이미지(404), 새로운 이미지(404)에 대응되는 카메라 위치 및 맵 포인트들의 리스트가, 새로운 이미지(404)의 미세조정된 포인트 위치들과 함께 사용된다. 카메라 위치는, 서브 스테이지(400)에서 획득된 초기 근사값결정(initial approximation), 맵 데이터베이스(304)에 저장된 포인트 좌표들 및 서브 스테이지(402)에서 획득된 미세조정된 포인트 위치들에 기초하여 계산된다. 이는 전형적인 Perspective-n-Point problem(PnP 문제)이다. 계산된 카메라 위치는 새로운 이미지(404)에 할당되고, 추가적인 변경 없이 다른 스테이지(또는 서브 스테이지)에서 사용될 수 있다. 카메라 위치의 미세조정(refinement) 이후에, 맵 포인트들의 3D 월드 좌표들 또한 미세조정될 수 있다.
- [0053] 서브 스테이지(408)에서 처리가 완료된 새로운 이미지(404)는, 서브 스테이지들(400 및 402)에서 추가적으로 이용되기 위하여 마지막(last) 카메라 이미지(406)로 설정될 수 있다.
- [0054] 새로운 이미지(404) 및 새로운 이미지(404)에 대응되는 카메라 위치는 이미지 큐(410)에 저장된다. 이미지 큐(410)의 이미지들은 이후의 검사 스테이지(42)에서 사용된다.
- [0055] 검사 스테이지(42)의 서브 스테이지(412)에서, 이미지 큐(410)로부터 입력되는 이미지(예를 들어, 새로운 이미지(404))가 키프레임인지 아닌지 여부가 검사된다.
- [0056] 새로운 이미지(404)가 키프레임인 경우, 새로운 이미지(404)는 대응되는 카메라 위치들과 함께 키프레임들(310)의 리스트에 저장된다. 그리고 나서, 서브 스테이지(414)에서, 새로운 이미지(404)에 대한 특징 추출(feature extraction)이 수행된다.
- [0057] 서브 스테이지(420) 단계에서, 현재의 평균 장면 깊이와 동일한 깊이 기대치(depth expectation)가 깊이-필터링(depth-filtering)을 위한 깊이-필터들과 함께, 각각의 특징점에 대해 생성되고(create), 능동형 깊이-필터들(active depth-filters)의 리스트(422)에 저장된다. 깊이-필터링에 관한 예시에 대해서는 예를 들어, George Vogiatzis 및 Carlos Hernandez의 "Video-based, real-time multi-view stereo" Image and Vision Computing, 29(7):434-441, 2011. 등에 설명되어 있다.

- [0058] 360도의 스캐닝 조건을 만족시키기 위하여, 깊이-필터들은 이전의 키프레임들로부터 연장된다(prolongated). 아직 처리되지 않은, 깊이-필터들의 패치들은 마지막 키프레임 상에 정렬되고, 깊이-필터의 상태가 변경되지 않는 동안에 새로운 패치는 각각의 깊이-필터에 결합된다(bind).
- [0059] 새로운 이미지(404)가 키프레임이 아닌 경우, 깊이-필터들을 업데이트하기 위한 서브 스테이지(416)가 수행된다. 서브 스테이지(416)에서, 리스트(422)에서의 특징점에 새로운 이미지(404)에서의 점을 정렬하고자, 에피폴라 라인(epipolar line) 주변에 대한 검색이 수행된다. 정렬이 완료되면, 점 깊이(point depth)가 삼각측량(triangulation) 기술들을 이용하여 계산된다. 점 깊이는 대응되는 해당 점의 깊이-필터에 등록된다. 그리고 나서, 서브 스테이지(418)에서, 각각의 깊이 필터는 병합(convergence)을 위해 검사된다(check). 만약, 깊이-필터가 병합되면(예를 들어, 깊이 변동(depth variation)이 충분히 작은 경우), 연관된 점이 맵 데이터베이스(304)에 추가되고 능동형 깊이-필터들의 리스트(422)로부터 제외된다.
- [0060] 움직임 추정 스테이지(41) 및 검사 스테이지(42)는 캡처된 이미지들 각각에 대해 수행된다. 모든 이미지들이 처리된 후, 키프레임들(310)의 모든 리스트는 추정된 위치들과 함께, 도 2b에서 설명된 다음의 서브 스테이지(204)(카메라 위치 미세조정)에서 이용된다.
- [0061] 도 5는 일 실시예에 따른 스페이스 이미지 정렬을 설명하기 위한 도면이다.
- [0062] 도 5를 참고하면, 500 단계에서, 마지막 카메라 이미지(406)에서 가시적 점들(visible points)이 맵 데이터베이스(304)로부터 추출된다. 가시적 점들은 360도의 스캐닝 조건을 만족시키기 위하여 필터링될 수 있다. 한편, 360도의 스캐닝 조건을 만족시키기 위하여, 보는 방향과 명백하게 다른 방향에서 캡처된 점들은 배제될 수 있다. 이는, 이와 같은 점들을 포함한 이미지들은 원래의 패치(original patch)와 많이 다르고, 추적이 신뢰될 수 없기 때문이다. 또한, 다른 점들에 의해 가려진 점들도 배제될 수 있다. z-버퍼 기술이 이와 같은 점들을 필터링하는데 이용될 수 있다.
- [0063] 502 단계에서, 필터링된 모든 점들의 패치들이 하나의 특징 벡터로 결합된다(combine). 504 단계에서, 특징 벡터는 새로운 이미지(404)에서 LKT를 이용하여 추적된다. LKT로부터 새로운 이미지의 카메라 위치(506)가 획득된다.
- [0064] 도 6은 일 실시예에 따른 스페이스 특징 정렬을 설명하기 위한 도면이다.
- [0065] 도 6을 참고하면, 600 단계에서, 추정된 카메라 위치를 고려하여, 마지막 카메라 이미지(406)에서 가시적 점들(visible points)이 맵 데이터베이스(304)로부터 추출된다. 602 단계에서, 각각의 특징점에 대해 패치가 독립적으로 생성된다(create). 604 단계에서, 각각의 패치는 새로운 이미지(404)에서 LKT를 이용하여 추적된다. 새로운 이미지(404)에서 각각의 특징점의 업데이트된 위치(606)가 획득된다.
- [0066] 도 7은 일 실시예에 따른 로컬 번들 조정을 설명하기 위한 도면이다.
- [0067] 도 7을 참고하면, 700 단계에서, 새로운 이미지의 카메라 위치(506)는, 도 4의 400 단계에서 획득된 초기 근사 값결정(initial approximation)에 기초한 PnP(Perspective-n-Points) 절차(procedure) 및 402 단계에서 획득된 새로운 이미지(404)에서 각각의 특징점의 업데이트된 위치(606)를 이용하여 업데이트된다. 이와 같은 방식으로, 새로운 이미지의 업데이트된 카메라 위치(706)가 획득된다. 업데이트된 카메라 위치(706)는 새로운 이미지(404)에 할당된다.
- [0068] 702 단계에서, 402 단계에서 몇 차례 사용되었던 점들은, 구조 최적화(structure optimization)를 위하여 선택된다. 704 단계에서, 선택된 점들의 월드 3D 좌표들은, 업데이트된 특징점들의 위치들(606) 및 업데이트된 카메라 위치들(706)에 기초한 번들 조정(bundle adjustment) 기술들을 이용하여 업데이트된다. 업데이트된 월드 3D 좌표들은 맵 데이터베이스(304)에 저장된다.
- [0069] 도 8은 일 실시예에 따른 깊이 필터들의 업데이트를 설명하기 위한 도면이다.
- [0070] 도 8을 참고하면, 이미지 큐(410)에서 키프레임이 아닌 이미지 각각에 대하여, 깊이-필터들이 업데이트된다. 리스트(422)의 깊이-필터들 각각에 대하여, 새로운 이미지의 에피폴라 라인이 계산된다. 800 단계에서, 에피폴라 라인을 따라 깊이 기대치 주변의 선형 검색이 수행된다. 802 단계에서, LKT를 이용하여, 최소의 에피폴라 검색(epipolar search) 주위의 깊이-필터로부터, 점들에 대한 최적의 정렬(optimal alignment)이 검색된다. 검색이 실패한 경우, 깊이-필터는 폴트 플래그(fault flag)(808)로 업데이트된다. 그러나, 804 단계에서, 최적의 정렬이 검색된 경우 삼각측량의 절차를 이용하여 점의 깊이가 결정된다. 806 단계에서, 깊이-필터는 이 깊이로 업데이트된다. 깊이-필터의 업데이트는 새로운 수학적 예측(mathematical expectation) 및 깊이-필터의 깊이 값의

변동(variance)을 판단하는데 사용된다.

[0071] 도 9는 일 실시예에 따른 카메라 위치 미세조정을 설명하기 위한 도면이다.

[0072] 서브 스테이지들(200 및 202)을 포함하는 실시간 스테이지(21)가 종료되면, 디바이스(106)의 CPU 및 GPU를 이용한 3D 재구성 프로세스의 후처리 스테이지(22)가 시작된다.

[0073] 서브 스테이지(204)는 디바이스(106)의 CPU를 이용하여 실행될 수 있다. SLAM 시스템에 의해 획득된 카메라 위치들은 급격한 드리프트를 포함할 수 있고, 추가적인 프로세싱 이전에 미세조정될 필요가 있다. 이러한 드리프트는, 카메라의 x, y 및 z 좌표들 및 요, 피치 및 롤 각도들의 추정에 있어서 시스템적 에러(systematic error)로 표현될 수 있다. 카메라 위치의 미세조정은 이러한 드리프트를 보상하기 위한 것이고, 그로써 카메라 위치들의 전반적인 정확도가 개선될 수 있다. 웨도(108)가 원형인 경우를 가정하면, 웨도(108)는 자연적인 루프 결합 점들(loop closure points)을 갖는다. 루프 결합 점이 검출되면, 카메라 위치의 누적된(accumulated) 드리프트를 추정하고 이를 보상하는 것이 가능하게 된다. 카메라 위치의 미세조정은 루프 결합의 900 단계, 루프 보정의 902 단계, 구조 재계산(structure re-calculation)의 904 단계 및 번들 조정의 906 단계의 네 단계들을 포함한다.

[0074] 900 단계에서, 웨도의 루프 점(loop point)이 구해지고(seek), 수신된 각각의 키프레임에 대하여 루프 점이 루프 결합 점의 후보인지 여부가 검사된다. 각각의 키프레임에 대하여, 다른 키프레임들 각각에 대한 SE3 스페이스에서의 상대적인 유클리드 거리가 계산된다. 계산된 거리가 임계값 ξ_{sim} 이하인 경우, 키프레임들이 공통의 맵 점들(common map points)을 얼마나 많이 갖는지가 검사된다. 두 키프레임들 KF_i 및 KF_k 가 n_{KFPt} 보다 많은 맵 점들을 갖는 경우, 상대적인(relative) 포즈 제약(pose constraint) ξ_{ik} 이 공통의 점 투영들(common point projections)로부터 PnP 알고리즘을 이용하여 계산된다. 공간적으로 근접한 두 프레임들은 공통의 점들이 적거나 또는 없다. 따라서, 이 두 프레임들은 루프 결합 점의 후보로 간주된다. 루프 결합(loop closure)을 수행하고자, PnP 알고리즘을 이용하여 특징 대응성들(feature correspondences)을 요구하는 위치 제약(position constraint)이 계산될 필요가 있다. 키프레임 KF_i 의 점 투영들의 위치들에서, 스케일링 및 회전(rotation)에 대해 불변인 BRISK(Binary Robust Invariant Scalable Keypoints) 특징들 ft_i 의 세트가 추출되고, 키프레임 KF_k 에서는 유사한 특징들 ft_k 의 세트가 FAST 코너들(FAST corners)의 위치들에서 추출된다. BRISK에 대해서는 예를 들어, Stefan Leutenegger, Margarita Chli, 및 Roland Yves Siegwart의 Brisk(Binary robust invariant scalable keypoints) In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2548-2555. IEEE, 2011에 소개되어 있다. BRISK 특징들은 시점 변화 불변(viewpoint change invariance)을 제한한다(limit). 웨도가 원형이기 때문에, 웨도의 동일한 부분을 다시 고려할(revising) 때에도 시점 변화는 매우 작다.

[0075] 앞서 설명된 두 세트들로부터, 특징 대응성들(feature correspondences)이 획득된다. 세트 ft_i 에서 특징들의 3D 좌표들은 알려져 있기 때문에, 키프레임 KF_i 에 상대적인 키프레임 KF_k 의 위치로부터, PnP 알고리즘을 이용하여, 포워드 제약(forward constraint) ξ_{ik}^{fc} 이 계산될 수 있다. 동일한 절차가 키프레임 KF_k 에 대해서도 수행되고, 백워드 제약(backward constraint) ξ_{ki}^{bc} 이 계산될 수 있다. 특징 매칭(feature matching)은, 잘못된 PnP 해(wrong PnP solution)로 인하여 잘못된 대응성들(wrong correspondences)을 야기할 수 있다. 카메라 위치 미세조정에 있어서 잘못된 PnP 해들이 포함되는 것을 방지하기 위하여, 역수 검사(reciprocal check)가 수행된다. 리 군(Lie group)의 속성들에 따라, 두 카메라 위치들이 에러를 포함하지 않는 경우, 포워드 제약 및 백워드 제

약은 $\xi_{ik}^{ft} = (\xi_{ki}^{ft})^{-1}$ 와 같이 일치하여야 한다. 그러므로, 포워드 제약 및 백워드 제약 간의 불일치는

$$err_{rc} = \left\| \ln \left(\xi_{ik}^{ft} \cdot (\xi_{ki}^{ft})^{-1} \right) \right\| \text{ 와 같이 계산될 수 있다. 불일치 } err_{rc} \text{가 임계값 } err_{rc}^{th} \text{ 이하인 경우,}$$

포워드 제약 ξ_{ik}^{fc} 와 백워드 제약 ξ_{ki}^{bc} 은 최적화에 추가된다. 그러나, 불일치 err_{rc} 가 임계값 err_{rc}^{th} 초과인 경우, 이와 같은 포워드 제약 및 백워드 제약은 무효(Invalid)로서 간주되어 무시된다.

[0076] 900 단계에서 계산된 제약들(constraints)은 902 단계에서 사용된다. 제 1 키프레임 및 루프 결합 키프레임(마

지막 키프레임)의 드리프트 때문에, 포워드 제약 및 백워드 제약은 불일치한다(disagree). 이와 같은 드리프트를 보상하기 위하여, 불일치(disagreement)가 프레임들로 분산되도록, 각각의 키프레임에 대응되는 카메라 위치가 재추정(re-estimate)된다. SLAM 방식은 각각의 카메라 위치에 변동(variance)을 제공한다. 이와 같은 변동은, 카메라 위치가 측정 노이즈 및 카메라 드리프트에 의해 얼마나 오류가 발생하였는지를 나타낸다. 많은 위치 변동(position variance)을 갖는 키프레임들은 더 많은 위치 보정이 제공되어야 하고, 반면에 적은 위치 변화를 갖는 키프레임들은 더 적은 위치 보정이 제공되어야 한다. 그 결과, 드리프트의 영향이 제거되기 때문에, 카메라 위치들의 정확도는 개선될 수 있다.

[0077] 904 단계에서, 카메라 위치들의 변화들로 인하여 3D 좌표들이 업데이트됨으로써, 최적화된 카메라 위치들이 계산된다. 다만, 제 1 키프레임에 대한 점의 깊이(z 좌표)는 루프 보정 전과 동일하도록, 최적화된 카메라 위치들이 계산된다.

[0078] 906 단계에서, 측정 노이즈를 감소시키기 위하여, 최종적인 카메라 위치들의 미세조정이 수행된다. 각각의 키프레임 KF_i 에 대하여, SLAM 방식의 정렬을 통해 획득된 점 m 의 투영 uv_m^i 은 예측 투영 $uvp_m^i = \pi(SE3_i * p_m)$ 과 비교되고, 점 재투영 에러(point re-projection error)는

$$err(m, i) = \|uv_m^i - uvp_m^i\|$$

$$err_{ba} = \sum_{i=0, m=0}^{i=L, m=M} err(m, i)$$

와 같이 계산된다. 총 재투영 에러는 err_{ba} 와 같이

계산된다. 이 에러는 카메라 위치들에서 나타난 측정 노이즈를 설명한다. 번들 조정은 카메라 위치들 및 점들의 3D 좌표들을 함께 최적화시킴으로써, 총 재투영 에러를 최소화한다. 이는 점들의 3D 좌표들 및 재투영 에러를 감소시키는 방향의 카메라 위치들의 반복적인 보정(iterative correction)을 통한, 가우시안(Gaussian) 최적화를 수행함으로써 행해진다. 이와 같은 최적화 후에, 키프레임들의 세트는 미세조정된 카메라 위치들과 함께, 추가적인 프로세싱을 위해 908 단계로 전달된다.

[0079] 도 10은 일 실시예에 따른 깊이 맵 추정을 설명하기 위한 도면이다.

[0080] 서브 스테이지(204)의 출력으로부터 획득된, 키프레임들 및 미세조정된 카메라 위치들은 서브 스테이지(206)의 입력으로 사용된다. 스테레오 쌍 선택(stereo pair selection)을 위한 1000 단계에서, 각 키프레임에 대하여, 다른 프레임이 선택되고 미정류(non-rectified) 스테레오 쌍이 형성된다. 제 2 프레임의 선택은, 스테레오 쌍의 이미지들 사이에 너무 크지 않은 충분한 병진운동(translational motion)을 보장(ensure)하기 위한 카메라 위치들에 기초한다. 스테레오 쌍 및 상대적인 카메라 위치는 깊이 맵 계산에 이용된다.

[0081] 깊이 맵은 단계적인(coarse-to-fine) 피라미드 방식(pyramidal approach)으로 계산된다. 거친 깊이 추정(coarse depth estimation)을 위한 1002 단계에서, 스테레오 쌍의 이미지들에 대해 저해상도(low resolution) 샘플링이 사용되고, 그 결과 저해상도 깊이 맵이 생성된다. 깊이 필터링을 위한 1004 단계에서, 저해상도 깊이 맵이 추가적으로 필터링될 수 있다. 그리고 나서, 다음의 피라미드 레벨에서, 더 미세한(finier) 해상도의 샘플링이 사용된다. 1006 단계에서, 이전 단계의 저해상도 결과는 업스케일(upscale)되고, 현재 피라미드 레벨의 깊이 맵을 추정하기 위해 사용된다. 1008 단계에서, 깊이 맵의 추정은 현재 피라미드 레벨 상에서 더 미세하게 샘플링된 이미지들 이용하여 미세조정되고, 1004 단계에서 더 필터링된다. 이와 같은 업-샘플링(up-sampling) 절차는, 원하는 해상도의 깊이 맵이 획득되거나 또는 정해진 런타임에 도달할 때까지, 루프(1010)를 거쳐 몇 회 수행될 수 있다.

[0082] 제 1 피라미드 레벨(가장 거친(coarsest))에 대하여는 오직 적은 수의 픽셀들만 처리되고, 이에 따라 보다 복잡하고 정확한 깊이 맵 추정 방법이 사용될 수 있다. 예를 들어, 블록-매칭(block-matching) 또는 코스트-볼륨 필터링(cost-volume filtering) 방법들의 경우에, 더 큰 런타임의 코스트에서 보다 높은 정확성을 제공하는, 보다 큰 윈도우 사이즈가 이용될 수 있다.

[0083] 거친 깊이 추정을 위한 1002 단계는, 검색 범위(search range)의 추정 및 샘플링, 매칭 코스트들의 계산, 코스트들의 집계(aggregation), 코스트 최소화에 기초한 차이 추정(disparity estimation) 및 깊이 삼각측량의 세부 단계들을 포함할 수 있다. 검색 범위는, 제 1 이미지의 현재 픽셀에 대한 가능한(possible) 가까운(near) 깊이 값들 및 먼(far) 깊이 값들의 주어진 추정들(given estimations)로부터 추정된다. 여기서, 현재 픽셀은 SLAM 방식이 수행되는 동안 생성된 재구성된 장면의 성긴(sparse) 포인트 클라우드(point cloud)로부터 획득될 수 있

다. 이와 같은 깊이 값들을 이용하여, 현재 픽셀에 대응되는 에피폴라 라인 상의 검색 세그먼트(search segment)가 판단된다. 그리고 나서, 제 1 이미지의 현재 픽셀에 대응되는 제 2 이미지의 점의 가능한 위치들을 결정하기 위하여, 제 2 이미지 상의 검색 세그먼트에 대한 픽셀 샘플링이 사용될 수 있다. 매칭 코스트는, 이미지 컬러 또는 계조도(gradient) 상의 픽셀 단위의 컬러 차이(pixel-wise color difference) 또는 거리 메트릭(distance metric)으로부터 계산된다. 코스트 집계는 현재 픽셀 주변의 몇몇 영역에서의 코스트들의 가중화된 축적(weighted accumulation)에 의해 수행된다. 차이(disparity)는 에피폴라 라인 상의 검색 세그먼트들을 따라 승자독식 코스트 최소화 전략(winner-takes-all (WTA) cost minimization strategy)에 의해 추정된다. 깊이는 에피폴라 제약들(epipolar constraints)을 이용하여 스테레오 쌍의 양 이미지들에서 대응되는 점 위치들로부터 삼각측량된다.

[0084] 스테레오 쌍의 이미지들이 정류되지 않는다는 사실에도 불구하고, 현재 알고리즘은 디바이스(106)의 GPU에서 매우 효율적인 병렬 구현(efficient parallel implementation)을 제공한다. 메모리 소비는 또한 감소될 수 있고, 이는 메모리에 모든(full) 코스트-볼륨을 저장할 필요가 없기 때문이다. 이미지는 작은 영역들에서 처리될 수 있고, 영역들 각각에 대하여 매칭 코스트 밸류들(matching cost values)은 GPU 로컬 메모리에 저장될 수 있다.

[0085] 계산된 깊이 맵들은 깊이 맵 합성(fusion)에 이용되기 때문에, 가능한 한 잘못된(wrong) 깊이 맵 추정들을 필터링하여, 오직 높은 신뢰도를 갖는 정확한 깊이 값들만 남기는 것이 바람직하다. 왜냐하면, 합성 방법들은 대개, 존재하는 값들로부터 인터폴레이션 또는 프로퍼게이션(propagation)에 의해 결측 데이터(missing data)를 다룰 수 있고, 심한 이상치들(heavy outliers)에 보다 민감하기 때문이다. 깊이 이상치들은, WTA(winner-takes-all) 코스트 최소화 동안에 현재 픽셀에 대한 코스트의 최소값에 대한 코스트의 누적된 비율들을 분석함으로써 효율적으로 필터링될 수 있다. 추가적인 필터링은 좌-우 일관성 검사(left-right consistency check)를 이용하여 수행될 수 있다. 또한, 이상치 필터링을 각각의 피라미드 레벨에 대해 수행함으로써 이상치 필터링은 단계적 전략(coarse-to-fine strategy)으로 포함될 수 있다. 이에 따라, 더 미세한 피라미드 레벨들에 대한 계산량이 감소될 수 있다.

[0086] 깊이 필터링을 위한 1004 단계는, 광도 모호성(photometric ambiguity)에 의한 필터링 및 좌-우 일관성 검사(left-right consistency check)에 의한 필터링의 세부 단계들을 포함한다. 광도 모호성은 WTA(winner-takes-all) 코스트 최소화 동안에 추정되고, 현재 픽셀에 대한 코스트의 최소값에 대한 코스트의 누적된 비율들을 분석함으로써 계산된다. 텍스처가 없거나 또는 모호한(즉, 에피폴라 라인을 따라 주기적인) 경우, 비율들이 1에 근접하도록 야기하여, 많은 코스트들이 최소 코스트(minimal cost)에 유사한 값들을 갖게 될 것이다. 이에 따라 비율들을 분석함으로써, 모호성들이 필터링될 수 있다. 좌-우 일관성 검사(left-right consistency check)는 스테레오 쌍의 좌우 이미지들에 대한 두 깊이 맵들의 일관성을 분석함으로써 수행된다. 일관성은 두 깊이 맵들의 깊이 값들을 이용하여 각각의 픽셀에 대한 재투영 에러들을 검사함으로써 판단될 수 있다.

[0087] 깊이-필터링은 다음 피라미드 레벨에서 처리될 픽셀들의 개수를 상당히 감소시킬 수 있다. 즉, 각각의 피라미드 레벨에 대해 깊이-필터링을 수행하는 것은 런타임을 상당히 줄일 수 있다. 왜냐하면, 더 미세한(finier) 피라미드 레벨들은 전형적으로, 거친 피라미드 레벨들보다 느리게 처리되기 때문이다.

[0088] 다음 피라미드 레벨에 대한 깊이 맵 업데이트는, 거친 깊이 맵 업스케일링(coarse depth map upscaling)을 위한 1006 단계 및 깊이 맵 미세조정을 위한 1008 단계를 포함한다. 거친 깊이 맵은, 다양한 필터링 방법들, 예를 들어 최근접 이웃법(nearest-neighbor), 2중선형 필터링법(bilinear filtering), 박스 필터링법(box filtering), 가우스 필터링법(Gauss filtering), 양방향 필터링(bilateral filtering) 또는 에지-보존 필터링법(edge-preserving filtering) 등을 이용하여, 더 미세한 해상도로 업스케일링될 수 있다. 깊이 업스케일링에 있어서, 결측치들(missing values)을 이용한 단순(simple) 2중선형 필터링이 성능 향상을 위해 사용된다. 업스케일링된 깊이는, 깊이 맵 미세조정(1008 단계) 이전에 사용된다. 각각의 단계에서, 매칭 코스트는 현재 피라미드 레벨에 대응되는 더 미세한(finier) 샘플링을 이용하여 재평가된다(re-evaluated). 대응 점(corresponding point)은 모든 단계들 중에서 미세조정된 코스트 값들의 최소로부터 발견되고, 깊이는 다시 삼각측량된다. 더 미세한 피라미드 레벨 각각에 대한 추정된 픽셀을 갖는 영역의 감소는, 스무딩(smoothing)의 영향을 감소시키고, 보다 정확한 깊이 값들을 생성해 낸다.

[0089] 서브 스테이지(206)의 결과는 깊이 맵들의 세트 및 카메라 위치들이고, 깊이 맵들의 세트는 매 키프레임에서 캡처된 오브젝트의 깊이에 관한 정보를 포함한다.

[0090] 도 11은 일 실시예에 따른 깊이 맵 합성(fusion)을 설명하기 위한 도면이다.

[0091] 도 11을 참고하면, 볼륨 깊이 맵 통합(volumetric depth map integration)을 위한 1100 단계에서, 서브 스테이지(206)에서 계산된 카메라 위치들과 함께 깊이 맵들의 세트는 볼륨 깊이 맵 통합 또는 합성을 획득하기 위해 처리된다. 이와 같은 합성의 결과는 1102 단계에서 3D 복셀 히스토그램 표현(3D voxel histogram representation)을 생성하기 위해 사용된다. 매 복셀에 대하여, 깊이 맵들에 의해 유도된 표면들에 대한 근사화된 부호 거리(signed distance)가 계산되고, 사용자-공급된(user-supplied) 임계값에 클램핑되고(clamped), 인터벌 (-1, 1)로 스케일링되고, 주파수들 n_j 각각의 히스토그램으로 저장된다. 각각의 복셀은 빈 센터(bin center) c_j 으로부터 고르게 간격을 둔(evenly spaced) N개의 빈들(N bins)을 포함하는 히스토그램을 갖는다. N-2개의 빈들(N-2 bins)은 인터벌 (-1, 1)의 내에서 값들의 분포, 이른바 표면에 가까운 복셀들을 나타내는데 사용된다. 두 개의 별도의 빈들(two separate bins)은 표면에 대해 더 먼 거리를 나타내는 값들을 위해 마련되고, 하나의 빈(one bin)은 가려진 복셀로서, -1 이하의 스케일링된 거리들로 간주되고, 하나의 빈은 빈 복셀로서, 1 이상의 값들을 나타낸다. 히스토그램 수치들의 총 개수는 이용 가능한 깊이 맵들의 개수에 의해 제한된다.

[0092] 볼륨 히스토그램 최적화를 위한 1104 단계에서, 변분적(variational) 방식이 사용된다. 1106 단계에서, 에너지 함수를 최소화하는 반복해(iterative solution)로서, 3D 복셀 SDF 표현(3D voxel signed distance function representation)(또는 3D 복셀 TSDF(truncated signed distance function) 표현)이 획득된다. 여기서, 에너지 함수는, 예를 들어 Christopher Zach의 "Fast and high quality fusion of depth maps", In Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT), volume 1, Citeseer, 2008.에서 설명된 바와 같이, 다음의 수학적 식 1과 같다.

수학적 식 1

$$E^{TV-L^1} = \int_{\Omega} \left\{ |\nabla u| + \lambda \sum_j n_j |u - c_j| \right\} dx$$

[0093]

[0094] 서브 스테이지(208)의 결과는, 캡처된 오브젝트(102)의 TSDF 표현이고, 이는 컴퓨터 그래픽스 분야에서 임의의 3D 표면을 나타내는 하나의 방법일 수 있다.

[0095] 도 12는 일 실시예에 따른 표면 재구성을 설명하기 위한 도면이다.

[0096] 도 12를 참고하면, 복셀-투-옥트리(voxel-to-octree) 표현을 위한 1200 단계가 수행된다. 서브 스테이지(208)에서 계산된 TSDF 표현은 오브젝트(102)에 대한 복셀-투-옥트리(voxel-to-octree) 표현을 획득하기 위하여 처리된다. 1202 단계에서, 복셀-투-옥트리(voxel-to-octree) 표현은, 옥트리 기반(octree-based) TSDF 표현을 획득하기 위해 사용된다. 최대 옥트리 깊이는, 오브젝트(102)의 3D 메쉬의 복잡도(폴리곤들의 개수)를 제어한다.

[0097] 등위면(isosurface) 추출을 위한 1204 단계 동안에, 3D 메쉬(1206)가 재구성된다. 이와 같은 방식에 대해서는, 예를 들어 Michael Kazhdan, Allison Klein, Ketan Dalal, 및 Hugues Hoppe의 "Unconstrained isosurface extraction on arbitrary octrees", In Symposium on Geometry Processing, volume 7, 2007.를 참고할 수 있다.

[0098] 도 13은 일 실시예에 따른 텍스처 매핑을 설명하기 위한 도면이다.

[0099] 3D 메쉬(1206) 및 키프레임들(908)과 함께 미세조정된 카메라 위치들은 텍스처 매핑을 위한 입력으로 사용된다. z-버퍼 면 가시성 검사(z-buffer face visibility check)를 위한 1300 단계에서, 어느 카메라 위치들에서 각각의 면(face)이 가시적인지가 검사된다. 카메라는, 카메라 이미지 및 카메라 위치, 다시 말하면 키프레임(908) 및 미세조정된 위치에 의해 각각 특징지어진다(characterized).

[0100] 각각의 가시적인 메쉬 면(mesh face)은 카메라 이미지들 중 어느 하나에 대한 투영에 의하여 텍스처링되어야 한다. 면 라벨링(face labeling)을 위한 1302 단계에서, 카메라 이미지들 중 어느 것이 이러한 목적에서 정의된다. 하나의 이미지로부터 텍스처들에 제공되는 메쉬 면들의 폐 영역들(closed areas)인 많은 패치들이 있다. 인접한 패치들 사이의 경계들(seams)은 매우 뚜렷할(noticeable) 수 있다. 글로벌 컬러 조정(global color adjustment)을 위한 1304 단계에서, 인접한 패치들 간의 컬러 불연속들(color discontinuities)은 경계들

(seams)이 덜 가시적이도록(less visible) 조정된다.

- [0101] 홀 필링(hole filling)을 위한 1308 단계에서, 비가시적인(invisible) 메쉬 면들에 대한 텍스처들이 생성된다.
- [0102] 텍스처 아틀라스 생성(texture atlas generation) 및 메쉬 파라미터화(mesh parametrization)를 위한 1310 단계에서, 카메라 이미지들로부터 텍스처 아틀라스가 생성되고, 비가시적 면들에 대한 텍스처들이 생성된다. 각각의 메쉬 면의 텍스처 아틀라스에 대한 텍스처 좌표들을 정의하는 메쉬 파라미터화는, 서브 스테이지(214)의 텍스처링된 메쉬를 획득하기 위하여 수행된다.
- [0103] 도 14는 일 실시예에 따른 z-버퍼 면 가시성 검사(z-buffer face visibility check)를 설명하기 위한 도면이다.
- [0104] 메쉬(1206) 및 키프레임들(908)은 미세조정된 카메라 위치들과 함께, z-버퍼 가시성 검사를 위한 입력으로 사용된다. 각각의 카메라 위치에 대하여, 어떤 메쉬 면들이 가시적인지가 판단된다.
- [0105] 1400 단계에서 z-버퍼가 생성되고, 카메라 이미지와 동일한 크기를 갖는 이미지, 픽셀에 투영된 표면의 최소 깊이를 포함하는 각 픽셀이 생성된다. 메쉬 면의 깊이는, 현재 카메라 좌표들에서 메쉬 면의 버텍스들의 최대 깊이이다.
- [0106] 1402 단계에서, 다음 점들은 가시적인 것들과 비가시적인 것들로 구분된다. 만약 p가 카메라 위치 r의 이미지 상의 픽셀 px1에 투영될 수 있고 $z_p \leq z_{buf}(px1)$ 를 만족한다면, 점 p는 카메라 위치 r로부터 가시적인 것으로 설정된다. 여기서, z_p 는 카메라 위치 r의 좌표들에서 점 p의 깊이이고, $z_{buf}(px1)$ 는 픽셀 px1에서 z-버퍼의 값이다.
- [0107] 만약 1404 단계에서 메쉬 면 f_1 의 모든 버텍스들이 카메라 위치 r로부터 가시적이고 $\cos \theta < 0$ 를 만족하는 것으로 판단된다면, 1408 단계에서 각각의 메쉬 면 f_1 은 카메라 위치 r로부터 가시적인 것으로 할당된다. 그렇지 않다면, 1406 단계에서, 메쉬 면 f_1 은 카메라 위치 r로부터 비가시적인 것으로 할당된다. 여기서, θ 는 카메라 위치 r의 시야 방향(view direction) 및 메쉬 면 f_1 의 페이스 노멀(face normal) 간의 각도이다.
- [0108] 그 결과, 두 개의 리스트들이 형성된다. 즉, 메쉬 면들이 보여지는(visible) 카메라 위치들에 대한 정보와 함께 가시적 메쉬 면들(1410)의 리스트 및 비가시적 메쉬 면들(1412)의 리스트가 형성된다.
- [0109] 도 15는 일 실시예에 따른 면 라벨링(face labeling)을 설명하기 위한 도면이다.
- [0110] 1302 단계에서, 리스트(1410)의 각각의 가시적 메쉬 면 f_i 에 대하여 키프레임들(908)의 카메라 이미지 I_i 의 텍스처들을 정의하는, 라벨링 벡터(labeling vector) l_i 가 계산된다. 라벨링 벡터 l_i 은 MRF(Markov random field) 에너지 최소화(energy minimization)에 의해 구해질 수 있고, 예를 들면 다음의 수학적 식 2가 이용될 수 있다.

수학적 식 2

$$E(l) = \sum_{f_i \in Faces} w_i^{l_i} + \alpha \sum_{(f_i, f_j) \in Edges} w_{ij}^{l_i, l_j} \rightarrow \min_l$$

- [0111]
- [0112] 수학적 식 2에서 첫 번째 항은, 텍스처링에 사용되는 텍스처 프래그먼트들(texture fragments)의 품질을 정의한다. 이와 같은 경우, $w_i^{l_i} = \sin^2 \phi$ 이다. 여기서, ϕ 는 카메라 이미지 I_i 의 시야 방향과 메쉬 면 f_i 의 페이스 노멀(face normal) 간의 각도이다.
- [0113] 수학적 식 2에서 두 번째 항은, 인접한 텍스처 프래그먼트들 간의 경계들의 가시성(visibility of seams)을 정의한다. 만약 메쉬 면들 f_i 및 f_j 는 인접한 에지 E_{ij} 를 갖는다면, 경계들의 가시성은, 대응되는 카메라 이미지 I_i 및 I_j 의 텍스처 프래그먼트들에서 에지 E_{ij} 의 컬러들의 적분 차(integral difference)로 측정된다. 이에 대해서는 다음의 수학적 식 3이 이용될 수 있다.

수학식 3

$$w_{i,j}^{I_i, I_j} = \int_{E_{ij}} d_{RGB}(Pr^{I_i}(X), Pr^{I_j}(X)) dX$$

[0114]

[0115] 수학식 3에서, Pr^{I_i} 는 카메라 이미지 I_i 의 시야 방향에 대한 투영 연산자(projection operator)를 의미하고, $d_{RGB}(\cdot, \cdot)$ 는 RGB 컬러 스페이스에서의 유클리드 거리를 의미한다.

[0116] 1500 단계에서, 단항 포텐셜(unary potential) w_i 및 쌍 포텐셜(pairwise potential) w_{ij} 은 모든 가시적 메쉬 면들(1410) 및 키프레임들(908)의 카메라 이미지들에 대해 계산된다. 표준화된 포텐셜들이 사용되고, α 는 두 에너지 항들(energy terms) 사이의 트레이드-오프(trade-off)를 나타내고, 실험상 50으로 설정된다.

[0117] 다음으로, 서브 스텝(1502)에서 SRMP(sequential reweighted message passing) 알고리즘이 MRF 문제(Markov random field (MRF) problem)를 해결하기 위하여 실행된다. 그 결과, 면 라벨들(face labels)을 특징짓는 라벨링 벡터 l (1504)이 구해진다.

[0118] 1304 단계에서, 글로벌 컬러 조정이 수행된다. 여기서, 메쉬(1206), 키프레임들(908), 미세조정된 카메라 위치들 및 라벨링 벡터 l (1504)이 입력으로 사용된다. 라벨링 벡터 l (1504)은 텍스처 패치들의 개수를 정의한다. 이후에 텍스처 패치들 간의 경계들이 덜 뚜렷하게 된다.

[0119] f 가 RGB 컴포넌트들 중 어느 하나에 대응되는 텍스처 세기 함수(texture intensity function)인 것으로 정의된다. f 는 각각의 텍스처 패치에 대해 연속적인 것으로 간주되고, 텍스처 패치들 간의 경계들(boundaries) 상에서는 불연속들(discontinuities)을 갖는다. f 가 g 가 패치들 및 경계들 상에서 연속적일 수 있도록 레벨링 함수 g 를 찾는 것이 목표이다.

[0120] 각각의 경계 벡터 v 는 I_1 로부터 텍스처링된 v_{I_1} 및 I_2 로부터 텍스처링된 v_{I_2} 등으로 분할된다. 이는, v_s 를 포함하는 각각의 패치의 하나이다. 레벨링 함수 g 는 메쉬의 각각의 벡터에서 구해진다.

수학식 4

$$\sum_{\substack{\text{lies on seam.} \\ \text{splits into } v_{I_1}, v_{I_2}}} (f'_{v_{I_1}} + g_{v_{I_1}} - (f'_{v_{I_2}} + g_{v_{I_2}}))^2 + \frac{1}{\lambda} \sum_{\substack{v_i, v_j \text{ are adjacent.} \\ \text{from the same patch}}} (g_{v_i} - g_{v_j})^2 \rightarrow \min_g$$

[0121]

[0122] 수학식 4에서, $f'_{v_{I_i}}$ 는 카메라 이미지 I_i 로부터 경계 벡터 v 를 포함하는 심 에지들(seam edges)을 따르는 컬러들의 가중합의 평균을 의미한다. 가중합에서, 경계 벡터 v 의 가중치는 1이고, 에지를 따라 선형적으로 감소한다.

[0123] 이와 같은 경우, MRF 문제는 RGB의 각각의 채널에 대해 별도로 Eigen의 공액 구배법(conjugate gradient)을 이용함으로써 해결된다. 모든 메쉬 벡터스들에 대해 레벨링 함수 g 의 값들이 계산된 후, 벡터스들 간의 값들은 인터폴레이션을 이용하여 계산된다. 레벨링 함수 g 는 카메라 이미지들(키프레임들(908))의 대응되는 프로그래먼트들에 적용된다.

[0124] 도 16은 일 실시예에 따른 홀 필링(hole filling)을 설명하기 위한 도면이다.

[0125] 도 16을 참고하면, 1600 단계 내지 1610 단계에서는 비가시적 점들에 대해 컬러가 할당되고, 1612 단계에서는

비가시적 메쉬 면들에 대한 텍스처들이 기록된다(write).

- [0126] 비가시적 점 컬러들(Invisible point colors)은 알려진 점 컬러들(known point colors)의 인터플레이션을 이용하여 계산된다. 가시적인 점 p_{vis} 의 컬러는, 점 p_{vis} 을 포함하는 메쉬 면을 텍스처링하기 위한 동일한 카메라 이미지로부터 제공된다. 만약, 서로 다르게 라벨링된 점 p_{vis} 을 포함하는 몇 개의 면들이 있다면, 그 면들 중 어느 하나가 선택된다.
- [0127] 1600 단계에서, 모든 비가시적 점들(리스트(1412)의 비가시적 메쉬 면들의 버텍스들)이 프로세싱 세트로서 추가된다.
- [0128] 다음으로, 1602 단계에서 프로세싱 세트가 비어있지 않은 동안에, 인터플레이션 프로세스가 수행된다.
- [0129] 1604 단계에서, 인터플레이션 프로세스 동안, 하나 이상의 인접한 가시적인 점들을 갖는 프로세싱 세트의 각각의 점에 대하여 컬러들이 계산된다. 1606 단계에서, 점 p 의 컬러는 인접한 가시적인 점들의 컬러들의 평균에 의해 획득된다. RGB 채널들 각각은 별개로 평균을 낸다. 그리고 나서, 1608 단계에서 점 p 는 가시적인 것으로 설정되고, 1610 단계에서 점 p 는 프로세싱 세트로부터 삭제된다. 인터플레이션 프로세스는 1602 단계에서 프로세싱 세트가 비어있지 않은 동안에 수행된다.
- [0130] 1612 단계에서, 비가시적 면들에 대한 텍스처들(1614)은 5x5 픽셀들 크기의 블록들로 구성된다. 여기서, 하나의 블록은 각각의 면에 대한 것이다. 블록의 구조는 도 17에 도시되어 있다. v_1 , v_2 및 v_3 에 해당되는 셀들은 인터플레이션 프로세스 동안에 발견된 면 버텍스들(face vertices)의 컬러들을 갖는다. 다른 셀들의 컬러들은 화살표들이 있는 셀들로부터 각각의 셀까지의 컬러들의 평균에 의해 계산된다.
- [0131] 서브 스테이지(212)의 텍스처 매핑 프로세서의 마지막 단계는, 텍스처 아틀라스 생성 및 파라미터화(texture atlas generation and parametrization)를 위한 1310 단계이다. 우선, 카메라 이미지들의 최소 바운딩 박스에 대하여, 텍스처 매핑에 사용될 영역들이 선택된다. 그리고 나서, 이 영역들의 일부들은, 텍스처들을 이용하여 리스트(1614)의 비가시적 면들 및 어떤 크기의 텍스처 아틀라스를 생성하고자 스케일링되고 결합된다(combine). 다음으로, 메쉬 파라미터화가 수행된다. 이는, 예를 들어 각각의 메쉬 면 버텍스(mesh face vertex)에 대한 텍스처 좌표들을 계산하는 것일 수 있다.
- [0132] 도 18은 일 실시예에 따른 디바이스(106)의 하드웨어 구성을 도시한 도면이다.
- [0133] 도 18을 참고하면, 디바이스(106)는 디스플레이(1800), CPU(1802), 하나 이상의 사용자 인터페이스부(1804), 메모리(1806), GPU(1808), 배터리(1810) 및 카메라(1812)를 포함한다.
- [0134] CPU(1802) 및 GPU(1808)는 앞서 설명된 3D 재구성 프로세스를 실행한다. 3D 재구성 프로세스는 저 성능 및 저 전력 프로세서에 의해서도 실행될 수 있다. 앞서 설명된 3D 재구성 프로세스는 하나의 프로세서에 의해 실행되거나 또는, 둘 이상의 프로세서들에 의해 실행될 수도 있다.
- [0135] 메모리(1806)는, CPU(1802) 및 GPU(1808)에 의해 사용될, 3D 재구성 프로세스의 실행에 요구되는 서로 다른 소프트웨어들을 저장한다. 나아가서, 메모리(1806)는 3D 재구성 프로세스의 실행에 의해 생성된 중간(intermediate) 데이터, 최종 결과 등을 저장한다. 한편, 본 실시예들에 따른 3D 재구성 프로세스는 적은 메모리 사이즈로도 구현될 수 있다.
- [0136] 카메라(1812)(또는 스캐닝 모듈)은 오브젝트(102)를 스캔하고, 이미지를 캡처하는데 사용된다.
- [0137] 디스플레이(1800)는 카메라(1812)가 오브젝트(102)를 정확히 향할 수 있도록 사용자를 도와주며, 3D 재구성 프로세스의 진행 정도, 예를 들어 오브젝트(102)의 텍스처링된 메쉬와 같은 최종 결과에 대한 정보를 표시한다.
- [0138] 사용자 인터페이스부(1804)는 하나 이상의 버튼들, 마이크론들, 스피커들 등을 포함하고, 스캐닝을 시작하거나 멈추는데 사용될 수 있다. 또한, 사용자 인터페이스부(1804)는 오브젝트(102)의 텍스처링된 메쉬와 인터랙션하기 위해 이용될 수 있다.
- [0139] 배터리(1810)는 디바이스(106)에 전원을 공급하는데 이용된다.
- [0140] 디바이스(106)의 하드웨어 구성은 도 18에 의해 제한되지 않는다. 디바이스(106)는 도 18에 도시된 바와 달리, 추가적인 구성들을 포함하거나 또는 일부 구성들을 포함하지 않을 수 있다. 예를 들어, 디바이스(106)는 사용자 인터페이스부(1804)가 구비되지 않을 수 있다. 또한, 디스플레이(1800)는 오브젝트(102)의 스캐닝을 제어하기 위한 사용자 인터페이스(UI)를 표시하는 터치 스크린으로 구현될 수 있다. 이때, 사용자 인터페이스(UI)는 사용

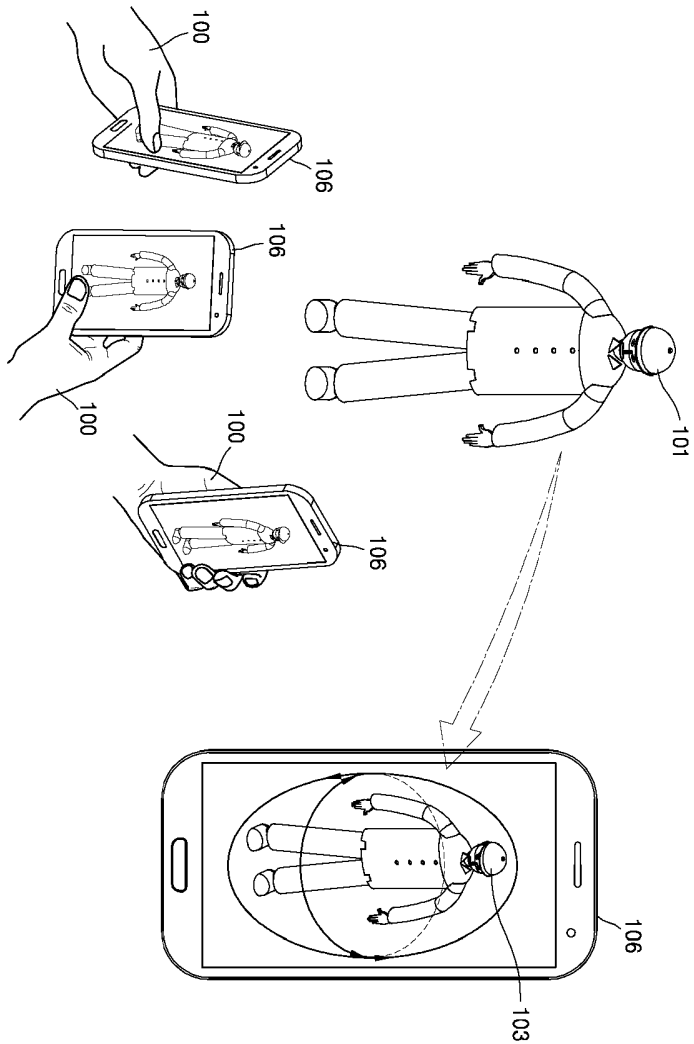
자가 스캐닝 각도 또는 스캐닝 시간 등을 변경하는데 이용될 수 있고, 3D 재구성 프로세스에 필요한 다른 기능들을 실행시키는데 이용될 수 있다.

- [0141] 디바이스(106)는 무선 또는 유선으로 하나 이상의 인쇄 수단들, 다른 휴대용 디바이스들과 연결될 수 있다. 예를 들어, 디바이스(106)는 오브젝트(102)의 텍스처링된 메쉬(3D 모델)를 3D 프린팅할 수 있는 인쇄 수단에 연결될 수 있다. 한편, 디바이스(106)는 다른 휴대용 디바이스들에 연결되어, 다른 휴대용 디바이스들로 오브젝트(102)의 3D 모델을 전송할 수 있다.
- [0142] 3D 재구성 프로세스가 수행되는 동안, 사용자(100)는 다음과 같은 액션들을 취할 수 있다.
- [0143] 1. 사용자(100)는 디바이스(106)에 설치된 소프트웨어를 실행하고, 카메라(1812)를 이용하여 오브젝트(102)의 스캐닝을 시작한다. 이때, 사용자(100)는 사용자 인터페이스부(1804)를 이용할 수도 있다. 예를 들어, 오브젝트(102)의 스캐닝은 사용자의 보이스 신호에 의해 활성화될 수 있다.
- [0144] 2. 사용자(100)는 오브젝트(102) 둘레의 궤도(108)를 따라 이동한다. 소프트웨어는 디스플레이(1800)에 표시된 메시지들로, 적절한 3D 재구성을 위하여 오브젝트(102) 둘레로 카메라(1812)를 얼마만큼 이동시켜야 하는지를 사용자에게 안내할 수 있다.
- [0145] 3. 사용자(100)는 오브젝트(102)의 스캐닝을 멈추고, "재구성 시작" 버튼을 터치한다. 디바이스(106)는 CPU(1802) 및 GPU(1808)를 이용하여 앞서 설명된 서브 스테이지들(204, 206, 208, 210, 212)를 포함하는 후처리(post processing)를 수행하고, 메모리(1806)에는 그 과정에서 처리된 결과들을 저장한다. 그 결과, 사용자(100)는 디스플레이(1800)를 통해 캡처된 오브젝트(102)의 3D 모델(텍스처링된 메쉬)(214)을 볼 수 있다. 사용자(100)는 3D 모델을 회전시키거나 스케일링할 수 있다.
- [0146] 4. 사용자(100)는 3D 모델을 저장할 수 있고, 다른 컴퓨터들에 전송하거나, 렌더링, 텍스처 변경 등과 같은 추가적인 작업들을 할 수 있다.
- [0147] 본 실시예들에 따르면, 3D 재구성에 사용될 키프레임들이 자동적으로 선택되기 때문에 빠르고 사용자 친화적인(user-friendly) 오브젝트 캡처링의 방식이 사용자에게 제공될 수 있다. 그리고, 클라우드 컴퓨팅 서비스를 이용하지 않기 때문에, 이미지를 수동적으로 저장하거나 클라우드로 전송할 필요가 없다.
- [0148] 한편, 본 실시예들에서는 후처리가 디바이스(106)에서만 실행되는 것으로 설명되었으나, 이에 제한되지 않고 후처리를 위해 클라우드 컴퓨팅 서비스가 이용될 수도 있다.
- [0149] 나아가서, 디바이스(106)가 호스트 PC로 실시간으로 이미지들을 전송함으로써, 호스트 PC에서도 카메라 위치 추정 및 후처리 등이 수행될 수도 있다.
- [0150] 카메라(1812)에 의해 캡처된 이미지들만이 서브 스테이지(202)의 카메라 위치 추정에 사용되는 것이 아닐 수도 있다. 예를 들어, 디바이스(106)는 관성 센서들(inertial sensors)과 같은 다른 추가적인 디바이스들을 구비할 수 있고, 이와 같은 디바이스들에 의해 획득된 데이터가 카메라 위치 추정에 추가적으로 이용될 수도 있다.
- [0151] 텍스처링된 메쉬는 3D 재구성의 결과로서 획득된다. 3D 재구성을 위한 서로 다른 두 모드들이 있을 수 있다. 즉, 일반 오브젝트 모델 모드(general object model mode) 및 정면 모델 모드(face model mode)이다. 두 모드들의 가장 큰 차이점은 스캐닝 방식에 있다. 일반 오브젝트 모델 모드의 경우에는, 360도의 스캐닝이 수행된다. 하지만, 정면 모델 모드의 경우에는 오직 오브젝트의 정면만을 스캐닝하고자 180도의 스캐닝이 수행된다. 그러나, 다른 형태의 3D 모델 표현(3D model representation)도 가능하다. 예를 들어, 조밀한 3D 포인트 클라우드가 3D 재구성에 사용될 수 있다. 또한, 텍스처를 입히지 않은 다각형 메쉬의 형태로 3D 모델이 표현될 수도 있다.
- [0152] 도 18에서 설명된 하드웨어 구성을 갖는 한, 스마트폰, 태블릿, 스마트 워치 등과 같은 어떠한 종류의 모바일 디바이스들도, 앞서 설명된 3D 재구성 프로세스를 위해 이용될 수 있다.
- [0153] 본 실시예에 따른 3D 재구성 프로세스에 의해 생성된 3D 모델에 기초하여 3D 프린팅이 수행될 수도 있다.
- [0154] 본 실시예들에 따르면, 사람 얼굴에 대한 재구성된 3D 모델을 디지털 파일로 저장함으로써, 3D 포토가 활용되거나 또는 인터넷 상에 공유될 수 있다.
- [0155] 도 19는 일 실시예에 따른 오브젝트의 3차원 모델을 구성하는 방법의 흐름도이다. 도 19를 참고하면, 3차원 모델의 구성 방법은 앞서 설명된 도면들에서 시계열적으로 처리되는 과정들이므로, 이하 생략된 내용이라 하더라도 앞서 도면들에서 설명되었던 내용은 도 19의 3차원 모델의 구성 방법에도 적용될 수 있다.

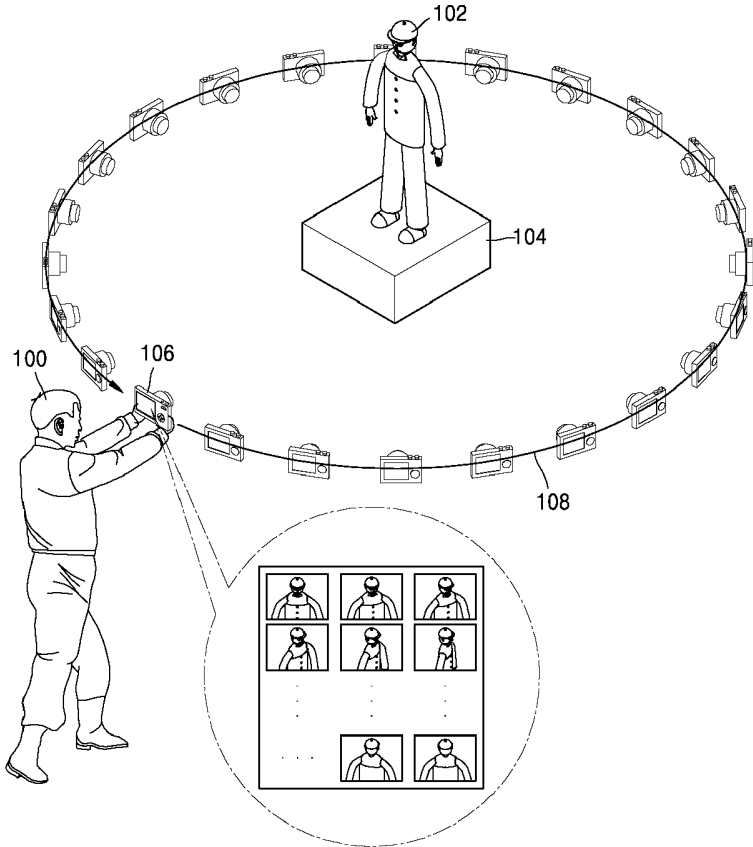
- [0156] 1901 단계에서, 디바이스(106)는 오브젝트 주위의 궤도를 따라 오브젝트를 스캐닝함으로써 오브젝트의 이미지들을 캡처한다.
- [0157] 1902 단계에서, 디바이스(106)는 캡처된 이미지들 각각에 대응되는 스캐닝 모듈(1812)의 위치들을 추정한다.
- [0158] 1903 단계에서, 디바이스(106)는 궤도 상의 적어도 둘 이상의 위치들에 기초하여 스캐닝 모듈(1812)의 추정된 위치들을 미세조정한다.
- [0159] 1904 단계에서, 디바이스(106)는 스캐닝 모듈(1812)의 미세조정된 위치들에 대응되는 깊이 맵들을 추정한다.
- [0160] 1905 단계에서, 디바이스(106)는 추정된 깊이 맵들을 합성함으로써, 오브젝트의 표면 메쉬를 생성한다.
- [0161] 1906 단계에서, 디바이스(106)는 생성된 표면 메쉬에 텍스처들을 매핑함으로써, 오브젝트의 3D 모델을 구성하여 표시한다.
- [0162] 한편, 상술한 방법은 컴퓨터에서 실행될 수 있는 프로그램으로 작성 가능하고, 컴퓨터로 읽을 수 있는 기록매체를 이용하여 상기 프로그램을 동작시키는 범용 디지털 컴퓨터에서 구현될 수 있다. 또한, 상술한 방법에서 사용된 데이터의 구조는 컴퓨터로 읽을 수 있는 기록매체에 여러 수단을 통하여 기록될 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)와 같은 저장매체를 포함한다.
- [0163] 본 실시예와 관련된 기술 분야에서 통상의 지식을 가진 자는 상기된 기재의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 방법들은 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.

도 1

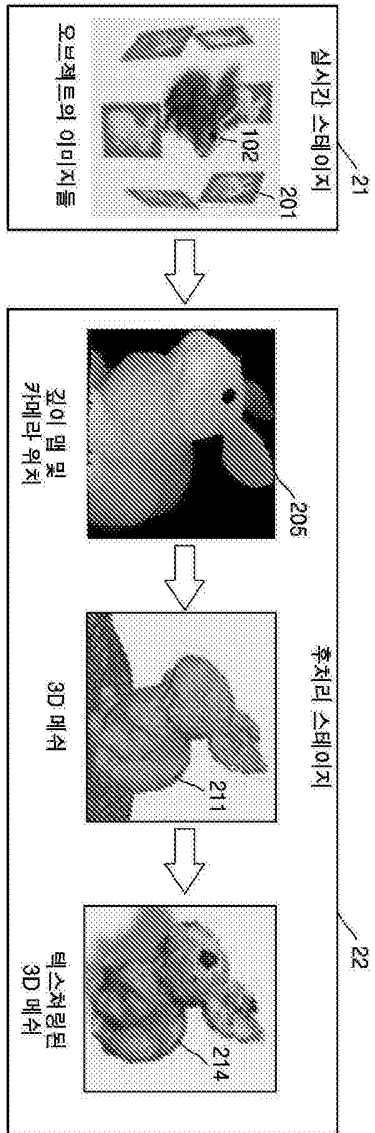
도 1a



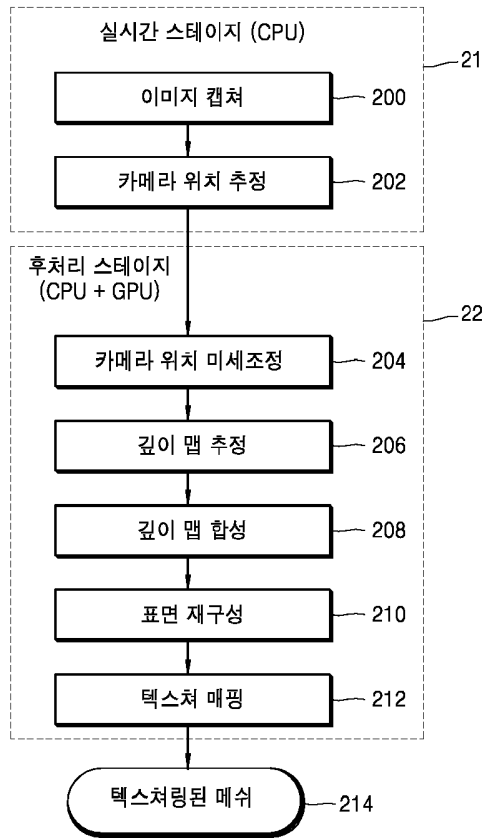
도면 1b



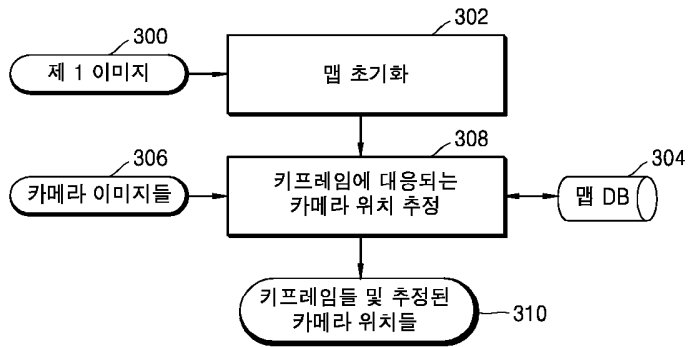
도 22



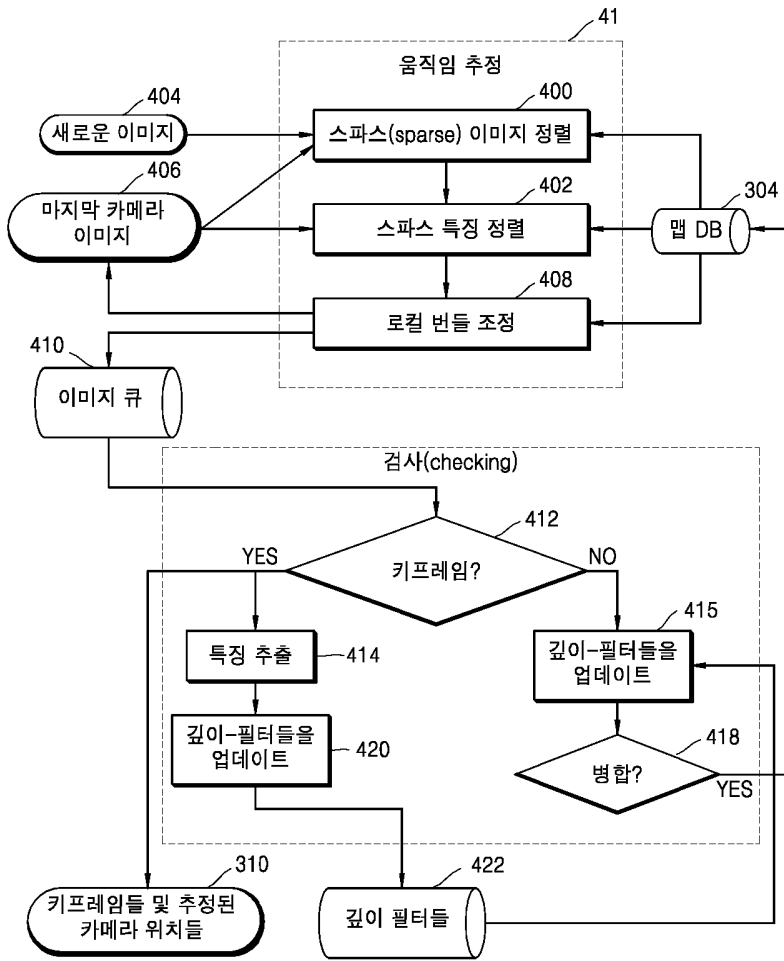
도면28



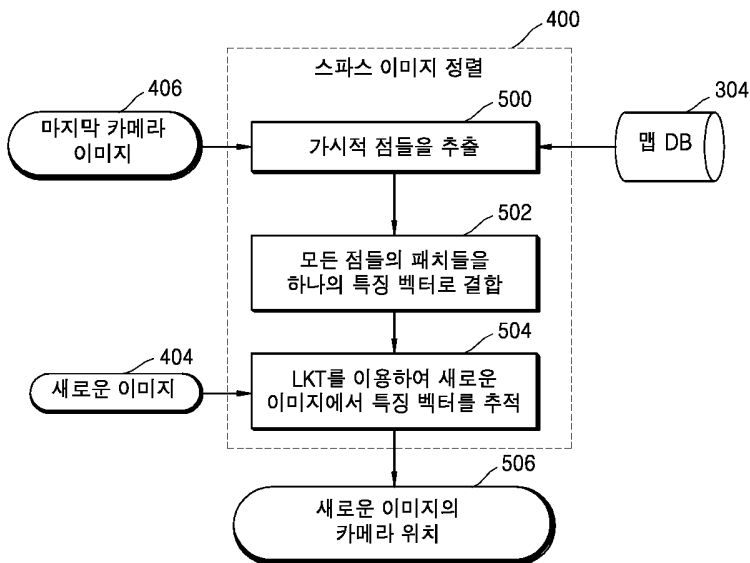
도면29



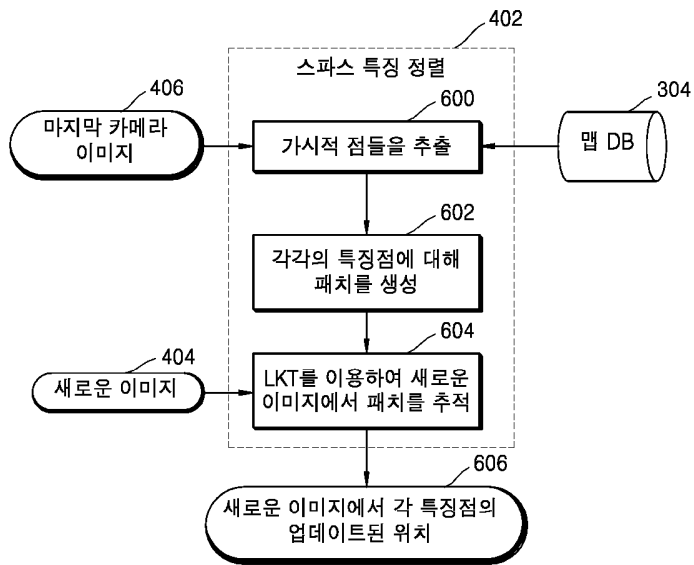
도면4



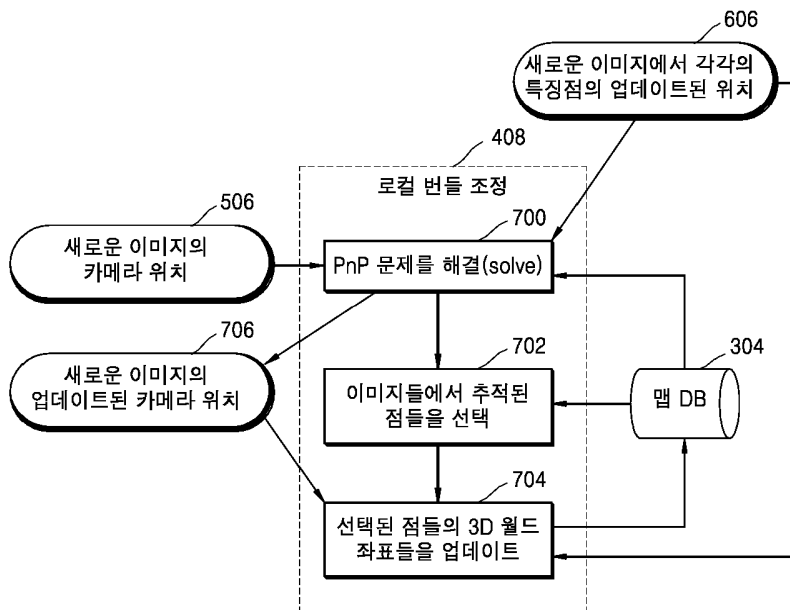
도면5



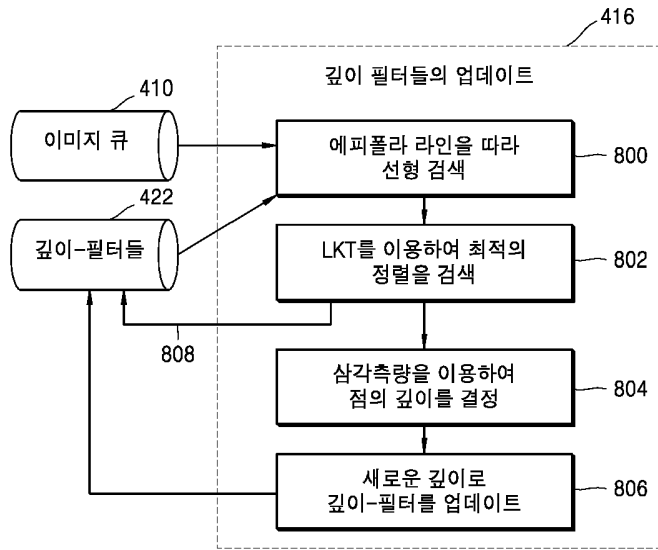
도면6



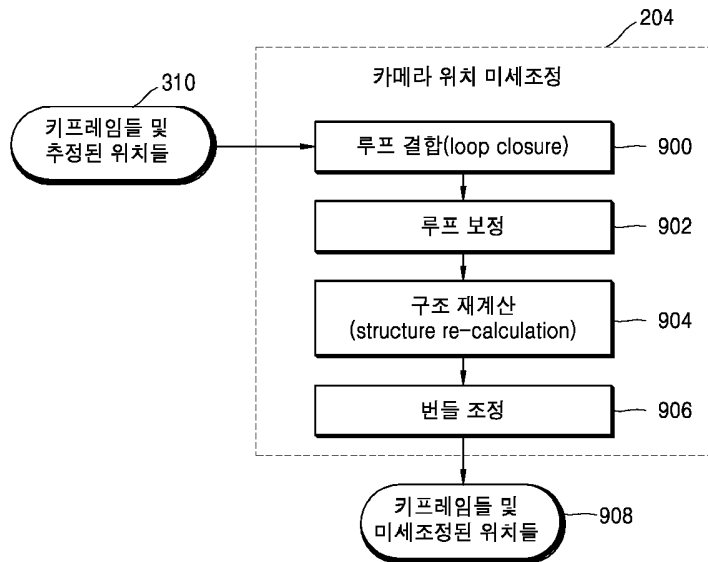
도면7



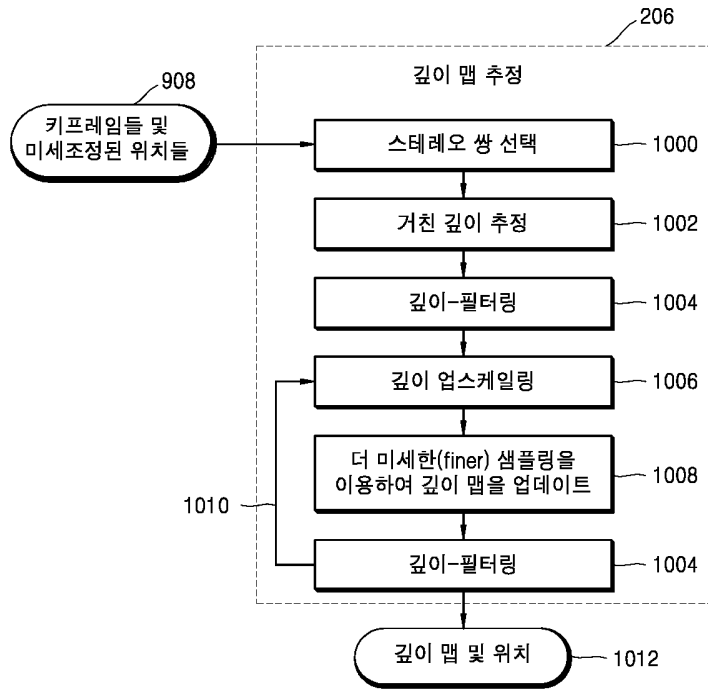
도면8



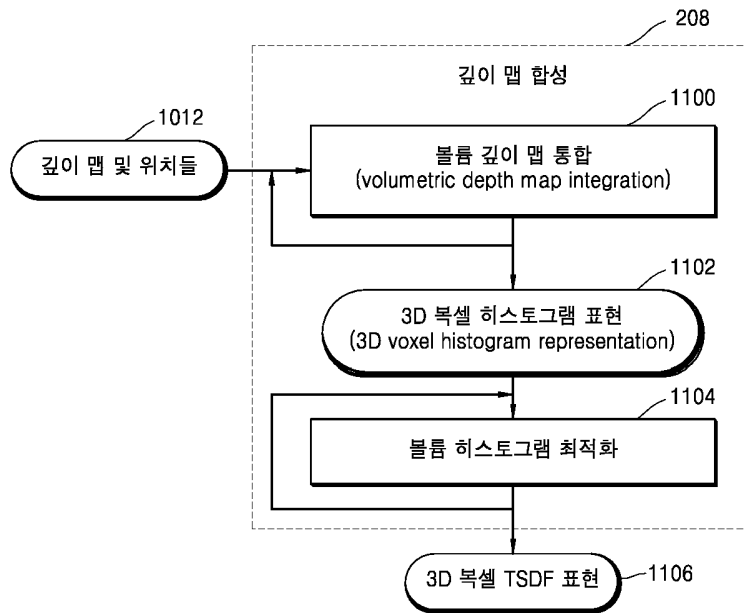
도면9



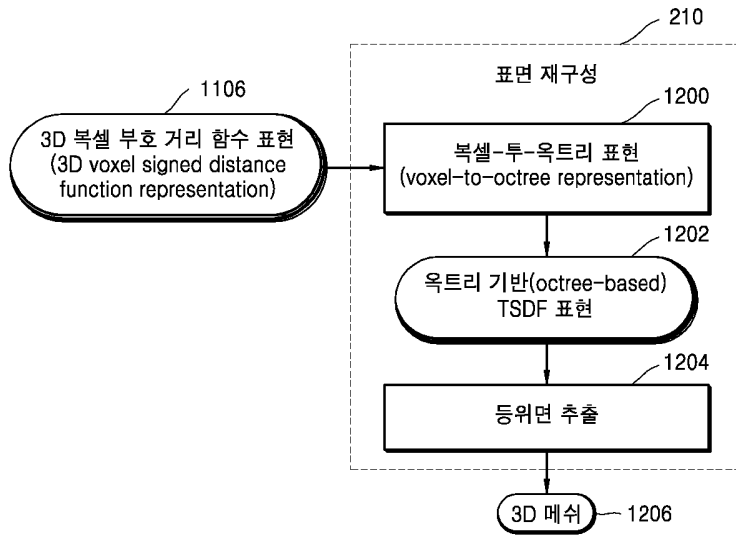
도면10



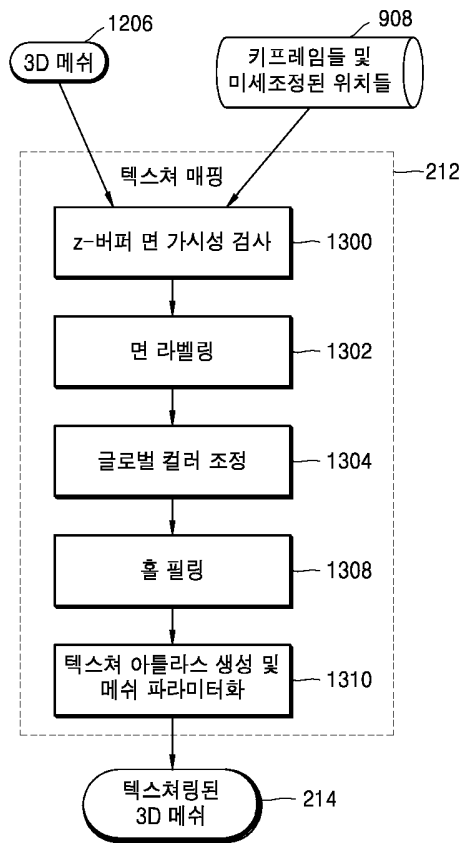
도면11



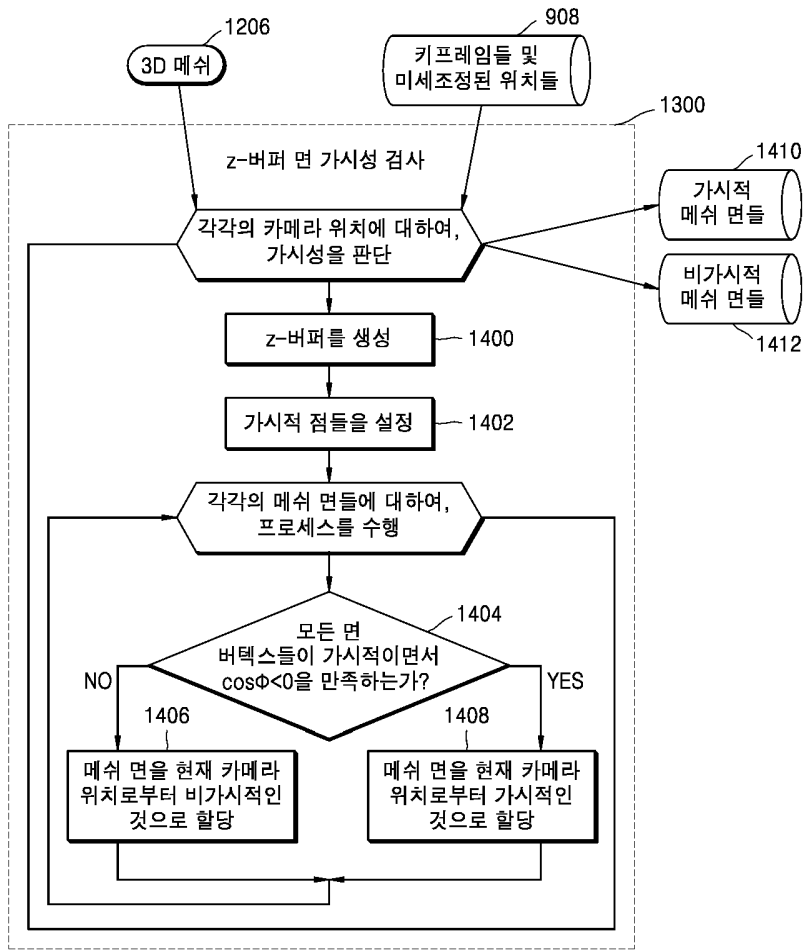
도면 12



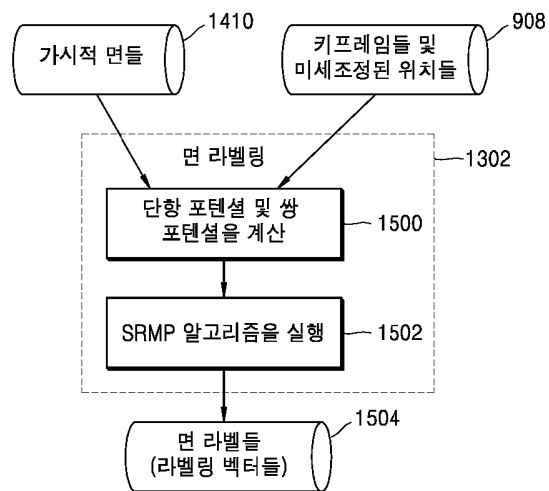
도면 13



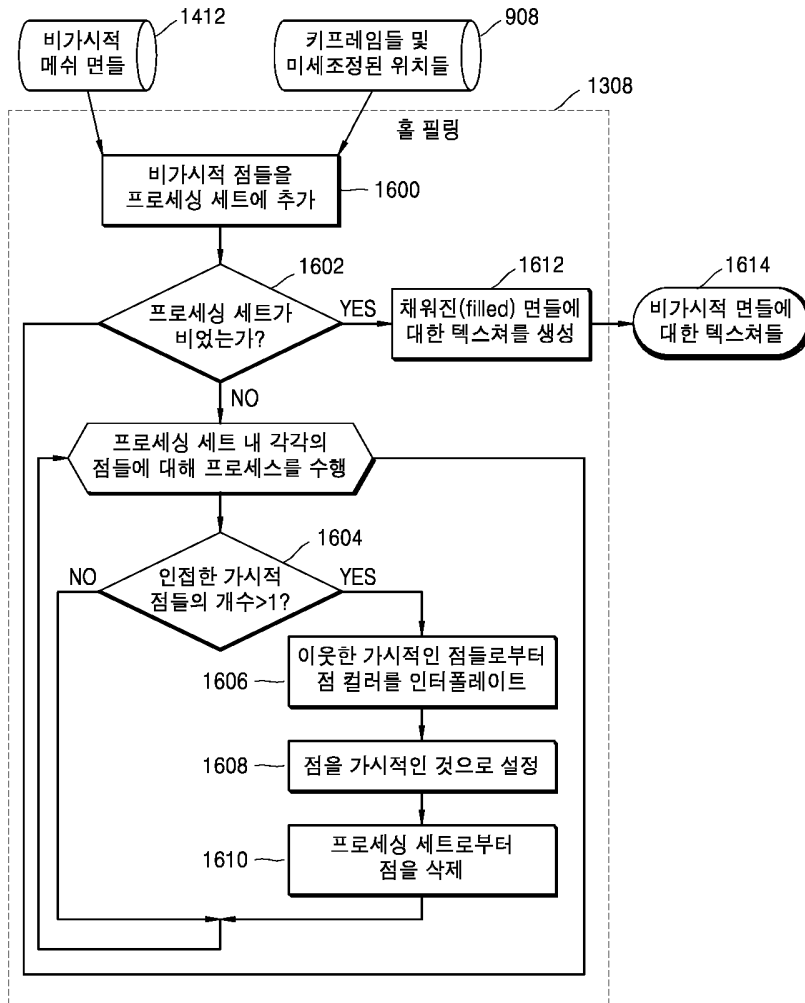
도면14



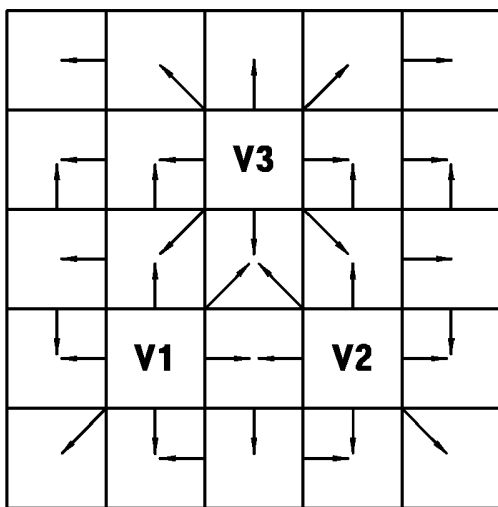
도면15



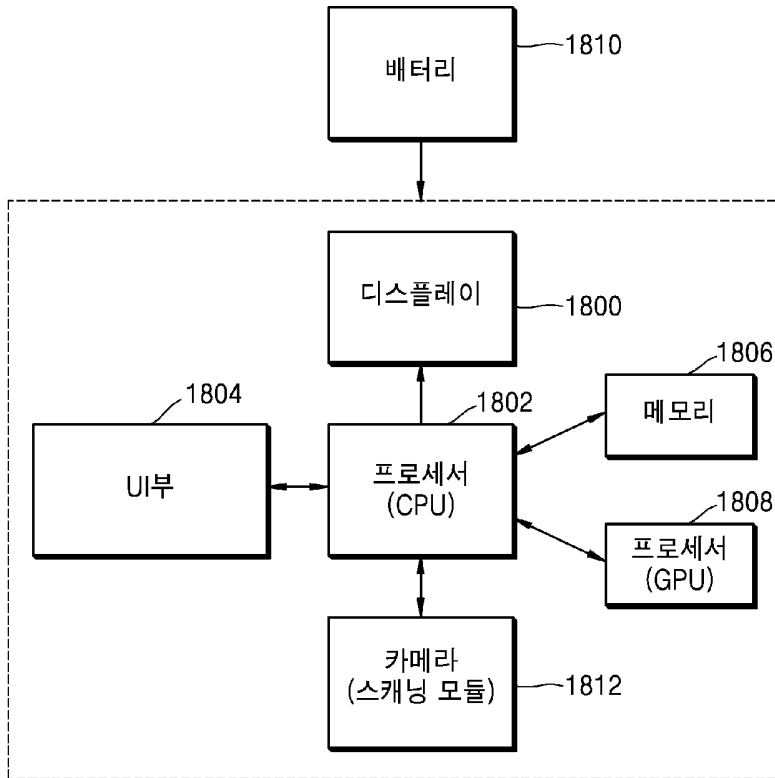
도면16



도면17



도면18



도면19

